TRƯỜNG ĐẠI HỌC BÁCH KHOA ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN

——— Â ———

GIÁO TRÌNH MÔN HỌC

MẠNG MÁY TÍNH

Ths. NGUYỄN TẤN KHÔI

(Lưu hành nội bộ)
Đà Nẵng – 2004

MỤC LỤC

Chương 1

MỞ ĐẦU

1.1 Giới thiệu

Mạng máy tính là tập hợp nhiều máy tính điện tử và các thiết bị đầu cuối được kết nối với nhau bằng các thiết bị liên lạc nhằm trao đổi thông tin, cùng chia sẻ pcứng, phần mềm và dữ liệu với nhau

Mạng máy tính bao gồm phần cứng, các giao thức và các phần mềm mạng.

Khi nghiên cứu về mạng máy tính, các vấn đề quan trọng được xem xét là giao thức mạng, cấu hình kết nối của mạng, và các dịch vụ trên mạng.

Mạng máy tính có những công dụng như sau :

1. *Tập trung tài nguyên tại một số máy và chia sẻ cho nhiều máy khác*

- Nhiều người có thể dùng chung một phần mềm tiện ích.
- Dữ liệu được quản lý tập trung nên an toàn hơn, trao đổi giữa những người sử dụng thuận lợi hơn, nhanh chóng hơn.
- Mạng máy tính cho phép người lập trình ở một trung tâm máy tính này có thể sử dụng các chương trình tiện ích của một trung tâm máy tính khác đang rỗi, sẽ làm tăng hiệu quả kinh tế của hệ thống.

2. *Khắc phục sự trở ngại về khoảng cách địa lý.*
3. *Tăng chất lượng và hiệu quả khai thác thông tin.*
4. *Cho phép thực hiện những ứng dụng tin học phân tán*
5. *Độ an toàn tin cậy của hệ thống tăng lên nhờ khả năng thay thế khi có sự cố với máy có sự cố :* An toàn cho dữ liệu và phần mềm vì phần mềm mạng sẽ khoá các tập tin khi có những người không đủ quyền hạn truy xuất các tập tin và thư mục đó.
6. *Phát triển các công nghệ trên mạng:* Người sử dụng có thể trao đổi thông tin với nhau dễ dàng và sử dụng hệ mạng như là một công cụ để phổ biến tin tức, thông báo về một chính sách mới, về nội dung buổi họp, về các thông tin kinh tế khác như giá cả thị trường, tin rao vặt (muốn bán hoặc muốn mua một cái gì đó), hoặc sắp xếp thời khoá biểu của mình chen lẫn với thời khoá biểu của những người khác , . . .

1.2 Phân loại mạng

1.2.1 Dựa theo khoảng cách địa lý

Mạng máy tính có thể phân bổ trên một khu vực nhất định hoặc có thể trong một quốc gia hay toàn cầu. Dựa vào phạm vi phân bố, người ta có thể phân ra các loại mạng như sau:

a. LAN (Local Area Network - Mạng cục bộ) : LAN thường được sử dụng trong nội bộ một cơ quan/tổ chức..., kết nối các máy tính trong một khu vực bán kính khoảng 100m-10km. Kết nối được thực hiện thông qua các môi trường truyền thông tốc độ cao, ví dụ cáp đồng trục hay cáp quang.

b. MAN (Metropolitan Area Network - Mạng đô thị) : Kết nối các máy tính trong phạm vi một thành phố. Kết nối này được thực hiện thông qua các môi trường truyền thông tốc độ cao (50-100 Mbit/s).

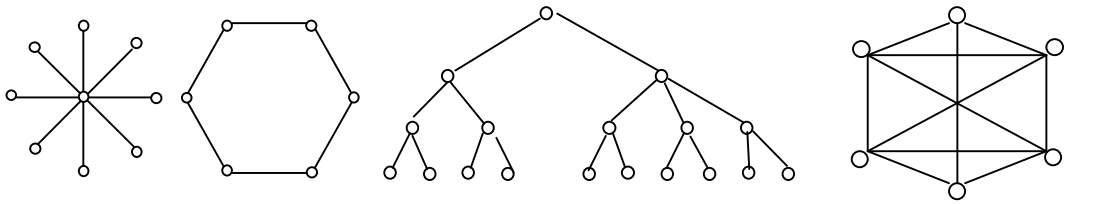
c. WAN (Wide Area Network) - Mạng diện rộng, kết nối máy tính trong nội bộ các quốc gia hay giữa các quốc gia trong cùng một châu lục. Thông thường kết nối này được thực hiện thông qua mạng viễn thông. Các WAN có thể được kết nối với nhau thành GAN hay tự nó đã là GAN.

d. GAN (Global Area Network) : Mạng toàn cầu, kết nối máy tính từ các châu lục khác nhau. Thông thường kết nối này được thực hiện thông qua mạng viễn thông và vệ tinh.

Trong các khái niệm nói trên, WAN và LAN là hai khái niệm hay được sử dụng nhất.

1.2.2 Dựa theo cấu trúc mạng

1.2.2.1 Kiểu điểm - điểm (point - to - point)



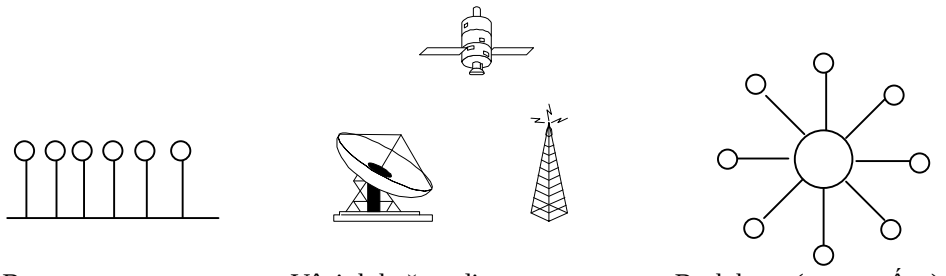
H×nh sao *Chu tr×nh* *H×nh c©y* *D¹ng ®Çy ®ñ*

Hình 1-1. Cấu trúc mạng kiểu điểm-điểm.

Đường truyền nối từng cặp nút mạng với nhau. Thông tin đi từ nút nguồn qua nút trung gian rồi gởi tiếp nếu đường truyền không bị bận. Do đó còn có tên là mạng lưu trữ và chuyển tiếp (*store and forward*).

1.2.2.2 Kiểu khuyếch tán

Bản tin được gởi đi từ một nút nào đó sẽ được tiếp nhận bởi các nút còn lại (còn gọi là broadcasting hay point to multipoint). Trong bản tin phải có vùng địa chỉ cho phép mỗi nút kiểm tra xem có phải tin của mình không và xử lý nếu đúng bản tin được gởi đến.



Hình 1-2. Sơ đồ kết nối theo kiểu khuyếch tán.

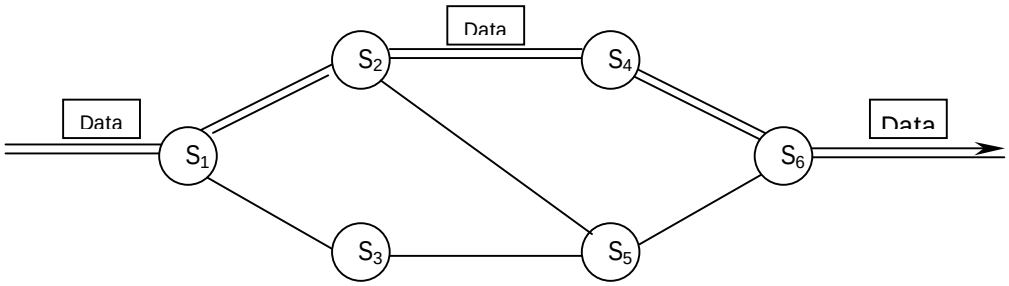
Trong cấu trúc dạng Bus và Vòng cần cơ chế "*trọng tài*" để giải quyết các xung đột (collision) xảy ra khi nhiều nút muốn truyền tin đồng thời. Trong cấu trúc vệ tinh hoặc radio, mỗi nút cần có ăng-ten thu và phát.

1.2.3 Theo phương pháp chuyển mạch

- Mạng chuyển mạch kênh (Line switching network), ví dụ như mạng điện thoại.
- Mạng chuyển mạch thông báo (Message switching network)
- Mạng chuyển mạch gói (Packet switching network)

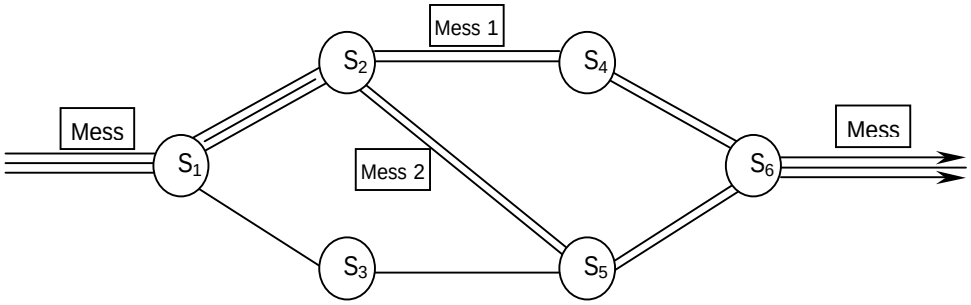
1.2.3.1 Chuyển mạch kênh

Chuyển mạch kênh (line switching) được dùng trong mạng điện thoại. Một kênh cố định được thiết lập giữa cặp thực thể cần liên lạc với nhau. Mạng này có hiệu suất không cao vì có lúc kênh bỏ không.



Hình 1-3. Mạng chuyển mạch kênh.

1.2.3.2 Mạng chuyển mạch bản tin



Hình 1-4. Phương pháp chuyển mạch thông báo.

Các nút của mạng căn cứ vào địa chỉ đích của "bản tin" để chọn nút kế tiếp. Như vậy các nút cần lưu trữ và đọc tin nhận được, quản lý việc truyền tin. Trong trường hợp bản tin quá dài và nếu sai phải truyền lại thì hiệu suất không cao. Phương pháp này giống như cách gởi thư thông thường.

- Ưu điểm so với phương pháp chuyển mạch kênh:

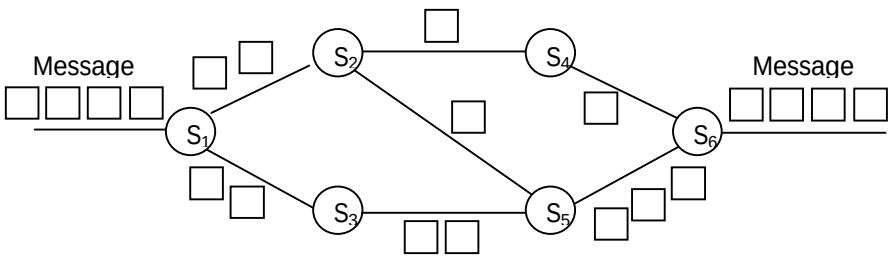
- Hiệu suất sử dụng đường truyền cao vì không bị chiếm dụng độc quyền mà được phân chia giữa nhiều thực thể.
- Mỗi nút mạng (hay nút chuyển mạch thông bao) có thể lưu trữ message cho tới khi kênh truyền rỗi mới gửi bản tin đi. Do đó giảm được tình trạng tắc nghẽn (congestion) trên mạng.
- Điều khiển việc truyền tin bằng cách sắp xếp độ ưu tiên cho các bản tin.
- Có thể tăng hiệu suất sử dụng giải thông của mạch bằng cách gán địa chỉ quảng bá (broadcast) để gởi bản tin đồng thời đến nhiều đích.

- Nhược điểm:

- Do không hạn chế kích thước của bản tin nên có thể dẫn đến phí tổn lưu trữ tạm thời cao và ảnh hưởng đến thời gian hồi đáp và chất lượng truyền đi.

Mạng chuyển mạch thông báo thích hợp với các dịch vụ thông tin kiểu thư điện tử (Email) hơn là đối với các ứng dụng có tính thời gian thực vì tồn tại độ trễ nhất định do lưu trữ và xử lý thông tin điều khiển tại mỗi nút.

1.2.3.3 Mạng chuyển mạch gói



Hình 1-5. Mạng chuyển mạch gói.

Bản tin được chia thành nhiều gói tin (packet) độ dài 512 bytes, phần đầu là địa chỉ đích, mã để tập hợp các gói. Các gói của các bản tin khác nhau có thể được truyền độc lập trên cùng một đường truyền. Vấn đề phức tạp ở đây là tạo lại bản tin ban đầu, đặc biệt khi được truyền trên các con đường khác nhau.

Chuyển mạch gói mềm dẻo, hiệu suất cao. Xu hướng phát triển hiện nay là sử dụng hai kỹ thuật chuyển mạch kênh và chuyển mạch gói trong cùng một mạng thống nhất gọi là mạng ISDN (*Integrated Services Digital Network* - Mạng thông tin số đa dịch vụ).

1.3 Kiến trúc phân tầng và chuẩn hoá mạng

Tình trạng không tương thích giữa các mạng đặc biệt là các mạng trên thị trường gây trở ngại cho những người sử dụng khác nhau. Do đó cần phải xây dựng mô hình chuẩn làm cơ sở cho các nhà nghiên cứu thiết kế mạng để tạo ra các sản phẩm mới về mạng, dễ phổ cập, sản xuất, sử dụng. Các chuẩn có vai trò quan trọng trong công tác thiết kế và xây dựng các hệ thống kỹ thuật và công nghệ.

Chuẩn hóa mạng máy tính là nêu ra các tiêu chuẩn cơ bản thống nhất về cấu trúc mạng giúp cho các mạng khác nhau có thể trao đổi thông tin được với nhau.

Để mạng hoạt động đạt khả năng tối đa, các tiêu chuẩn được chọn phải cho phép mở rộng mạng để có thể phục vụ những ứng dụng không dự kiến trước trong tương lai tại lúc lắp đặt hệ thống và điều đó cũng cho phép mạng làm việc với những thiết bị được sản xuất từ nhiều hãng khác nhau.

1.3.1 Các tổ chức chuẩn hoá mạng

Hai tổ chức chính thực hiện chuẩn hóa mạng là ISO và CCTTT.

1. ISO (*International Standards Organization*) - Tổ chức chuẩn hóa quốc tế. ISO hoạt động dưới sự bảo trợ của LHQ. Thành viên của ISO là các cơ quan tiêu chuẩn hóa của các quốc gia và các Ban chuyên môn. Ban TC97 được chia ra thành các tiểu ban và các nhóm công tác.
2. IEEE (*Institute of Electrical and Electronic Engineers*) - Viện nghiên cứu các vấn đề về kỹ thuật điện và điện tử của Mỹ. IEEE chịu trách nhiệm về tầng Data Link và Physical. Phân ban các chuẩn này là phân ban 802 (thành lập tháng Hai năm 1980).
3. CCITT (*Commité Consultatif Internatinal pour Télégraphe et Téléphone*) - Tổ chức tư vấn quốc tế về điện báo và điện thoại hoạt động dưới sự bảo trợ của LHQ, chuyên nghiên cứu nhằm công bố các khuyến nghị thống nhất về mạng

máy tính. Bao gồm các khuyến nghị liên quan đến việc truyền dữ liệu trên mạng, mạng ISDN.

4. ANSI (*American National Standards Institute*) :Viện nghiên cứu các chuẩn quốc gia của Mỹ.
5. ECMA (*European Computer Manufactures Association*) : Hiệp hội máy tính châu âu
6. ATM Forum (*Asynchronous Transfers Mode*) - Thực hiện các giải pháp cho mạng ISDN.
7. IETF (*Internet Enggineering Task Force*) : Sản xuất các chuẩn liên quan đến Internet (SNMP, TCP/IP ...)

1.3.2 Kiến trúc phân tầng

Để giảm độ phức tạp thiết kế, kiến trúc mạng được tổ chức thành một cấu trúc đa tầng, mỗi tầng được xây trên tầng trước nó, tầng dưới sẽ cung cấp dịch vụ cho tầng cao hơn. Tầng N trên một máy thực hiện việc giao tiếp với tầng N trên một máy khác. Các qui tắc, luật lệ được sử dụng cho việc giao tiếp này được gọi là các giao thức của tầng N.

Các thực thể (entity) nằm trên các tầng tương ứng trên những máy khác nhau gọi là các tiến trình đồng mức. Các tiến trình đồng mức giao tiếp với nhau bằng cách sử dụng các giao thức trong tầng của nó.

Giữa 2 tầng kề nhau tồn tại một giao diện (*interface*) xác định các hàm nguyên thủy và các dịch vụ tầng dưới cung cấp cho tầng trên.

Tập hợp các tầng và các giao thức được gọi là kiến trúc mạng (*Network Architecture*).

Cấu trúc phân tầng của mạng máy tính có ý nghĩa đặc biệt như sau :

- Thuận tiện trong công tác thiết kế, xây dựng và cài đặt các mạng máy tính, trong đó mỗi hệ thống thành phần được xem như là một cấu trúc đa tầng.
- Mỗi tầng được xây dựng dựa trên cơ sở tầng kề liền trước đó. Như vậy tầng dưới sẽ cung cấp dịch vụ cho tầng trên.
- Số lượng, tên gọi và chức năng của mỗi tầng sẽ được người thiết kế mạng máy tính cụ thể quy định.
- Tập hợp các giao thức, các vấn đề kỹ thuật và công nghệ cho mỗi tầng có thể được khảo sát, nghiên cứu triển khai độc lập với nhau.

- Giao thức : Mỗi khi trao đổi thông tin như điện thoại, telex, viết . . . người ta phải tuân theo một số quy luật. Các quy luật này được nhóm lại và gọi là giao thức (*protocol*).

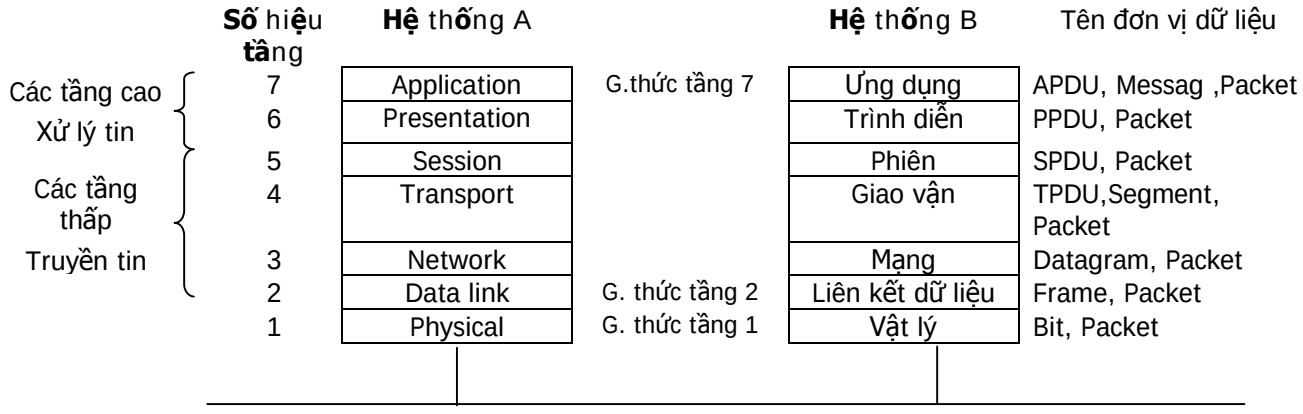
Giao thức có các chức năng chính như sau :

1. Định nghĩa cấu trúc khung một cách chính xác cho từng byte, các ký tự và bản tin.
2. Phát hiện và xử lý các lỗi, thông thường là gửi lại bản tin gốc sau khi phát hiện lần trước bị lỗi
3. Quản lý thứ tự các lệnh để đếm các bản tin, nhận dạng, tránh mất hoặc thu thừa bản tin.
4. Đảm bảo không nhầm lẫn giữa bản tin và lệnh
5. Chỉ ra các thuộc tính đường dây khi lập các đường nối đa điểm hoặc bán song công (cho biết ai đối thoại với ai).
6. Giải quyết vần đề xung đột thâm nhập (yêu cầu đồng thời), gửi khi chưa có số liệu, mất liên lạc, khởi động.

1.4 Mô hình OSI

1.4.1 Kiến trúc của mô hình OSI

Dựa trên kiến trúc phân tầng, ISO đã đưa ra mô hình 7 tầng (layer) cho mạng, gọi là mô hình kết nối hệ thống mở hoặc mô hình OSI (Open Systems Interconnection model), vào năm 1984.



Hình 1-6. Mô hình OSI 7 tầng.

Nhóm các tầng thấp (*physical, data link, network, transport*) liên quan đến các phương tiện cho phép truyền dữ liệu qua mạng. Các tầng thấp đảm nhiệm việc truyền dữ liệu, thực hiện quá trình đóng gói, dẫn đường, kiểm duyệt và truyền từng nhóm dữ liệu. Các tầng này không cần quan tâm đến loại dữ liệu mà nó nhận được từ hay gửi cho tầng ứng dụng, mà chỉ đơn thuần là gửi chúng đi.

Nhóm các tầng cao (*session, presentation, application*) liên quan chủ yếu đến việc đáp ứng các yêu cầu của người sử dụng để triển khai các ứng dụng của họ trên mạng thông qua các phương tiện truyền thông cung cấp bởi các nhóm tầng thấp.

Hệ thống kết nối mở OSI là hệ thống cho phép truyền thông tin với các hệ thống khác, trong đó các mạng khác nhau, sử dụng những giao thức khác nhau, có thể thông báo cho nhau thông qua chương trình để chuyển từ một giao thức này sang một giao thức khác.

Mô hình OSI đưa ra giải pháp cho vấn đề truyền thông giữa các máy tính không giống nhau. Hai hệ thống, dù khác nhau đều có thể truyền thông với nhau một cách hiệu quả nếu chúng đảm bảo những điều khiển chung sau đây :

1. Các hệ thống đều cài đặt cùng một tập hợp các chức năng truyền thông.
2. Các chức năng đó được tổ chức thành cũng một tập các tầng. Các tầng đồng mức phải cung cấp các chức năng như nhau, nhưng phương thức cung cấp không nhất thiết phải giống nhau.
3. Các tầng đồng mức phải sử dụng một giao thức chung.

Để đảm bảo những điều trên cần phải có các chuẩn xác định các chức năng và dịch vụ được cung cấp bởi một tầng (nhưng không cần chỉ ra chúng phải cài đặt như thế nào). Các chuẩn cũng phải xác định các giao thức giữa các tầng đồng mức. Mô hình OSI chính là cơ sở để xây dựng các chuẩn đó.

1.4.2 Sự ghép nối giữa các mức

Trong thực tế dữ liệu không truyền trực tiếp từ tầng i máy này sang tầng i máy kia (trừ tầng thấp nhất). tầng thấp nhất có đường truyền thông vật lý tới tầng thấp nhất của máy tương ứng từ đó dữ liệu và thông tin điều khiển lại được chuyển ngược lên tầng trên. Tầng trên chỉ xác định đường truyền thông lôgic (truyền thông ảo).

- Các Header của giao thức : Thông thường, thông tin điều khiển giao thức được gói thành một khối và được đặt trước dữ liệu nó đi kèm và được gọi là *Header* hay *Protocol Header*, được dùng để truyền thông tin giữa các tầng và giữa các máy tính với nhau. Các header của giao thức được phát triển theo các luật được cho trong tập tài liệu ASN.1 của ISẶ.
- Khi máy A gửi tin đi, các đơn vị dữ liệu đi từ tầng trên xuống dưới. Qua mỗi tầng nó được bổ sung thông tin điều khiển của tầng đó.
- Khi nhận tin, thông tin đi từ dưới lên. Qua mỗi tầng thông tin điều khiển được khử bỏ dần và cuối cùng máy B nhận được bản tin của A.

1.4.3 Chức năng của mỗi tầng

1. Tầng Vật lý

Cung cấp phương tiện truyền tin, thủ tục khởi động, duy trì huỷ bỏ các liên kết vật lý. Giữ nhiệm vụ chuyển tải các bit thông tin trên kênh truyền thông. Tầng Vật lý làm việc với các giao diện cơ, điện và giao diện thủ tục (chức năng) trên môi trường vật lý, không quan tâm đến nội dung biểu diễn của các bit.

Thực chất tầng này thực hiện nối liền các phần tử của mạng thành một hệ thống bằng các phương pháp vật lý, ở mức này sẽ có các thủ tục đảm bảo cho các yêu cầu về chuyển mạch hoạt động nhằm tạo ra các đường truyền thực cho các chuỗi bit thông tin.

2. Tầng liên kết dữ liệu

Thiết lập, duy trì, huỷ bỏ các liên kết dữ liệu kiểm soát luồng dữ liệu, phát hiện và khắc phục sai sót truyền tin

Tiến hành chuyển đổi thông tin dưới dạng chuỗi các bit ở mức mạng thành từng đoạn gọi là khung tin (frame). Sau đó đảm bảo truyền liên tiếp các khung tin tới tầng vật lý, đồng thời xử lý các thông báo từ trạm thu gửi trả lại. Bit thông tin trong khung tin đều mang những ý nghĩa riêng, bao gồm các *trường địa chỉ, trường kiểm tra, dữ liệu và kiểm tra lỗi* dùng cho các mục đích riêng.

Nhiệm vụ chính của mức 2 này là khởi tạo, tổ chức các khung tin và xử lý các thông tin liên quan tới khung tin.

3. Tầng mạng

Tầng mạng được xây dựng dựa trên kiểu nối kết *điểm - điểm* do tầng LKDL cung cấp, bảo đảm trao đổi thông tin giữa các mạng con trong một mạng lớn, mức này còn được gọi là mức thông tin giữa các mạng con với nhau.

Có nhiệm vụ gán địa chỉ cho các bản tin và chuyển đổi địa chỉ logic hay các tên thành các địa chỉ vật lý.

Thực hiện chọn đường truyền tin, cung cấp dịch vụ định tuyến (chọn đường) cho các gói dữ liệu trên mạng. Tầng này chỉ ra dữ liệu từ nguồn tới đích sẽ đi theo tuyến nào trên cơ sở các điều kiện của mạng, độ ưu tiên dịch vụ và các nhân tố khác.

Kiểm soát luồng dữ liệu, khắc phục sai sót, cắt/hợp dữ liệu, giúp loại trừ sự tắc nghẽn cũng như điều khiển luồng thông tin.

4. Tầng Giao vận

Tầng giao vận giúp đảm bảo độ tin cậy khi chuyển giao dữ liệu và tính toàn vẹn dữ liệu từ nơi gửi đến nơi nhận. Điều này được thực hiện dựa trên cơ chế kiểm tra lỗi do các tầng bên dưới cung cấp. Tầng giao vận còn chịu trách nhiệm tạo ra nhiều kết nối cục bộ trên cùng một kết nối mạng gọi là ghép kênh (multiplexing), phân chia thời gian xử lý (time sharing), cắt hợp dữ liệu.

Nhiệm vụ của mức này là xử lý các thông tin để chuyển tiếp các chức năng từ tầng phiên đến tầng mạng và ngược lại. Thực chất mức truyền này là để đảm bảo thông tin giữa các máy chủ với nhau. Mức này nhận các thông tin từ tầng phiên, phân chia thành các đơn vị dữ liệu nhỏ hơn và chuyển chúng tới mức mạng.

5. Tầng phiên

Thiết lập, duy trì, đồng bộ hoá và huỷ bỏ các phiên truyền thông. Liên kết phiên phải được thiết lập thông qua đối thoại và trao đổi các thông số điều khiển.

Dùng tầng giao vận để cung cấp các dịch vụ nâng cao cho phiên làm việc như: kiểm soát các cuộc hội thoại, quản lý thẻ bài (*token*), quản lý hoạt động (*activity management*).

Nhận dạng tên và thủ tục cần thiết cũng như là các công việc bảo mật, để hai ứng dụng có thể giao tiếp với nhau trên mạng. Nhờ tầng phiên, những người sử dụng lập được các đường nối với nhau, khi cuộc hội thoại được thành lập thì mức này có thể quản lý cuộc hội thoại đó theo yêu cầu của người sử dụng. Một kết nối giữa hai máy cho phép người sử dụng được đăng ký vào một hệ thống phân chia thời gian từ xa hoặc chuyển tập tin giữa 2 máy.

6. Tầng trình diễn

Quản lý cách thức biểu diễn thông tin theo cú pháp dữ liệu của người sử dụng, loại mã sử dụng (ASCII, QBCDIC, ...) và thực hiện các vấn đề nén dữ liệu.

Nhiệm vụ của mức này là lựa chọn cách tiếp nhận dữ liệu, biến đổi các ký tự, chữ số của mã ASCII hay các mã khác và các ký tự điều khiển thành một kiểu mã nhị phân thống nhất để các loại máy khác nhau đều có thể thâm nhập vào hệ thống mạng.

7. Tầng ứng dụng

Tầng này là giao diện giữa người sử dụng và môi trường hệ thống mở.

Tầng này có nhiệm vụ phục vụ trực tiếp cho người sử dụng, cung cấp tất cả các yêu cầu phối ghép cần thiết cho người sử dụng, yêu cầu phục vụ chung như chuyển các File, sử dụng các Terminal của hệ thống,.... Mức sử dụng bảo đảm tự động hoá quá trình thông tin, giúp cho người sử dụng khai thác mạng tốt nhất.

1.4.4 Các giao thức chuẩn của OSI

1.4.4.1 Các hàm nguyên thủy

Mỗi thực thể truyền thông với các thực thể ở tầng trên và dưới nó qua một *giao diện* (interface). Giao diện này gồm một hoặc nhiều điểm truy cập dịch vụ (SAP - Service Access Point). Thực thể tầng N-1 cung cấp dịch vụ cho thực thể tầng N thông qua việc gọi các hàm dịch vụ nguyên thủy (primitive).

Hàm nguyên thuỷ chỉ rõ chức năng cần thực hiện và được dùng để chuyển dữ liệu và thông tin điều khiển. Bốn hàm nguyên thuỷ được sử dụng để định nghĩa tương tác giữa các tầng kề nhau như sau :

request	*Ịêu cầu*
indication	*Chỉ báo*
response	*Trả lời*
confirm	*Xác nhận*

request được gởi bởi người sử dụng dịch vụ ở tầng N+1 trong hệ thống A để gọi thủ tục của giao thức ở tầng N. Ịêu cầu này được cấu tạo dưới dạng một hoặc nhiều đơn vị dữ liệu giao thức (PDU - *Protocol Data Unit*) để gởi tới B.

Khi nhận được PDU, một thủ tục của giao thức ở tầng N của B sẽ thông báo yêu cầu đó lên tầng N+1 bằng hàm nguyên thủy *indication*. Sau đó *response* được gởi từ N + 1 của B xuống N gọi thủ tục giao thức tầng N để trả lời tới A.

Khi nhận được trả lời này một thủ tục giao thức tầng N sẽ gởi hàm *confirm* lên N+1 để hoàn tất chu trình yêu cầu thiết lập liên kết của người sử dụng ở tầng N+1 của A.

Các chu trình của người sử dụng khác nhau được phân biệt nhờ khái niệm điểm thâm nhập dịch vụ (SAP - *Service Access Point*) ở ranh giới của 2 tầng N + 1 và N.

1.4.4.2 Các phương thức truyền thông

Tại mỗi tầng trong mô hình OSI có 2 phương thức hoạt động chính được sử dụng : phương thức có liên kết (*connection oriented*) và phương thức không liên kết (*connectionless*).

Với các phương thức truyền không liên kết thì chỉ có một giai đoạn truyền dữ liệu. Các gói tin dữ liệu (còn được gọi là datagram) được truyền độc lập với nhau theo một con đường xác định dần bằng địa chỉ đích được đặt trong mỗi datagram. Có 3 giai đoạn phân biệt :

- *Thiết lập liên kết* : hai thực thể cùng tầng ở hai đầu của liên kết sẽ thương lượng với nhau về tập các tham số sử dụng trong giai đoạn truyền dữ liệu.
- *Truyền dữ liệu* : các cơ chế kiểm soát sai sót, luồng dữ liệu, ghép kênh, cắt hợp dữ liệu được thực hiện để tăng cường độ tin cậy và hiệu suất của việc truyền dữ liệu.
- *Kết thúc truyền* : giải phóng các tài nguyên hệ thống đã được cấp phát cho liên kết để dùng vào mục đích khác.

Tương ứng với 3 giai đoạn trao đổi trên, có 3 loại thủ tục cơ bản được sử dụng : CONNECT, DATA, DISCONNECT.

Ví dụ đối với giao thức tầng N ta có các thủ tục :

N_CONNECT	Thiết lập liên kết
N_DATA	Truyền dữ liệu
N_DISCONNECT	Huỷ bỏ liên kết

Ngoài ra có một số các thủ tục phụ được sử dụng tuỳ theo chức năng của mỗi tầng.

Ví dụ:	Thủ tục N_RESTART	Dùng để khởi động lại hệ thống ở tầng 3
	Thủ tục T_EXPEDITED_DATA	Dùng cho việc truyền dữ liệu nhanh tầng 4
	Thủ tục S_TOKEN_GIVE	Dùng để chuyển điều khiển ở tầng 5

Mỗi thủ tục trên sẽ dùng các hàm nguyên thuỷ (*request, indication, response, confirm*) để tạo thành các hàm cơ bản của mô hình OSI.

1.5 Hệ điều hành mạng

Việc lựa chọn hệ điều hành mạng (NOS - Network Operating System) làm nền tảng cho mạng tuỳ thuộc vào kích cỡ của mạng hiện tại và sự phát triển trong tương lai, ngoài ra còn tuỳ thuộc vào những ưu điểm và nhược điểm của từng hệ điều hành.

Một số hệ điều hành mạng phổ biến hiện nay:

- Hệ điều hành mạng UNIX: Đây là hệ điều hành do các nhà khoa học xây dựng và được dùng rất phổ biến trong giới khoa học, giáo dục. Hệ điều hành mạng UNIX là hệ điều hành đa nhiệm, đa người sử dụng, phục vụ cho truyền thông tốt. Nhược điểm của nó là hiện nay có nhiều Version khác nhau, không thống nhất gây khó khăn cho người sử dụng và là hệ điều hành này phức tạp.
- Hệ điều hành mạng Windows 2000: Đây là hệ điều hành của hãng Microsoft, cũng là hệ điều hành đa nhiệm, đa người sử dụng. Được xây dựng dựa trên công nghệ của hệ điều hành Windows NT. Đặc điểm của nó là tương đối dễ sử dụng, hỗ trợ mạnh cho các phần mềm WINDOWS. Windows 2000 có thể

liên kết tốt với máy chủ Novell Netware, Unix. Tuy nhiên, để chạy có hiệu quả, Windows 2000 Server đòi hỏi cấu hình máy tương đối mạnh.

- Hệ điều hành mạng NetWare của Novell: Đây là hệ điều hành phổ biến trên thế giới trong thời gian cuối, nó có thể dùng cho các mạng nhỏ (khoảng từ 5-25 máy tính) và cũng có thể dùng cho các mạng lớn gồm hàng trăm máy tính. Netware là một hệ điều hành LAN dùng cho các máy tính theo chuẩn của IBM hay các máy tính Apple Macintosh, chạy trên hệ điều hành MS-DOS hoặc OS/2.

1.6 Mạng Internet

1.6.1 Lịch sử ra đời và phát triển

Vào những năm 60, Bộ Quốc phòng Mỹ cho triển khai khẩn trương một mạng lưới thông tin với yêu cầu: Nếu như một trạm trung chuyển nào đó trong mạng bị phá huỷ, toàn bộ hệ thống thông tin vẫn phải làm việc bình thường... Cơ quan Nghiên cứu Dự án Cao cấp (ARPA - Advanced Research Projects Agency) thuộc Bộ Quốc phòng Mỹ được giao trách nhiệm thực hiện việc nghiên cứu kỹ thuật liên mạng (internet) nhằm đáp ứng yêu cầu trên. Đây là mạng chuyển mạch gói (packet switching) đầu tiên trên thế giới, lấy tên là ARPAnet. Ban đầu, ARPAnet chỉ gồm một vài mạng nhỏ được chọn lựa của các trung tâm nghiên cứu và phát triển khoa học. Giao thức truyền thông lúc bấy giờ là kiểu điểm - điểm, rất chậm và thường xuyên gây tắc nghẽn trên mạng. Để giải quyết vấn đề này, vào năm 1974 Vinton G. Cerf và Robert O. Kahn đưa ra ý tưởng thiết kế một bộ giao thức mạng mới thuận tiện hơn, đó chính là tiền thân của giao thức TCP/IP.

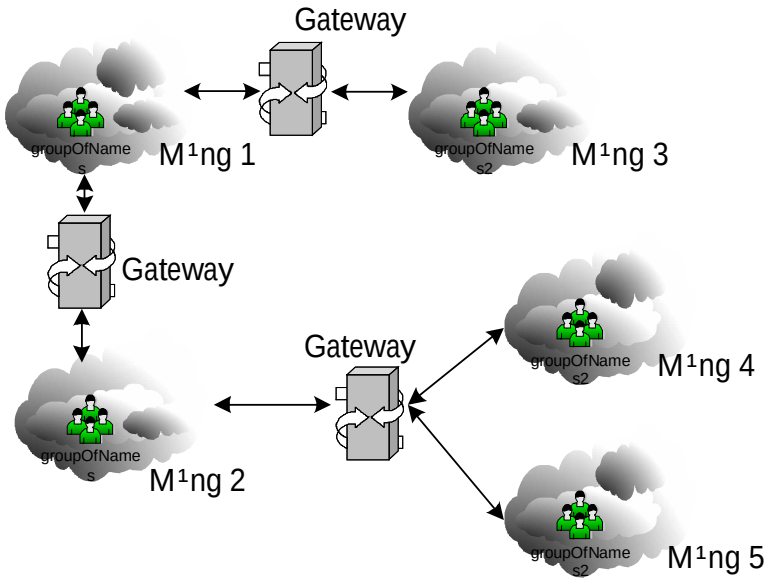
Tháng 09/1983, dưới sự tài trợ của Bộ Quốc phòng Mỹ, Berkeley Software Distribution đưa ra bản Berkeley UNIX 4.2BSD có kết hợp giao thức TCP/IP, biến TCP/IP thành phương tiện kết nối các hệ thống UNIX. Trên cơ sở đó, mạng ARPANỌT nhanh chóng lan rộng và chuyển từ mạng thực nghiệm sang hoạt động chính thức: nhiều trường đại học, viện nghiên cứu ghi tên gia nhập để trao đổi thông tin. Đến năm 1984, mạng ARPANỌT được chia thành hai nhóm mạng nhỏ hơn là MILNET, dành cho quốc phòng, và nhóm mạng thứ hai vẫn gọi là ARPANET, dành cho nghiên cứu và phát triển. Hai nhóm này vẫn có mối liên hệ trao đổi dữ liệu với nhau qua giao thức TCP/IP và được gọi chung là Enternet.

Mạng Internet đã và đang trở thành phương tiện trao đổi thông tin toàn cầu, là phương thức thông tin nhanh với lưu lượng truyền tải dữ liệu rất lớn. Thông qua Internet mà các nhà nghiên cứu khoa học kỹ thuật, các cơ quan giáo dục đào tạo, các nhà doanh nghiệp... có thể trao đổi thông tin với nhau, hoặc truy cập thông tin

của nhau về các công trình, các lĩnh vực nghiên cứu mới nhất; về các phương pháp, hình thức giáo dục và đào tạo, về các thông tin kinh tế, thị trường giá cả... một cách nhanh chóng, thuận tiện và dễ dàng.

1.6.2 Cấu trúc của mạng Internet

Mạng Internet không phải một mạng đơn mà là bao gồm nhiều mạng con (sub-network) được kết nối với nhau thông qua các cổng (gateway) như trên hình. Thuật ngữ mạng con ở đây mang nghĩa một *đơn vị mạng hoàn chỉnh* trong hệ thống mạng lớn. Mạng con hoàn toàn có thể là một mạng WAN với quy mô quốc gia, và có khả năng hoạt động độc lập với Internet. Do giao thức TCP/IP không phụ thuộc lớp vật lý, các mạng con có thể sử dụng những công nghệ ghép nối khác nhau (như Ọthernet, X.25,...) mà vẫn giao tiếp được với nhau.

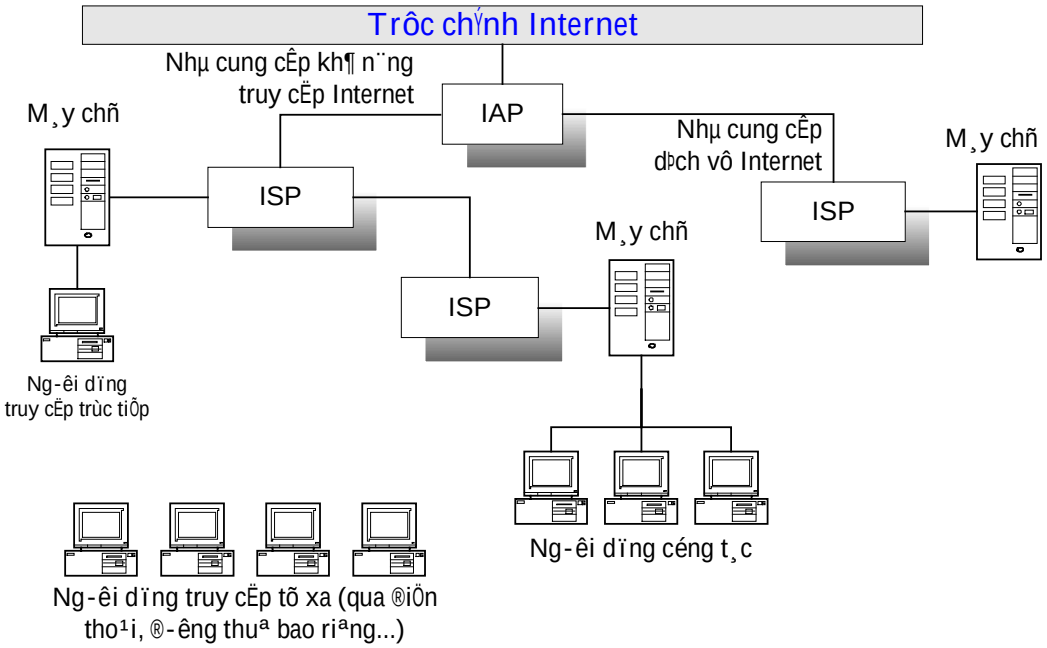


Hình 1-7. Cấu trúc của mạng Internet.

Các cổng được dùng để nối các mạng con tạo thành một mạng lớn.

Có 2 cách kết nối với Internet như sau :

- Máy con nối trong mạng LAN (hay WAN) và mạng này nối với Internet
- Máy con nối đến một trạm cung cấp dịch vụ Internet (Internet Service Provider), thông qua đó kết nối với Internet. Trong hình trên, ta có thể thấy các trạm ISP lại kết nối với Internet thông qua IAP (Internet Access Provider). Một IAP có thể làm luôn chức năng của ISP nhưng ngược lại thì không.



Hình 1-8. Sơ đồ kết nối của các trung tâm cung cấp dịch vụ (ISP)

1.6.3 Các kiến trúc khác

Level	ISO	ARPANET	SNA	DECNET
7	Application	User	End User	Application
6	Presentation	Telnet, FTP	NAU Services	
5	Session	(none)	Data Flow Control	(none)
			Transmision Control	
4	Transport	Host - Host		Network Services
		SRC to DESI - IMP	Path Control	
3	Network			Transport
		IMP - IMP		
2	Datalink		Data Link Control	Data Link Control
1	Physical	Physical	Physical	Physical

ARPANET: Advanced Research Projects Agency
FTP: File Transfer Protocol
SNA: System Network Architecture cña IBM
IMP: Interface Message Procesor
NAU: Network Addressable Unit

Chương 2

TẦNG VẬT LÝ

Nhiệm vụ của tầng vật lý là chuyển các bit tin từ máy này đến máy kia. Tốc độ truyền tin phụ thuộc vào môi trường truyền tin. Tín hiệu truyền có thể ở dạng tương tự (*analog*) hoặc ở dạng số (*digital*). Hướng phát triển hiện nay :

- Truyền tin bằng cáp quang, bằng vệ tinh.
- Hệ thống nối nhanh (Fast - Connect), hệ thống chuyển mạch gói
- Mạng thông tin số đa dịch vụ (Integrated Services Digital Network)

2.1 Môi trường truyền tin

2.1.1 Phương tiện truyền

Mục đích lắp đặt cáp là đảm bảo dung lượng (tốc độ) cần thiết cho các nhu cầu truyền thông trong mạng. Hệ thống cáp cần phải ổn định. Để đạt được mục tiêu này, người quản trị mạng phải cân đối bốn yếu tố sau:

- Tốc độ truyền lớn nhất của hệ thống cáp hiện hành, khả năng nâng cấp.
- Nhu cầu về tốc độ truyền thông trong vòng 5-10 năm tới là bao nhiêu.
- Chọn trong số những loại cáp đang có trên thị trường.
- Chi phí để lắp đặt thêm cáp dự phòng.

Việc kết nối vật lý một máy tính vào mạng được thực hiện bằng cách cắm một card giao tiếp mạng NIC (Network Interface Card) vào khe cắm của máy tính và nối với cáp mạng. Sau khi kết nối vật lý đã hoàn tất, quản lý việc truyền tin giữa các trạm trên mạng tuỳ thuộc vào phần mềm mạng.

NIC sẽ chuyển gói tín hiệu vào mạng LAN, gói tín hiệu được truyền đi như một dòng các bit dữ liệu thể hiện bằng các biến thiên tín hiệu điện. Khi nó chạy trong cáp dùng chung, mọi trạm gắn với cáp đều nhận được tín hiệu này, NIC ở mỗi trạm sẽ kiểm tra địa chỉ đích trong tín hiệu đầu của gói để xác định đúng địa chỉ đến, khi gói tín hiệu đi tới trạm có địa chỉ cần đến, đích ở trạm đó sẽ sao gói tín hiệu rồi lấy dữ liệu ra khỏi khung tin và đưa vào máy tính.

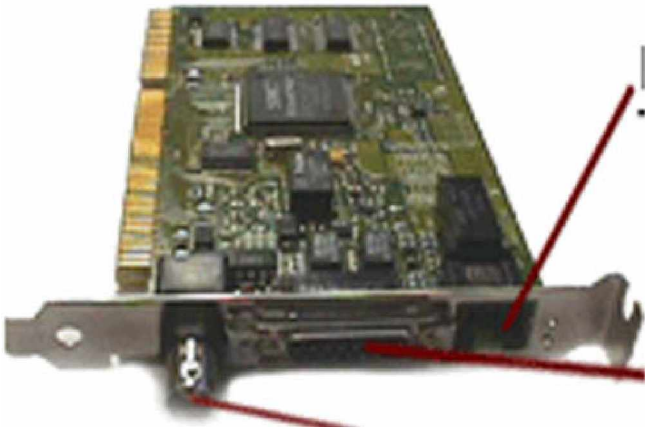
Có hai kỹ thuật truyền tín hiệu đã mã hóa lên mạng : Truyền ở dải tần gốc (baseband) và truyền ở dải tần rộng (broadband).

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 10Mbit/s đến 100Mbit/s và hơn nữa.

Có 3 nhóm cáp chính được dùng để nối hầu hết các mạng :

- Cáp đồng trục (Coaxial)
- Cáp xoắn đôi (Twisted-Pair) : gồm có cáp xoắn đôi trần (Unshielded Twisted-Pair) và cáp xoắn đôi có bọc (Shielded Twisted-Pair).
- Cáp sợi quang (Fiber-Optic)

2.1.1.1 Card mạng



Hình 2-1. Card mạng (NIC)

Card mạng còn được gọi là card giao tiếp mạng NIC (Network Interface Card) được lắp đặt trong mỗi máy tính trong mạng cục bộ, Card này có nhiệm vụ chuyển dữ liệu từ máy tính vào cáp mạng và ngược lại. Quá trình này chính là sự chuyển đổi từ tín hiệu số của máy tính thành các tín hiệu điện hay quang được truyền dẫn trên cáp mạng. Đồng thời nó cũng thực hiện chức năng tổ hợp dữ liệu thành các gói và xác định nguồn và đích của gói.

- Các loại đầu nối cho card mạng :

Một vài loại card mạng có nhiều đầu nối để nối với cáp mạng, để xác định đầu nào dùng ta có thể thay đổi các jump hay công tắc chuyển DIP ngay trên card mạng hoặc sử dụng phần mềm.

- Mạng thin Ethernet sử dụng các đầu nối cáp đồng trục BNC (British Naval Connector)
- Mạng thicknet dùng giắc nối AUI 15 chân để cắm vào đầu DB15 của card mạng.
- Mạng Ethernet twisted-pair (10 Base T) sử dụng đầu nối RJ45.

2.1.1.2 Cáp đồng trục

Cáp đồng trục được chế tạo gồm một dây đồng ở giữa chất cách điện, chung quanh chất cách điện được quán bằng dây bện kim loại dùng làm dây đất. Giữa dây đồng dẫn điện và dây đất có một lớp cách ly, ngoài cùng là một vỏ bọc bảo vệ.

Cáp đồng trục có hai loại : loại nhỏ (Thin) và loại to (Thick). Dây cáp đồng trục loại nhỏ được thiết kế để truyền tin cho băng tần cơ bản (Base Band) hoặc băng tần rộng (broadband). Dây cáp loại to dùng cho đường xa, dây cáp nhỏ dùng cho đường gần, tốc độ truyền tin qua cáp đồng trục có thể đạt tới 35 Mbit/s.

2.1.1.3 Cáp dây xoắn (Twisted Pair)

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 (khoảng 328 feet). Cáp xoắn đôi có hai loại: cáp xoắn đôi không bọc (UTP) và cáp xoắn đôi có bọc (STP).

Cáp xoắn thường được dùng trong hệ thống điện thoại để truyền tín hiệu tương tự (analog) cũng như tín hiệu số (digital). Trong khoảng cách vài km thì không cần bộ khuyếch đại và có tốc độ ở mức megabit/giây.

2.1.1.4 Cáp quang (Fiber Optics)

Khi các tín hiệu số được điều chế thành các tín hiệu xung ánh sánh thì được truyền tải qua cáp quang. Cáp sợi quang bao gồm một sợi thuỷ tinh cực mảnh gọi là lõi (core), được bao bọc bởi một lớp thuỷ tinh đồng tâm gọi là lớp vỏ bọc hay còn gọi là lớp phủ (cladding). Đôi khi các sợi được làm bằng chất dẻo. Chất dẻo dễ lắp đặt hơn nhưng không thể mang xung ánh sáng đi xa như thuỷ tinh.

Mỗi sợi thuỷ tinh chỉ truyền tín hiệu theo một hướng nhất định, do đó cáp có 2 sợi nằm trong vỏ bọc riêng biệt : một sợi truyền và một sợi nhận. Cáp sợi quang có thể truyền tín hiệu đi xa hơn với tốc độ cực nhanh (theo lý thuyết cáp quang có thể truyền tín hiệu với tốc độ tối đa 200.000Mbit/s).

Cáp quang có dải thông lớn hơn cáp đồng, ưu điểm mạnh của cáp quang là khoảng cách truyền dẫn lớn, giá rẻ, dung lượng truyền cao.

2.1.1.5 Vệ tinh thông tin

Vệ tinh truyền thông (communication satellites) nhận thông tin mặt đất, khuyếch đại tín hiệu thu được và phát lại xuống mặt đất ở tần số khác để tránh giao thoa (interference) với tín hiệu thu được. Các vệ tinh có vai trò như những trạm lặp tin giữa các trạm mặt đất với nhau. Một vệ tinh đều phủ sóng rất rộng và có thể có nhiều trạm mặt đất, thường hoạt động ở tần số 12 - 14Ghz. Truyền tin qua vệ tinh có dãi truyền rất rộng, do đó những khoảng cách xa (hàng trăm km) được bảo đảm chất lượng tin. Ngoài ra giá của truyền vệ tinh đang giảm nhanh.

Uỷ ban kỹ thuật điện tử (IEEE) đề nghị dùng các tên sau đây để chỉ 3 loại dây cáp dùng với mạng Ethernet chuẩn 802.3 :

1. Dây cáp đồng trục sợi to (thick coax) gọi là 10BASE5, có tốc độ 10 Mbps, tần số cơ sở, ≤ 500m.

2. Dây cáp đồng trục sợi nhỏ (thin coax) gọi là 10BASE2, có tốc độ 10 Mbps, tần số cơ sở, ≤ 200m.
3. Dây cáp đôi xoắn không vỏ bọc (twisted pair) gọi là 10BASET, có tốc độ 10 Mbps, tần số cơ sở, sử dụng cáp sợi xoắn.
4. Dây cáp quang (Fiber Optic Inter-Repeater Link) gọi là FOIRL .

2.1.2 Các thông số cơ bản của môi trường truyền tin

2.1.2.1 Độ suy giảm

Tín hiệu trên đường dây bị suy giảm trong quá trình truyền tin. Để khắc phục ta dùng các bộ khuyếch đại (amplifiers). Độ suy giảm được tính bằng đơn vị decibel. Nếu điện thế ban đầu là V_1 và sau đó giảm xuống V_2 thì số decibel của độ suy giảm được định nghĩa như sau:

$$\mathrm{S\,(decibel)} = 20\log_{10}\frac{V_1}{V_2}$$

2.1.2.2 Độ nhiễu

Điện từ trường trong môi trường truyền tin gây nhiễu cho các tín hiệu mang thông tin. Để khắc phục ta dùng các bộ lọc nhiễu (*filters*). Để đặc trưng độ nhiễu trên đường dây, ta dùng tỉ số tần số tín hiệu/tạp âm (Signal/Noise - S/N) :

$$SN(decibel) = 10\log_{10}\frac{S}{N} \quad (S: Signal;\ \ N: Noise)$$

2.1.2.3 Tốc độ truyền

$$C = B\log_2\left(1+\frac{S}{N}\right) bit/s$$

Trong đó B là độ rộng dải tần tính bằng Hz. C là tốc độ tính bằng bit/giây (b/s). Nếu mạng điện thoại có dải tần 3000Hz, tỉ số S/N = 20dB thì tốc độ truyền cực đại là :

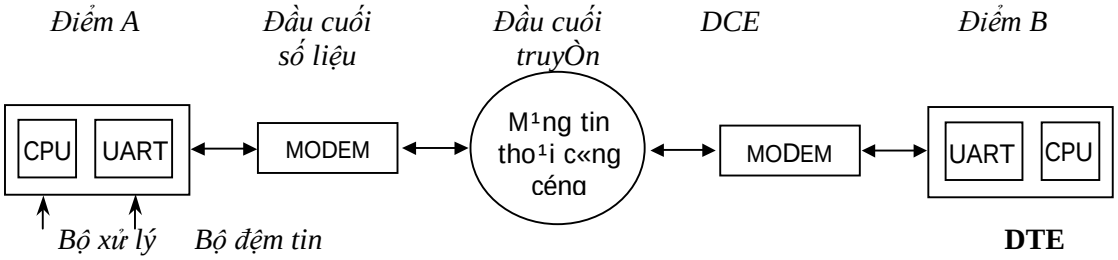
$$\frac{S}{N} = 10\log_{10}\frac{S}{N} = 20 \rightarrow \frac{S}{N} = 100 \qquad C = B\log_2\left(1+\frac{S}{N}\right) = 3000\times\log_2(1+100) = 19963b/s$$

Các tín hiệu trên kênh truyền có thể là tín hiệu tương tự hoặc tín hiệu số và tương ứng sẽ tạo thành kênh tương tự hoặc kênh số.

2.2 Chuẩn giao diện

2.2.1 Modem

Modem là bộ điều chế và giải điều chế biến đổi các tín hiệu số thành các tín hiệu tương tự và ngược lại trên mạng điện thoại.



Hình 2-2. Sơ đồ truyền tin giữa hai điểm A và B.

Tín hiệu số từ máy tính đến modem, được modem biến đổi thành tín hiệu tương tự để có thể đi qua mạng điện thoại. Tín hiệu này đến modem ở điểm B được biến đổi ngược lại thành tín hiệu số đưa vào máy tính ở B.

Các kỹ thuật điều chế cơ bản là điều chế biên độ AM, điều chế tần số FM, điều chế pha PM .

- Điều chế biên độ : Các tín hiệu 1 và 0 được phân biệt bởi biên độ, còn tần số của tín hiệu là giống nhau. Điều chế biên độ để thực hiện nhưng dễ bị nhiễu.
- Điều chế tần số : Các tín hiệu 1 và 0 được phân biệt bởi tần số, còn biên độ các tín hiệu giống nhau.

  Kỹ thuật điều tần phức tạp hơn nhưng tính chống nhiễu cao.
- Điều chế theo pha : Các tín hiệu 1 và 0 được phân biệt bởi các pha của dao động, còn biên độ và tần số của các tín hiệu giống nhau. Điều pha cũng phức tạp nhưng ít bị nhiễu.

Để tăng tốc độ truyền tin người ta kết hợp điều pha với điều biên gọi là điều pha biên.

Hiện nay có rất nhiều loại modem hiện đại từ loại thấp: 300, 600, 1200, 2400 bit/s, đến loại 9600 bit/s. Với tốc độ truyền tương đối cao trên đường truyền băng hẹp (băng thoại) nên đòi hỏi những phương pháp điều biên phức tạp.

Các *phương thức truyền dữ liệu* giữa hai điểm có thể là:

- Một chiều đơn (simplex)
- Hai chiều luân phiên (half - duplex)
- Hai chiều đầy đủ (duplex)

Truyền một chiều đơn chỉ cho phép truyền một hướng. Truyền hai chiều luân phiên cho phép truyền hai hướng, nhưng mỗi thời điểm chỉ có một hướng được truyền, sau đó phải thực hiện chuyển mạch để truyền ngược lại. Truyền hai chiều đầy đủ có thể nhận hoặc phát cùng một lúc. Các modem hiện nay đều có thể hoạt động ở hai chế độ bán song công và song công.

2.2.2 DTE và DCE

Trước khi nghiên cứu các chuẩn cho giao diện tầng Vật lý, chúng ta có hai khái niệm mới : đó là DTE và DCE.

- DTE (Data Terminal Equipment - Đầu cuối số liệu) : là khái niệm được sử dụng để chỉ các máy mà người sử dụng bình thường thao tác trực tiếp lên đó. Các máy này có thể là máy tính hay trạm cuối.
- DCE (Data Communication Equipment - Đầu cuối truyền) : là khái niệm chỉ các thiết bị cuối kênh dữ liệu có chức năng nối các DTE với các đường truyền vật lý và chuyển đổi dữ liệu. DCE có thể là các Modem, Tranducer, Multiplexer...

ISO qui định các chuẩn quy ước phương thức ghép nối giữa đầu cuối số liệu DTE và đầu cuối truyền DCE.

2.2.3 Chuẩn RS-232C

Đầu những năm 50, chuẩn RS-232(Recommended Standard 232C, của EIA) được phát triển để truyền tin giữa các thiết bị đầu cuối dữ liệu. Chuẩn này hiện nay đang được sử dụng, nó chính là các cổng COM1, COM2 trên các máy PC.

- *Phần cơ học :* là một bộ có 25 chân độ rộng tính ở giữa là 47,05mm ± 13 hàng trên đánh số 1 ÷ 13 (trái qua phải) hàng dưới 14 ÷ 25 (trái qua phải).
- *Phần điện :* gồm qui ước logic 1 <-3V và logic 0 >+ 3V.

Tốc độ truyền cho phép 20 *kbps* qua dây cáp 15m (thường là 9,6 *kbps*)

Từ năm 1987, RS-232-C đã được sửa đổi và đặt tên lại là EIA-232-D. Ngoài ra còn có một số chuẩn mở rộng khác như RS-422-A, RS-423-A RS-449, các khuyến nghị loại X của CCITT như X21. . .. Mặc dầu RS-232-C vẫn là chuẩn thông dụng nhất cho giao diện DTE/DCE nhưng các chuẩn mới nói trên được áp dụng phổ biến hiện nay.

Đối với các máy tính, thông thường người ta sử dụng hai cổng COM1, COM2 để *kết nối trực tiếp*. Cổng COM1 có địa chỉ vào/ra là 3F8_3FF hex và ngắt là IRQ4, cổng COM2 có địa chỉ vào/ra là 2F8_2FF hex và ngắt là IRQ3. Các chân cắm của hai cổng cũng được chuẩn hóa để tiện lợi hơn cho việc sử dụng.

Chương 3

TẦNG LIÊN KẾT DỮ LIỆU

3.1 Chức năng

Tầng liên kết dữ liệu thực hiện các công việc chính như sau :

- Định danh các thiết bị trên mạng, cấu hình logic của mạng.
- Điều khiển luồng dữ liệu và việc truy nhập ở tầng vật lý.
- Phát hiện và chỉnh sửa các lỗi xuất hiện trong quá trình truyền dữ liệu.

Chức năng chính của tầng LKDL là tách rời các khung thành các bít để truyền đi và kiến tạo các khung (frames) từ các dòng bít nhận được.

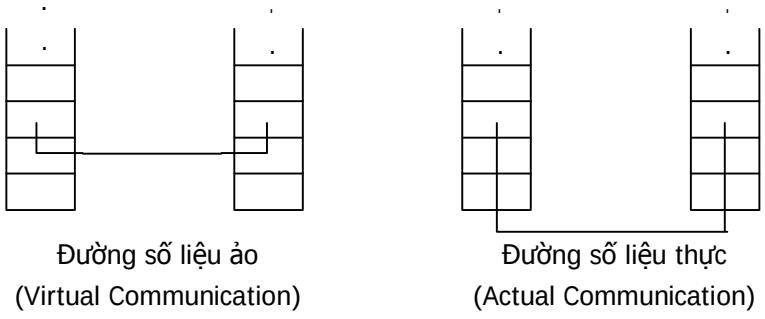
Tầng LKDL nghiên cứu các thuật toán thực hiện thông tin hiệu suất, tin cậy giữa hai máy cạnh nhau ở tầng 2. Đưa ra các thủ tục truyền tin có lưu ý đến lỗi có thể xẩy ra do nhiễu trên đường dây, sự trễ do lan truyền.

Thông thường, tầng LKDL có liên quan đến nhiễu của tín hiệu của phương tiện truyền vật lý, cho dù là truyền qua dây đồng, cáp quang hay truyền thông qua sóng ngắn. Nhiễu là một vấn đề rất thông thường và có thể do rất nhiều nguồn khác nhau, trong đó có cả nhiễu của các tia vũ trụ, nhiễu do tạp âm của khí quyển và từ các nguồn khác nhau.

3.2 Các vấn đề của tầng liên kết dữ liệu

3.2.1 Cung cấp dịch vụ cho tầng mạng

Tầng 2 chuyển dữ liệu từ mức 3 ở máy nguồn tới mức 3 ở máy nhận.



Hình 3-1. đường truyền dữ liệu trong tầng LKDL.

Các dịch vụ tầng 2 có thể là:

1. Dịch vụ không kết nối, không biên nhận (*Unacknowledged Connectionless Service*)
2. Dịch vụ không kết nối, có biên nhận (*Acknowledged Connectionless Service*)
3. Dịch vụ có kết nối (*Connection Oriented Service*)

Dịch vụ kết nối có hướng có 3 giai đoạn: *kết nối, truyền số liệu, tách bỏ liên kết* (kết thúc) : CONNECT, DATA, DISCONNECT. Truyền tin giữa 2 tầng kề nhau dùng các hàm dịch vụ nguyên thủy (request, indication, response và confirm).

Dịch vụ không kết nối được thể hiện bằng một bước duy nhất là truyền tin, không cần thiết lập liên kết logic. Các đơn vị dữ liệu truyền độc lập với nhau.

3.2.2 Khung tin - Nhận biết gói tin

Để cung cấp dịch vụ cho tầng mạng, tầng LKDL phải dùng dịch vụ được cung cấp từ tầng Vật lý. Tầng Vật lý tiếp nhận dòng bít và giao cho nơi nhận. Dòng bit này có thể có lỗi. Tầng LKDL sẽ kiểm tra và nếu cần sẽ sửa lỗi.

Tầng LKDL tách dòng bit thành các khung tin (frame) và tính thông số kiểm tra tổng (checksum) cho mỗi khung tin này, nếu kết quả tính được khác với checksum chứa trong khung tin, nghĩa là có lỗi và khi đó lỗi sẽ được thông báo cho nơi gửi.

Muốn tách các khung tin, có thể chèn các đoạn phân cách (timegaps) vào giữa các khung tin, giống như khoảng trống (*space*) giữa các từ trong văn bản. Nhưng điều này khó thực hiện nên người ta thường dùng các phương pháp sau :

- Đếm số ký tự : Hiện nay ít được dùng, vì từ đếm cũng bị lỗi khi truyền.
- Dùng ký tự bắt đầu (STX) và kết thúc (ETX) với ký tự đệm (DLE).
- Dùng các cờ (*flags*) đánh dấu bắt đầu và kết thúc với các bit đệm.

3.2.3 Kiểm tra lỗi

Các cách để kiểm tra lỗi trong quá trình truyền :

- Dùng thông số trả lời có biên nhận (ACK) hoặc không biên nhận (NAK) để biết đã nhận đúng bản tin hay phải phát lại.
- Dùng bộ định thời gian, nếu quá thời gian quy định không có trả lời nghĩa là bản tin chưa nhận được.
- Dùng phương pháp đánh số thứ tự các khung tin (frame) được gửi đi.

Quá trình kiểm tra lỗi đồng thời với quản lý thời gian và số thứ tự của các khung tin nhằm bảo đảm mỗi khung tin chỉ nhận được một lần duy nhất. Đây là chức năng quan trọng của tầng LKDL.

3.2.4 Điều khiển luồng dữ liệu

Trong quá trình truyền dữ liệu, nếu tốc độ bên phát nhanh hơn bên thu thì xảy ra hiện tượng mất tin do không nhận kịp. Vì vậy cần phải điều khiển luồng truyền

(*flow control*) để quá trình thu phát được phối hợp nhịp nhàng và đồng bộ với nhau. Chức năng có tại một vài cấp giao thức, kể cả tầng con LLC.

Các giao thức phải chứa các quy tắc xác định rõ khi nào nơi gởi có thể phát các khung tin kế tiếp.

3.2.5 Quản lý liên kết

Một chức năng khác của tầng LKDL là quản lý các kết nối như tách, nối, đánh số khung tin, bắt đầu lại khi lỗi, quản lý các thiết bị đầu cuối thứ cấp hoặc sơ cấp bằng khung tin thăm dò (poll).

3.2.6 Nén dữ liệu khi truyền

Nén dữ liệu là một vấn đề quan trọng đơn vị việc truyền dữ liệu trên mạng. Về cơ bản, nén dữ liệu là ép chúng lại để đỡ tốn chỗ khi lưu trữ trên đĩa và đỡ tốn thời gian khi truyền trên đường dây. Thực tế, các dữ liệu số chứa nhiều đoạn lặp đi lặp lại, nén dữ liệu sẽ thay thế các thông tin lặp lại bằng một ký hiệu hoặc một đoạn mã để rút ngắn độ dài của tập tin. Các kỹ thuật nén dữ liệu cơ sở bao gồm :

- *Null compression* : Thay thế một dãy các dấu cách bằng một mã nén và một giá trị số lượng các dấu cách.
- *Run-length compression* : Mở rộng kỹ thuật trên bằng cách nén bất kỳ một dãy nào có từ 4 ký tự lặp. Các ký tự này được thay thế bằng một mã nén, là một trong các ký tự này, và một giá trị bằng đúng số lần lặp.
- *Keyword encoding* : Tạo ra một bảng mã cho các từ hoặc các cặp ký tự thường xuyên xuất hiện và thay thế.
- *Phương pháp thống kê Huffman* : Kỹ thuật nén này giả thiết rằng sự phân bố của các ký tự trong dữ liệu là không đồng nhất. Tức là một số ký tự xuất hiện nhiều hơn các ký tự khác. Ký tự nào càng xuất hiện nhiều thì càng ít tốn bit để mã hóa nó. Một bảng được tạo ra để ghi lại lược đồ mã hóa và bảng này có thể chuyển cho modem nhận để nó biến đổi trở lại các ký tự đã mã hóa.
- Ngoài ra còn một thuật toán nén nữa được gọi là nén ngẫu nhiên. Thuật toán này được sử dụng trong một chuẩn nén dữ liệu V.24bits

3.3 Phát hiện và hiệu chỉnh lỗi

Trong khi truyền đi một byte trong hệ thống máy tính thì khả năng xảy ra một lỗi do hỏng hóc ở phần nào đó hoặc do nhiễu gây nên là khá lớn. Các kênh vào-ra thường xẩy ra nhiều lỗi, đặc biệt là khi truyền số liệu. Phần lớn các hệ thống đều có các phương pháp phát hiện và sau đó sửa lỗi. Quá trình sửa lỗi thường khó hơn rất nhiều so với phát hiện lỗi. Có thể chia phương pháp xử lý lỗi ra làm hai nhóm:

- Phát hiện lỗi và thông báo cho bên phát biết để phát lại tin.
- Phát hiện lỗi và tự sửa.

3.3.1 Phương pháp bit chẵn lẻ (Parity)

Đây là phương pháp thường dùng nhất để phát hiện lỗi. Bằng cách thêm 1 bit (được gọi là bit chẵn lẻ) vào từ nhị phân phụ thuộc vào tổng số các bit 1 trong một từ là một số chẵn hay lẻ, và nhờ vào phép toán logic XOR, ta sẽ biết được bit thêm vào đó là bit chẵn hay bit lẻ.

Mạch kiểm tra sẽ xác định các số bit 1 có đúng tính chẵn lẻ hay không. Phương pháp tương đối đơn giản và có hai cách như sau :

- Kiểm tra ngang (VRC - Vertical Redundancy Checking) : Thêm một bit chẵn lẻ vào mỗi byte để phát hiện lỗi. Cách này làm mất đi khoảng 12,5% dung lượng bản tin. Để khắc phục ta có thể dùng phép kiểm tra tổng các byte.
- Kiểm tra dọc (LRC - Longitudinal Redundancy Checking) : lỗi được phát hiện trong các khối byte thay cho việc tìm lỗi trong từng byte. Trong phương pháp này người ta thêm mỗi khối *1 byte ở cuối*, byte này mang các thông tin về tính chất đặc thù của khối (Characteristic Redundancy Checking - CRC). Byte này đơn giản có thể tính bằng phép logic XOR của tất cả các byte trong khối hoặc tính theo đa thức chuẩn để được FCS.

  Ví dụ :

Vị trí bit trong ký tự	Khối ký tự truyền đi					LRC
	A	S	C	I	I	
0	1	1	1	1	1	1
1	0	0	0	0	0	0
2	0	1	0	0	0	1
3	0	0	0	1	1	0
4	0	0	0	0	0	0
5	0	1	1	0	0	0
6	1	1	1	1	1	1
VRC	0	0	1	1	1	1

Kiểm soát lỗi 2 chiều : VRC-LRC.

Bên nhận sẽ kiểm tra parity theo cả hai chiều để phát hiện và định vị lỗi cho từng ký tự. ($1 \oplus 1 = 0$ $0 \oplus 0 = 0$ $1 \oplus 0 = 0$ $0 \oplus 1 = 1$)

3.3.2 Tính theo đa thức chuẩn

Cách tính check sum như sau :

- Giả sử ta nhận được bản tin M(x).

- Nếu đa thức chuẩn G(x) có bậc là r, ta bổ sung thêm r bit 0 vào cuối bản tin và được m+r bit tương ứng đa thức xrM(x).
- Chia xrM(x) theo module 2 cho G(x). Kết quả ta được số dư T(x) là checksum được phát đi.

Các đa thức chuẩn thường được dùng để tính biến kiểm tra tổng là :

CRC-12 = $x^{12} + x^{11} + x^3 + x^2 + x + 1$ (dùng cho ký tự 6 bit)

CRC-16 = $x^{16} + x^{15} + x^2 + 1$ (dùng cho ký tự 8 bit)

CRC-CCTTT= $x^{16} + x^{12} x^5 + 1$ (dùng cho ký tự 8 bit)

Ví dụ Khung tin ban đầu 1101011011, G(x) = $x^4 + x + 1$, vậy r = 4, chuỗi bit thêm : 10011. Ta có xrM(x) = 1101011011 0000. Chia xrM(x) theo module 2 cho G(x), ta được thông số kiểm tra tổng T(x) = 1110

```
   11010'1'1011 0'00'0'
⊕  10011
   01001 1
    10011
     0000010110
           10011
           0010100
              10011
               001110 → Số dư là 1110
```

Khung tin được truyền đi: 1101011011 1110

3.3.3 Mã sửa sai

Để sửa sai một bit, ta dùng tập mã Hamming dựa trên các "bit chẵn lẻ" được rải vào các bit số liệu trong từng byte theo nguyên lý cân bằng chẵn lẻ để chỉ ra các bit lỗi.

Nếu trong bản tin có k bit và số "bit chẵn lẻ" là r, thì số bit tin và "bit chẵn lẻ" phát đi sẽ là n=k+r. r bit kiểm tra luôn các vị trí 1, 2, 4, 8,..., 2r-1 và được tạo bởi cộng module 2 giá trị nhị phân của các vị trí có bit '1' của từ mã. Vì các bit kiểm tra chiếm vị trí 2^i với i = 0, 1, 2,..., r-1 nên độ dài cực đại của các từ mã Hamming là n $\leq 2^r - 1$ và từ dãy số cực đại của các bit tin được bảo vệ là : k $\leq (2^r - 1 - r)$. Từ đây ta xác định được r.

Ví dụ: Bản tin 11 bit (10101011001) được bảo vệ bởi mã Hamming.

Từ điều kiện 11 ≤ 2' - 1-r, ta cần 4 bit kiểm tra (r =4) để tạo mã Hamming (n=11+4=15)

1	0	1	0	1	0	1	C	1	0	0	C	1	C	C
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Các bít kiểm tra C được tính như sau:

VÞ trÝ bit 1 Gi¸ trÞ nhÞ ph©n

15	1111
13	1101
11	1011
9	1001
7	0111
	1101

-> TËp m· c¸c bit kiÓm tra Hamming: 0100

Tõ m· Hamming: 10101010010011100

NÕu b¶n tin nhËn ®-îc lµ:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1	0	1	0	0	0	1	0	1	0	0	1	1	0	0

↑ bit error

Vị trí bit 1	Giá trị nhị phân
15	1111
13	1101
9	1001
7	0111
4	0100
3	0011
	1011 (11)

→ Vb trÝ sai ë bit 11

3.4 Thủ tục liên kết dữ liệu cơ bản

Để truyền tin có độ tin cậy cao ta dùng dịch vụ liên kết (Connection Oriented Service).

Ví dụ máy A gửi số liệu cho máy B, khi tầng 2 đã được nối, số liệu từ tầng 3 máy A chuyển xuống tầng 2 nhờ chương trình con “FromNetworkLayer”. Tầng 2 bổ sung phần đầu thông tin điều khiển và tính cờ kiểm tra tổng (FCS).

Khung tin (Frame) =	Đầu tin (Header)	Thông tin (Information)	FCS

Khung tin được phát sang tầng 2 máy B nhờ chương trình con ToPhysicalLayer.

Máy B đợi tin bằng chương trình con Procedure CallWait(Event). Khi khung tin tới bên nhận, máy B tính cờ kiểm tra tổng, nếu không đúng cờ sẽ báo event = CKsumErr, nếu khung tin đúng nó báo event=FrameArrival và thu nhận khung tin từ tầng Vật lý nhờ chương trình con FromPhysicalLayer.

Sau đó đầu tin chứa các thông tin điều khiển (*header*) sẽ được kiểm tra và nếu tất cả đều đúng cả, phần số liệu được chuyển lên tầng 3 nhờ chương trình con ToNetworkLayer.

- Giao thức đơn công với kênh không lỗi và không chờ : Trong giao thức này do tin chỉ truyền theo một hướng, đường kênh không có lỗi nên số liệu luôn sẵn sàng không phải chờ.
- Giao thức đơn công với kênh không lỗi và phải đợi : Bên thu bộ nhớ hạn chế và tốc độ vật lý hữu hạn, do đó bên phát phải chờ.

3.4.1 Giao thức đơn công với kênh có lỗi

- *Bên nhận*

Khi nào đường kênh có lỗi, bên nhận sẽ chỉ gửi tín hiệu biên nhận nếu gói tin nhận được là đúng, nếu gói tin nhận được là sai thì sẽ bị bỏ đi. Quá thời hạn qui định, bên phát sẽ gửi lại gói tin. Quá trình này lặp lại cho đến khi nhận được gói tin đúng. Trong trường hợp này, tầng 3 ở máy B không biết được gói tin bị mất hay nhận hai lần, tầng 2 phải nhận biết được điều này.

Có thể xảy ra các trường hợp :

- Tầng 3 ở máy A gửi gói tin X xuống tầng 2 của nó và phát đi.
- Máy B nhận được và trả lời bằng tín hiệu biên nhận ACK.
- Tín hiệu biên nhận bị mất trên đường đi.
- Quá thời gian qui định mà máy A không nhận được tín hiệu biên nhận, nó sẽ phát lại gói tin X. Dẫn đến máy B nhận được hai gói tin X

Để giải quyết vấn đề này người ta đánh dấu gói tin gửi đi và bên nhận gửi tín hiệu cho biết đã nhận gói tin số mấy.

- *Bên phát*

Bên phát sau khi phát gói tin, có 3 khả năng xảy ra: nhận được tín hiệu biên nhận đúng, tín hiệu biên nhận bị mất hoặc quá thời gian mà chưa nhận được trả lời. Nếu tín hiệu biên nhận đúng, máy A nhận tiếp gói tin từ tầng mạng đặt vào vùng đệm (*buffer*), xoá gói tin trước, tăng số thứ tự gói tin phát. Nếu tín hiệu biên nhận bị mất hoặc đã quá thời gian mà chưa nhận được thì phát lại gói tin với số thứ tự gói tin không thay đổi.

Bên nhận nếu nhận đúng gói tin thì tiếp nhận và chuyển đến tầng mạng và phát tín hiệu biên nhận. Nếu gói tin sai hoặc nhận 2 lần thì không được chuyển lên tầng mạng.

3.5 Điều khiển dòng truyền

Để tận dụng đường dây, các tín hiệu biên nhận (ACK) được ghép cùng với gói tin. Khi gói tin đến, thay cho việc trả lời ngay tín hiệu biên nhận, bên thu nhận tiếp gói tin từ tầng mạng để ghép cùng cùng tín hiệu biên nhận và gửi trả lời. Kỹ thuật này được gọi là Piggybacking (ghép thêm).

Ưu điểm của phương pháp này là tận dụng đường kênh. Nếu quá thời gian (vài μs) mà không có gói tin mới thì bên thu cũng phải trả lời tín hiệu biên nhận để bên phát không phải phát lại gói tin cũ.

Để tận dụng đường kênh, bên phát và bên thu phải đồng bộ để bên thu kịp nhận các gói tin và bên phát cũng không lãng phí đường truyền, người ta dùng cơ chế cửa sổ trượt (sliding windows). Cửa sổ mở to thì số gói tin đưa lên đường kênh nhiều hơn(tốc độ nhanh), cửa sổ mở bé thì số gói tin đưa lên đường kênh ít lại (tốc độ chậm lại). Tương tự như cửa chắn đập nước.

3.5.1 Cơ chế cửa sổ

Người ta dùng số bit để đặc trưng cho độ rộng cực đại của cửa sổ. Trong thủ tục này, mỗi gói tin đi sẽ được đánh số từ 0 đến Max (Max là 2^n - 1) thông qua một dãy gồm các số 0, 1. Chẳng hạn cửa sổ 3 bit sẽ quản lý các gói tin có số từ 0 $\rightarrow$7. Ta có thể dùng *n* tuỳ ý.

Danh sách các gói tin gửi đi giữ trong cửa sổ phát. Danh sách các gói tin nhận được giữ trong cửa sổ nhận. Cửa sổ phát và nhận không bắt buộc phải có kích thước, giới hạn trên và dưới giống nhau.

Mặc dầu thủ tục này cho phép tầng liên kết dữ liệu linh hoạt hơn về thứ tự gởi, nhận gói tin nhưng nó yêu cầu phải đảm bảo tầng mạng đích ở bên nhận có cùng thứ tự với tầng mạng nguồn ở bên gửi.

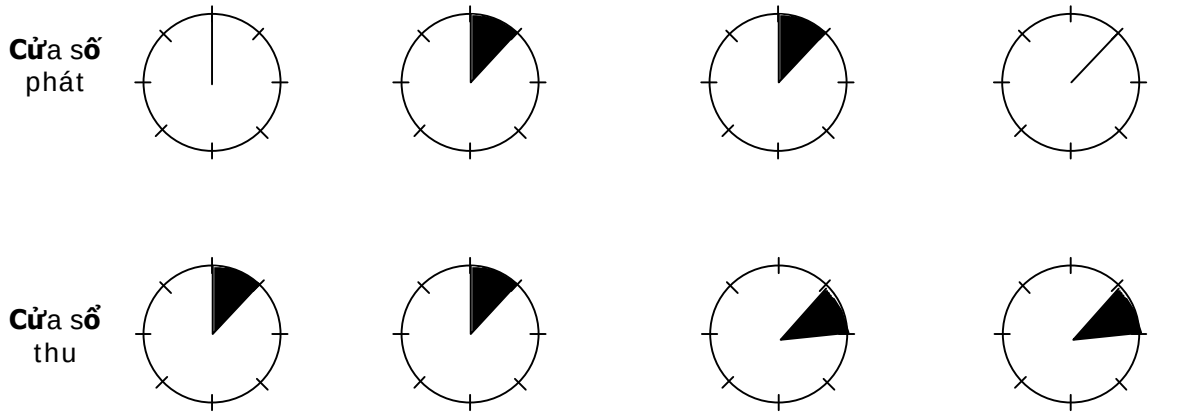
- *Cửa sổ bên phát*

Trong cửa sổ bên phát đặt các gói tin gửi đi nhưng chưa nhận được tín hiệu biên nhận. Khi nhận được gói tin mới đến từ tầng mạng để phát đi, biên trên cửa sổ tăng 1, và khi có tín hiệu biên nhận, biên dưới của cửa sổ tăng 1. Bên phát luôn giữ trong bộ nhớ các gói tin đã phát đi nhưng chưa nhận được tín hiệu biên nhận vì có thể phát lại. Như vậy nếu Max bằng n thì bên phát cần n vùng đệm để giữ các gói tin đã phát đi nhưng chưa nhận được trả lời. Nếu cửa sổ đã tới Max thì tầng liên kết dữ liệu bên phát ngừng nhận tin từ tầng 3 cho đến khi có bộ đệm tự do.

- *Cửa sổ bên nhận*

Cửa sổ bên nhận chứa các gói tin được chuyển đến. Khi gói tin có số thứ tự trùng với biên dưới của cửa sổ được nhận, cửa sổ chuyển tin lên tầng ba, phát tín hiệu biên nhận và quay một đơn vị. Không như cửa bên phát, cửa sổ bên nhận luôn duy trì cùng một kích thước. Khi kích thước cửa sổ = 1, tầng 2 nhận gói tin theo thứ tự. Nhưng nếu kích thước cửa sổ lớn hơn thì không phải như vậy.

Hoạt động của cửa sổ có kich thước là 3 bit với độ trượt 1 bit như sau :

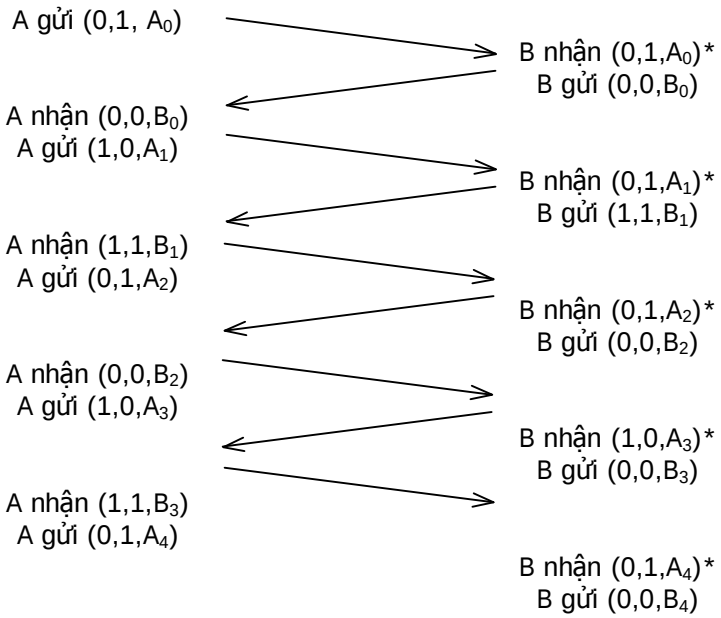


Hình 3-2. điều khiển dòng truyền theo cơ chế cửa sổ.

3.5.2 Trao đổi bản tin với cửa sổ 1 bit

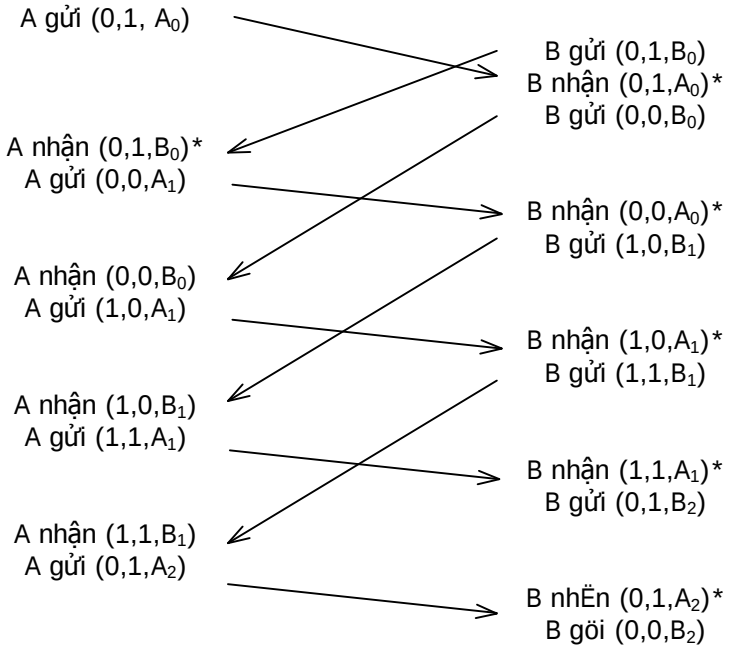
Bản tin gồn có gói tin với phần điều khiển (Header). Phần điều khiển gồm có số gói tin, số thứ tự phát *seq*, số gói tin, số thứ tự nhận là *ack*.

Trong trường hợp bình thường máy A gửi trước như sau :



Hình 3-3. Trao đổi bản tin với cửa sổ 1 bit bình thường.

Trong trường hợp bất thường máy A và B cùng gửi như sau :



Hình 3-4. Trao đổi bản tin với cửa sổ 1 bit bất thường.

Máy A ở tầng 2 nhận gói tin ở tầng 3, tạo bản tin và gửi đi. Khi bản tin này đến tầng 2 máy B, nó sẽ được kiểm tra xem có bị lặp lại không. Nếu đúng là bản tin đang mong đợi thì nó được chuyển lên tầng 3 và cửa sổ nhận dịch đi 1 nấc.

Vùng tín hiệu biên nhận chứa số bản tin cuối cùng đã được nhận mà không có lỗi. Nếu số này trùng với số bản tin vừa gửi. Bên phát sẽ lấy bản tin tiếp theo từ tầng mạng. Nếu số không đúng nó phải gửi lại bản tin cũ.

3.5.3 Vận chuyển liên tục

Thực tế cho ta thấy thời gian từ lúc phát gói tin đến lúc nhận trả lời biên nhận ACK là không đáng kể. Khi đó, nếu đường kênh vệ tinh có tốc độ 50Kbp/s với trễ lan truyền 500 ms, ta dùng thủ tục điều khiển dòng truyền gửi gói tin là 1000 bit qua vệ tinh. Thời gian phát gói tin là 20ms, vậy sau 520ms mới nhận được tín hiệu biên nhận trả lời. Như vậy bên phát phải chờ đến 96% thời gian (500/520), chỉ có 4% độ rộng băng được dùng đến.

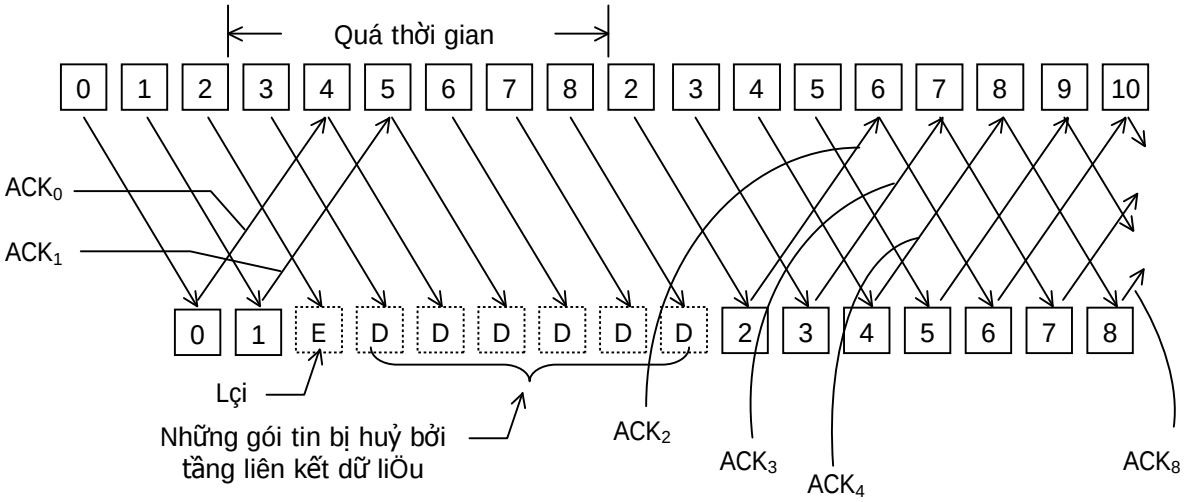
Để nâng cao hiệu suất đường truyền ta không chờ tín hiệu biên nhận mà cứ phát tiếp. Ví dụ, với thời gian phát 20ms cho một gói tin, ta sẽ gửi liên tục 26 gói tin. Như thế khi gửi hết 26 gói tin thì mất khoảng thời gian là 520 ms, đúng lúc tín hiệu biên nhận cho gói tin 0 cũng vừa đến. Kỹ thuật này gọi là Pipe-Lining (vận chuyển liên tục).

Khi có gói tin ở đoạn giữa bị hỏng thì làm thế nào ?, có bỏ những gói tin đúng đi tiếp sau nó không?. Có hai phương pháp như sau :

- Phát lại tất cả các gói tin kể từ gói tin hỏng (*go back n*)
- Phát lại chỉ riêng gói tin bị hỏng, còn gọi là phát có chọn lọc .
- Phát lại từ gói tin hỏng

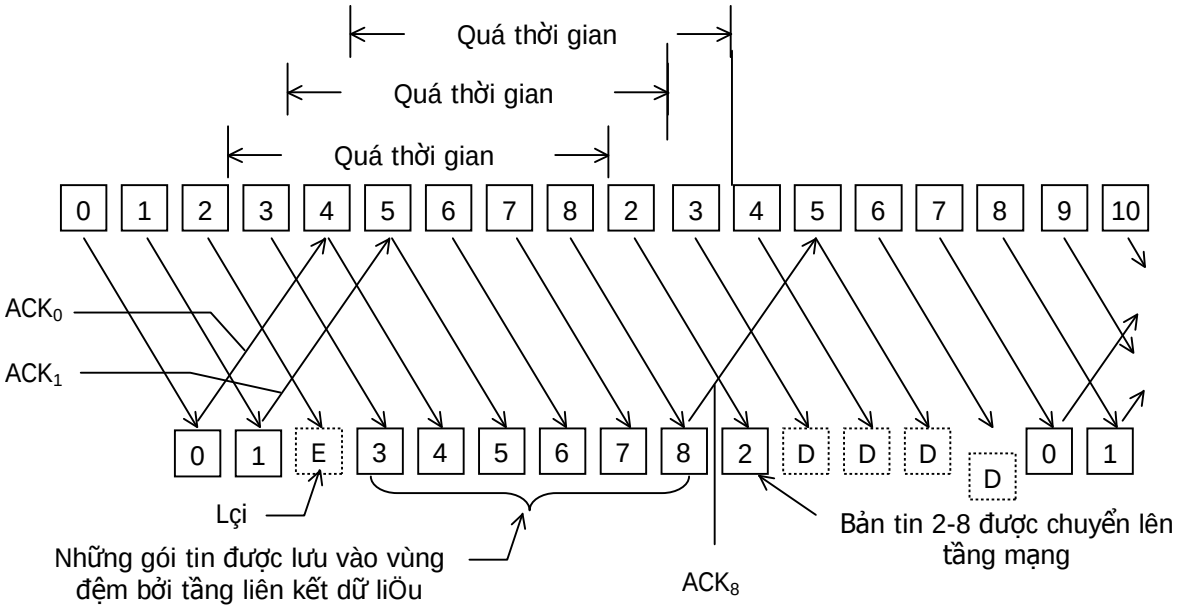
Trong trường hợp này, bên thu huỷ bỏ các gói tin tiếp theo gói tin bị hỏng. Bên phát phát lại tất cả các gói tin chưa được biên nhận bắt đầu từ gói tin bị hỏng.

Phương pháp này lãng phí đường truyền vì phải phát lại nhiều gói tin.



Hình 3-5. Cơ chế vận chuyển liên tục.

3.5.3.1 Phát lại có chọn läc



Hình 3-6. Cơ chế phát bản tin có chọn lọc.

Trong phương pháp này, các gói tin nhận được có thể không theo thứ tự nhưng sẽ được sắp xếp lại để chuyển lên tầng mạng theo đúng thứ tự. Khi có gói tin bị lỗi, bên thu tiếp tục thu các gói tin đúng sau gói tin hỏng ở tầng 2. Bên phát chỉ phát lại

gói tin hỏng. Phương pháp này ứng với cửa sổ bên thu lớn hơn 1 và đòi hỏi bộ nhớ lớn để giữ các gói tin sau gói tin hỏng.

3.6 Các giao thức của tầng Liên kết dữ liệu

Tầng LKDL cung cấp các phương tiện để truyền thông tin qua liên kết vật lý đảm bảo tin cậy thông qua các cơ chế đồng bộ hóa, kiểm soát lỗi và kiểm soát luồng dữ liệu. Các giao thức được xây dựng cho tầng LKDL (DLP - Data Link Protocol) được phân thành hai loại :

1. Giao thức dị bộ (asynchronous DLP) : Cho phép một ký tự dữ liệu được truyền đi bất kỳ lúc nào mà không cần quan tâm đến các tính hiệu đồng bộ trước đó.
2. Giao thức đồng bộ (synchronous DLP) : Chèn các ký tự điều khiển hoặc các cờ giữa các dữ liệu của người sử dụng để báo cho bên nhận. Có hai nhóm giao thức đồng bộ :
    a. Đồng bộ hướng ký tự (character -oriented)
    b. Đồng bộ hướng bit (bit - oriented)

Các hệ thống truyền thông đòi hỏi hai mức đồng bộ hóa :

- Mức vật lý : để giữ đồng bộ giữa các đồng hồ người gửi và người nhận
- Mức LKDL : để phân biệt dữ liệu của người sử dụng với các 'cờ'và các vùng thông tin điều khiển khác

Sau đây ta xét hai loại giao thức đồng bộ là giao thức truyền tin đồng bộ nhị phân BSC (Binary Synchronous Control) và giao thức điều khiển liên kết dữ liệu mức cao HDLC (Highlevel Data Link Control).

3.6.1 Giao thức BSC

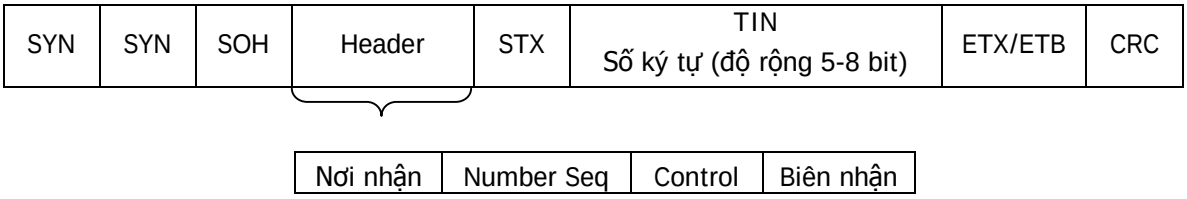
Đây là giao thức *hướng ký tự* (COP - Character Oriented Protocol) được xây dựng dựa trên các ký tự đặc biệt của một bộ mã chuẩn nào đó (như ASCII hoặc EBCDIC) hoạt động theo phương thức hai chiều luân phiên.

3.6.1.1 Tập ký tự điều khiển

ENQ (05): Enquire - Yêu cầu trả lời từ một trạm xa
ACK (06): Acknowledgement - Thông báo tiếp nhận tốt thông tin
NAK (15): Negative ACK - Thông báo tiếp nhận không tốt thông tin
STX (02): Start of text - Kết thúc phần Header và bắt đầu phần dữ liệu
ETX (03): End of text - Kết thúc phần dữ liệu
ETB (17): End of transmission block - Kết thúc đoạn tin (khối dữ liệu)

SOH (01): Start of heading - Bắt đầu phần header của bản tin
EOT (04): End of transmission - Kết thúc quá trình truyền tin và giải phóng liên kết
DLE (10): Data Link Escape - Để thay đổi ý nghĩa của các ký tự điều khiển truyền tin khác
SYN (16): Synchronous - Ký tự đồng bộ bản tin dùng để duy trì đồng bộ giữa 2 bên

3.6.1.2 Khuôn dạng tổng quát bản tin của giao thức BSC

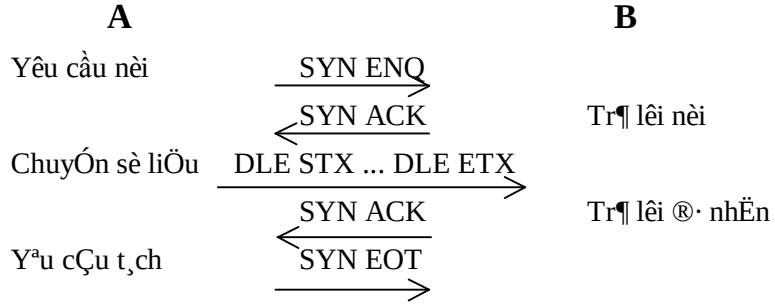


Để thông suốt bản tin, có thể dùng thêm các byte đệm :

SYN	SYN		DLE	DLE		DLE	ETX/ETB	CRC

Khi phát nếu ký tự phát trùng với DLE thì ta chèn thêm DLE. Khi thu, DLE chèn thêm sẽ được khử bỏ.

Ví dụ về thủ tục BCS



3.6.2 Giao thức HDLC

HDLC là giao thức hướng bit (Bit Oriented Protocol - BOP) có các phần tử của giao thức (đơn vị dữ liệu, thủ tục) được xây dựng từ các cấu trúc nhị phân (xâu bit) và khi nhận dữ liệu sẽ được tiếp nhận lần lượt từng bit một.

Đây là giao thức có vị trí quan trọng nhất, được ISO phát triển để sử dụng trong cả hai trường hợp : điểm - điểm và nhiều điểm, cho phép truyền thông hai chiều đồng thời.

3.6.2.1 Khuôn dạng tổng quát bản tin của giao thức HDLC

<--- *Hướng truyền*

	8 bits	8 bits	128-1024 bytes		
Flag	HEADER		INFORMATION	FCS	Flag
01111110	Address	Control	Số các bit	(16 bit)	01111110

Trong đó :

- *Flag* (01111110): là cờ dùng để nhận biết điểm bắt đầu và kết thúc bản tin.

Để tránh sự xuất hiện của mã cờ trong nội dung của bản tin, người ta cài đặt cơ chế '*cứng*' có các chức năng sau :

- Khi truyền tin cứ sau năm bit 1 liên tiếp thì thêm một bit 0 để không nhầm với Flag : 01101111111110010

  01101111011110010

  ↑ *bit chèn thêm (khi thu thì bit này sẽ được khử bỏ)*
- Khi nhận tin, nếu phát hiện có bit 0 sau 5 bit 1 liên tiếp thì tự động loại bỏ bít 0 đó đi.

- *Address* : vùng chứa địa chỉ trạm đích của khung tin.
- *Information* : vùng ghi thông tin truyền đi, có kích thước không xác định.
- *FCS (Frame Check Sequence)* : vùng để ghi mã kiểm soát lỗi (checksum) cho nội dung khung tin, dùng phương pháp CRC với đa thức sinh là CRC-CCITT = $x^{16} + x^{12} + x^5 + 1$
- *Control* : vùng định danh cho các loại khung tin khác nhau của HDLC, có ba dạng như sau :

  Dạng I : hiệu lực truyền tin tức - Information
  Dạng S : hiệu lực điều hành sự nối - Supervisor
  Dạng N : chức năng phụ của điều hành nối – Unnumbered

3.6.2.2 Phương thức trao đổi thông tin

Giao thức HDLC có 3 phương thức trao đổi thông tin chính, ứng với mỗi phướng thức có các giao thức khung tin tương ứng là SNRM, SARM hoặc SABM :

- *Phương thức trả lời chuẩn SNRM (Set Normal Response Mode):* Được sử dụng trong trường hợp cấu hình không cân bằng, có một trạm điều khiển chung (master), các trạm còn lại (slave) chỉ có thể truyền tin khi trạm chủ cho phép.
- *Phương thức trả lời dị bộ SARM (Set Asynchronous Response Mode):* Cũng được sử dụng trong trường hợp cấu hình không cân bằng như trường hợp trên, nhưng các trạm slave được phép truyền tin mà không cần sự cho phép của trạm master. Phương thức này được sử dụng trong trường hợp điểm-điểm với liên kết 2 chiều, cho phép trạm slave gửi các gói tin (frame) không đồng bộ với trạm master.

- *Phương thức trả lời dị bộ cân bằng SABM* (Set Asynchronous Balanced Mode) : Sử dụng trong trường hợp điểm-điểm, liên kết 2 chiều. Trong đó các trạm đều có vai trò tương đương.

3.6.2.3 Các giao thức dẫn xuất của HDLC

- LAP (Link Access Procedure) : tương ứng với phương thức trả lời dị bộ (ARM).
- LAPB (Link Access Protocol-Balanced) : tương ứng với phương thức trả lời dị bộ cân bằng (ABM), được dùng hầu hết trong các mạng truyền dữ liệu công cộng X25.
- LAP-D (Link Access Procedure, D Channel) : Được xây dựng từ LAP-B và được dùng như giao thức liên kết dữ liệu cho các mạng ISDN
- SDLC, ADCCP

3.6.2.4 So sánh BOP và COP

- BOP nhận lần lượt từng bit một, do đó mềm dẻo, dễ dàng tương thích với các hệ khác nhau.
- BOP có overhead (phụ trội) ngắn, số bít bổ sung và số tín hiệu điều khiển ít do đó có tốc độ cao.
- Thủ tục điều khiển trên bit nhị phân đảm bảo không phụ thuộc mã dùng. Cách giải quyết này mềm dẻo và cho phép giải quyết vô số yêu cầu khác.
- Thủ tục HDLC được coi là chuẩn quốc tế và sẽ thông trị trong thời gian tới, nó thích ứng với các hệ thống phức tạp. Đối với các thiết bị ít phức tạp có thể dùng HDLC đơn giản hoá để đảm bảo sự tương thích với HDLC và sự phát triển mở rộng hệ thống sau này.

BÀI TẬP

1. Tìm hiểu thêm về chuẩn giao tiếp RC232 và các chuẩn khác được phát tiển từ chuẩn này.
2. Tìm hiểu các chuẩn mở rộng của giao thức HDLC.

-

Chương 4

MẠNG CỤC BỘ

Mạng cục bộ (LAN) là hệ truyền thông tốc độ cao được thiết kế để kết nối các máy tính và các thiết bị xử lý dữ liệu khác cùng hoạt động với nhau trong một khu vực địa lý nhỏ như ở một tầng của toà nhà, hoặc trong một toà nhà.... (100m đến vài km), có tốc độ truyền dữ liệu cao (có thể tới 100Mbps), tỷ lệ sai số dữ liệu nhỏ (10^{-8} ->10^{-11}). Một số mạng LAN có thể kết nối lại với nhau trong một khu làm việc.

Mạng LAN thường bao gồm một hoặc một số máy chủ (file server, host), còn gọi là máy phục vụ) và một số máy tính khác gọi là trạm làm việc (Workstations, Client) hoặc còn gọi là nút mạng (Network Node) - một hoặc một số máy tính cùng nối vào một thiết bị nút.

4.1 Các cấu hình của mạng LAN

Cấu hình (topology) của mạng là cấu trúc hình học không gian mà thực chất là cách bố trí phần tử của mạng cũng như cách nối giữa chúng với nhau. Thông thường mạng có 3 dạng cấu trúc là: Mạng dạng hình sao (Star Topology), mạng dạng vòng (Ring Topology) và mạng dạng tuyến (Linear Bus Topology). Ngoài 3 dạng cấu hình kể trên còn có một số dạng khác biến tướng từ 3 dạng này như mạng dạng cây, mạng dạng hình sao - vòng, mạng hỗn hợp,v.v....

4.1.1 Mạng dạng hình sao (Star Topology)

Mạng dạng hình sao bao gồm một trung tâm và các nút thông tin. Các nút thông tin là các trạm đầu cuối, các máy tính và các thiết bị khác của mạng. Trung tâm của mạng điều phối mọi hoạt động trong mạng với các chức năng cơ bản là:

- Xác định cặp địa chỉ gửi và nhận được phép chiếm tuyến thông tin và liên lạc với nhau.
- Cho phép theo dõi và xử lý sai trong quá trình trao đổi thông tin.
- Thông báo các trạng thái của mạng...

Ưu điểm :

- Hoạt động theo nguyên lý nối song song nên nếu có một thiết bị nào đó ở một nút thông tin bị hỏng thì mạng vẫn hoạt động bình thường.
- Cấu trúc mạng đơn giản và các thuật toán điều khiển ổn định.
- Mạng có thể mở rộng hoặc thu hẹp tuỳ theo yêu cầu của người sử dụng.

Nhược điểm:

- Khả năng mở rộng mạng hoàn toàn phụ thuộc vào khả năng của trung tâm. Khi trung tâm có sự cố thì toàn mạng ngừng hoạt động.

- Mạng yêu cầu nối độc lập riêng rẽ từng thiết bị ở các nút thông tin đến trung tâm. Khoảng cách từ máy đến trung tâm rất hạn chế (100 m).

Nhìn chung, mạng dạng hình sao cho phép nối các máy tính vào một bộ tập trung (HUB) bằng cáp xoắn, giải pháp này cho phép nối trực tiếp máy tính với HUB không cần thông qua trục BUS, tránh được các yếu tố gây tắc nghẽn mạng. Gần đây, cùng với sự phát triển switching hub, mô hình này ngày càng trở nên phổ biến và chiếm đa số các mạng mới lắp.

4.1.2 Mạng hình tuyến (Bus Topology)

Theo cách bố trí hành lang các đường như hình vẽ thì máy chủ (host) cũng như tất cả các máy tính khác (workstation) hoặc các nút (node) đều được nối về với nhau trên một trục đường dây cáp chính để chuyển tải tín hiệu.

Tất cả các nút đều sử dụng chung đường dây cáp chính này. Phía hai đầu dây cáp được bịt bởi một thiết bị gọi là terminator. Các tín hiệu và gói dữ liệu (packet) khi di chuyển lên hoặc xuống trong dây cáp đều mang theo địa chỉ của nơi đến.

Loại hình mạng này dùng dây cáp ít nhất, dễ lắp đặt. Tuy vậy cũng có những bất lợi đó là sẽ có sự ùn tắc giao thông khi di chuyển dữ liệu với lưu lượng lớn và khi có sự hỏng hóc ở đoạn nào đó thì rất khó phát hiện, một sự ngừng trên đường dây để sửa chữa sẽ ngừng toàn bộ hệ thống.

4.1.3 Mạng dạng vòng (Ring Topology)

Mạng được bố trí theo dạng vòng tròn, đường dây cáp được thiết kế làm thành một vòng khép kín, tín hiệu chạy quanh theo một chiều nào đó. Các nút truyền tín hiệu cho nhau mỗi thời điểm chỉ được một nút mà thôi. Dữ liệu truyền đi phải có kèm theo địa chỉ cụ thể của mỗi trạm tiếp nhận.

Mạng Token Ring có thể chạy ở tốc độ 4Mbps hoặc 16Mbps. Phương pháp truy cập dùng trong mạng Token Ring gọi là Token passing. Token passing là phương pháp truy nhập xác định, trong đó các xung đột được ngăn ngừa bằng cách ở mỗi thời điểm chỉ một trạm có thể được truyền tín hiệu. Điều này được thực hiện bằng việc truyền một bó tín hiệu đặc biệt gọi là Token (mã thông báo) xoay vòng từ trạm này qua trạm khác. Một trạm chỉ có thể gửi đi bó dữ liệu khi nó nhận được Token, khi đó nó sẽ chiếm được quyền ưu tiên hoạt động trên mạng.

Mạng dạng vòng có thuận lợi là có thể nới rộng ra xa, tổng đường dây cần thiết ít hơn so với hai kiểu trên. Nhược điểm là đường dây phải khép kín, nếu bị ngắt ở một nơi nào đó thì toàn bộ hệ thống cũng bị ngừng.

4.1.4 Mạng dạng kết hợp

4.1.4.1 Kết hợp hình sao và tuyến (star/Bus Topology)

Cấu hình mạng dạng này có bộ phận tách tín hiệu (spitter) giữ vai trò thiết bị trung tâm, hệ thống dây cáp mạng có thể chọn hoặc Ring Topology hoặc Linear Bus Topology.

Ưu điểm của cấu hình này là mạng có thể gồm nhiều nhóm làm việc ở cách xa nhau, ARCNET là mạng dạng kết hợp Star/Bus Topology. Cấu hình dạng này đưa lại sự uyển chuyển trong việc bố trí đường dây tương thích dễ dàng đối với bất cứ toà nhà nào.

4.1.4.2 Kết hợp hình Sao và Vòng (Star/Ring Topology)

Cấu hình dạng kết hợp Star/Ring Topology, có một "thẻ bài" (token) được chuyển vòng quanh một cái HUB trung tâm. Mỗi trạm làm việc được nối với HUB - là cầu nối giữa các trạm làm việc và để tăng khoảng cách cần thiết.

Một hub thông thường có nhiều cổng nối với người sử dụng để gắn máy tính và các thiết bị ngoại vi. Mỗi cổng hỗ trợ một bộ kết nối dùng cặp dây xoắn 10BASET từ mỗi trạm của mạng. Khi bó tín hiệu Ethernet được truyền từ một trạm tới hub, nó được lặp lại trên khắp các cổng khác của hub. Các hub thông minh có thể định dạng, kiểm tra, cho phép hoặc không cho phép bởi người điều hành mạng từ trung tâm quản lý hub. Có ba loại hub:

- Hub đơn (stand alone hub)
- Hub modun (modular hub) : Modular hub rất phổ biến cho các hệ thống mạng vì nó có thể dễ dàng mở rộng và luôn có chức năng quản lý, modular có từ 4 đến 14 khe cắm, có thể lắp thêm các modun Ethernet 10BASET.
- Hub phân tầng (stackable hub) : thuận tiện cho những cơ quan muốn đầu tư tối thiểu ban đầu nhưng lại có kế hoạch phát triển LAN sau này.

4.2 Các giao thức điều khiển truy nhập đường truyền

Giao thức dùng để đánh giá khả năng của một mạng được phân chia bởi các trạm như thế nào. Hệ số này được quyết định chủ yếu bởi hiệu quả sử dụng môi trường truy xuất (medium access) của giao thức.

Mọi kênh phương tiện chỉ có thể hỗ trợ một lần tín hiệu. Nếu hai máy tính truyền trên kênh cùng một lúc, các tín hiệu của chúng sẽ gây nhiễu cho nhau (ví dụ như hai người cùng nói một lúc). Có hai phương pháp điều khiển việc truy nhập phương tiện để không xảy ra sự cố gây nhiễu : truy nhập ngẫu nhiên và truy nhập có điều khiển.

- *Loại truy nhập ngẫu nhiên*

Trạm có thể truy nhập phương tiện truyền tuỳ theo ý muốn, bất kỳ ở thời điểm ngẫu nhiên nào.

a. Kỹ thuật truy cập ngẫu nhiên đối với dạng bus

- Phương pháp đa truy nhập sử dụng sóng mang (CSMA - Carrier Sense Multiple Access).
- Phương pháp đa truy nhập sử dụng sóng mang với phát hiện xung đột (CSMA/CD - with Collision Detection)

b. Kỹ thuật truy cập ngẫu nhiên đối với dạng vòng

- Phương pháp chèn thanh ghi (Register insertion)
- Phương pháp vòng có ngăn (Slotted-ring)

- *Loại truy nhập có điều khiển*

Phương pháp điều khiển tranh chấp thường thích hợp với các mạng có sự trao đổi dữ liệu không liên tục và tương đối ít máy tính. Đây là dạng thông dụng trong cấu trúc mạng cục bộ.

- Kỹ thuật bus với thẻ bài (Token Bus) : dùng cho các mạng LAN
- Kỹ thuật vòng với thẻ bài (Token Ring) : dùng cho các mạng LAN
- Kỹ thuật tránh xung đột : dùng cho các mạng cục bộ tốc độ cao.

4.2.1 Phương pháp CSMA

Còn được gọi là phương pháp LBT (Listen Before Talk - Nghe trước khi nói). Một trạm có dữ liệu cần truyền trước hết phải 'nghe' xem phương tiện truyền rỗi hay bận. Nếu rỗi thì bắt đầu truyền tin, còn nếu bận thì thực hiện một trong ba giải thuật sau :

- Giải thuật '*non-persistent*' : Tram rút lui (không kiên trì) chờ đợi một thời gian ngẫu nhiên nào đó rồi lại bắt đầu 'nghe' đường truyền. Giải thuật này có hiểu quả tránh xung đột nhưng có thời gian chết.
- Giải thuật '*1-persistent*' : Trạm tiếp tục nghe đến khi phương tiện truyền rỗi thì tiến hành truyền dữ liệu đi (với xác xuất 1). Giải thuật này giảm thời gian chết, xong nếu có nhiều trạm cùng chờ và tiến hành phát dữ liệu cùng một lần thì sẽ xảy ra xung đột.
- Giải thuật '*p-persistent*' : trạm tiếp tục nghe, đến khi phương tiện truyền rỗi thì tiến hành phát tin với một xác xuất nhất định nào đó (mỗi trạm có gắn một hệ số ưu tiên). Ngược lại trạm 'rút lui' trong một thời gian cố định rồi

truyền với xác suất p hoặc tiếp tục chờ đợi với xác xuất 1-p. Giải thuật này phức tạp nhưng giảm được tối đa xung đột và thời gian chết.

Phương pháp CSMA chỉ 'nghe trước khi nói', không có khả năng phát hiện xung đột trong quá trình truyền, dẫn đến lãng phí đường truyền.

4.2.2 Phương pháp CSMA/CD

Phương pháp CSMA/CD có nguồn gốc từ hệ thống radio đã phát triển ở trường đại học Hawai vào khoảng năm 1970, gọi là ALOHANET, còn được gọi là phương pháp LWT (Listen While Talk - Nghe cả trong khi nói). Các va chạm luôn xảy ra tại một cấp nào đó trên các mạng, với số lượng gia tăng theo tỉ lệ thuận khi các phiên truyền gia tăng.

Phương pháp CSMA/CD ngoài các chức năng của CSMA còn bổ sung các quy tắc sau :

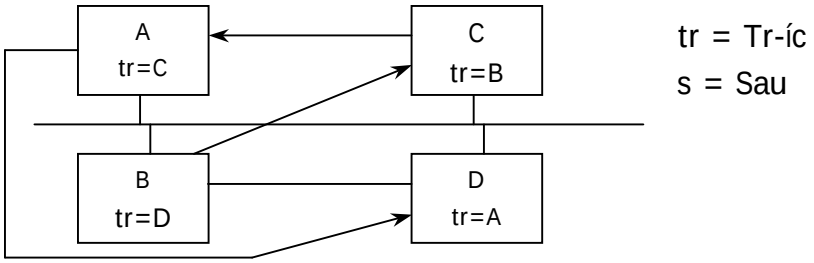
1. Khi đang truyền vẫn tiếp tục nghe đường dây.
2. Nếu phát hiện có xung đột thì *ngừng truyền* và *tiếp tục gởi sóng mang* thêm một thời gian nữa để bảo đảm các trạm đều có thể nghe được sự kiện xung đột.
3. Sau khi chờ đợi một thời gian ngẫu nhiên thì trạm thử truyền lại bằng cách sử dụng các phương pháp của CSMA.

Với phương pháp CSMA/CD thời gian chiếm dụng vô ích đường truyền giảm xuống bằng thời gian dùng để phát hiện một đụng độ. CSMA/CD sử dụng ba giải thuật 'persistent' ở trên. Trong đó giải thuật '*1-persistent*' được sử dụng trong mạng Ethernet, Mitrenet và được chọn cả trong chuẩn IEEE.802. Ngoài ra mỗi chuẩn LAN còn có thêm các cơ chế bổ sung.

4.2.3 Điều khiển truy nhập bus với thẻ bài

Các trạm trên bus tạo nên một vòng logic, được xác định vị trí theo một dãy thứ tự, trong đó trạm cuối sẽ tiếp liền ngay sau trạm đầu tiên. Mỗi trạm được biết địa chỉ của các trạm kề sau và kề trước nó.

Thẻ bài dùng cấp phát quyền truy nhập, được lưu chuyển trong vòng logic. Khi trạm nhận được thẻ bài thì được trao quyền sử dụng phương tiện trong một thời gian xác định để truyền dữ liệu. Khi truyền xong hoặc hết thời hạn, trạm sẽ chuyển thẻ bài đến trạm kế tiếp trong vòng logic. Các trạm không sử dụng thẻ bài vẫn có mặt trên bus nhưng chúng chỉ có thể trả lời cho yêu cầu xác nhận (nếu chúng là đích của gói tin nào đó). Thứ tự vật lý của trạm trên bus là không quan trọng, độc lập với thứ tự logic.



Hình 4-1. Điều khiển truy nhập bus với thẻ bài.

Các chức năng :

- Khởi tạo vòng logic : khi thiết lập mạng hoặc khi vòng logic bị gãy.
- Bổ sung trạm vào vòng logic (xem xét định kỳ) bằng cách mời nút đứng sau nhập vòng. Loại bỏ một trạm ra khỏi vòng logic bằng cách nối trạm trước và sau nó với nhau
- Quản lý sai sót : trùng địa chỉ, gãy vòng (các trạm bị treo, rơi vào trạng thái chờ lẫn nhau), bởi nút giữ Token.
- Khi đang giữ thẻ mà có trạm khác nhận được gói tin thì chứng tỏ nút khác đã có thẻ, lúc đó nó sẽ bỏ thẻ bằng cách chuyển sang trạng thái 'nghe'.
- Khi nút đã hoàn thành công việc, nó gửi thẻ đến nút đứng sau, nếu nút tiếp sau hoạt động thì nó gởi thẻ chuyển sang trạng thái bị động. Nếu ngược lại, nó gửi thẻ cho nút kế tiếp lần nữa. Nếu hai lần gửi không được thì xem như nút kế tiếp hỏng và gửi đi gói tin "tìm nút kế tiếp" để tìm nút tiếp theo.
- Nếu không thành công thì nút bị xem là có sự cố. Nút ngừng hoạt động và 'nghe' trên bus.

- Dạng bản tin của mạng Token bus

Bắt đầu tin	Điều khiển gói tin	Địa chỉ nguồn	Địa chỉ đích	TIN	FSC	Kết thúc tin
1 byte	1 byte	2-6 bytes	2 - 6 bytes		4 bytes	1 byte
	Khung tin cực đại 8191 bytes				Tốc độ có thể là 1; 5; 10Mbps	

- So sánh CSMA/CD và Token Bus

- Token bus quản lý phức tạp hơn so với CSMA/CD. Trong trường hợp tải nhẹ thì không hiệu quả bằng CSMA/CD (do phải qua nhiều trạm)
- Tuy nhiên Token Bus có hiệu quả trong trường hợp tải nặng, dễ điều hoà lưu thông trên mạng Token Bus. Không quy định độ dài tối thiểu của gói tin, không cần nghe trước khi nói.

4.2.4 Điều khiển truy nhập vòng với thẻ bài

Đây là giao thức thông dụng được dùng trong các LAN có cấu trúc vòng (Ring). Phương pháp này sử dụng một khối tín hiệu đặc biệt gọi là Token di chuyển vòng quanh mạng theo một chiều xác định. Một trạm muốn truyền phải đợi cho đến khi nhận được thẻ bài. Khi một trạm đang chiếm Token thì nó có thể phát đi một gói dữ liệu. Khi đã phát hết gói dữ liệu cho phép hoặc không còn gì để phát nữa thì trạm đó chuyển khung thẻ bài đến cho trạm kế tiếp trên mạng. Trong token có chứa một địa chỉ đích và được luân chuyển tới các trạm theo một trật tự đã định trước. Đối với cấu hình mạng dạng xoay vòng thì trật tự của sự truyền token tương đương với trật tự vật lý của các trạm xung quanh vòng.

Các chuẩn mạng sử dụng phương pháp điều khiển truy nhập thẻ bài :

- Chuẩn IEEE 802.5, còn gọi là chuẩn Token Ring.
- FDDI là chuẩn sợi quang 100 Mps sử dụng phương pháp chuyển thẻ bài và vòng tròn.

Phương pháp chuyển thẻ bài thích hợp trong các điều kiện như sau :

- Khi mạng đang tải dữ liệu quan trọng về thời gian do phương pháp này cung cấp khả năng bàn giao.
- Khi mạng được sử dụng nhiều, do tránh được xung đột.
- Khi một vài trạm có mức ưu tiên cao hơn so với các trạm khác. Phương pháp chuyển thẻ bài có thể áp dụng các mức ưu tiên cho trạm để ngăn cấm một trạm bất kỳ không được độc quyền về mạng.
- Do thẻ bài luân chuyển quanh mạng nên mỗi trạm có thể truyền theo quãng thời gian tối thiểu.

Phương pháp chuyển thẻ bài đòi hỏi cơ chế điều khiển phức tạp và chi phí đầu tư phần cứng cao, nhưng được thiết kế với độ tin cậy cao. Tuy vậy hiện nay Ethernet vẫn là chuẩn LAN thông dụng, chứng tỏ được ưu điểm của phương pháp tranh chấp khi sử dụng trên các mạng LAN.

Giao thức truyền token có trật tự hơn nhưng cũng phức tạp hơn CSMA/CD, có ưu điểm là vẫn hoạt động tốt khi lưu lượng truyền thông lớn. Giao thức truyền token tuân thủ đúng sự phân chia của môi trường mạng, hoạt động dựa vào sự xoay vòng tới các trạm. Việc truyền token sẽ không thực hiện được nếu việc xoay vòng bị đứt đoạn. Giao thức phải chứa các thủ tục kiểm tra token để cho phép khôi phục lại token bị mất hoặc thay thế trạng thái của token và cung cấp các phương tiện để sửa đổi logic (thêm vào, bớt đi hoặc định lại trật tự của các trạm).

Khung tin cực đại là 16KB ở chế độ truyền 16Mbps và 4KB ở chế độ truyền 4Mps.

Dạng bản tin với mạng Token Ring :

Bắt đầu tin	Điều khiển thâm nhập	Điều khiển gói tin	Địa chỉ nguồn	Địa chỉ đích	TIN	FSC	Kết thúc gói tin	Trạng thái gói tin
1 byte	1 byte	2-6 bytes	2 - 6 bytes	2 - 6 bytes		4 bytes	1 byte	1 byte

4.2.4.1 Phương pháp điều khiển truy nhập dò báo

Dò báo (*polling*) là một phương pháp điều khiển truy cập sử dụng một thiết bị trung tâm để điều khiển toàn bộ việc truy cập mạng. Đây là phương pháp được sử dụng phổ dụng nhất trên các mạng máy tính lớn.

Thiết bị trung tâm có tên là thiết bị chính sẽ yêu cầu dữ liệu từ các thiết bị khác trên mạng có tên là thiết bị thứ cập (*secondaries*). Sau khi được dò báo, thiết bị thứ cấp có thể truyền một lượng dữ liệu được xác định bởi các giao thức dùng trên mạng. Một thiết bị thứ cấp không thể truyền trừ phi nó được thiết bị chính dò báo.

Phương pháp dò báo có nhiều ưu điểm của phương pháp chuyển thẻ bài như :

- Dự đoán được các lần truy cập định sẵn.
- Gán được các mức ưu tiên, tránh được va chạm.

So sánh phương pháp dò báo và phương pháp chuyển thẻ bài : kỹ thuật dò báo tập trung hóa quyền điều khiển. Nhìn dưới góc độ quản lý thì đây là một ưu điểm, nhưng nếu cơ chế điều khiển trung tâm bị hỏng, mạng sẽ ngừng hoạt động. Phương pháp chuyển thẻ bài sử dụng các chức năng điều khiển phân phối hơn do đó ít bị hỏng tập trung tại một điểm. Bên cạnh đó, phương pháp dò báo đôi khi lãng phí các lượng băng thông lớn do phải dò báo từng thiết bị thứ cấp, cho dù các thiết bị không có gì để truyền.

4.3 Chuẩn hóa mạng cục bộ

Các chuẩn LAN là các tiêu chuẩn công nghệ cho Lan được phê chuẩn bởi các tổ chức chẩun hóa quốc tế, nhằm hướng dẫn các nhà sản xuất thiết bị mạng đi đến sự thống chung khả năng sử dụng chung các sản phẩm của họ, vì lợi ích của người sử dụng và tạo điều kiện thuận lợi cho các nghiên cứu phát triển.

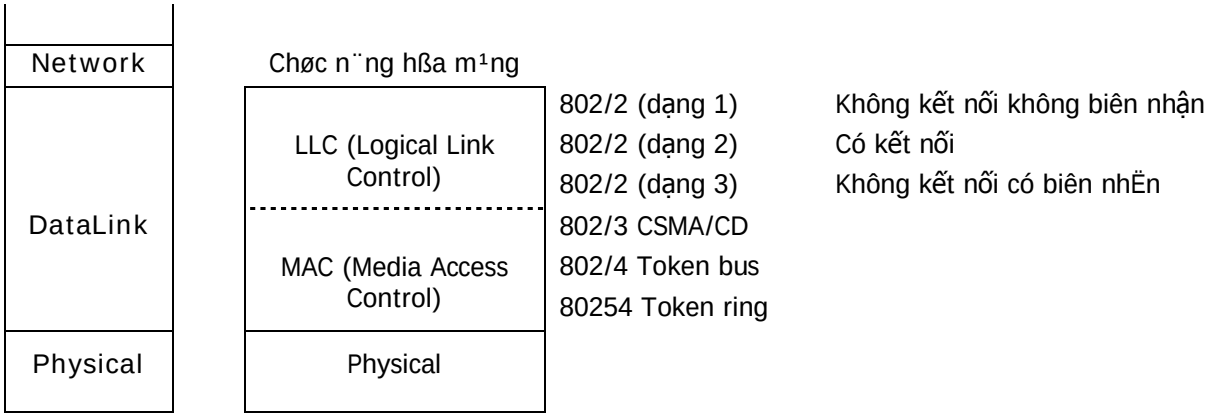
Các chuẩn này quy định môi trường truyền dẫn cũng như cách thức sử dụng chúng trong kết nối LAN; Các giao thức truyền thông ở các tầng vật lý và tầng liên kết dữ liệu của mạng theo mô hình OSI.

Các giao thức truyền thông ở các tầng trên của mô hình OSIhiện tại được xác định qua một số giao thức phổ biến như TCP/IP, IPX/SPX, NetBIOS,. . .

Uỷ ban IEEE phát triển tiêu chuẩn IEEE LAN và đề xuất phân chia hai tầng thấp nhất của mô hình OSI như dưới đây.

Theo chuẩn 802 thì tầng LKDL được chia thành 2 tầng con:

- Tầng con điều khiển logic LLC (Logical Link Control Sublayer) : giữ vai trò tổ chức dữ liệu, tổ chức thông tin để truyền và nhận. Thủ tục tầng LLC không bị ảnh hưởng khi sử dụng các đường truyền dẫn khác nhau, nhờ vậy mà linh hoạt hơn trong khai thác.
- Tầng con điều khiển xâm nhập mạng MAC (Media Access Control Sublayer). làm nhiệm vụ điều khiển việc xâm nhập mạng.



Hình 4-2. Các tầng con LLC và MAC.

Chuẩn 802.2 ở mức con LLC tương đương với chuẩn HDLC của ISO hoặc X.25 của CCITT.

Chuẩn 802.3 xác định phương pháp thâm nhập mạng tức thời có khả năng phát hiện lỗi chồng chéo thông tin CSMA/CD. Phương pháp CSMA/CD được đưa ra từ năm 1993 nhằm mục đích nâng cao hiệu quả mạng. Theo chuẩn này các mức được ghép nối với nhau thông qua các bộ ghép nối.

Chuẩn IEEE 802.3 dùng cho mạng Ethernet (sử dụng giao thức truy nhập CSMA/CD) bao gồm cả 2 phiên bản băng tần cơ bản và băng tần mở rộng.

Chuẩn IEEE 802.4 liên quan tới sự sắp xếp tuyến token, thực chất là phương pháp thâm nhập mạng theo kiểu phát tín hiệu thăm dò token qua các trạm và đường truyền bus.

Chuẩn IEEE 802.5 dùng cho mạng dạng vòng và trên cơ sở dùng tín hiệu thăm dò token. Mỗi trạm khi nhận được tín hiệu thăm dò token thì tiếp nhận token và bắt đầu quá trình truyền thông tin dưới dạng các frame. Các frame có cấu trúc tương tự như của chuẩn 802.4. Phương pháp xâm nhập mạng này quy định nhiều mức ưu tiên khác nhau cho toàn mạng và cho mỗi trạm, việc quy định này vừa cho người thiết kế vừa do người sử dụng tự quy định.

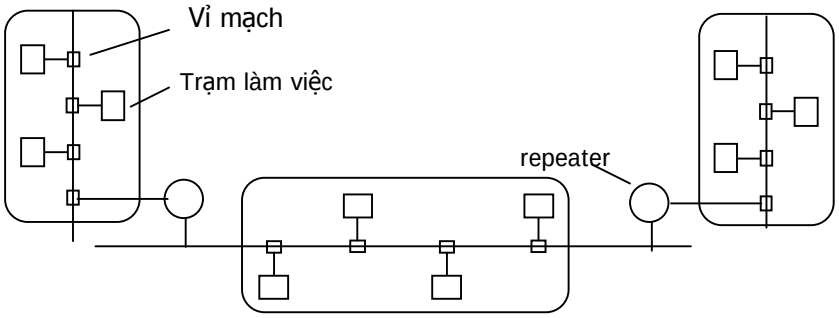
Chuẩn IEEE 802.11 dùng cho mạng không dây (Wireless).

4.3.1 Chuẩn Ethernet

Chuẩn Ethernet được sử dụng phổ biến nhất, đến mức đôi khi được hiểu đồng nghĩa với LAN. Tuy nhiên nó đã được xây dựng và phát triển qua các giai đoạn với các tên gọi là DIX standard Ethernet và IEE 802.3 standard. Chuẩn Ethernet do các công ty Xerox, Intel và Digital equipment xây dựng và phát triển. Ethernet LAN được xây dựng theo chuẩn 7 lớp trong cấu trúc mạng của ISO, mạng truyền số liệu Ethernet cho phép đưa vào mạng các loại máy tính khác nhau kể cả máy tính mini. Ethernet có các đặc tính kỹ thuật chủ yếu sau đây:

- Có cấu trúc dạng tuyến phân đoạn, đường truyền dùng cáp đồng trục, tín hiệu truyền trên mạng được mã hoá theo kiểu đồng bộ (Manchester), tốc độ truyền dữ liệu là 10 Mb/s.
- Chiều dài tối đa của một đoạn cáp tuyến là 500m, các đoạn tuyến này có thể được kết nối lại bằng cách dùng các bộ chuyển tiếp và khoảng cách lớn nhất cho phép giữa 2 nút là 2,8 km.
- Sử dụng tín hiệu băng tần cơ bản, truy xuất tuyến (bus access) hoặc tuyến token (token bus), giao thức là CSMA/CD, dữ liệu chuyển đi trong các gói. Gói tin dùng trong mạng có độ dài từ 64 đến 1518 byte.

- Cấu trúc của mạng Ethernet : Mạng Ethernet có cấu trúc dạng bus như sau :



Hình 4-3. Cấu trúc của mạng Ethernet.

Số trạm cực đại trong mạng là 1024, số lượng segment của mạng giới hạn nhỏ hơn 5 segment, khoảng cách tối đa giữa hai trạm là 2,5km. Mạng sử dụng cáp đồng trục tốc độ 10Mps. Cấu trúc khung tin Ethernet có khuôn dạng như sau :

Cờ	Địa chỉ đích	Địa chỉ nguồn	Loại tin	TIN	CRC	Cờ
	6 bytes	6 bytes	2 bytes	46 – 1500 bytes	4 bytes	

Chương 5

TẦNG MẠNG

Tầng mạng đảm bảo truyền tin thông suốt giữa hai nút đầu cuối trong mạng. Trên cơ sở cấu hình của mạng, tầng mạng sẽ kiểm tra sơ đồ kết nối (*topology*) của toàn mạng để quyết định đường đi tối ưu truyền gói dữ liệu, tránh quá tải trên một đường truyền trong khi một số đường truyền rỗi. Thực hiện cắt/ hợp dữ liệu khi qua mạng và liên kết mạng khi có nhiều mạng nối với nhau.

5.1 Các vấn đề của tầng mạng

5.1.1 Định địa chỉ cho tầng mạng

Tầng mạng sử dụng các kiểu địa chỉ bổ sung sau :

1. Địa chỉ mạng logic (Logical network addresses), định tuyến các gói tin theo các mạng cụ thể trên liên mạng. Dùng để định danh một mạng cụ thể trên liên mạng dưới dạng một nguồn hay đích của một gói tin.
2. Địa chỉ dịch vụ (Service addresses), định tuyến các gói tin theo các tiến trình cụ thể đang chạy trên thiết bị đích, dùng định danh một giao thức hay tiến trình trên máy tính là nguồn hay đích của một gói tin.
3. Địa chỉ mạng vật lý (MAC) định danh một thiết bị cụ thể dưới dạng một nguồn hay đích của một khung.

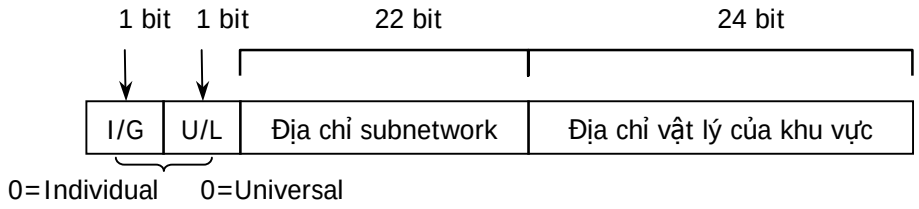
Địa chỉ vật lý của máy trạm :

Mỗi thiết bị trên một mạng có một địa chỉ vật lý duy nhất để giao tiếp với các thiết bị khác, còn gọi là địa chỉ phần cứng. Trên tất cả các mạng hiện nay, mỗi địa chỉ xuất hiện một lần duy nhất (nghĩa là mỗi thiết bị chỉ có một địa chỉ duy nhất). Đối với phần cứng, địa chỉ thường được mã hoá trong thiết bị card mạng (Network Interface Card), có thể được đặt bằng chuyển mạch hoặc bằng phần mềm. Trong mô hình OSI thì địa chỉ này được đặt ở lớp vật lý.

Độ dài của địa chỉ vật lý phụ thuộc vào từng mạng, chẳng hạn với mạng Ethernet và một số mạng khác thì dùng địa chỉ vật lý dài 48 bit. Để trao đổi thông tin thì cần có địa chỉ của nơi gởi, và địa chỉ của nơi nhận.

Hiện nay IEEE đang đảm nhiệm việc ấn định địa chỉ vật lý tổng thể (*universal physical address*) cho các subnetwork. Đối với mỗi subnetwork, IEEE ấn định một phần địa chỉ đồng nhất đối với tất cả các subnetwork gọi là OUI (Organization Unique Identifier) phần này có độ dài là 24 bit, cho phép IEEE ấn định phần địa chỉ 24 bit còn lại theo yêu cầu. (Trên thực tế, hai trong 24 bit địa chỉ OUI là các bit điều khiển, do đó 22 bit là để xác định subnetwork đó. Dó đó chỉ có khoảng 2^{22} địa chỉ

được dùng, nếu với tốc độ phát triển như hiện nay có thể sẽ thiếu địa chỉ trong tương lai). Sau đây là cấu tạo của địa chỉ OUI :



Hình 5-1. Cấu tạo của địa chỉ vật lý ĂUI.

5.1.2 Dịch vụ cung cấp cho tầng giao vận

- Các dịch vụ phải độc lập với công nghệ được dùng trong mạng.
- Tầng giao vận phải độc lập với công nghệ được dùng trong mạng.
- Các địa chỉ mạng phải thống nhất để tầng giao vận có thể dùng cả mạng LAN và WAN.

Có 2 loại dịch vụ :

- Dịch vụ truyền tin có liên kết (*Connection Ăriennted Service*)
- Dịch vụ truyền tin không liên kết (*Connectionless Service*)

Sự khác nhau giữa hai dịch vụ

Vấn đề	**Dịch vụ có liên kết**	**Dịch vụ không liên kết**
Khởi động kênh	Cần thiết	Không
Địa chỉ đích	Chỉ cần lúc khởi động	Cần ở mọi gói tin
Thự tự gói tin	Được đảm bảo	Không đảm bảo
Kiểm soát lỗi	ở tầng mạng	ở tầng giao vận
Điều khiển thông lượng	ở tầng mạng	ở tầng giao vận
Thảo thuận tham số	Có	Không
Nhận dạng liên kết	Có	Không

Các hàm cơ bản của dịch vụ liên kết tầng mạng :

N-CONNECT. Request (callce, caller, acks wanted, exp wanted, qos, user data)
N-CONNECT. Indication (callce, caller, acks wanted, exp wanted, qos, user data)
N-CONNECT. Response (response acks wanted, exp wanted, qos, user data)
N-CONNECT. Confirmation (response acks wanted, exp wanted, qos, user data)
N-DISCONNNECT. Request (originator, reason, user data, responding address)
N-DISCONNNECT. Indication (originator, reason, user data, responding address)
N-DATA. Request (user data)
N-DATA. Indication (user data)
N-DATA-ACKNOWLEDGED. Request ()
N-DATA-ACKNOWLEDGED. Indication ()
N-EXPEDITED-DATA. Request (user data)
N-EXPEDITED-DATA. Indication (user data)
N-RESET. Request (originator, reason)

N-RESET. Indication (originator, reason)
N-RESET. Response()
N-RESET. Confim()

Các hàm cơ bản của dịch vụ không liên kết tầng mạng

N-UNITDATA. Request (source address, destination address, qos, user_data)
N-UNITDATA. Indication (source address, destination address, qos, user_data)
N-FACILITY. Request (qos)
N-FACILITY. Indication (destination address, qos, reason)
N-FACILITY. Indication (destination address, qos, reason)

Hàm N_FACILITY.request cho phép NSD dịch vụ mạng biết tỷ lệ phần trăm gói tin đang được giao vận.

Hàm N_REPORT.indication cho phép tầng mạng thông báo lại cho NSD dịch vụ mạng.

5.1.3 Tổ chức các kênh truyền tin trong tầng mạng

Có hai loại kênh truyền tin hoạt động trong mạng :

5.1.3.1 Kênh ảo (virtual circuit)

Tương đương kênh điện thoại trong tầng vật lý sử dụng trong mạng có liên kết. Kênh ảo được thiết lập cho mỗi liên kết. Một khi đã được thiết lập thì các gói tin được chuyển đi tương tự trong mạng điện thoại cho đến khi liên kết bị hủy bỏ.

- Mỗi nút mạng chứa một kênh ảo, với cửa vào cho một kênh ảo
- Khi một liên kết được khởi động, một kênh ảo chưa dùng sẽ được chọn
- Nút chọn kênh ảo chứa đường dẫn đến trạm tiếp theo và có số thấp nhất

Khi gói tin khởi động đến nút đích, nút chọn kênh ảo có số thấp nhất thay thế số trong gói tin và chuyển vào trạm đích. Số kênh ảo nối với trạm đích có thể khác số kênh ảo mà trạm nguồn sử dụng.

5.1.3.2 Mạng Datagram

Tương đương với điện báo sử dụng trong mạng không liên kết. Trong mạng này, không có tuyến đường nào được thiết lập. Các gói tin có thể đi theo nhiều đường khác nhau mà không nhất thiết theo một trình tự xác định. Thông tin vào là địa chỉ đích, thông tin ra là nút mạng phải tới.

Mạng Datagram phức tạp về điều khiển nhưng nếu kênh hỏng thì dễ dàng đi theo kênh khác. Do đó có thể giải quyết được vấn đề tắc nghẽn dữ liệu.

- Các đặc trưng của mạng Datagram và mạng kênh ảo

Vấn đề	Mạng datagram	Mạng kênh ảo
Khởi động kênh	Không	Cần thiết
Địa chỉ (đ/c) hoá	Gói tin phải có đ/c nguồn và đ/c đích	Gói tin chỉ cần số của kênh ảo
Thông tin tìm đường	Không cần bất cứ thông tin nào.	Mỗi kênh ảo cần một vùng trong bảng
Tìm đường	Mỗi gói tin tìm đường độc lập. Phải tìm đường mỗi khi có gói tin tới nút mạng.	Được thiết lập khi khởi động kênh ảo mới. Liên kết sẽ được duy trì cho cả phiên.
Điều khiển	Chỉ mất gói tin ở trong nút hỏng	Kênh ảo đi qua nút hỏng sẽ bị huỷ
Hỏng nút	Khó khắc phục	Dễ khắc phục hơn
Độ phức tạp	Trong tầng giao vận	Trong tầng mạng
Thích hợp	Các dịch vụ liên kết và không liên kết	Các dịch vụ liên kết

5.1.4 Tìm đường đi trong mạng

Chức năng quan trọng nhất của tầng mạng là dẫn đường cho các gói tin từ trạm nguồn tới trạm đích. Thuật toán tìm đường là qui trình để quyết định chọn đường ra khỏi nút mạng nhằm gởi gói tin đi tiếp tới nút khác.

- Yêu cầu của thuật toán tìm đường

- Chính xác, ổn định, đơn giản và tối ưu.
- Thuật toán tìm đường phải có khả năng cập nhật lại cấu hình và đường vận chuyển để không phải khởi động lại mạng khi có một nút hỏng hoặc phải ngừng hoạt động của các máy ở trạm.

- Các thuật toán chia làm hai nhóm chính:

- Nhóm không thích nghi (*non adaptive*) : việc chọn đường không dựa vào việc đánh giá tình trạng giao thông và cấu hình trong thời gian thực.
- Nhóm thích nghi (*adaptive*) : việc tìm đường phải thích nghi với tình trạng giao thông hiện tại.

Sơ đồ mạng được biểu diễn dưới dạng đồ thị, mỗi nút của đồ thị là một nút mạng, cung của đồ thị biểu diễn đường truyền nối giữa hai nút. Việc chọn đường giữa hai nút mạng là tìm đường ngắn nhất giữa chúng.

Mỗi cung được gán một nhãn cho biết thời gian trung bình phải đợi và thời gian truyền một gói tin chuẩn. Thời gian này được thử mỗi giờ hay mỗi ngày một lần. Đường ngắn nhất là đường có ít bước chuyển tiếp qua nút nhất và có số đo độ dài nhỏ nhất, mất ít thời gian.

Có nhiều thuật toán để tìm đường ngắn nhất giữa 2 điểm, ví dụ như thuật toán Dijkstra (1959). Ta xây dựng đồ thị cho các nút mạng và tìm khoảng cách giữa các nút mạng.

5.1.5 Tắc nghẽn trong mạng

Khi có quá nhiều gói tin trong mạng hay một phần của mạng làm cho hiệu suất của mạng giảm đi vì các nút mạng không còn đủ khả năng lưu trữ, xử lý, gửi đi và chúng bắt đầu bị mất các gói tin. Hiện tượng này được gọi là sự tắc nghẽn (*congestion*) trong mạng.

Hàng đợi sẽ bị đầy (phải lưu tập tin, tạo các bảng chọn đường ...) nếu khả năng xử lý của nút yếu hoặc khi thông tin vào nhiều hơn khả năng của đường ra

Điều khiển dòng dữ liệu là xử lý giao thông giữa điểm với điểm, giữa trạm thu và phát. Trong khi đó điều khiển tránh tắc nghẽn là một vấn đề tổng quát hơn bao gồm việc tạo ra hoạt động hợp lý của các máy tính của các nút mạng, quá trình lưu trữ bên trong nút, điều khiển tất cả các yếu tố làm giảm khả năng vận chuyển của toàn mạng.

- Các biện pháp ngăn ngừa

- Bố trí khả năng vận chuyển, lưu trữ, xử lý của mạng dư so với yêu cầu.
- Huỷ bỏ các gói tin bị tắc nghẽn quá thời hạn.
- Hạn chế số gói tin vào mạng nhờ cơ chế cửa sổ (*flow control*).
- Chặn đường vào khi của các gói tin khi mạng quá tải.

5.2 Kết nối liên mạng

Nhu cầu trao đổi thông tin và phân chia các tài nguyên dùng chung đòi hỏi hoạt động truyền thông không chỉ ở phạm vi cục bộ mà ở cả khuôn khổ quốc gia và quốc tế. Từ đó dẫn đến sự nối kết các mạng viễn thông tin học được đặt ở các vị trí địa lý khác nhau và chịu sự quản lý của các tổ chức hoặc quốc gia khác nhau.

Sự nối kết mạng (*Networks Interconnection*) giống như ghép nối mạng đơn lẻ nhưng phức tạp hơn nhiều do tính chất không thuần nhất của các mạng con được kết nối. Chúng có thể có kiến trúc khác nhau bao gồm các máy tính nút mạng. Đường truyền khác nhau, chiến lược quản lý khác nhau.

Người ta thường xem xét các vấn đề sau để kết nối các mạng con lại với nhau :

- Xem mỗi nút của mạng con như là một hệ thống mở : mỗi nút mạng con có thể truyền thông trực tiếp với một nút của mạng con khác bất kỳ. Như thế yêu cầu phải xây dựng một chuẩn chung cho các mạng.
- Xem mỗi mạng như là một hệ thống mở : Hai nút thuộc hai mạng con không bắt tay trực tiếp với nhau mà phải thông qua một phần tử trung gian gọi là *giao diện kết nối (interconnection interface)* dat giữa hai mạng con đó.

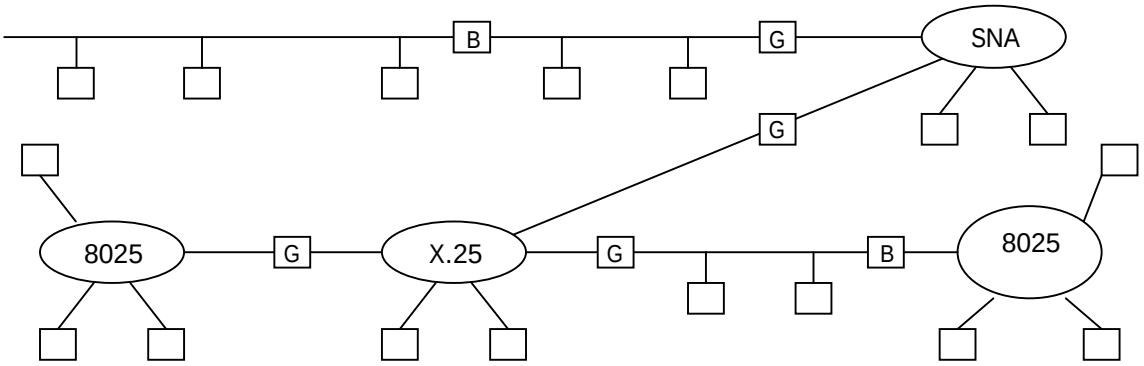
Chức năng của giao diện kết nối phụ thuộc vào sự khác biệt kiến trúc của mạng con : sự khác biệt càng lớn thì chức năng của giao diện càng phức tạp.

Có thể có các kết nối mạng như sau :

LAN-LAN	: Nối các mạng cục bộ.
LAN-WAN	: Nối các mạng cục bộ với mạng đường dài.
WAN-WAN	: Nối các mạng đường dài
LAN-WAN- LAN	: Nối mạng đường dài với mạng cục bộ.

Nếu máy nguồn và máy đích không ở cùng một mạng phải tìm đường từ mạng này sang mạng khác. Nếu trạm nguồn và đích không ở hai mạng liền kề thì giải quyết tìm đường qua nhiều trạm.

Các mạng khác nhau có các giao thức khác nhau, dẫn đến khác nhau về dạng khuôn của gói tin, đầu gói tin, điều khiển dòng dữ liệu và qui tắc xác nhận.



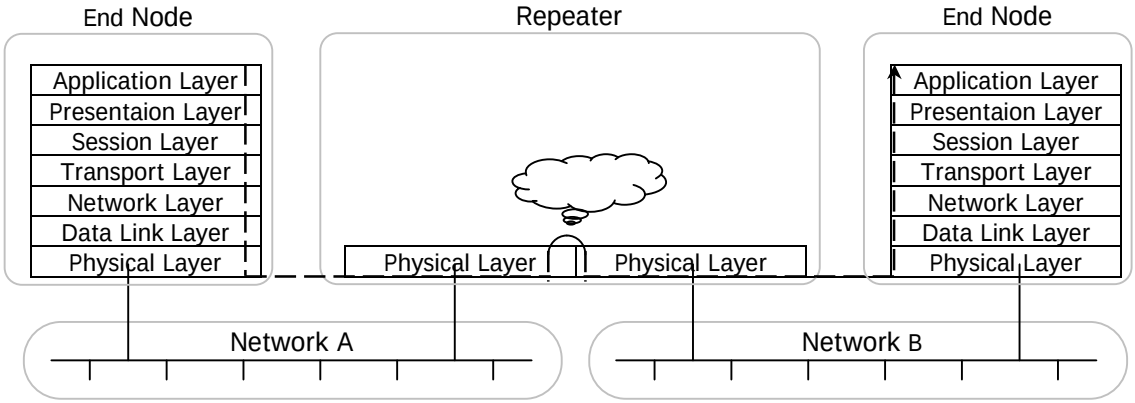
Hình 5-2. Kết nối liên mạng.

5.2.1 Các thiết bị dùng để kết nối liên mạng

Việc kết nối các LAN riêng lẻ thành một liên mạng chung gọi là Internetworking, sử dụng các thiết bị kết nối thông dụng như sau :

5.2.1.1 Bộ lặp

Bộ lặp (repeater) thực hiện chức năng ở tầng vật lý để khuyếch đại tín hiệu khi tín hiệu truyền đi xa. Bộ lặp được sử dụng để kết nối các đoạn mạng lại với nhau. Bộ lặp nhận tín hiệu từ một đoạn mạng, tái tạo và truyền tín hiệu này đến đoạn mạng khác. Nhờ có bộ lặp mà tín hiệu bị suy yếu do phải truyền qua một đoạn cáp dài có thể trở lại dạng ban đầu và truyền đi được xa hơn.



Hình 5-3. Sơ đồ kiến trúc của Repeater trong mô hình OSI.

Bộ lọc không có khả năng xử lý lưu lượng. Tất các tín hiệu điện, bao gồm cả nhiễu điện từ và các lỗi khác cũng được lặp và khuếch đại. Để bộ lặp hoạt động, cả hai đoạn mạng nối tới bộ lặp phải sử dụng cùng một phương thức truy nhập đường truyền. Ví dụ: bộ lặp không thể nối một đoạn mạng sử dụng phương thức CSMA/CD và một đoạn mạng sử dụng phương thức chuyển thẻ bài.

Bộ lặp có thể di chuyển gói dữ liệu từ phương tiện truyền dẫn này sang phương tiện truyền dẫn khác. Ví dụ có thể nhận gói dữ liệu từ một đoạn mạng dùng cáp đồng trục và chuyển gói đó sang đoạn mạng sử dụng cáp quang.

5.2.1.2 Hub

HUB là một thiết bị liên kết mạng được sử dụng rộng rãi. HUB còn là thành phần trung tâm trong cấu trúc mạng hình sao (Star). Mạng Star sử dụng sự phân chia tín hiệu trong HUB để đưa các tín hiệu ra các đường cáp khác nhau. Do vậy, có 3 loại HUB có thể sử dụng trong mạng là: HUB chủ động, HUB thụ động và HUB lai.

- ***HUB chủ động:*** Hầu hết các HUB đều là HUB chủ động, chúng tái tạo và truyền lại tín hiệu giống như bộ lặp. HUB thường có nhiều cổng nên thỉnh thoảng chúng còn được gọi là bộ lặp đa cổng. HUB chủ động đưa ra các tín hiệu mạnh hơn do đó cho phép đoạn cáp dài hơn.

Hình 5-4. Thiết bị kết nối mạng HUB.

- *HUB thụ động:* Các HUB thụ động hoạt động như các điểm kết nối, chúng không tái tạo hoặc khuếch đại tín hiệu.
- *HUB lai:* Các HUB thích ứng với nhiều loại cáp khác nhau được gọi là HUB lai.

5.2.1.3 Cầu nối (Bridge)

Cầu nối là một thiết bị hoạt động ở tầng liên kết dữ liệu. Dùng để nối hai hoặc nhiều đoạn (*segment*) của mạng LAN khác nhau.

Hình 5-5. Cầu nối.

- Chức năng của cầu nối :

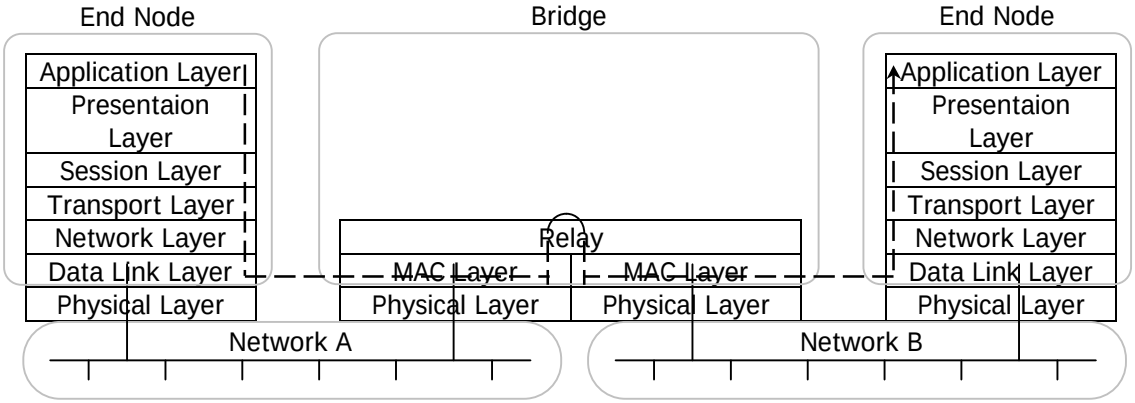
- Mở rộng khoảng cách của phân đoạn mạng, tăng số lượng máy tính trên mạng.
- Lọc những gói dữ liệu để gửi đi (hay không gửi) cho đoạn nối, hoặc gửi trả lại nơi xuất phát.
- Phân chia một mạng lớn thành hai mạng nhỏ nhằm cô lập lưu lượng, tăng tốc độ mạng. Nếu lưu lương từ một nhóm máy tính trở nên quá tải và làm giảm hiệu suất toàn mạng thì cầu nối có thể cô lập máy tính hoặc bộ phận này.
- Làm giảm hiện tượng tắc nghẽn do số lượng máy tính nối vào mạng qua lớn : Cầu nối có thể tiếp nhận một mạng quá tải và chia nó thành hai mạng riêng biệt, nhằm giảm bớt lưu lượng truyền trên mỗi đoạn mạng và do đó mỗi mạng sẽ hoạt động hiệu quả hơn.
- Kết nối các phương tiện truyền dẫn khác nhau, như cáp xoắn đôi và cáp quang.
- Kết nối các đoạn mạng sử dụng phương thức truy nhập đường truyền khác nhau, chẳng hạn CSMA/CD và chuyển thẻ bài.

- Nguyên lý hoạt dộng

- Cầu nối không phân biệt giữa giao thức này với giao thức khác, chỉ có nhiệm vụ chuyển lưu lượng của tất cả các giao thức dọc theo mạng. Vì giao thức nào cũng di chuyển ngang qua cầu nối, nên tuỳ thuộc vào từng máy tính quyết định chúng có thể nhận diện được giao thức nào.

- Cầu nối hoạt động trên nguyên tắc mỗi nút mạng có một địa chỉ riêng. Cầu nối chuyển gói dữ liệu dự trên địa chỉ của nút đích (địa chỉ MAC). Khi dữ liệu truyền qua cầu nối, thông tin địa chỉ của máy tính được lưu trong RAM của cầu nối dùng để xây dựng bảng địa chỉ dựa trên địa chỉ nguồn của gói tin.

Giao diện Bridge chỉ chứa tầng 1 và tầng con MAC, có chức năng chuyển đổi khuôn dạng của các đơn vị dữ liệu (frame) của các giao thức khác nhau và gửi chúng tới các mạng cục bộ đích có kèm theo phối hợp tốc độ.



Hình 5-6. Sơ đồ kiến trúc của Bridge trong mô hình OSI.

Ví dụ một Bridge nối giữa IEEE 820.3 và IEEE 820.5. Cầu nối này có hai card mạng: card Token Ring và card Ethermet để giao tiếp với hai mạng.

5.2.1.4 Bộ dẫn đường (router)

Trong môi trường gồm nhiều đoạn mạng với giao thức và kiến trúc mạng khác nhau, cầu nối không thể đảm bảo truyền thông nhanh trong tất cả các đoạn mạng. Mạng có độ phức tạp như vậy cần một thiết bị không những biết địa chỉ của mỗi đoạn mạng, mà còn quyết định tuyến đường tốt nhất để truyền dữ liệu và lọc lưu lượng quảng bá trên các đoạn mạng cục bộ. Thiết bị như vậy được gọi là bộ định tuyến.

Hình 5-7. Bộ định tuyến.

- Chức năng của bộ định tuyến :

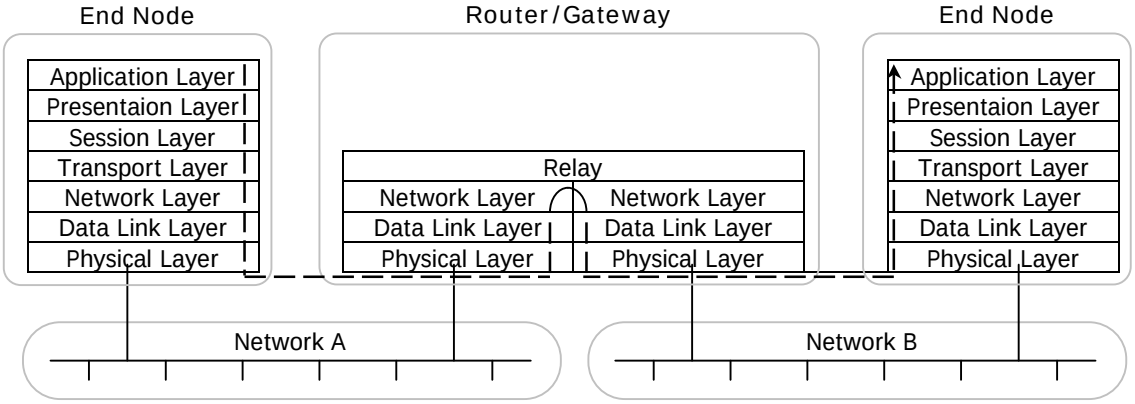
- Chuyển đổi và định tuyến gói dữ liệu qua nhiều mạng dựa trên địa chỉ phân lớp của mạng, cung cấp các dịch vụ như bảo mật, quản lý lưu thông...
- Phân chia một mạng lớn thành nhiều mạng nhỏ, và có thể kết nối nhiều đoạn mạng với nhau.
- Lọc gói tin và cô lập lưu lượng mạng : hoạt động như một rào cản an toàn giữa các đoạn mạng (do có thể lọc dữ liệu).
- Ngăn chặn tình trạng quảng bá vì chúng không chuyển tiếp các gói tin quảng bá, cải thiện việc phân phát gói dữ liệu.
- Các bộ định tuyến có thể chia sẽ thông tin trạng thái và thông tin định tuyến với nhau và sử dụng thông tin này để bỏ qua các kết nối hỏng hoặc chậm.

- Nguyên lý hoạt động :

Trong bộ định tuyến có một bảng định tuyến chứa các địa chỉ mạng. Tuy nhiên, địa chỉ mạng có thể được lưu trữ tuỳ thuộc vào giao thức mạng đang chạy. Bộ định tuyến sử dụng bảng định tuyến để xác định địa chỉ đích cho dữ liệu nhận được. Bảng này liệt kê các thông tin sau:

- Địa chỉ mạng đã kết nối.
- Cách kết nối tới các mạng khác.
- Phí tổn truyền dữ liệu qua các lộ trình đó.

Khi bộ định tuyến nhận được một gói dữ liệu cần gởi đến mạng ở xa, nó kiểm tra bảng đinh tuyến và chọn đường đi tối ưu (theo một tiêu chuẩn nào đó) để gởi gói dữ liệu đến đích.



Hình 5-8. Sơ đồ kiến trúc của Router trong mô hình OSI.

- Truyền dữ liệu qua bộ định tuyến

Trong mọi trường hợp, khi một trạm xác định rằng nó phải gởi một gói dữ liệu tới một trạm trên một mạng khác. Công việc đầu tiên trạm này cần làm là lấy địa chỉ vật lý MAC của Router (địa chỉ cổng nối ngầm định). Sau đó nó điền thông tin trong trường địa chỉ vật lý đích của gói dữ liệu bằng địa chỉ vật lý MAC của Router, và trường thông tin địa chỉ đích ở tầng mạng (chẳng hạn địa chỉ IP nếu dùng giao thức TCP/IP) bằng địa chỉ của trạm đích.

Khi Router kiểm tra địa chỉ đích, nó xác định xem nó biết hay không biết cách chuyển tiếp gói dữ liệu đến bước nhảy tiếp theo (Router kế tiếp trên đường đi) bằng cách kiểm tra địa chỉ. Nếu địa chỉ mạng đích nằm trong gói dữ liệu không có bảng định tuyến, Router thường bỏ gói dữ liệu đi. Trong trường hợp địa chỉ mạng đích có bảng định tuyến, Router thay địa chỉ vật lý đích bằng địa chỉ vật lý của bước nhảy tiếp theo và truyền gói dữ liệu đến bước nhảy tiếp theo.

Như vậy, khi một gói tin được chuyển qua liên mạng, địa chỉ vật lý đích của nó thay đổi, nhưng địa chỉ của giao thức không đổi.

Bộ định tuyến được chia thành 2 loại, tuỳ theo cách sử dụng chúng. Bộ định tuyến cục bộ (Local Router) nối các đoạn mạng ở gần nhau. Hai bộ định tuyến ở xa nhau (Remote Router) nối hai đoạn mạng ở xa qua các kênh truyền thông.

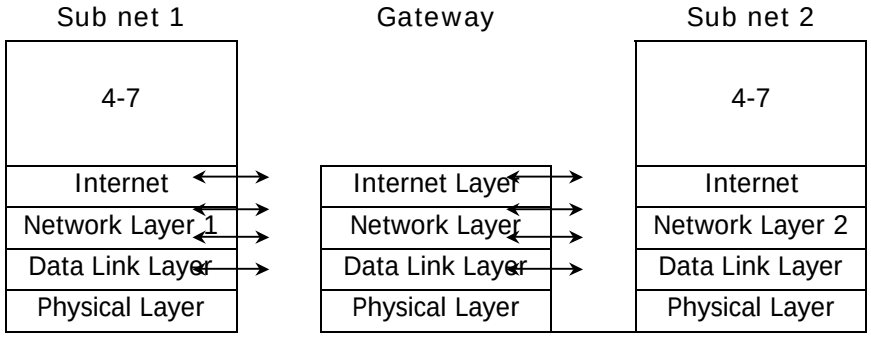
5.2.1.5 Bộ chuyển mạch

Chức năng chính của bô chuyển mạch (switch) là cùng một lúc duy trì nhiều cầu nối giữa các thiết bị mạng bằng cách dựa vào một loại đường truyền xương sống (backbone) nội tại tốc độ cao. Switch có nhiều cổng, mỗi cổng có thể hỗ trợ toàn bộ Ọthernet LAN hoặc Token Ring. Bộ chuyển mạch kết nối một số LAN riêng biệt và cung cấp khả năng lọc gói dữ liệu giữa chúng.

Các bộ chuyển mạch là loại thiết bị mạng mới, hiện đang được sử dụng rộng rãi vì Switch cho phép chuyển sang chế độ truyền không đồng bộ ATM.

5.2.1.6 Gateway

Hoạt động ở mức mạng, thực hiện ghép nối với WAN. Nguyên lý chung của nối kết này là tạo ra 1 tầng "*liên mạng*" (internet) chung trong tất cả các kiến trúc của mạng con tham gia nối kết. Tầng liên mạng thường là tầng con nằm ngay trên tầng 3 mô hình OSI.



Hình 5-9. Sơ đồ kiến trúc của gateway trong mô hình OSI.

Tầng con Internet được cài đặt trong tất cả các trạm cũng như trong các giao diện kết nối (gateway), Tầng này cung cấp dịch vụ truyền thông liên mạng với hai chức năng chính :

- Chuyển đổi các đơn vị dữ liệu của giao thức (Protocol Data Unit - PDU)
- Chọn đường đi cho các PDU này.

Các gói tin ở tầng con Internet lưu thông trong mạng theo phương pháp '*gói/bóc*' (encapsulation/decasulation). Khi một datagram được truyền từ mạng con này sang mạng con khác thông qua gateway thì nó được bổ sung thêm vào (hoặc tách ra) các phần thông tin điều khiển cần thiết tương ứng với các mạng con.

5.3 Giao thức liên mạng IP

Giao thức IP (Internet Protocol) hoạt động ở tầng mạng, cung cấp dịch vụ dữ liệu không liên kết (connectionless) cho nhiều giao thức liên kết dữ liệu khác. Đơn vị dữ liệu dùng trong giao thức IP được gọi là *datagram*, hay còn gọi là khung tin IP.

- Chức năng của giao thức IP :

- Định nghĩa gói tin Datagram là đơn vị dữ liệu cơ bản của việc truyền tin trên mạng Internet.
- Xác định mô hình đánh địa chỉ cho các khung tin và quản lý các quá trình trao đổi, xử lý các khung tin này.
- Chọn đường cho các datagram trên mạng
- Cung cấp cơ chế trên gói tin trên mạng hiệu quả nhất.
- Phân đoạn và tổng hợp các gói tin.

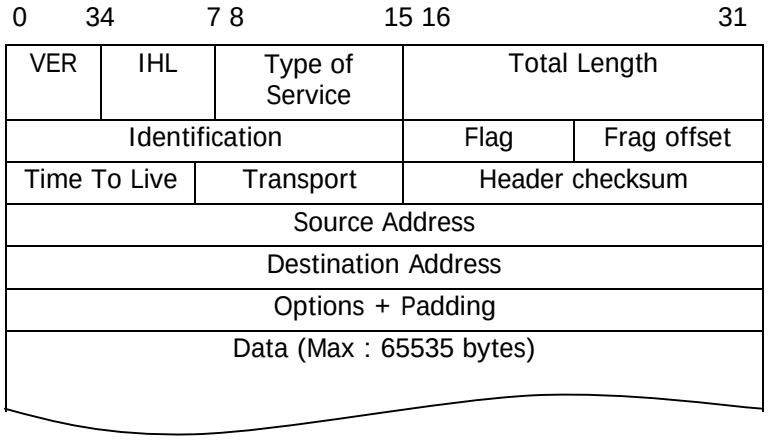
- Tính chất của giao thức IP :

- Hoạt động theo phương thức không kết nối : IP không chuyển các thông tin điều khiển trước khi truyền dữ liệu.

- Không tin cậy : giao thức IP không có khả năng phát hiện và khắc phục lỗi., không quan tâm đến vấn đề dữ liệu có được nhận một cách chính xác hay không. Do đó, các gói dữ liệu có thể bị thất lạc, bị trùng lặp, bị chuyển chậm hoặc đi không đúng thứ tự, mỗi gói dữ liệu được xử lý độc lập với nhau và có thể gửi theo những đường định tuyến khác nhau.

5.3.1 Cấu trúc khung tin IP

IP Header được gắn cho mỗi datagram, chứa các thông tin cần thiết cho sự hoạt động của gói tin trên mạng. Cấu trúc khung tin IP như hình sau :



Hình 5-10. Cấu trúc khung tin IP.

VER (4 bit)

Chứa phiên bản giao thức IP đang dùng. Phiên bản hiện nay là IPV4.

Một phần của giao thức IP quy định rằng phần mềm nhận dữ liệu trước tiên phải kiểm tra phiên bản của IP trong các khung tin đến, trước khi phân tích tiếp phần còn lại của Header và dữ liệu. Nếu như không đúng phiên bản thì lớp IP của máy nhận sẽ từ chối và bỏ qua toàn bộ nội dung của khung tin đến.

IHL (Internet Header length) (4 bit)

Chứa chiều dài của Header IP do máy gửi dữ liệu tạo nên, chiều dài này được tính theo các word có chiều dài 32 bit. Header ngắn nhất có chiều dài là 5 word (20 byte), nhưng do việc dùng các trường lựa chọn có thể làm tăng chiều dài của Header lên đến 6 word (24 byte). IHL dùng để giao thức IP được vị trí kết thúc của Header và bắt đầu phần dữ liệu của khung tin.

Type of Service - Loại dịch vụ (8 bits)

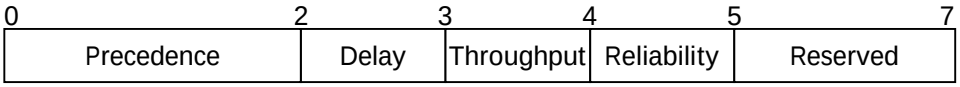
Trường này chứa các thông tin về quyền ưu tiên của việc truyền datagram và các ảnh hưởng có thể xảy ra trong quá trình truyền các datagram

đó. IP chuẩn không yêu chỉ ra các hành động cụ thể dựa trên giá trị của trường *Type of Service*. IP chỉ định sử dụng nó trong việc thiết lập các tuỳ chọn cho các mạng con và nó sẽ truyền qua trong bước nhảy tới.

Ví dụ, việc truy nhập vào mạng Token Ring cần thiết có các mức độ ưu tiên được xác định. IP có thể chuyển các mức độ ưu tiên của nó sang các mức độ ưu tiên tương ứng của mạng Token Ring.

Một số máy tính và bộ chọn đường (*router*) không quan tâm đến giá trị của trường này trong khi một số khác lại dựa vào đây để quyết định đường truyền.

Cấu trúc của trường như sau :



Cấu trúc của trường Type of Service

Precedence (3 bít) : chỉ thị về quyền ưu tiên gửi datagram, cụ thể là :

111 - Network Control (cao nhất) 011 - Flash
110 - Internetwork Control 10 - Immediate
101 - CRITIC/ECP 001 - Priority
100 - Flag Overrride 000 - Routime (thấp nhất)

D (Delay) - 1 bit : chỉ độ trễ yêu cầu
D = 0 độ trễ bình thường
D = 1 độ trễ thấp

T (Throughput) - 1 bit : chỉ thông lượng yêu cầu
T = 0 thông lượng bình thường
T = 1 thông lượng cao

R (Reliability) - 1 bít chỉ độ tin cậy yêu cầu
R = 0 độ tin cậy bình thường
R = 1 độ tin cậy cao

Ba bit đầu tiên của trường này là để chỉ ra quyền của khung tin đó, với các giá trị từ 0 (bình thường) đến 7 (Mạng điều khiển). Nếu giá trị của phần này càng cao thì khung tin đó càng quan trọng và trên lý thuyết thì khung tin này phải được chuyển đến đích nhanh hơn. Nhưng trên thực tế thì TCP/IP và các phần cứng dùng giao thức TCP/IP đều bỏ qua trường này và coi tất cả các khung tin có độ ưu tiên như nhau.

Ba bit tiếp theo là ba cờ 1-bit để điều khiển thời gian trễ, độ tin cậy, và thông lượng (throughput) của khung tin. Nếu tất cả các bit đều là 0 thì có nghĩa là đặt ở chế độ bình thường. Nếu bit thứ nhất là 1 thì có nghĩa là thời gian trễ thấp, truyền nhanh và độ tin cậy cao cho từng cờ. Còn hai bit còn lại của trường này không dùng. Phần lớn các bit của trường này đều bị bỏ qua khi thực hiện IP, và tất cả các khung tin đều được đặt thời gian trễ, thời gian truyền, và độ tin cậy như nhau.

Trong thực tế, hầu hết tất cả các bit của trường loại dịch vụ đều được đặt về giá trị 0 bởi vì sự khác nhau về quyền, thời gian trễ, thời gian truyền, độ tin cậy giữa các máy hầu như không tồn tại trừ khi một mạng mới được thành lập.

Total Length (16 bits) - Chiều dài gói tin

Trường này cho biết toàn bộ chiều dài của khung tin (datagram) bao gồm phần Header và phần dữ liệu, đơn vị tính bằng byte. Độ lớn của trường này là 16 bit do đó mà chiều dài của khung tin tối đa là 65535 byte.

Indentification (16 bits) -Trường định danh

Trường này chứa một giá trị đặc trưng do máy gửi khung tin tạo ra, cùng với các tham số khác (như Source Address và Destination Address), tham số này dùng để định danh duy nhất một khung tin trong khoảng thời gian nó tồn tại trên liên mạng.

Số trong trường này được cần đến khi sắp xếp các khung tin để đảm bảo rằng các khung tin không bị lẫn lộn với nhau. Khi lớp IP nhận được một đoạn dữ liệu từ các lớp cao hơn thì nó sẽ gán các số định danh này vào. Nếu như khung tin đã được tách (bằng kỹ thật tách thông tin) thì tất cả các khung tin sẽ mang cùng một số định danh như nhau.

Flags (3 bits) - Các cờ

Trường này có chiều dài 3 bit, liên quan đến sự phân đoạn các datagram.

Bit 0 : Dùng để dự trữ - chưa sử dụng, luôn có giá trị 0

Bit 1 : (DF) = 0 (May Fragment)

=1 (Don’t Fragment)

Bit 2 : (MF) = 0 (Last Fragment)

=1 (More Fragment)

Nếu như cờ DF có giá trị là 1 thì có nghĩa là khung tin không thể tách ra được trong bất cứ trường hợp nào. Nếu như mà phần mềm của lớp IP hiện tại không thể gửi khung tin đến nới nhận nếu như không tách ra, mà hiện tại bit cờ đang là 1 thì khi đó khung tin sẽ bị huỷ bỏ và một thông báo lỗi được gửi đến thiết bị phát.

Nếu router không thể truyền nguyên cả một datagram mà bit này được thiết lập bằng 1 thì datagram đó sẽ bị loại bỏ và nó sẽ có một thông báo lỗi gửi đến máy phát. Bất kì một người quản lý mạng nào cũng có thể sử dụng cách này để kiểm tra độ lớn của các datagram có thể được truyền trên các phần khác nhau trên mạng kết hợp.

Nếu như cờ MF là 1 có nghĩa là khung tin hiện tại vẫn đang còn các gói tin khác nữa đang đến, do đó mà phải cần đến việc sắp xếp lại để khôi phục lại message ban đầu. Khung tin cuối cùng đến sẽ lớn hơn các khung tin bình thường vì nó còn chứa thêm phần MF=0 để báo cho máy nhận biết là đã hết các khung tin cần thiết không cần phải đợi thêm nữa. Có thể là các khung tin đến không đúng với thứ tự chúng đã được phát đi, do đó cờ MF còn được dùng cùng với trường Fragment Off để chỉ cho máy nhận được thứ tự của toàn bộ message ban đầu.

Fragment Offset (13 bits)

Nếu mà cờ MF bằng 1 (tức là có sự tách thông tin từ một khung tin lớn), khi đó fragment offset chứa .vị trí của các message con trong message ban đầu trong khung tin hiện thời. Điều này cho phép IP sắp xếp lại các khung tin thành message ban đầu t theo đúng trật tự.

Offset thường được để ở đầu message. Trường này có chiều dài là 13 bit, do vậy offset được tính theo đơn vị 8 byte, tương ứng với gói lớn nhất là 65535 byte. Việc dùng số định danh để chỉ rằng khung tin đến là thuộc bản tin nào, lớp IP ở máy nhận có thể dùng fragment offset để sắp xếp lại message ban đầu.

TTL (Time to Live - Thời gian sống)

Trường này cho biết khoảng thời gian tính bằng giây mà một khung tin có thể tồn tại trên mạng trước khi nó bị huỷ bỏ. Giá trị này được nút gửi khung tin đi ấn định.

Các chuẩn của TCP/IP quy định rằng trường TTL phải được giảm đi ít nhất là 1 giây cho mỗi nút xử lý khung tin đó, thậm chí là thời gian xử lý có thể nhỏ hơn 1 giây. Khi một gateway nhận được một khung tin thì thời gian đến được đính vào khung tin do đó nếu như khung tin đó phải chờ để được xử lý. Bởi vậy nếu một gateway nào đó mà bị quá tải và không thể lấy khung tin về, khi đó bộ đếm thời gian của trường TTL sẽ tự động giảm đi trong quá trình chờ để được xử lý. Nếu trường TTL giảm về 0 thì khi đó khung tin đó phải được nút hiện thời hủy bỏ, sẽ có một thông báo gửi về máy gửi.

Hầu hết các TCP/IP cài đặt giá trị trường TTL khoảng 60 hoặc cao hơn, nghĩa là datagram có thể đi qau 60 router hay hop để đến đích. Trường TTL được thiết kế để tránh việc các gói dữ liệu cứ chuyển vòng quanh trên mạng mà không có đường ra.

Giao thức giao vận (Transport Protocol)

Trường này chứa số định danh của giao thức giao vận mà đã xử lý khung tin. Số định danh này do trung tâm thông tin mạng Internet NIC ấn định. Hiện nay đã có khoảng 50 giao thức giao vận được ấn định. Hai giao thức quan trọng nhất là : ICMP (Internet Control message Protocol) và TCP.

Header checksum

Dùng để tính checksum của trường Header để làm cho quá trình xử lý thông tin được nhanh hơn. Do trường TTL bị giảm đi 1 giây mỗi khi được xử lý, trường checksum cũng thay đổi tại các máy mà khung tin đi qua. Thuật toán checksum là một thuật toán nhanh và có hiệu quả, nhưng có một số trường hợp bị sai chẳng hạn mất hoàn toàn một từ 16 bit mà 16 bit này đều bằng 0. Tuy nhiên trường checksum do cả TCP và UDP để đóng gói, các lỗi này sẽ được phát hiện khi khung tin được tập hợp để truyền trên mạng.

Source Address (32 bits) : chứa địa chỉ IP 32 bit của máy gửi.

Destination Address (32 bits) : chứa địa chỉ IP 32 bit của máy nhận.

Hai trường trên được tạo ra cùng với khung tin và không bị thay đổi trong quá trình truyền.

Options (32 bits) - Phần lựa chọn

Phần lựa chọn được tạo ra từ một vài mã mà các mã này có độ dài có thể thay đổi được. Nếu như có nhiều lựa chọn trong khung tin, thì các lựa chọn đó được đặt liên tục nhau trong phần Header của IP. Tất cả các lựa chọn này được điều khiển bằng một byte có ba trường: ***Cờ copy*** có độ dài 1 bit, ***loại lựa chọn*** có độ dài 2 bit, và ***trường số lựa chọn*** có độ dài 5 bit. Trường cờ copy được dùng để quy định là lựa chọn sẽ được thực hiện như thế nào nếu ở một gateway nào đó cần đến kỹ thuật tách thông tin. Nếu như cờ này có giá trị là 0 thì có nghĩa là lựa chọn đó sẽ được copy vào khung tin thứ nhất mà không copy vào các khung tin tiếp theo sau. Nếu như cờ này có giá trị là 1 thì có nghĩa là lựa chọn đó sẽ được sao chép vào tất cả các khung tin.

Các lựa chọn quan trọng là Record route và Timestamp.

Record route

Trường Record Route (*Bản ghi chọn đường)* chứa danh sách dự trữ của các route mà datagram đã đi qua trên đường tìm tới đích. Mỗi lần đi qua một router thì trường này sẽ bổ sung một địa chỉ của router đó vào danh sách của nó. Độ dài của trường này do máy nguồn xác lập, do đó rất có thể là nó sẽ bị

đầy trước khi datagram tìm được đến đích. Trong trường hợp này thì các địa chỉ của các router sau sẽ không được thêm vào danh sách của nó.

Timestamp : Có 3 định danh cho trường Timestamp. Trường này có thể chứa:

- Danh sách của 32 bit Timestamp.
- Danh sách của địa chỉ IP và các cặp Timestamp tương ứng.

Danh sách của các địa chỉ cho trước bởi máy nguồn. Một nút bất kỳ được ghi vào trường này chỉ khi địa chỉ của nó là mục kế tiếp trong danh sách này. Trường này có thể bị đầy nếu rơi vào hai trường hợp đầu, trong trường hợp này sẽ có trường ghi tràn (overllow field) dùng để đếm số nút mà không thể ghi vào timestamp được.

Padding (Độ dài thay đổi)

Nội dung của phần **Padding** phụ thuộc vào phần **Options** như thế nào. Phần Padding thường được dùng để bảo đảm rằng chiều dài Header của khung tin luôn là một số nguyên bội số của 32.

Data : Vùng dữ liệu có độ dài thay đổi, nhưng luôn là bội số của 8 bits, và tối đa là 65535 bytes.

5.3.2 Địa chỉ IP

Mỗi thiết bị nối vào mạng TCP/IP được gán một địa chỉ IP duy nhất (mỗi card mạng sẽ có địa chỉ IP riêng). Khi sử dụng mạng cục bộ không kết nối với các mạng khác, người sử dụng có thể gán địa chỉ IP tuỳ ý cho các máy trạm. Tuy nhiên, đối với các site Internet thì địa chỉ IP phải được cung cấp từ trung tâm quản lý thông tin mạng trên thế giới (NIC - Network Information Center).

Địa chỉ của IP có độ dài 32 bit, được chia làm 4 phần, mỗi phần 1 byte, phân cách nhau bằng dấu chấm. Dạng tổng quát : ***x.y.z.t*** với $0 \leq x,y,z,t \leq 255$

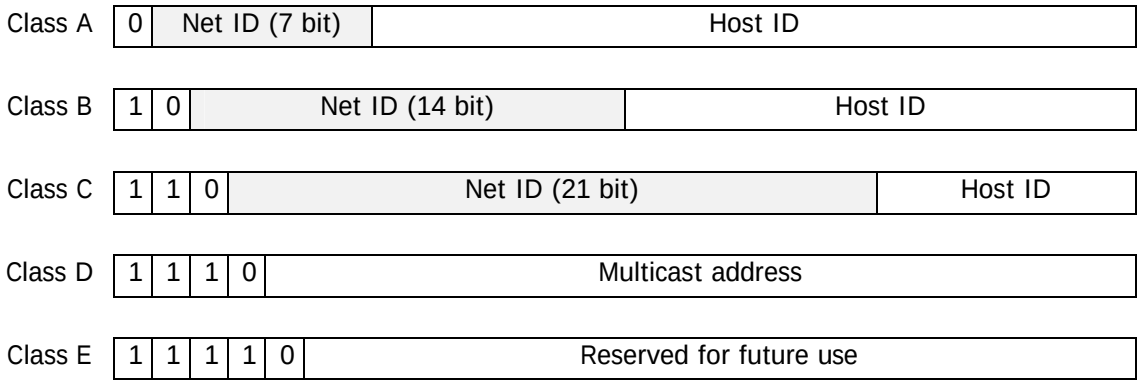
Ví dụ: 128.83.12.14 hoặc 0x80530C0E Hex.

Địa chỉ IP bao gồm hai phần thông tin: địa chỉ mạng (network address) và địa chỉ máy (host address): NetworkID.HostID

Khi đề nghị NIC cung cấp địa chỉ IP ta sẽ không nhận được địa chỉ tương ứng của máy trạm, thay vào đó là địa chỉ mạng và ta có quyền gán địa chỉ cho các máy trạm của mạng trong phạm vi địa chỉ được cung cấp.

5.3.2.1 Các lớp địa chỉ IP

Địa chỉ IP thuộc một trong E lớp địa chỉ, từ lớp A đến E. Các lớp địa chỉ nhằm để phân loại các mạng có quy mô khác nhau.



Hnh 5-11. Các lớp địa chỉ IP.

1. Lớp A (1≤ x ≤ 126) : NetworkID= x, HostID=y.z.t

- Cho phép định danh 126 mạng, với tối đa 2^{24} (= 167.772) máy trạm trên mỗi mạng, lớp A giới hạn số subnetwork trong Internet.
- Các mạng lớp A thuộc loại mạng diện rộng (very large), như mạng quốc gia

2. Lớp B (128≤ x ≤ 191) : NetworkID= x.y, HostID=z.t

- Cho phép định danh đến 16384 mạng, với tối đa 2^{16} (=65.536) host trên mỗi mạng.
- Mạng lớp B thuộc loại mạng trung bình như mạng University Campuses.

3. Lớp C (192 ≤ x ≤ 223) : NetworkID= x.y.z, HostID=t

- Giới hạn số trạm trong mạng lớn nhất là 256, có 21 bit cho địa chỉ mạng. Cho phép định danh đến 2 triệu mạng, với tối đa 254 host trên mỗi mạng.
- Mạng lớp C được sử dụng cho các loại LAN, như các mạng Enterprise-wide.

4. Lớp D (224≤ x ≤ 239)

- Địa chỉ lớp D dùng cho các giao thức đặc biệt (Internet Group management Protocol - IGMP) và các giao thức khác.

5. Lớp E (240≤ x ≤ 255) : Để dành cho sự phát triển về sau.

- Các máy trong cùng một mạng phải có địa chỉ mạng giống nhau.
- Các mạng khác nhau có địa chỉ mạng khác nhau.

5.3.2.2 Các địa chỉ IP đặc biệt

1. Địa chỉ quay vòng : 127.y.z.t

Tất cả các gói tin được gửi đến địa chỉ 127.0.0.0 sẽ được gửi ngược trở lại máy tính. Gói tin này được sao chép từ nơi truyền đến bộ đệm nơi nhận trên cùng một máy tính. Địa chỉ loopback có thể được sử dụng như một địa chỉ kiểm tra

nhanh xem phần mềm TCP/IP có được cấu hình thích hợp. Trên hệ điều hành Windows địa chỉ loopback là 127.0.0.1 còn Unix là 127.1.*.

2. Mặt nạ mạng (Netmask)

Mặt nạ mạng của một địa chỉ IP là một giá trị 32 bits trong đó các bit tương ứng với phần địa chỉ mạng bằng 1, các bit của phần máy bằng 0.

Ví dụ : Địa chỉ IP lớp B có mặt nạ mạng là 255.255.255.0 sẽ cho địa chỉ mạng con là 180.10.15.0

3. Địa chỉ quảng bá (broadcast address)

Địa chỉ này có các bit của phần HostID bằng 1, được sử dụng khi muốn chuyển một gói tin đến mọi máy tính trong mạng con.

Ví dụ một mạng con có địa chỉ là 180.10.0.0 sẽ có địa chỉ quảng bá là 180.10.255.255. Tương tự, một mạng con có địa chỉ là 180.10.15.0 sẽ có địa chỉ quảng bá là 180.10.15.255.

Đặc biệt địa chỉ 255.255.255.255 quảng bá cục bộ (local broadcast) hay còn gọi là limited broadcast có thể sử dụng trong các LAN.

Địa chỉ 0.0.0.0 cũng được sử dụng trong bảng định tuyến để chỉ đến điểm vào mạng cho địa chỉ bộ định tuyến mặc định.

5.4 Phân chia mạng con

Để thuận tiện cho việc quản lý và định hướng dữ liệu trên mạng lớn, người ta thường tổ chức mạng IP theo cơ chế địa chỉ phân cấp : mỗi mạng được chia nhỏ thành nhiều mạng con, mỗi mạng con thực hiện các dvc về địa chỉ trong nội bộ mạng đó. Sự phân cấp này cho phép giảm khối lượng công việc chọn đường cho các gói tin trong toàn liên mạng.

Mỗi mạng con chịu trách nhiệm cho việc chọn đường cho các gói tin IP trong mạng của mình, các gói tin này được nhận ra nhờ phần địa chỉ mạng của nó. Trong các mạng loại A, B, C thì phần địa chỉ này có độ dài cố định. Tuy nhiên, để tạo sự linh hoạt trong việc phân chia mạng con thì địa chỉ mạng có thể mở rộng sang các bit của địa chỉ máy. Đó là kỹ thuật phân chia mạng con.

Ví dụ một mạng loại B có địa chỉ mạng là 203.160.9.0 và mặt nạ mạng là 255.255.255.0 (địa chỉ mạng dài 24 bit). Người ta cần chia mạng này thành 4 mạng cục bộ riêng, do đó sẽ lấy thêm 2 bit cho địa chỉ mạng (26 bit). Vậy ta có địa chỉ các mạng con này là :

Địa chỉ mạng 1 :

203	160	9	0
11001011	10100000	00001001	00000000

Địa chỉ mạng 2 :

203	160	9	64
11001011	10100000	00001001	01000000

Địa chỉ mạng 3 :

203	160	9	128
11001011	10100000	00001001	10000000

Địa chỉ mạng 4 :

203	160	9	192
11001011	10100000	00001001	11000000

Mặt nạ của các mạng con này là : 255.255.255.192

255	255	255	192
11111111	11111111	11111111	11000000

Việc phân chia mạng được tiến hành bởi người quản trị hệ thống và thường dựa trên ranh giới vật lý giữa các nhánh mạng. Khi có gói dữ liệu cần chuyển đi, bộ định tuyến sẽ dùng mặt nạ mạng để kiểm tra gói dữ liệu này thuộc mạng con nội bộ hay thuộc mạng ngoài. Sự phân chia mạng riêng thành các mạng con chỉ có ý nghĩa bên trong mạng đó.

Nếu kết nối Internet thông qua một mạng LAN, điều quan trọng là phải sử dụng đúng mặt nạ mạng. Cũng giống như địa chỉ IP, một mặt nạ mạng con có thể được gán một cách riêng lẻ hay có thể tự động thông qua DHCP (Dynamic Host Configuration Protocol).

5.5 Hoạt động của giao thức IP

Nếu địa chỉ đích của gói tin IP không nằm trên cùng mạng với máy chủ nguồn thì giao thức IP trong máy chủ hướng gói tin đến bộ định tuyến nội bộ. Nếu bộ định tuyến này không được nối đến mạng đích, gói tin sẽ được gởi đến một bộ định tuyến khác. Cứ thế cho đến khi tới trạm đích. Việc quy định truyền theo đường truyền nào của router dựa trên bảng đường truyền (*routing table*). Các bộ định tuyến có thể phát hiện :

- Một mạng mới đã được thêm vào liên mạng
- Đường dẫn đến trạm đích đã bị hỏng

Các bước thực hiện bởi một thực thể IP như sau :

- *Đối với thực thể IP ở trạm nguồn*

- Khi nhận được lệnh SEND từ tầng trên, nó thực hiện các bước như sau:
- Tạo một IP datagram dựa trên các tham số của lệnh SEND
- Tính checksum và ghép vàp phần đầu của datagram
- Ra quyết định chọn đường
- Chuyển datagram xuống tầng dưới

• *Đối với gateway*

- Khi nhận được datagram quá cảnh, nó thực hiện các tác động như sau :
- Tính checksum, nếu không đúng thì loại bỏ datagram
- Giảm giá trị tham số thời gian tồn tại. Nếu hết thời gian thì loại bỏ datagram
- Ra quyết định chọn đường
- Phân loại datagram nếu cần
- Kiến tạo lại phần đầu IP bao gồm giá trị mới của vùng TTL, checksum, Fragmentation.
- Chuyển datagram xuống tầng dưới để truyền qua mạng.

• *Tại trạm đích*

- Tính checksum, nếu không đúng thì loại bỏ datagram.
- Tập hợp các đoạn của datagram.
- Chuyển dữ liệu và các tham số điều khiển lên tầng trên.

Như vậy, do gói tin IP không sửa đổi, đơn giản nên hiệu suất đường truyền cao. Vì gói tin IP cung cấp dịch vụ giao nhận gói tin không tin cậy nên cần có giao thức ICMP để hỗ trợ, các bản tin ICMP được đóng gói và chuyển tải trong các gói tin IP. Tầng TCP đảm nhận việc bảo đảm các datagram được truyền đến đích một cách an toàn và đầy đủ.

5.6 Các giao thức liên quan đến IP

5.6.1 Giao thức phân giải địa chỉ ARP

Địa chỉ IP được dùng để *định danh các host và mạng ở tầng mạng* của mô hình OSI, và chúng không phải là các địa chỉ vật lý (hay địa chỉ MAC) của các trạm đó trên cùng một mạng cục bộ (Ethernet, Token Ring, ...). Trên một LAN như vậy, hai trạm chỉ có thể liên lạc với nhau nếu chúng biết địa chỉ vật lý của nhau.

Vấn đề đặt ra là phải thực hiện ánh xạ địa chỉ IP (32 bits) và địa chỉ vật lý (48 bits) của một trạm. Giao thức ARP (Address Resolution Protocol) đã được xây dựng để chuyển đổi từ địa chỉ IP sang địa chỉ vật lý khi cần thiết. Ngược lại, giao

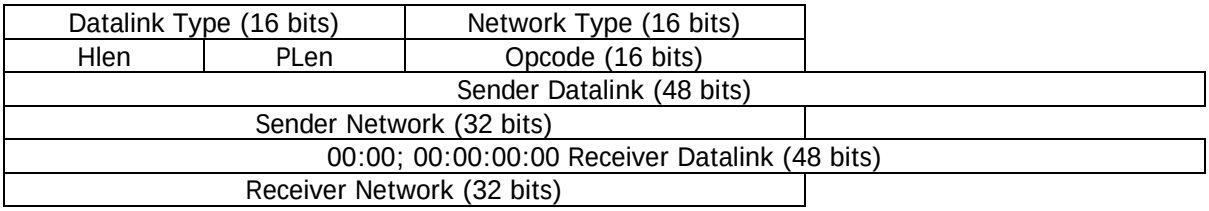
thức RARP (Reserse Address Resolution Protocol) được dùng để chuyển đổi từ địa chỉ vật lý sang địa chỉ IP.

Cả hai giao thức ARP và RARP đều không phải là bộ phận của IP, IP sẽ dùng đến chúng khi cần.

Mỗi ghép nối mạng có *địa chỉ giao thức mạng* (IP address) và *địa chỉ giao thức liên kết dữ liệu* (Datalink Protocol Address) riêng. Do đó cần có bảng ánh xạ giữa hai địa chỉ này (địa chỉ ảo và địa chỉ vật lý). Bảng địa chỉ này có thể làm bằng tay, nhưng do khối lượng địa chỉ lớn, tăng khá nhanh, nên người ta giải quyết thông qua thủ tục "*Tìm giải pháp cho địa chỉ*" (Address Resolution Protocol -ARP).

Các gói tin ARP được đóng gói trong khung dữ liệu liên kết (data link frame). Đối với mạng Ethernet, kiểu trường (type field) sẽ là 0x0806.

ARP ánh xạ địa chỉ IP sang địa chỉ liên kết dữ liệu (datalink address). Trạm tin sẽ gửi gói tin yêu cầu ARP (request packet) với khuôn dạng gói tin như hình sau.



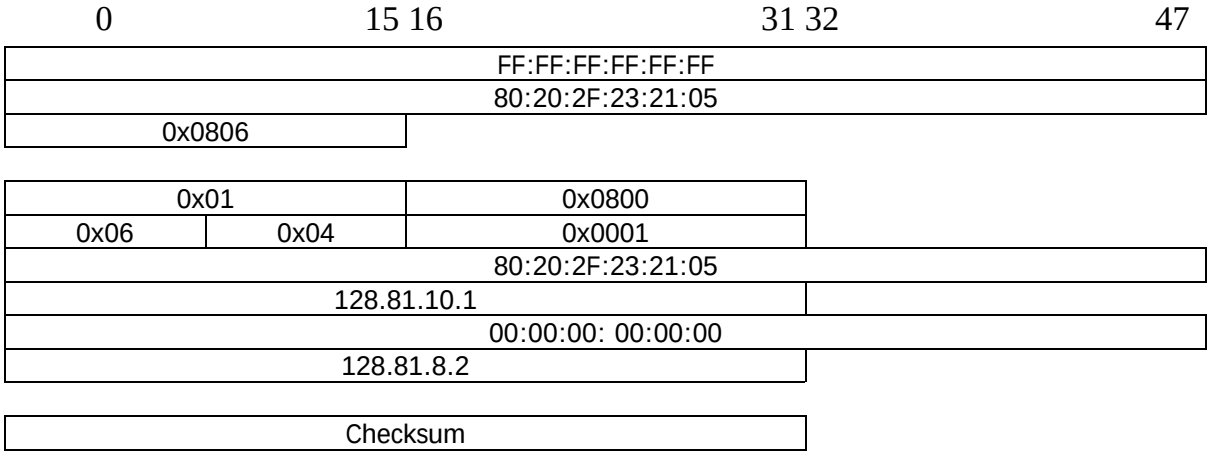
Hình 5-12. Khuôn dạng gói tin ARP.

- Data link type: Loại dữ liệu liên kết, với mạng Ethernet thì trường này có giá trị là 0x0001
- Network type : Loại địa chỉ mạng, Ethernet type used for IP (0x0800)
- Hlen : Độ rộng của phần địa chỉ dữ liệu liên kết, với mạng Ethernet độ rộng là 6 bytes
- Plen : Độ rộng của địa chỉ mạng, trong giao thức IP, phần này là 4 byte
- Opcode : Có giá trị là 0x0001 cho thủ tục yêu cầu ARP, 0x0002 cho ARP trả lời.
- Sender datalink and sender network : Địa chỉ vật lý và địa chỉ ảo (địa chỉ mạng) của người gửi
- Receive datalink and receive network : Địa chỉ vật lý và địa chỉ ảo (địa chỉ mạng) của người nhận

Ví dụ: Trạm A muốn gửi trạm B một gói tin IP. Cả hai máy A, B đều có cùng có địa chỉ mạng IP và cùng kết nối vào mạng Ethernet như hình sau :

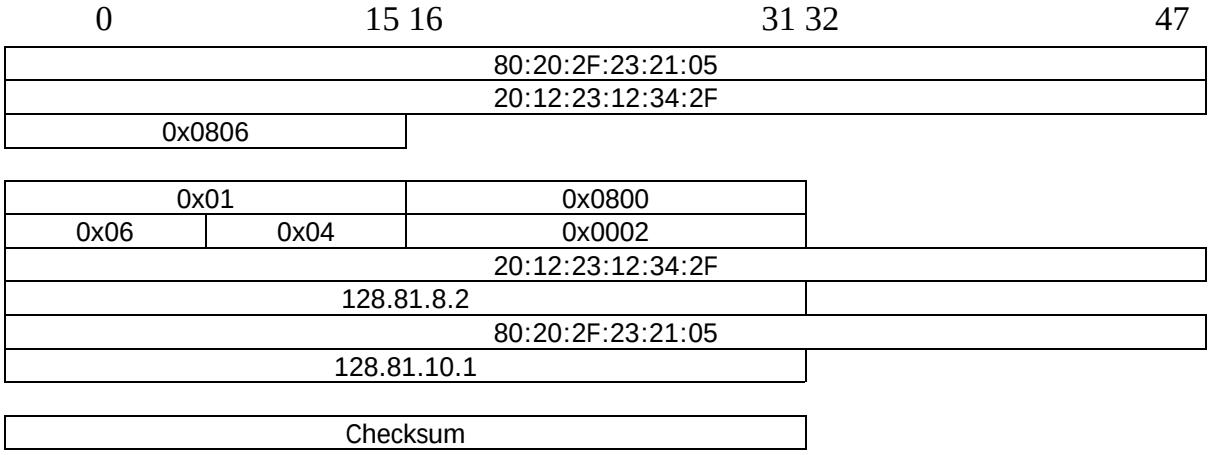
Trạm A biết được địa chỉ mạng của trạm B nhưng không biết địa chỉ vật lý của trạm B. Trạm A cần hỏi địa chỉ vật lý của trạm B để gửi tin. Khi đó trạm A phát đi một gói tin ARP yêu cầu (ARP request packet) đóng gói trong khung tin Ethernet.

- Quá trình gởi yêu cầu ARP



Hình 5-13. Khuôn dạng gói tin ARP yêu cầu.

Gói tin yêu cầu ARP (ARP request packet) được gửi tới các trạm, chỉ trạm B là đúng địa chỉ IP. Trạm B sẽ tạo ARP trả lời :



Hình 5-14. Khuôn dạng gói tin ARP trả lời.

Trạm B bổ sung IP_ to_Ethernet Address entry của host A và ARP cache của B

Trạm A bổ sung IP_to_Ethernet Address entry của host B và ARP cache của A

Như vậy bảng ánh xạ tự động bổ sung những đường dẫn (entry) mới mà nó biết, đồng thời cũng huỷ bỏ những đường dẫn (entry) mà nó không dùng đến.

5.6.2 Giao thức RARP (Reverse Address Resolution Protocol)

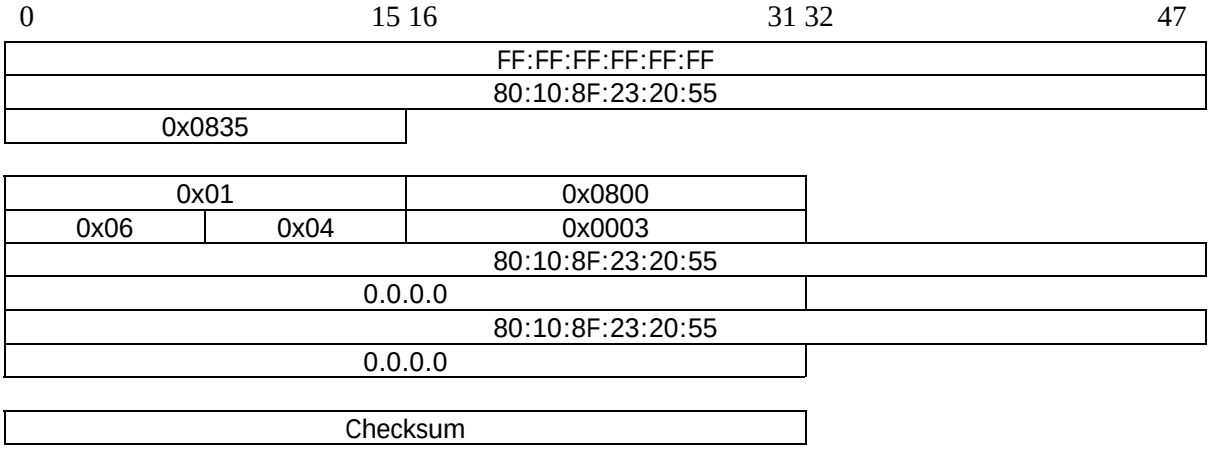
Đôi khi ta cần ánh xạ ngược lại.

Ví dụ một trạm không ổ đĩa, biết địa chỉ vật lý (datalink address) tức là địa chỉ card mạng giữ ở bộ nhớ ROM, nhưng không biết địa chỉ IP vì không có ổ đĩa. Khi này cần ánh xạ từ địa chỉ vật lý sang địa chỉ mạng.

Ta cũng làm như trên, nhưng thay kiểu trường từ 0x0806 bằng 0835.

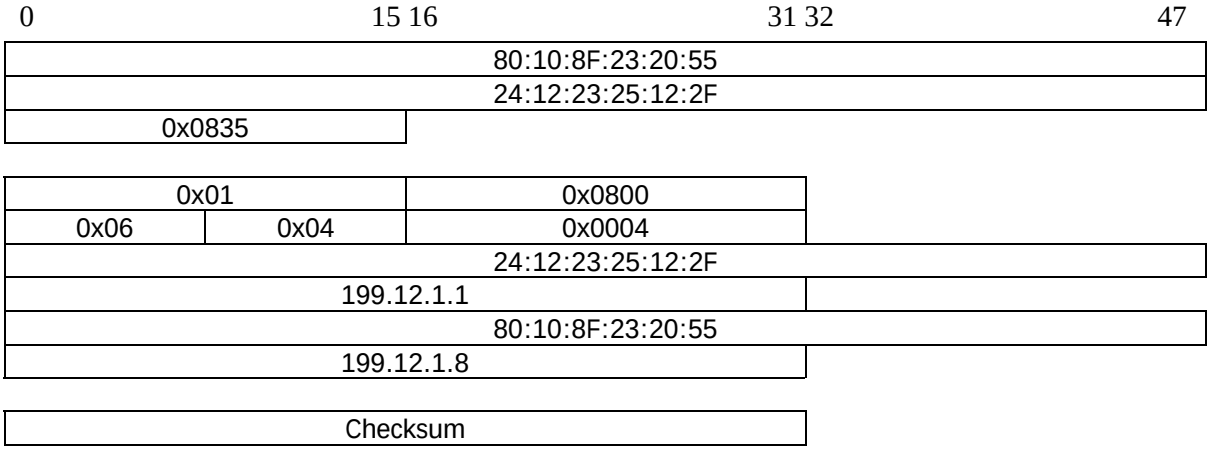
Yêu cầu chuyển đổi (reverse request) là 0x0003 và trả lời chuyển đổi (reverse reply) là 0x0004.

- Quá trình gởi yêu cầu RARP



Hình 5-15. Khuôn dạng gói tin RARP yêu cầu.

- Quá trình gởi trả lời RARP



Hình 5-16. Khuôn dạng gói tin trả lời RARP .

5.6.3 Giao thức ICMP

Giao thức ICMP (Internet Control Message Protocol) thực hiện truyền các thông tin điều khiển (các báo cáo về các tình trạng lỗi trên mạng, ...) giữa các

gateway hoặc các máy chủ trên liên mạng theo giao thức IP. Tình trạng lỗi có thể là: một datagram không thể đến được đích của nó, hoặc một router không đủ bộ nhớ đệm để lưu và chuyển một datagram, ... Một thông báo ICMP được khởi tạo và chuyển cho IP. IP sẽ bọc (*encapsulate*) thông báo đó với một IP header và truyền đến cho router hoặc trạm đích.

5.6.3.1 Ccác thành phần của thông báo ICMP hỗ trợ xác định lỗi và truy vấn

Thông báo ICMP được chia làm 2 loại: thông báo lỗi ICMP và thông báo truy vấn ICMP.

Các thông báo ICMP khác nhau về định dạng tuỳ vào chức năng của từng loại, nhưng kiến trúc tổng quát bao gồm 2 phần: phần đầu (ICMP header) và phần dữ liệu (ICMP data).

Phần đầu của thông báo ICMP luôn bắt đầu bằng 3 trường:

a. TYPE: 8 bits, xác định loại thông báo ICMP.

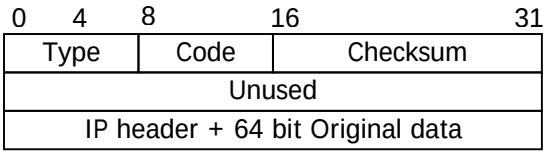
b. CODE: 8 bits, cung cấp thông tin chi tiết của từng loại thông báo ICMP.

c. CHECKSUM: 16 bits, xác định sự toàn vẹn dữ liệu trong quá trình truyền.

1. Các thông báo lỗi ICMP

Về mặt kỹ thuật, ICMP được thiết kế để cung cấp các thông tin về trạng thái không ổn định và thực hiện thông báo các trường hợp lỗi phát sinh của hệ thống phần cứng cũng như phần mềm làm ngăn chặn, huỷ bỏ quá trình gửi, nhận hoặc xử lý các datagram trên mạng Internet trước khi được chuyển đến đích cuối cùng.

Có 5 loại thông báo lỗi ICMP trong bảng I.1 và các thông báo có dạng chung như hình sau :



Hình 5-17. Dạng chung thông báo lỗi của ICMP

Type	Thông báo lỗi ICMP
3	Destination Unreachable
4	Source Quench
5	Redirect
11	Time Exceeded
12	Parameter Problem

Bảng I.1: Các loại thông báo lỗi của ICMP

Original IP header: 20-60 bytes chứa IP header của gói bị lỗi.

Original data: 8 bytes, chứa nội dung 64 bits đầu tiên của gói dữ liệu bị lỗi.

- *Destination Unreachable*

Các thông báo ICMP Destination Unreachable được tạo ra khi không thể chuyển đến 1 đích được xác định trong IP datagram. Bao gồm các loại lỗi sau:

Code	Nội dung thông báo ICMP
0	Network Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation needed and DF flag set
5	Source Route Fail
6	Destination Network unknown
7	Destination Host unknown
8	Source Host Isolated
9	Communication with Destination Network is Administratively Prohibited
10	Communication with Destination Host is Administratively Prohibited

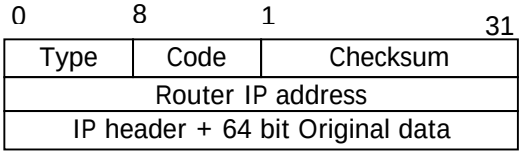
Bảng 5-1. Các lỗi của ICMP Destination Unreachable

- *Source Quench :* Khi vùng đệm của hệ thống nhận không đủ chỗ trống lưu trữ, hệ thống sẽ phát ra thông báo Source Quench. Trường CẶDQ của thông báo này luôn luôn nhận giá trị 0.
- *Redirect :* Một thông báo ICMP Redirect được tạo ra bởi 1 router trong trường hợp nó nhận thấy rằng một máy tính đang sử dụng con đường định tuyến không tối ưu.

Trường CẶDQ nhận 4 giá trị trong bảng và có định dạng như hình sau:

Code	Nội dung
0	Redirect for the network (or subnet)
1	Redirect for the host
2	Redirect for the type of service and network
3	Redirect for the type of service and host

Bảng 5-2. Các lỗi của ICMP Redirect



Hình 5-18. Dạng ICMP Redirect

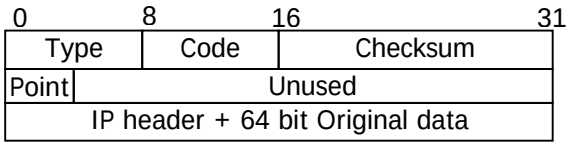
Router ip address là địa chỉ của bộ định tuyến mà máy nguồn sẽ dùng để trỏ máy đích.

- *Time Exceeded :* Router sẽ huỷ bỏ, không xử lý 1 datagram khi giá trị TTL của nó bằng 0 và phát ra một thông báo ICMP Time Exceeded. Có 2 loại ICMP Time Exceeded như sau:

Code	Nội dung
0	Bộ đếm thời gian sống TTL của 1 datagram bằng 0
1	Quá thời gian đợi để kết hợp các gói bị phân mảnh

Bảng 5-3. Các lỗi của ICMP Time Qxceeded.

- *Parameter Problem :* Thông báo này được gửi đi khi có lỗi xuất hiện ở phần các tham số chọn lựa của datagram gửi đến. Trường CẶDỌ của thông báo này nhận 3 giá trị trong bảng và có định dạng như hình sau :



Hình 5-19. Dạng ICMP Parameter Problem

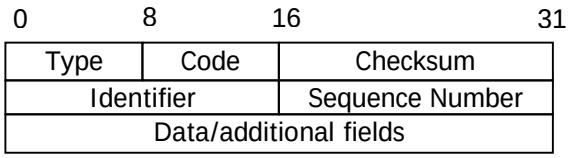
Code	Giải thích
0	Có một lỗi đặc biệt trong lược đồ dữ liệu.
1	Phần option của IP header chưa định nghĩa.
2	Lỗi Header Length và (hoặc) Total Packet Length trong IP header.

Bảng 5-4. Các lỗi của ICMP Parameter Problem

Pointer: xác định vị trí gây ra lỗi trong datagram.

2. Các thông báo truy vấn ICMP

ICMP được sử dụng trong việc khảo sát các đặc trưng chung của mạng với 2 loại thông báo request và reply. Có 8 loại thông báo truy vấn ICMP được liệt kê trong bảng và có định dạng như hình sau :



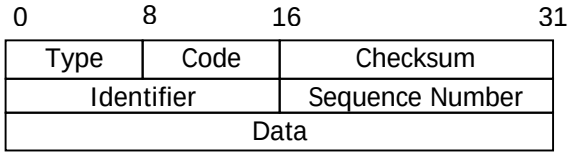
Hình 5-20. Dạng ICMP truy vấn.

Type	Loại thông báo
0	Echo Reply
8	Echo Request
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

Bảng 5-5. Các loại thông báo truy vấn ICMP.

- Identifier được sử dụng để phân biệt các thông báo được gởi đến các host khác nhau.
- Sequence number được sử dụng để phân biệt các thông báo được gửi đến cùng một host.
- Data/additional fields được dùng theo từng loại thông báo truy vấn ICMP.

- *Echo Request vµ Echo Reply*



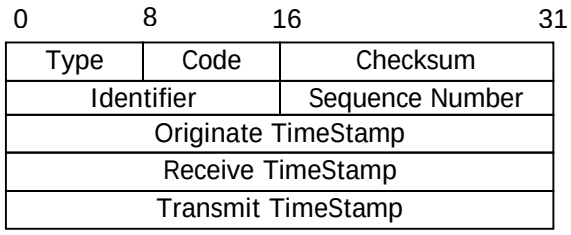
Hình 5-21. Dạng ICMP Echo Request & Reply.

Người ta sử dụng ICMP Echo để xác định xem một địa chỉ IP đích còn hoạt động hay không bằng cách gửi thông báo ICMP Echo Request đến hệ thống đích và chờ xem nếu nhận được thông báo ICMP Echo Reply thì sẽ xác định đích đấy vẫn còn hoạt động ngược lại thì đã bị down. Định dạng thông báo như trong hình sau :

Kích thước của DATA thay đổi tuỳ thuộc vào từng loại hệ điều hành. Trong hệ điều hành UNIX, kích thước của nó là 56 bytes, trong Microsoft Windows là 32 bytes,...

- *Timestamp Request và Timestamp Reply*

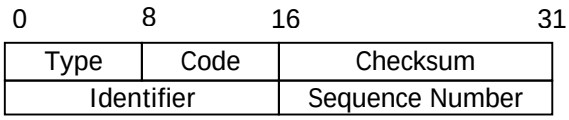
Mỗi máy đều có 1 đồng hồ riêng xác định thời gian vận hành của nó, quá trình hoạt động trong những hệ thống phần mềm phân tán thì sự khác biệt nhau lớn về thời gian giữa các máy tính sẽ gây ra nhiều vấn đề khó khăn. ICMP cung cấp một cơ chế cho phép lấy thời gian từ một máy khác và có định dạng như hình sau.



Hình 5-22. ICMP Timestamp Request & Reply

- Originate timestamp là thời gian máy nguồn thực hiện gửi báo.
- Receive timestamp là thời gian đầu tiên máy đích nhận được thông báo.
- Transmit timestamp là thời gian cuối bên đích xử lý thông báo và gửi đi.

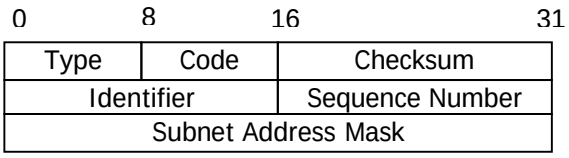
- *Information request và reply*



Hình 5-23. ICMP information request & reply

Được sử dụng nhằm hỗ trợ các hệ thống máy trạm không đĩa khi khởi động; cho phép các máy tính tìm ra địa chỉ Internet của chúng lúc khởi động hệ thống.

- *Address Mask Request và Reply*



H×nh 5-24. ICMP Ađress Mask Request & Reply

Để biết subnet mask, máy sẽ gửi một thông báo ICMP Address Mask Request đến 1 router và chờ nhận thông báo ICMP Address Mask Reply. Subnet Address Mask chứa địa chỉ của mặt nạ con của mạng.

Các bộ định tuyến phát bản tin ICMP để báo cho các trạm biết : gói tin không tới, hoặc tồn tại đường đi tốt hơn. Một số trường hợp có thể xảy ra là :

- *Destination unreachable* (không tới được đích): Bản tin không tới được đích do có lỗi hoặc không tìm được đường đi.
- *Routing redirect* (đổi đường đi): Thay đổi đường đi của bản tin do tồn tại đường đi tối ưu hơn (yêu cầu đổi đường đi).
- *Time expirect* (hết thời gian): Hết thời hạn khi TTL về 0 (timeout).
- *Echo request và cho echo reply* : Xuất hiện yêu cầu và trả lời.

  ICMP được dùng vào việc gỡ rối mạng cho biết tình trạng của mạng.

Lệnh Ping ***(Packet Internet Oroper)*** được dùng để hỏi (query) hệ thống (máy tính) khác để đảm bảo rằng một kết nối vẫn đang hoạt động (active). Lệnh Ping hoạt động bằng cách gửi ra một yêu cầu phản hồi (echo request) ICMP (Internet Control Message Protocol). Nếu như phần mềm IP của máy tính nhận được yêu cầu ICMP đó, nó đưa ra một trả lời phản hồi (echo reply) ngay lập tức. Máy gửi lại tiếp tục gửi một yêu cầu phản xạ cho đến khi lệnh ping được kết thúc bằng một tổ hợp phím thoát (Ctrl+C hoặc phím Delete trên UNIX).

5.7 Phiên bản IPv6

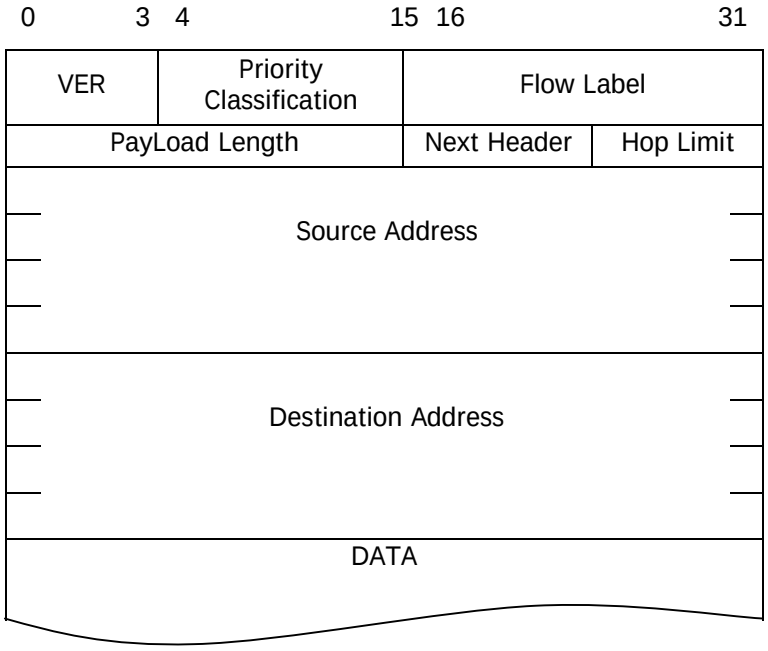
Với sự phát triển nhanh chóng của Internet thì địa chỉ IP 32 bit không thể đáp ứng được nhu cầu sử dụng Internet. Để khắc phục điều này phiên bản IP6 (IP Next Generation) đang được phát triển. Phiên bản IPv6 có các thay đổi như sau :

- Sử dụng 128 bit địa chỉ mạng thay cho 32 bit địa chỉ như phiên bản IPv4.
- Mở rộng phần Header cho ứng dụng và lựa chọn của khung tin.
- Hỗ trợ các loại dữ liệu audio và video.

- Có các giao thức mở rộng : cho phép bổ sung nhiều thông tin vào một datagram.

5.7.1 Khung tin IPng v6

Phần Header của các khung tin Ipng đã được thay đổi so với phiên bản 4. Phần lớn sự thay đổi của IPng là địa chỉ IP 128 bit và bỏ các trường không cần thiết. Cấu tạo của khung tin IPng như sau :



H×nh 5-25. CÊu t¹o cña gãi tin IPv6.

5.8 Định tuyến trên Internet

5.8.1 Bảng chọn đường

Một số phương thức thông thường xây dựng một bảng chọn đường (routing *table*) như sau :

- Bảng cố định được tạo ra dựa vào sơ đồ của mạng, bảng này liên tục được thay đổi và được cập nhật lại mỗi khi có sự thay đổi vật lý ở bất cứ nơi nào của mạng.
- Bảng động được dùng để ước lượng về đường truyền và các thông điệp từ các nút khác để điều chỉnh lại thông tin của bảng bên trong.
- Bảng dẫn đường cố định chính được tải về từ một trung tâm của các nút mạng trong một khoảng thời gian nhất định hoặc được tải về khi cần thiết.

Mỗi một phương thức đều có các ưu, nhược điểm của nó. Bảng động được đặt ở từng nút mạng hoặc được tải về trong những khoảng thời gian nhất định từ một

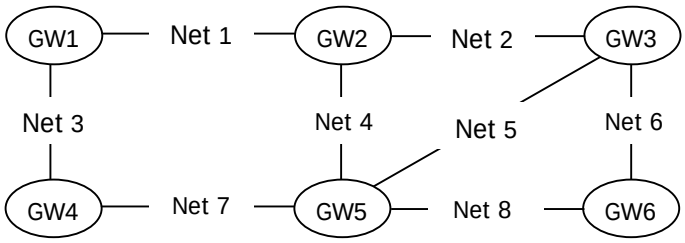
nơi chứa bảng cố định, nó không phức tạp và thích ứng với những thay đổi nhanh chóng trên mạng. Bảng chính thường là tốt hơn bảng cố định bởi vì quản lý một bảng ở trung tâm sẽ dễ dàng hơn quản lý từng bảng được đặt tại mỗi nút mạng.

5.8.2 Xây dựng bảng chọn đường cho các Router/Gateway

Trong liên mạng, tại mỗi cổng phải có một bảng chọn đường để chỉ ra muốn đến mạng đích nào thì phải đến cổng tiếp theo là cổng nào. Bảng chọn đường gồm hai phần : phần bên trái là mạng đích, nơi muốn đến, phần bên phải là khoảng cách tới đó và cổng tiếp theo.

Để xây dựng bảng chọn đường, từ cổng đang đứng ta xét các mạng cạnh đó, sau đó là các mạng ở cạnh các cổng tiếp theo và cứ thế cho đến hết các mạng trong liên mạng.

Ví dụ 1: Lập bảng chọn đường cho các router.gateway của liên mạng sau :



GW1	
Neti	D,G
1	0,1
2	1,2
3	0,1
4	1,2
5	2,2(4)
6	2,2
7	1,4
8	2,2(4)

GW2	
Neti	D,G
1	0,2
2	0,2
3	1,1
4	0,2
5	1,3(5)
6	1,3
7	1,5
8	1,5

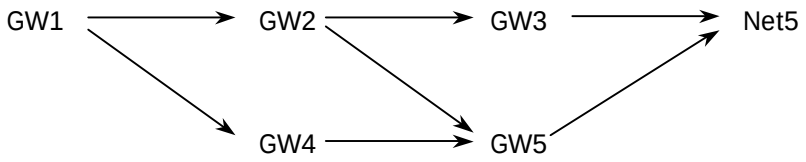
GW3	
Neti	D,G
1	1,2
2	0,3
3	2,2
4	1,2(5)
5	0,3
6	0,3
7	1,5
8	1,5(6)

GW4	
Neti	D,G
1	1,2
2	2,1(5)
3	0,4
4	1,5
5	1,5
6	2,5
7	0,4
8	1,5

GW5	
Neti	D,G
1	1,2
2	1,2
3	1,4
4	0,5
5	0,5
6	1,3(6)
7	0,5
8	0,5

GW6	
Neti	D,G
1	2,3(5)
2	1,3
3	2,5
4	1,5
5	1,3(5)
6	0,6
7	1,5
8	0,6

Dựa vào bảng chọn đường, tìm đường đi từ GW1 tới Net 5 như sau :



Đối với nhiều host, bảng dẫn đường tĩnh hoạt động như sau :

- Nếu đích nằm trong mạng cục bộ, dữ liệu được gởi đến máy đích
- Nếu đích nằm trên mạng ở xa, dữ liệu được chuyển tiếp đến gateway cục bộ.

Tuỳ thuộc vào kích cỡ của mạng mà các giao thức chọn đường khác nhau sẽ được sử dụng. Giao thức chọn đường trong một hệ thống nội bộ là RIP (Routing Information Protocol). Giao thức chọn đường giữa các hệ thống là EGP (External Gateway Protocol) và BGP (Border Gateway Protocol).

5.9 Mạng X.25

Vào những năm cuối thập niên 70, người ta phải cần đến một loạt các giao thức để cung cấp cho những người sử dụng mạng diện rộng WAN kết nối thông qua mạng dữ liệu công cộng (Public Data Networks - PDNs). Các loại hình PDNs như TELENET và TYMNET đã đạt được những thành công đáng ghi nhận, nhưng việc tiêu chuẩn hóa giao thức dường như còn ngoài tầm những người sử dụng mạng PDNs do việc đòi hỏi tính tương thích của thiết bị ngày một cao và đồng thời chi phí phải thấp. Kết quả của sự nỗ lực không ngừng này là sự ra đời của một loạt giao thức, trong đó X.25 được xem là giao thức phổ biến nhất.

Mạng X.25 và các giao thức liên quan do một tổ chức Quốc gia gọi là Hiệp hội Viễn thông Quốc tế (ITU) quản lý. Ban chịu trách nhiệm về các nghiệp vụ truyền tín hiệu âm thanh và dữ liệu của ITU gọi là ủy ban Tư vấn Quốc tế về Điện thoại và Điện báo (CCITT). Các thành viên của CCITT bao gồm FCC, PTTs Âu châu, các doanh nghiệp truyền thông và nhiều hãng máy tính, truyền dữ liệu khác. Do nhiều thành quả đóng góp trực tiếp có tính kế thừa, mạng X.25 thực sự được xem là mạng tiêu chuẩn có tính toàn cầu.

5.9.1 Cơ sở kỹ thuật

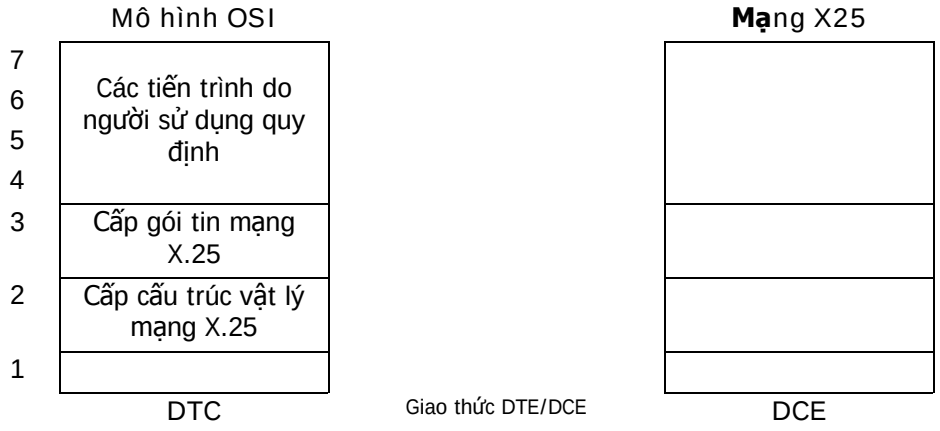
Mạng X.25 là một mạng điện thoại dùng để truyền dữ liệu. Để bắt đầu thực hiện quá trình giao tiếp, một máy tính cần phải liên kết với một máy khác để yêu cầu thực hiện giao tiếp. Máy được yêu cầu liên kết có thể chấp nhận hoặc từ chối việc giao tiếp. Nếu liên kết được chấp nhận, hai hệ thống có thể bắt đầu truyền tải thông tin qua lại hai chiều đồng thời với nhau. Cả hai bên đều có thể chấm dứt việc giao tiếp vào bất cứ thời điểm nào tùy ý.

Các đặc tính của mạng X.25 cho phép xác định quá trình tương tác từ nút-đến-nút (point-to-point) giữa các thiết bị truyền dữ liệu đầu cuối (Data Terminal Equipment - DTE) với các thiết bị kết cuối mạch truyền dữ liệu (Data Circuit-terminating Equipment - DCE). DTEs (bao gồm các trạm đầu cuối và máy chủ của người sử dụng mạng) kết nối với DCEs (bao gồm modem, các gói tin và các cổng truy cập PDN, thường đặt tại các trạm truyền thông), DCEs lại nối kết vào kênh chuyển mạch gói (Packet Switching Exchanges - PSEs) và các DCEs khác trong mạng PSNs và cuối cùng đến một DTE khác.

Một DTE có thể xem là một trạm đầu cuối nhưng không thực hiện đầy đủ các chức năng của mạng X.25. Các DTE được nối kết với DCE thông qua một thiết bị chuyển đổi gọi là thiết bị ghép/tách gói tin (Packet Assembler/Disassembler - PAD).

Quá trình hoạt động của mạch ghép nối từ trạm đầu cuối đến PAD, các dịch vụ do PAD cung cấp và các tương tác giữa PAD và các máy chủ do CCITT quy định.

Sơ đồ đặc tính của mạng X.25 kiểu phân tầng từ 1 tới 3 theo mô hình tham chiếu cho việc nối kết các hệ thống mở OSI. Tầng 3 của mạng X.25 mô tả các quy trình định dạng và chuyển mạch gói giữa các thành tố tầng 3 ngang cấp. Tầng 2 của mạng X.25 do các thủ tục truy cập liên kết cân bằng (Link Access Procedure Balance - LAPB) kiểm soát. LAPB xác lập các đơn vị gói tin (packet framing) cho các liên kết DTE/DCE. Tầng 1 của mạng X.25 xác lập các thủ tục về điện và cơ để kích hoạt và chấm dứt quá trình kết nối vật lý của DTE và DCE. Mối quan hệ này được minh họa theo hình vẽ dưới đây. Chú ý rằng tầng 2 và 3 cũng tham chiếu theo tiêu chuẩn ISO 7776 (LAPB) và ISO 8208 (các tầng gói tin mạng X.25).



Hình 5-26. Mối quan hệ giữa các tầng trong mạng X.25.

Quá trình giao tiếp từ nút-tới-nút (end-to-end) giữa các DTEs được thực hiện hoàn thiện thông qua một sự kết nối song phương gọi là liên kết truyền ảo (virtual circuit). Các liên kết ảo cho phép các hệ mạng khác nhau có thể giao tiếp được với nhau thông qua mọi nút liên kết trung gian mà không cần đến các bộ phận chuyên dụng để định rõ các liên kết vật lý. Các liên kết ảo hoặc có thể duy trì vĩnh viễn hoặc có thể tạm thời. Liên kết ảo vĩnh viễn được gọi là PVCs (Permanent Virtual Circuits), liên kết ảo tạm thời được gọi là SVCs (Switched Virtual Circuits). PVCs chủ yếu áp dụng cho phương thức truyền dữ liệu thường xuyên còn SVCs được áp dụng cho phương thức truyền dữ liệu không thường xuyên. Tầng 3 của mạng X.25 liên quan tới phương thức giao tiếp từ nút tới nút bao gồm cả hai liên kết ảo PVCs và SVCs.

Một khi đã thiết lập liên kết ảo, PTE có thể thực hiện truyền một gói tin đến một PTE khác bằng cách chuyển gói tin đến DCE thông qua một liên kết ảo thích hợp. Sau đó DCE sẽ tiến hành ền ưu tiên của liên kết ảo để định ra thức truyền gói tin lên mạng X.25. Các giao thức của tầng 3 mạng X.25 sẽ tiến hành chèn thông tin

vào giữa các DTE được kiểm soát bởi DCE của mạng phía nhận gói tin rồi sau đó được chuyển đến DTE đích.

5.10 Kỹ thuật FRAME RELAY

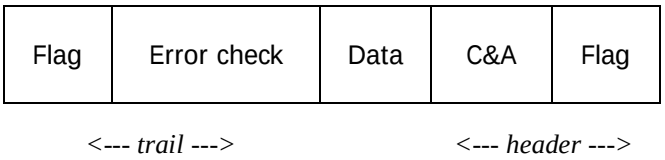
Bước sang thập kỷ 80 và đầu thập kỷ 90, công nghệ thông tin có những bước tiến đặc biệt là chế tạo và sử dụng cáp quang vào mạng truyền dẫn tạo nên chất lượng thông tin rất cao. Sử dụng giao thức X25 để truyền đa số liệu trên mạng cáp quang, dữ liệu nhận được có thể đánh giá là đạt yêu cầu. Tuy nhiên người ta nhận thấy rằng sử dụng giao thức này làm mất rất nhiều thời gian để truyền số liệu trên mạng cáp quang. Do đó công nghệ Frame Relay ra đời có thể chuyển nhận các khung lớn tới 4096 byte trong khi đó gói tiêu chuẩn của X25 khuyến cáo dùng là 128 byte, không cần thời gian cho việc hỏi đáp, phát hiện lỗi và sửa lỗi ở lớp 3 (*No protocol at Network Layer*) nên Frame Relay có khả năng chuyển tải nhanh hơn hàng chục lần so với X25 ở cùng tốc độ. Frame Relay rất thích hợp cho truyền số liệu tốc độ cao và cho kết nối LAN to LAN và cả cho âm thanh, nhưng điều kiện tiên quyết để sử dụng công nghệ Frame relay là chất lượng mạng truyền dẫn phải cao.

Frame-Relay bắt đầu được đưa ra như tiêu chuẩn của một trong những giao thức truyền số liệu từ năm 1984 trong hội nghị của ủy ban Tư vấn Quốc tế về Điệnthoại và Điện báo CCITT và cũng được Viện tiêu chuẩn quốc gia Mỹ ANSI đưa thành tiêu chuẩn của ANSI vào năm đó.

Mục tiêu chính của Frame-Relay cũng giống như của nhiều tiêu chuẩn khác, đó là tạo ra một giao diện chuẩn để kết nối thiết bị - của các nhà sản xuất thiết bị khác nhau - giữa người dùng và mạng UNI (*User to Network Interface*). Frame-Relay được thiết kế nhằm cung cấp dịch vụ chuyển khung nhanh cho các ứng dụng số liệu tương tự như X.25 hay ATM.

Mạng truyền số liệu theo công nghệ chuyển mạch gói X.25 chỉ có thể phục vụ cho các nhu cầu truyền số liệu tốc độ thấp (tối đa tới 128 Kbps) nhưng nó có tính an toàn cao, khắc phục được các yếu điểm của một mạng truyền dẫn chất lượng kém. Với các công nghệ truyền dẫn hiện nay, vấn đề nâng cấp chất lượng các đường truyền dẫn không còn quá phức tạp như trước kia. Vì vậy, chúng ta còn có thể chọn hướng phát triển là xây dựng mạng truyền số liệu theo công nghệ Frame-relay và tiến tới công nghệ ATM.

5.10.1 Khuôn dạng gói dữ liệu Frame-Relay



Hình 5-27. Khuôn dạng gói dữ liệu Frame-Relay.

- Flag: Cờ
- Error check: Trường kiểm tra lỗi
- Data: Trường dữ liệu
- C&A: Trường địa chỉ và điều khiển

Để thực hiện nhiệm vụ truyền số liệu, mạng Frame-Relay sẽ phải giải quyết vấn đề tắc nghẽn thông tin trên mạng, thực chất đây là vấn đề của tầng Mạng trong mô hình 7 tầng. Frame-Relay làm việc ở tầng Liên kết nhưng cũng phải giải quyết vấn đề này để đảm bảo khả năng lưu chuyển thông tin. Hầu hết các mạng truyền số liệu đều sử dụng kỹ thuật điều khiển luồng để giải quyết vấn đề tắc nghẽn. Có hai phương pháp được sử dụng khi xảy ra tắc nghẽn trong mạng: thông báo cho người dùng, router, chuyển mạch về sự cố tắc nghẽn xảy ra và thực hiện các công việc nhằm hiệu chỉnh luồng thông tin. Cả hai phương pháp này mạng Frame-Relay đều dùng đến các bit BECN (Backward Explicit Congestion Notification) và bit FECN (Forward Explicit Congestion Notification) trong trường điều khiển.

Bit FECN được thiết lập khi có tắc nghẽn để thông báo rằng thủ tục xử lý tắt nghẽn đã được khởi tạo, và tương ứng với lưu lượng bị nghẽn từ hướng của Frame có bit FECN tới. Ngược lại, bit BECN cũng được thiết lập khi có tắc nghẽn để thông báo rằng thủ tục xử lý nghẽn đã được khởi tạo, nhưng tương ứng với lưu l-ượng bị nghẽn từ hướng ngược với Frame có bit BECN tới. Khi các bit này được thiết lập thì mạng phải dùng đến một liên kết logic dự phòng để chuyển các thông tin để xử lý nghẽn, đó là liên kết với mã nhận dạng DLCI (Data Link Connection Identifier) số 1023. Các liên kết với mã nhận dạng nhỏ hơn được dùng để truyền số liệu của người dùng.

BÀI TẬP

1. Viết sơ đồ mô tả thuật giải hoạt động chọn đường trên mạng.
2. Khảo sát cấu trúc và hoạt động của giao thức điều khiển ICMP
3. Tìm hiểu các lệnh của hệ điều hành Windows và Linux để xem và thay đổi các thông số bảng chọn đường.

Chương 6

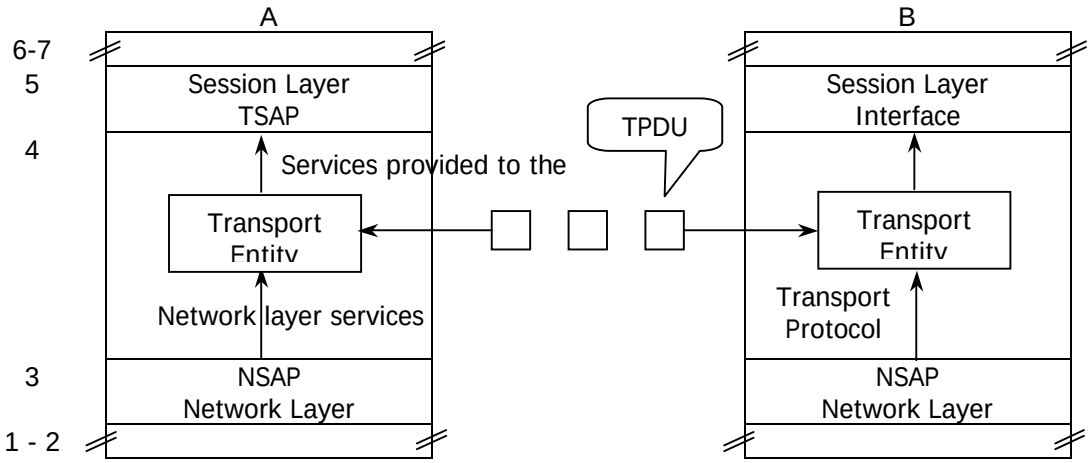
TẦNG GIAO VẬN

Tầng giao vận làm nhiệm vụ thiết lập, duy trì và huỷ bỏ các cuộc giao tiếp giữa hai máy, đảm bảo việc dữ liệu truyền giống hoàn toàn dữ liệu nhận. Dữ liệu qua các mạng con có thể bị lỗi, tập tin tầng giao vận 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. Tầng giao vận *quản lý dữ liệu gửi*, *xác định trật tự* của dữ liệu và *độ ưu tiên* của dữ liệu đó.

6.1 Các vấn đề của tầng giao vận

6.1.1 Cung cấp dịch vụ cho tầng phiên

Để thực hiện mục tiêu chuyển giao dữ liệu tin cậy, an toàn cho tầng 5, tầng 4 phải dùng các dịch vụ được cung cấp từ tầng 3 (network layer). Phần cứng và phần mền trong phần 4 để thực hiện công việc coi là thực thể giao vận (*transport entity*). Mối quan hệ giữa các lớp 3, 4, 5, được mô tả bởi hình sau:



Hình 6-1. Mối quan hệ giữa các thực thể trong tầng Phiên.

Có hai dịch vụ mạng nên cũng có hai dịch vụ giao vận: *dịch vụ có kết nối* và *không kết nối*.

Do dữ liệu qua các subnet có thể sai sót, người sử dụng không có được điều khiển trên subnet hoặc tăng cường quản lý lỗi ở tầng hai. Chỉ có khả năng đặt thêm một tầng trên lớp 3 để cải thiện chất lượng dịch vụ (QoS). Nếu giữa chúng một tầng giao vận được kết nối mạng được kết thúc đột ngột và không biết được sự cố gì đã xảy ra, nó có thể thiết lập một kết nối mới ở lớp mạng tới tầng giao vận ở xa và gởi yêu cầu hỏi số liệu nào đến, số liệu nào không tự nó biết được sai sót xảy ra ở đâu. Tầng 4 có thể phát hiện mất gói tin, số liệu bị biến đổi, N-RESET ở lớp mạng. Tầng 1 -> 4 cung cấp dịch vụ giao vận. Tầng 5 ->7 sử dụng dịch vụ giao vận

- Các hàm dịch vụ của tầng giao vận có kết nối

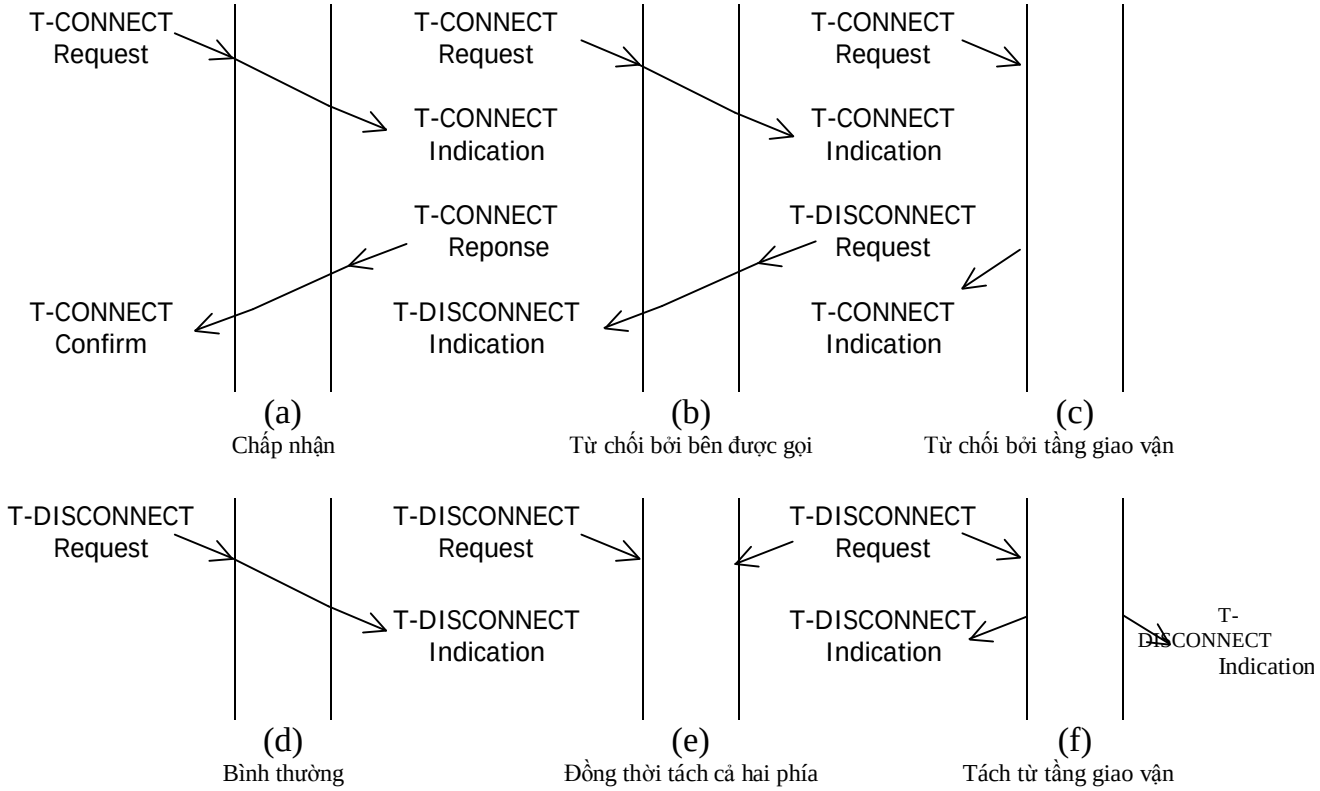
Ngoài phần giao thức chuẩn, ISO còn định nghĩa các dịch vụ mà tầng Giao vận cung cấp cho các thực thể ở tầng Phiên trong trường hợp có liên kết, dưới dạng một tập hợp các hàm dịch vụ nguyên thủy(services primitives) như sau :

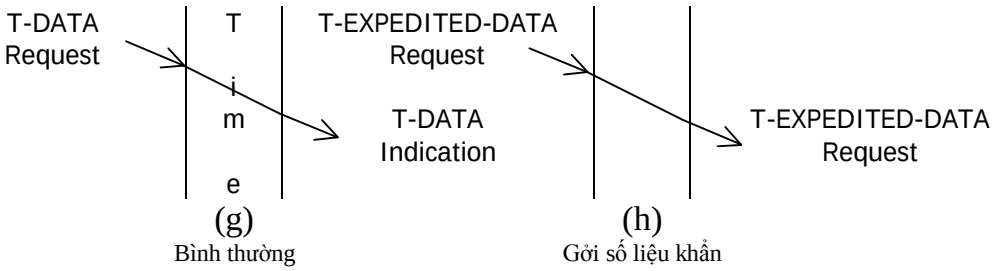
T-CONNECT request (callce, caller, exp wanted, qos, user data)
T-CONNECT indication (callce, caller, exp wanted, qos, user data)
T-CONNECT response (qos, responder, exp wanted, user data)
T-CONNECT confirm (qos, responder, exp wanted, user data)
T-DISCONNECT request (user data)
T-DISCONNECT indication (reason, user data)
T-DATA request (user data)
T-DATA indication (reason, user data)
T-EXPEDITED-DATA request (user data)
T-EXPEDITED-DATA indication (reason, user data)

- Các hàm dịch vụ của tầng giao vận không có kết nối : Chỉ có hai hàm dịch vụ được định nghĩa :

T-UNITDATA request (callce, caller, QoS, user data)
T-UNITDATA indication (callce, caller, QoS, user data)

- Quan hệ giữa các hàm OSI nguyên thủy : Quá trình nối, tách và trao đổi dữ liệu diễn ra như sau :





Hình 6-2. Quan hệ giữa các hàm OSI nguyên thủy.

Giải thích

(a) Quá trình nối được chấp nhận
(b) Quá trình nối bị từ chối bởi bên được gọi
(c) Quá trình nối bị từ chối bởi tầng Giao vận do lỗi của người sử dụng hoặc người chung cấp dịch vụ giao vận gây nên.
(d) Quá trình tách bình thường
(e) Quá trình tách đồng thời cả hai phía
(f) Quá trình tách từ tầng Giao vận
(g) Quá trình trao đổi dữ liệu bình thường
(h) Quá trình trao đổi dữ liệu khẩn

Trong hình (c) trên, việc từ chối có thể do lỗi của người sử dụng hoặc người cung cấp dịch vụ giao vận gây nên. Khi đó, không có gì được phát qua mạng vì vậy đầu kia không nghe được gì cả. Có những qui tắc cho người sử dụng các hàm dịch vụ giao vận. Ví dụ, không được dùng T-DISCONNECT.request khi tiếp nối chưa được thiết lập.

6.1.2 Chất lượng dịch vụ QoS

Chức năng cơ bản của tầng 4 là tăng cường chất lượng dịch vụ được cung cấp bởi tầng 3. Nếu lớp chất lượng chưa tốt, tầng Giao vận sẽ khắc phục khoảng ngăn cách giữa những gì mà người sử dụng tầng Giao vận muốn và những gì mà lớp mạng cung cấp. Các tham số của chất lượng dịch vụ QoS (Quality of Service) bao gồm :

- *Thời gian thiết lập liên kết* là thời gian từ khi gởi yêu cầu tới thời điểm nhận được xác nhận liên kết.
- *Xác nhận không thành công của thiết lập liên kết* - là tỷ lệ yêu cầu liên kết không được chấp nhận trong một thời hạn tối đa.
- *Lưu lượng của liên kết* do số byte hữu ích có thể truyền trong một giây, lưu lượng được tính *trong một cuộc trao đổi* hoặc dựa vào *khả năng của mạng theo 2 chiều.*

- *Thời gian trễ (*Độ trễ truyền dẫn - transmit delay) là khoảng thời gian giữa thời điểm mà người sử dụng dịch vụ của tầng Giao vận bên phát gởi thông báo tời thời điểm thực thể của tầng Giao vận bên thu nhận được. Đánh giá theo 2 chiều.
- *Tỷ lệ lỗi* là tỷ số giữa tin báo bị lỗi (hoặc mất) trên tổng số tin báo được truyền trong một chu kỳ định trước.
- *Xác nhận sự cố truyền:* tỷ số giữa thời gian có sự cố với thời gian cả chu kỳ quan sát.
- *Thời gian hủy liên kết* là thời gian từ khi một người sử dụng phát huy cầu huỷ liên kết đến khi liên kết được huỷ thật sự tại thiết bị đầu cuối từ xa.
- *Xác suất lỗi khi hủy liên kết* là tỷ lệ số yêu cầu hủy liên kết không được thực hiện trong thời gian lớn nhất.
- *Khả năng bảo vệ* là khả năng của người sử dụng cấm thiết bị đầu cuối bên ngoài truy nhập bất hợp pháp hay thay đổi dữ liệu truyền.
- *Thông số ưu tiên:* cho phép người sử dụng có quyền ưu tiên được phục vụ cao hơn đối với một liên kết.
- *Thông số hủy bỏ* cho phép tầng giao vận tự quyết định huỷ liên kết khi có tắc nghẽn hay các vấn đề bên trong mạng.

Người sử dụng khi yêu cầu liên kết sẽ gởi tất cả các thông số với các giá trị yêu cầu tới tầng giao vận và bắt đầu quá trình đàm thoại với các thông số đó.

So sánh các hàm cơ bản của dịch vụ giao vận và dịch vụ mạng, ta thấy các dịch vụ mạng và giao vận gần giống nhau. Sự khác nhau là dịch vụ mạng cho phép người sử dụng xử lý Acknowledgements và N-RQSQTS. Ngược lại, dịch vụ giao vận không quan tâm đến vì dịch vụ lớp giao vận là tin cậy, không có lỗi. Dịch vụ mạng được dùng bởi tầng giao vận.

6.1.3 Các lớp giao thức của tầng giao vận

Các dịch vụ tầng giao vận bảo đảm bằng các giao thức giữa 2 thực thể của tầng cũng tương tự như giao thức của tầng liên kết dữ liệu nó giải quyết vấn đề lỗi, điều khiển lưu lượng và bảo đảm trình tự mảng tin.

tầng liên kết dữ liệu, hai IMP truyền tin trực tiếp qua đường kênh vật lý. ở tầng giao vận, đường kênh vật lý này được thay bằng subnet. Sự khác nhau này kéo theo sự khác nhau về xây dựng các thủ tục. ở tầng giao vận phải xác định địa chỉ nơi nhận, ở tầng liên kết dữ liệu thì không cần vì chỉ có một đường truyền tin giữa hai điểm. Quá trình kết nối ở tầng giao vận cũng phức tạp hơn ở tầng liên kết giữ liệu.

Tầng giao vận đòi hỏi khả năng lưu trữ trong mạng (subnet) để giữ những gói tin bị sự cố và đòi hỏi thủ tục đặc biệt. tầng giao vận số các kết nối lớn hơn nên các vấn đề bộ đệm và điều khiển dòng phức tạp hơn.

Từ quan điểm thiết kế thủ tục giao vận, các dịch vụ được cho bởi mạng quan trọng hơn các tính chất thực tế của mạng, mặc dù cái sau bị ảnh hưởng mạnh bởi cái trước. Tuy vậy, trong một phạm vi nào đó, dịch vụ mức mạng có thể che những mặt ít được chú ý của mạng và cung cấp ghép nối tốt hơn. Để tiện lợi xem xét các thủ tục giao vận, ta chia các dịch vụ trên mạng thành 3 nhóm :

Nhóm	Ý nghĩa
Nhóm A	- Hoàn thiện, tỷ lệ các gói tin bị mất, trùng lặp hoặc bị hỏng không đáng kể. - Lệnh N-RESET có thể bỏ qua. - Tầng giao vận đơn giản, không cần các dịch vụ phục hồi và sắp xếp lại thứ tự gói tin. - Thường là mạng cục bộ.
Nhóm B	- Gói tin bị mất, nhưng kiểm soát được. - Thỉnh thoảng tầng mạng gởi lệnh N-RESET do tắc nghẽn, hỏng phần cứng, vấn đề phần mềm. - Thông thường là mạng đường dài • Giao thức tầng Giao vận có nhiệm vụ: - Thiết lập tại liên kết. Đồng bộ lại - Theo dõi toàn bộ yêu cầu khởi động lại cho NSD.
Nhóm C	- Truyền tin không tin cậy, không liên kết - Mạng đường dài, kết nối nhiều mạng con - Giao thức của tầng giao vận phức tạp, phải có khả năng phục hồi lỗi khi xảy ra sự cố và sắp xếp lại thứ tự các gói tin.

Bảng 6-1. Các nhóm dịch vụ của tầng Giao vận.

Dịch vụ mạng xấu thì giao thức của tầng giao vận sẽ phức tạp hơn. OSI đã nhận thức vấn đề này và chia giao thức của tầng giao vận thành 5 lớp ứng với các loại mạng như sau :

Lớp	Ý nghĩa
Lớp O *Mạng loại A*	- Lớp thủ tục đơn giản - Kết nối mạng khi có yêu cầu giao vận không phải giải quyết lỗi - Chủ yếu tạo ra trình tự, điều khiển dòng dữ liệu để tầng mạng hoạt động tốt. - Bao gồm cơ cấu thiết lập và huỷ liên kết ở tầng giao diện.
Lớp 1 *Mạng loại B*	Có tính chất tương tự lớp 0, ngoài ra còn thêm: - Khởi động lại mạng sau khi N-RESET. Giao thức có khả năng báo nhận (ACK) và truyền dữ liệu khẩn. - Đồng bộ lại và sau đó nối lại liên lạc giữa các thực thể giao vận đã bị gián đoạn - Lớp 1 không kiểm tra lỗi và kiểm soát dòng dữ liệu.
Lớp 2 *Mạng loại A*	Lớp 2 là phiên bản của lớp 0 và được xây dựng cho mạng tin cậy và có thêm một số chức năng như sau : - Sự ghép kênh : Hai hay nhiều liên kết của tầng giao vận có thể dùng chung một kết nối ở tầng mạng. - Sử dụng khi nhiều liên kết ở tầng giao vận được mở đồng thời, nối liên kết có lưu lượng nhỏ. Ví dụ như hệ thống đặt vé máy bay cho phép tiết kiệm đường truyền.
Lớp 3 *Mạng loại B*	Là tổ hợp lớp 1 và lớp 2 - Cho phép dồn kênh - Khởi động lại - Điều khiển dòng dữ liệu.
Lớp 4 *Mạng loại C*	Lớp 4 có hầu hết các chức năng của lớp trước và bổ sung thêm một số khả năng kiểm soát luồng dữ liệu. - Phải có biện pháp giải quyết vấn đề mất gói tin, gói tin bị hỏng - Phải giải quyết yêu cầu khởi động lại - Thủ tục Giao vận phức tạp nhất.

Bảng 6-2. Các lớp dịch vụ của tầng Giao vận.

Dịch vụ không có kết nối đặt tất cả sự phức tạp và thủ tục Giao vận.

6.1.4 Thủ tục giao vận trên X. 25

Thủ tục X. 25 là thủ tục có nối và tin cậy, coi như lớp mạng loại A. Do đó thủ tục giao vận trên X.25 là thủ tục giao vận lớp 0 mô hình OSI. Thủ tục này được thể hiện qua các hàm dịch vụ cơ bản và quá trình nối, tách, trao đổi số liệu của thủ tục.

6.1.4.1 Các hàm dịch vụ cơ bản

Các hàm dịch vụ cơ bản được thực hiện bằng các chương trình con minh họa bằng ngôn ngữ Pascal

1. Hàm Connect thực hiện T-CONNECT .request

   connum = CONNECT(local, remote)

Hàm dịch vụ này để thiết lập kết nối tầng giao vận giữa 2 máy. Nếu kết nối thành công, hàm trả về một số dương, ngược lại hàm trả về số âm.

2. Hàm Listen thực hiện T-CONNECT.indication

   connum = LISTEN (local)

Hàm này dùng để thông báo tiếp nhận yêu cầu kết nối

3. Hàm Disconnect thực hiiện T-DISCONNECT.request

   status = DISCONNECT (commun)

Hàm này dùng để kết thúc kết nối, tham số commun cho biết kết nối nào sẽ bị ngắt, kết quả thực hiện sẽ được gán cho biến status với giá trị OK hoặc error

4. Hàm Send thực hiện T-DATA.request

   status = SEND (commun, buffer, bytes)

Hàm này để phát nội dung ở buffer với kích thước là bytes cho số kết nối đặt ở commun. Kết quả đặt ở status.

5. Hàm Receive thực hiện T-DATA.indication

   status = RECEIVE (commun buffer, bytes)

Hàm này để nhận tin vào buffer với kích thước là giá trị ở biến bytes. Kết quả thực hiện đặt vào status giá trị OK hoặc error.

Chương 7

HỌ GIAO THỨC TCP/IP

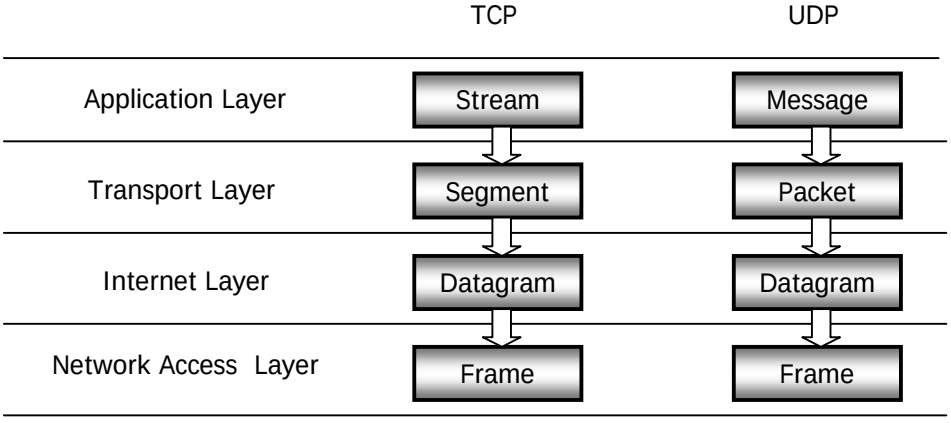
Do đặc tính của mô hình OSI là một mô hình tham chiếu, việc áp dụng mô hình OSI vào thực tế thường có hiệu suất kém do dữ liệu phải truyền qua tất cả các lớp của mô hình OSI ở cả hai máy, mô hình OSI là tiêu chuẩn để các nhà phát triển dựa vào mà phát triển các mô hình khác tối ưu hơn. Có rất nhiều mô hình khác nhau như NetBIOS, IPX/SPX, TCP/IP, tuy nhiên mô hình TCP/IP hiện nay đang được sử dụng phổ biến nhất.

TCP/IP thực chất là một họ giao thức cùng làm việc với nhau để cung cấp ptn truyền thông liên mạng. Mô hình TCP/IP có những tính chất chung như sau :

- TCP /IP độc lập với phần cứng mạng vật lý, điều này cho phép TCP/IP hoạt động trên nhiều mạng khác nhau như Ethernet, Token Ring, X25, dial up,...
- TCP/IP sử dụng sơ đồ đánh địa chỉ toàn cục duy nhất : mỗi máy tính trên mạng TCP/IP có một địa chỉ xác định duy nhất. Mỗi gói tin gởi trên mạng có một tiêu đề chứa địa chỉ nguồn và đích.
- Chuẩn giao thức mở : TCP/IP có thể thực hiện trên bất kỳ phần chứng hay hệ điều hành nào.
- Hoạt động theo mô hình Client/Server.
- Cung cấp các giao thức ứng dụng : cung cấp cho người lập trình phương thức truyền dữ liệu trên mạng giữa các ứng dụng mà còn cung cấp nhiều giao thức ở mức ứng dụng như giao thức truyền nhận mail, truyền file, . . .
- TCP/IP hỗ trợ cho liên mạng (internetworking) và định tuyến, các giao thức mức cao được chuẩn hoá thích hợp và cung cấp sẵn các dịch vụ người dùng.

7.1 Mô hình TCP/IP

Cấu trúc của bộ giao thức TCP/IP có bốn tầng, được mô tả như hình vẽ sau



Hình 7-1. Kiến trúc TCP/IP và các đơn vị dữ liệu.

Chức năng của các tầng như sau :

1. Tầng truy cập mạng NAL (Network Access Layer)

- Cung cấp cho hệ thống phương thức để truyền dữ liệu trên các thiết bị phần cứng vật lý khác nhau của mạng.
- Đóng gói các lược đồ dữ liệu IP (IP datagram) vào các *frame* truyền trên mạng và việc ánh xạ các địa chỉ IP thành các địa chỉ vật lý tương ứng dùng cho mạng trước khi truyền xuống kênh vật lý.
- Định nghĩa cách thức truyền các khối dữ liệu IP : Các giao thức ở lớp này phải biết chi tiết các phần cấu trúc vật lý mạng ở dưới nó để định dạng chính xác các dữ liệu sẽ được truyền phụ thuộc vào từng loại mạng vật lý cụ thể.

Lớp truy cập mạng NAL của mô hình kiến trúc TCP/IP tương đương với ba lớp thấp nhất của mô hình OSI là Network layer, Datalink layer, và Physical layer.

2. Tầng mạng

Tầng mạng chịu trách nhiệm định tuyến các thông báo (message) qua các mạng vật lý khác nhau, liên mạng, giao thức ở lớp này là IP là giao thức quan trọng nhất vì IP cung cấp dịch vụ giao nhận gói tin cơ bản trên các mạng TCP/IP, mọi giao thức ở các lớp trên và bên dưới tầng mạng đều sử dụng giao thức IP để thực hiện việc giao nhận dữ liệu. Hơn nữa IP bổ sung một hệ thống địa chỉ logic được gọi là địa chỉ IP, được sử dụng bởi lớp Internet và các lớp cao hơn để nhận diện các thiết bị và thực hiện định tuyến liên mạng.

3. Tầng Giao vận (Host to Host Transport Layer)

- Cung cấp phương tiện liên lạc từ một chương trình ứng dụng này đến chương trình ứng dụng khác, chịu trách nhiệm đảm bảo toàn vẹn dữ liệu đầu cuối.
- Trong lớp này có 2 giao thức quan trọng nhất:

- Transmission Control Protocol (TCP) : Về chức năng TCP tương đương với lớp giao thức đầy đủ nhất của giao thức chuẩn Transport của OSI. Tuy nhiên, khác với mô hình ISO, TCP sử dụng phương thức trao đổi các dòng dữ liệu (data stream) giữa người sử dụng.
- User Datagram Protocol (UDP) : cung cấp dịch vụ giao nhận dữ liệu theo kiểu “không liên kết” (connectionless), không cần phải thực hiện thiết lập liên kết logic giữa một cặp thực thể UDP trước khi chúng trao đổi dữ liệu với nhau.

4. Tầng ứng dụng (Application Layer)

Bao gồm tất cả các tiến trình sử dụng các giao thức của lớp Transport để truyền dữ liệu. Có nhiều giao thức ứng dụng ở lớp này, phần lớn là nhằm cung cấp cho người dùng các dịch vụ ứng dụng, sử dụng 2 giao thức chính TCP và UDP.

Tầng ứng dụng cung cấp các dịch vụ trên Internet như thư điện tử (SMTP), truyền file (FTP), v.v.. Tầng dưới là phần mạng để định tuyến địa chỉ đến.

<table>
<tr><td rowspan="2">Application</td><td>Ping</td><td>Telnet & Rlogin</td><td>FTP</td><td>SMTP</td><td>SNMP</td><td>Trace - Route</td></tr>
<tr><td>DNS</td><td>TFTP</td><td>BOOTP</td><td>RIP</td><td>OSPF</td><td>etc.</td></tr>
<tr><td>Transport</td><td colspan="2">TCP</td><td colspan="2">UDP</td><td colspan="2">ICMP</td></tr>
<tr><td>Network</td><td colspan="6">IP</td></tr>
<tr><td rowspan="2">DataLink</td><td colspan="2">LLC</td><td colspan="2">HDLC</td><td colspan="2">PPP</td></tr>
<tr><td>Ethernet</td><td>802.3</td><td>X.25</td><td>Token Ring</td><td>Frame Relay</td><td>ATM</td><td>SMDS</td><td>etc.</td></tr>
<tr><td>Physical</td><td>Fiber Optics</td><td>UTP</td><td>Coax</td><td>Microwave</td><td>Satelite</td><td>STP</td></tr>
</table>

Hình 7-2. Họ giao thức TCP/IP.

Telnet	Tele Comunication	Dịch vụ truy cập từ xa.
FTP	File Transfer Protocol	Dịch vụ truyền File.
SMTP	Simple Mail Transfer Protocol	Dịch vụ truyền thư đơn giản.
DNS	Domain Name System	Hệ thống tên miền
SNMP	Simple Network Management Protocol	Giao thức quản lý mạng đơn giản
RPC	Remote Procedure Call	Thủ tục gọi từ xa
RIP	Routing Information Protocol	Giao thức định tuyến thông tin
TCP	Transmission Control Protocol	Giao thức TCP
UDP	User Datagram Protocol	Giao thức dữ liệu của người dùng.
IP	Internet Protocol	Giao thức IP
ICMP	Internet Control Message Protocol	G.thức kiểm soát message giữa các mạng.
FDDI	Fiber Distributed Data Multiplexing	

7.2 Giao thức TCP

Tầng Giao vận sử dụng hai giao thức chính là TCP và UDP. Giao thức TCP (Transmission Control Protocol) đảm bảo độ tin cậy giữa nơi gởi và nơi nhận (end-to-end) trong điều kiện lớp mạng loại C không tin cậy. Dòng số liệu có chiều dài tuỳ ý được phân thành những đoạn không vượt quá 64KB, gửi đi đến đầu bên kia lại được gộp lại thành bản tin ban đầu.

- Chức năng của giao thức TCP :

Chức năng	Giải thích
Phát hiện lỗi	Bằng cách sử dụng một trường checksum để kiểm tra lỗi bất cứ khi nào datagram được cắt ra trong quá trình truyền.
Truyền lại	TCP sẽ truyền lại các gói tin bị mất hoặc bị sai hỏng trong quá trình truyền.
Đánh số thứ tự	Cho phép bên gởi đã phát đi các gói tin theo một trật tự, bên nhận đã nhận và kết hợp các gói tin theo một trật tự đã định
Báo nhận và kiểm soát luồng	Bên TCP nhận sẽ gửi một đoạn báo nhận xác định một số chức năng trong quá trình truyền tin.
Phát gói tin đến đúng ứng dụng yêu cầu	Mỗi đoạn gói tin TCP có một số hiệu cổng nguồn và đích, là giá trị duy nhất để xác định một phiên làm việc.

- Tính chất của giao thức TCP :

Tính chất	Giải thích
Tin cậy	TCP cung cấp khả năng tin cậy bằng cách gởi lại dữ liệu đến khi bên nhận có một báo nhận hỏng. Đơn vị dữ liệu mà TCP truyền đi là segment và được giao thức IP phân ra thành các datagram.
Hướng kết nối	TCP thiết lập kết nối logic giữa các máy khi truyền dữ liệu, hoạt động theo cơ chế "bắt tay" (handshake), và có nhiệm vụ đồng bộ việc kết nối giữa hai máy.
Dòng dữ liệu	TCP xử lý dữ liệu dưới dạng một dòng nối tiếp các byte, theo cơ chế đánh số thứ tự gói tin.

7.2.1 Khuôn dạng gói tin TCP

TCP là một giao thức có liên kết (*connection - oriented*) nghĩa là cần phải thiết lập liên kết logic giữa một cặp thực thể TCP trước khi chúng trao đổi dữ liệu với nhau, có 3 giai đoạn : ***thiết lập liên kết, truyền tải dữ liệu*** và ***hủy liên kết***. Đơn vị dữ liệu của TCP được gọi là **segment** (đoạn dữ liệu). Cấu trúc đơn vị dữ liệu của TCP được mô tả như hình sau :

Source Port - Số hiệu cổng nguồn (16 bits)

Xác định số hiệu cổng của trạm nguồn - User TCP cục bộ (thường là một chương trình ứng dụng trên lớp cao hơn).

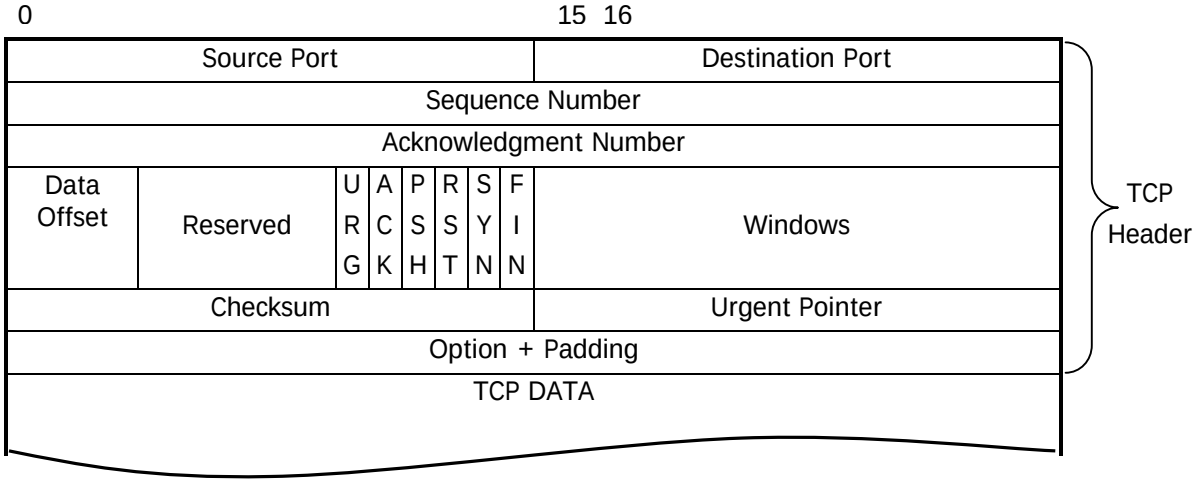
Destination Port - Số hiệu cổng đích (16 bits)

Xác định số hiệu cổng của trạm đích của máy ở xa. Dùng để nhận diện các tiến trình điểm đầu mút ở kênh ảo TCP.

Sequence Number - Số thứ thự (32 bits)

Trường này chứa một số chỉ vị trí hiện tại của khối tin trong Message. Số này cũng được các phiên bản khác nhau của TCP để cung cấp số thứ tự của khối tin ban đầu (ISN).

Đây là số hiệu byte đầu tiên của segment trừ khi bit SYN được thiết lập. Nếu bit SYN được thiết lập thì Sequence Number là số hiệu tuần tự khởi đầu (ISN) và byte dữ liệu đầu tiên là ISN+1.



Hình 7-3. Cấu trúc của gói tin TCP.

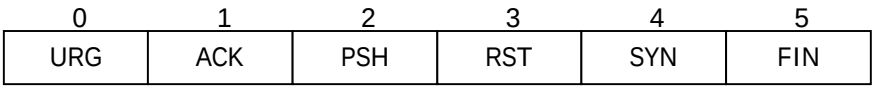
Acknowledgment Number - Số phúc đáp (32 bits)

Dùng để chỉ ra số hiệu của segment (khối tin) sắp được truyền tiếp theo mà trạm đích đang chờ để nhận. Dùng báo nhận tốt các Segment mà trạm nguồn đã gửi cho trạm đích. Ngoài ra nó cũng chỉ ra số thứ tự của khối tin nhận được sau cùng; nó chỉ ra số thứ tự của khối tin nhận được cộng thêm 1.

Data offset (32 bits) : Trường này dùng để chỉ ra vị trí bắt đầu của trường dữ liệu.

Reserved (6 bits) : Chưa dùng đến, dành sử dụng về sau. Các bit được đặt bằng 0.

Control Bits - Các bit điều khiển



Cờ URG : Nếu có giá trị là 1 thì thì trường urgent pointer rất quan trọng.
Cờ ACK : Nếu có giá trị là 1 thì trường Acknowledgment rất quan trọng.
Cờ PSH : Nếu thiết lập thì tức là chức năng PUSH sắp được thực hiện.
Cờ RST : Nếu được thiết lập thì kết nối hiện tại sắp được khởi tạo lại.
Cờ SYN : Chỉ ra số thứ tự của đoạn tin sẽ được đồng bộ hoá. Cờ này được dùng khi mà kết nối được thiết lập.
Cờ FIN : Nếu cờ này thiết lập, nó chỉ ra rằng phía gửi không còn dữ liệu để gửi nữa. Điều này tương đương với việc đánh dấu kết thúc quá trình truyền.

Window - Cửa sổ (16 bits)

Trường này cấp phát thẻ dùng để kiểm soát luồng dữ liệu theo cơ chế cửa sổ. Đây là số lượng các byte dữ liệu khối tin mà phía thu có thể chấp nhận được.

Checksum (16 bits)

Chứa mã kiểm soát lỗi (theo phương pháp CRC) cho toàn bộ segment.

Urgent Pointer - Con trỏ khẩn (16 bits)

Trường này được dùng khi mà cờ URG được thiết lập; con trỏ này trỏ tới số hiệu tuần tự của các byte đi theo sau dữ liệu khẩn, cho phép bên nhận biết được độ dài của dữ liệu khẩn.

Options (có độ dài thay đổi)

Trường này dùng để xác định các Option của TCP. Mỗi lựa chọn bao gồm một số (1 byte) để chỉ ra lựa chọn đó, một số chỉ giá trị của các byte trong trường Option, và các giá trị lựa chọn. Hiện nay với TCP mới có 3 Option được định nghĩa, như sau:

Số 0 : Cuối danh sách các lựa chọn
Số 1 : Không hoạt động (*No Operation*)
Số 2 : Kích cỡ lớn nhất của một Segment

Trường Options chỉ để xác định kích thước lớn nhất của bộ đệm mà TCP nhận có thể chấp nhận được. Bởi vì TCP dùng trường dữ liệu có chiều dài thay đổi được nên có thể có trường hợp là máy gửi sẽ tạo ra một đoạn tin mà phía nhận không thể chấp nhận được.

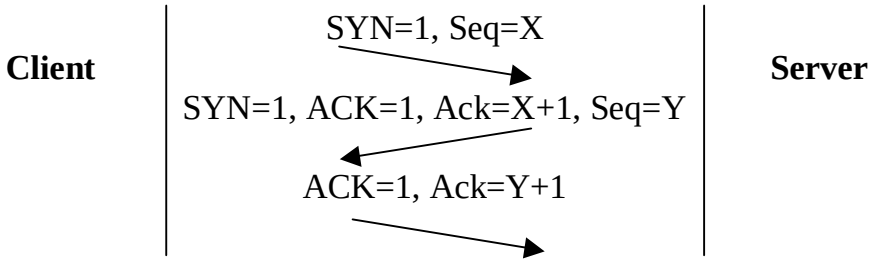
Padding :

Dùng để bổ sung vào Header để bảo đảm rằng phần Header luôn là bội số của 32 bit. Phần thêm vào bao gồm toàn số 0.

TCP Data (Có độ dài thay đổi)

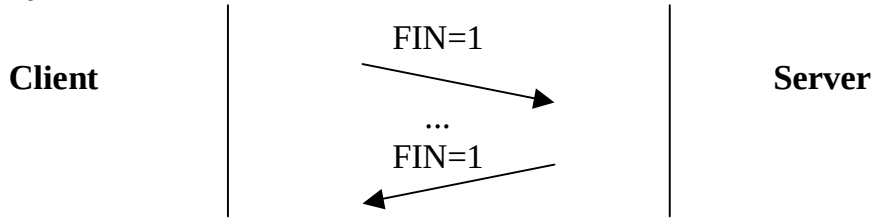
Chứa dữ liệu của tầng trên, độ dài tối đa ngầm định là 536 bytes. Giá trị có thể điều chỉnh bằng cách khai báo trong vùng Options.

7.2.2 Quá trình nối-tách



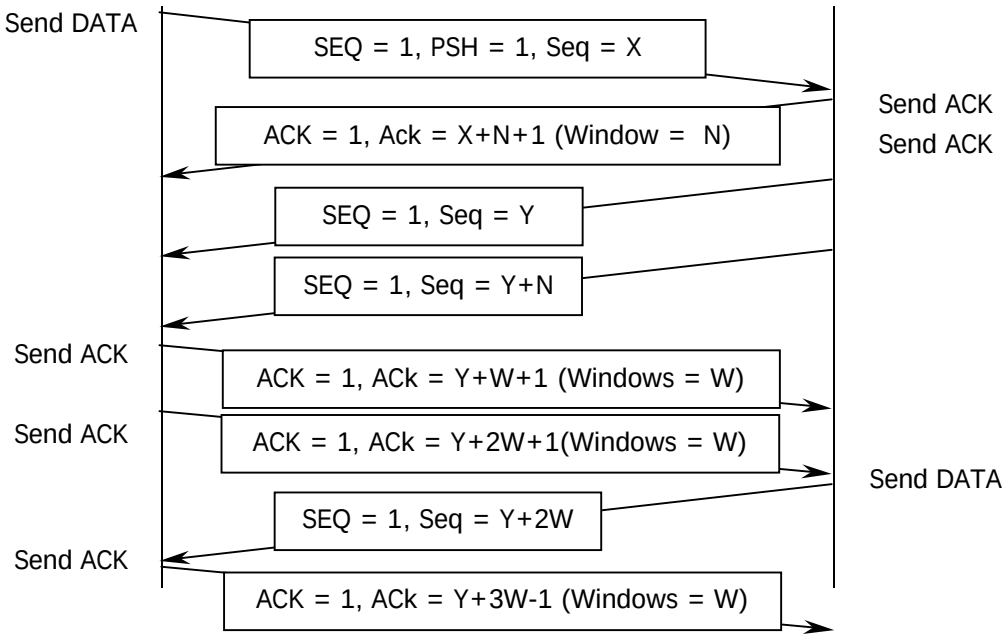
Quá trình thiết lập kết nối bằng thủ tục bắt tay 3 lần (three-way hand). Client gửi bản tin với SYN=1 (yêu cầu kết nối). Server nhận được, gửi bản tin với SYN=1 và ACK=1. Client lại đáp lại với bản tin ACK=1.

Kết thúc kết nối bằng thủ tục bắt tay hai lần (two-way hand). Bên kết thúc gởi số liệu, gởi bản tin với FIN=1, TCP cho phép nhận tiếp tục số liệu cho đến khi bên kia gửi bản tin FIN=1



Ngoài ra, thủ tục TCP/IP còn dùng để kết nối giữa LAN và WAN như một thủ tục cho mạng LAN.

7.2.3 Quá trình trao đổi dữ liệu



H×nh 7-4. S¬ ®å qu¸ tr×nh trao ®æi d÷ liÖu cña TCP.

W= maximun Segment size (W>N)

2W = Windows limit

7.2.4 Thứ tự thực hiện ứng dụng TCP/IP

Sự kết hợp của thủ tục TCP và IP thực sự là sự kết hợp giữa các mạng máy tính nối với nhau cho phép người dùng các mạng máy tính nối với nhau cho phép người dùng các mạng khác nhau liên lạc và làm việc được với nhau.

Thủ tục TCP là thủ tục tại đầu cuối, còn IP dùng để chạy trên mạng. Khi người sử dụng thủ tục TCP tạo được phân đoạn TCP và kết hợp vào IP để tạo thành IP datagram. Router căn cứ vào địa chỉ IP trong gói tin và thông tin chứa trong bảng định tuyến để chuyển gói này đi tới các router sau. Khi gói tin IP đến router cuối cùng, router này tìm và chuyển gói tin đến địa chỉ hệ thống đầu cuối.

Nếu IP datagram không chuyển tới đầu cuối được vì một lý do nào đó, nó sẽ bị hủy bỏ và giao thức IP không còn thông báo được điều này cho người sử dụng biết. Giao thức TCP cung cấp mối liên hệ tin cậy giữa các đầu cuối, đảm bảo dữ liệu phát đi đúng địa chỉ, không bị thiếu hay phát lặp nghĩa là tại điểm cuối cùng thủ tục TCP sẽ đọc số thứ tự trong phân đoạn TCP để biết gói tin bị thiếu hay gói đã nhận rồi và báo lại cho bên phát biết.

Gói tin IP không phụ thuộc vào các giao thức cụ thể của các mạng khác nhau mà nó đi qua (X.25 hay Frame relay v.v..). Với IP các mạng chỉ đơn thuần là đường dẫn các Router. Ta có thể hình dung IP datagram như một phong bì bình thường, người gửi thư không quan tâm đến bức thư đến được người nhận bằng ô tô, tàu hỏa hay máy bay.

Sự kết hợp giữa thủ tục TCP và IP giúp người dùng sử dụng được các dịch vụ trao đổi trên Internet thực hiện qua các bước chính sau đây:

Bước 1: Các dữ liệu ứng dụng kết hợp với số thứ tự để hình thành phân đoạn TCP.

Người sử dụng dùng dịch vụ trên mạng như thư điện tử, Telnet hay FTP v.v.. có nghĩa là đưa các dữ liệu của người dùng vào phần dữ liệu của gói tin TCP. Giao thức TCP sẽ đưa vào phần header của gói tin các thông tin sau:

- Số hiệu cổng quy định của Internet.
- Số thứ tự Segment gửi đi.
- Thông báo cho bên gửi biết đã nhận được Segment thứ mấy (ACK)
- Số byte cần phát.

Bước 2: Kiến tạo ra gói tin IP datagram

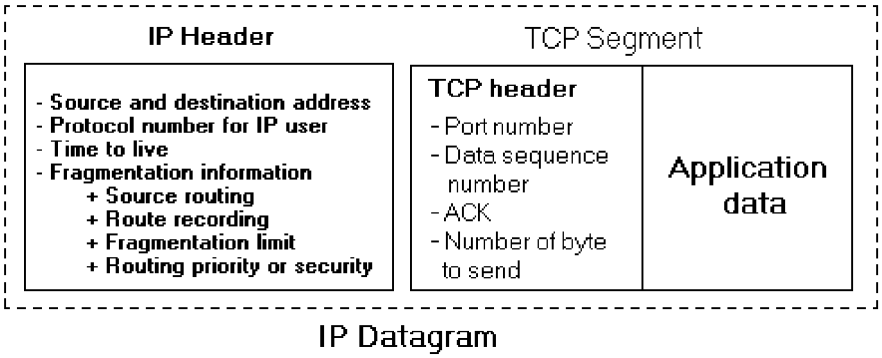
Trên cơ sở của gói tin TCP, IP thêm các thông tin sau đây vào để tạo thành IP Datagram.

- Địa chỉ phát và nhận : Router sử dụng địa chỉ này để định tuyến.
- Số thủ tục (Protocol number): định nghĩa thủ tục mà IP thực hiện.
- Thời gian tồn tại (Time to live): định nghĩa số Router bắt buộc Datagram phải đi qua trước khi nó bị hủy bỏ.
- Thông tin về các phân đoạn bị chia nhỏ trong quá trình chuyển đi trên mạng.

Kích thước gói tin thay đổi tùy thuộc vào mạng khác nhau, chẳng hạn như kích thước gói tin trong mạng Ethernet là 1500 bytes còn mạng X.25 chỉ có 128 bytes.

- Các thông tin tùy chọn

- Source Routing (Định tuyến bên phát): cung cấp danh sách các Router sử dụng.
- Route recording (Ghi lại tuyến đường đã đi qua): Thông tin sẽ yêu cầu mỗi Router ghi lại địa chỉ IP khi nó chuyển datagram qua, dùng để thống kê được số liệu của đường dẫn trong Internet.
- Fragmentation limit (Giới hạn phân mảnh): Định nghĩa cỡ lớn nhất (tính theo byte) của một datagram có thể chuyển đi mà không cần phải chia nhỏ.
- Routing priority or security (Ưu tiên hoặc bảo đảm an toàn cho Datagram): chỉ rõ tuyến nào dành ưu tiên hay tuyến nào bảo đảm được an toàn cho datagram.



Hình 7-5. Cấu trúc của IP Datagram.

Như vậy Datagram thực chất là hình thức một gói tin chứa dữ liệu thông tin được dùng trong internet.

Bước 4: Chuyển gói đến địa chỉ đích

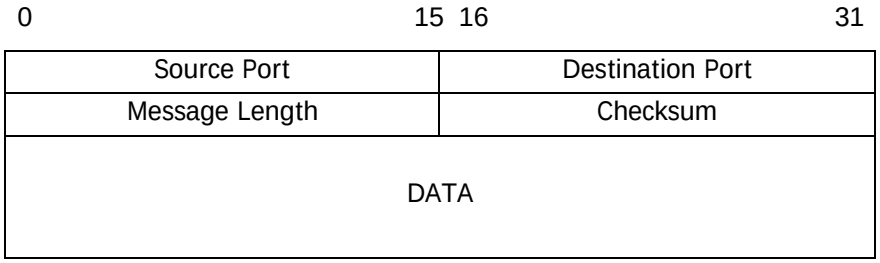
Các IP Datagram chuyển qua các lớp dưới đưa vào và định tuyến để tìm tới địa chỉ đến qua mạng căn cứ vào địa chỉ vật lý của mạng lưới ví dụ như địa chỉ mạng X.25, mạng Frame relay hoặc ngay bản thân của Internet. Tất cả các thông tin này đều nằm trong bảng định tuyến trong các Router. Các mạng X.25 hay Frame relay chỉ làm nhiệm vụ chuyển tải các Datagram.

Tại phía đầu cuối thu, TCP tách IP datagram để lấy phân đoạn TCP xử lý dữ liệu thông tin, đối chiếu số thứ tự, phát hiện những gói thiếu thiếu hay đã nhận được, đồng thời cũng nhận được thông báo (ACK) từ phía phát báo cho biết bên đấy đã nhận được gói thứ mấy do bên này phát đi.

Phía thu thông báo (ACK) cho bên phát biết số dữ liệu đã nhận được đồng thời cũng yêu cầu phát lại những gói tin thiếu nếu có.

7.3 Giao thức UDP

Giao thức UDP (User Datagram Protocol) cho phép người sử dụng gởi bản tin mà không cần thiết lập liên kết, do đó không bảo đảm việc giao nhận chính xác hoặc thứ tự bản tin. Giao thức UDP dùng cho dịch vụ không tin cậy 100%. Thực tế trong các mạng 99% bản tin UDP được giao nhận đúng đích. Do ít chức năng phức tạp nên UDP hoạt động nhanh hơn so với TCP.



Hình 7-6. Khuôn dạng của UDP Datagram.

Các trường có ý nghĩa như sau:

- *Source Port* - Số hiệu cổng nguồn (của máy gửi): Một trường có thể lựa chọn được với số hiệu cổng. Nếu một số hiệu cổng không xác định thì trường này có giá trị là 0.
- *Destination Port* - Số hiệu cổng trên máy nhận.
- *Message Length* - Chiều dài của dữ liệu trong đó cả phần Header và dũ liệu.
- *Trường Checksum*: là 16 bit bù một của phép tổng bù một của trường dữ liệu, có cả phần pseudoHeader giống như của TCP.

Trường checksum của UDP cũng có thể lựa chọn được, nhưng không được dùng. Không một checksum nào được dùng cho phần dữ liệu vì phần checksum của IP chỉ dùng cho phần Header IP mà thôi. Nếu phần checksum không được dùng thì các bit của trường này được thiết lập là 0.

Giao thức UDP được sử dụng trong một số tình huống đặc biệt :

- Khi truyền một dữ liệu nhỏ thì dùng UDP có hiệu quả hơn so với việc kết nối và hủy kết nối khi sử dụng TCP.
- Các ứng dụng hỏi đáp, mong muốn trả lời trong một thời gian ngắn sau khi người sử dụng gởi đi yêu cầu. Trả lời cũng là một cơ chế báo nhận. Người ta sử dụng giao thức UDP như trong các dịch vụ ứng dụng không yêu cầu độ chính xác cao như thông báo giờ hay các dịch vụ gởi nhắn tin, tỷ giá .. .

- Một số mô hình nén để truyền các thông tin audio, video, có thể chấp nhận được một vài gói dữ liệu bị hỏng hay thất lạc.
- Một vài ứng dụng có đô tin cậy riêng trong khi truyền dữ liệu thì nên dùng UDP hơn là TCP.

7.4 Cổng và Socket

7.4.1 Số hiệu cổng

Khi một máy khách kết nối vào máy chủ thì có thể yêu cầu nhiều dịch vụ khác nhau trên máy chủ. Mỗi dịch vụ đều có cách gởi và nhận dữ liệu theo quy ước riêng. TCP và UDP chỉ chịu trách nhiệm đưa dữ liệu từ một máy tính náy đến một máy tính khác, còn dữ liệu đó được gởi đến dịch vụ theo cách nào thì phải thông qua cổng của dịch vụ.

Cổng được đặc trưng bởi một số có giá trị từ 0 đến 65535. Các cổng chuẩn từ 0 - 1023 là cổng được dùng cho các dịch vụ phổ biến như FTP, eMAIL, POP3, HTTP, ... Không thể có hai tiến trình cùng sử dụng chung một số hiệu cổng.

Các số hiệu cổng (Port Numbers) được dùng thông dụng trong thực tế :

UDP Port		TCP Port	
0	Reversed	0	Reversed
7	Echo	1	TCP Multiplexor
37	Time	20	FTP_ Data Connection
42	Name Server	21	FTP_ Command Connection
53	Domain Name Server	23	TELNET
69	Trivial File Transfer Program (TFTP)	25	SMTP
514	System Log	42	Name Server
. . . . .		53	Domain Name Server
		79	Finger_ find a active user
		80	HTTP

7.4.2 Socket

Mỗi socket xác định một điểm cuối trong liên kết truyền thông hai chiều giữa các tiến trình giao tiếp trên mạng, là đối tượng mà qua đó các dịch vụ ứng dụng truyền hoặc nhận các gói dữ liệu trên mạng. Khi cần gởi dữ liệu đi, các tiến trình ghi dữ liệu vào socket, khi có dữ liệu đến, các tiến trình sẽ đọc socket để lấy dữ liệu.

Trong những năm 80, do nhu cầu cần có một giao diện lập trình ứng dụng API (Application Programming Interface) để phát triển các trình ứng dụng trên mạng TCP/IP, giao diện socket đã được xây dựng lần đầu tiên trên hệ điều hành UNIX. Loại Berkelay Socket (Berkeley Software Distribution - BSD, tại Trường Đại học

California ở Berkeley) được thiết kế với nguyên tắc truyền thông liên tiến trình (IPC - InterProcess Communication) trên mạng gắn liền với TCP/IP.

Các Socket cung cấp khả năng gửi và nhận dữ liệu thông qua kết nối mạng mà không đi sâu vào các mức và khuôn dạng gói tin, cơ chế quản lý bộ đệm. Cơ chế trao đổi Socket được sử dụng trong các mô hình mạng như sau :

- Mô hình Client/Server
- Mô hình mạng ngang hàng, như các trình ứng dụng "chat"
- Tạo các cuộc gọi thủ tục từ xa (Remote Procedure Calls - RPC)

Các kiểu socket :

1. Stream socket :

Sử dụng với giao thức TCP, có thiết lập kết nối giữa hai máy trước khi trao đổi dữ liệu.

Stream socket cung cấp cơ chế trao đổi dữ liệu theo hai chiều, tin cậy, có thứ tự và không phát lặp.

Không có biên giới giữa các record.

2. Datagram socket

Sử dụng với giao thức UDP, không cần thiết lập liên kết giữa hai máy trước khi truyền dữ liệu. Việc định vị Server và Client sẽ thông qua địa chỉ đích trong gói tin.

Datagram socket cung cấp cơ chế trao đổi dữ liệu theo hai chiều, không bảo đảm tin cậy và có thứ tự , có thể phát lặp.

Biên giới giữa các record trong dữ liệu gởi đi được bảo toàn.

3. Raw socket

Cung cấp sự truy xuất vào các giao thức giao tiếp nền có hỗ trợ socket. Các Socket này tình huống là các gói tin có định hướng mặc dù độ tin cậy phụ thuộc vào giao diện được cung cấp bởi giao thức.

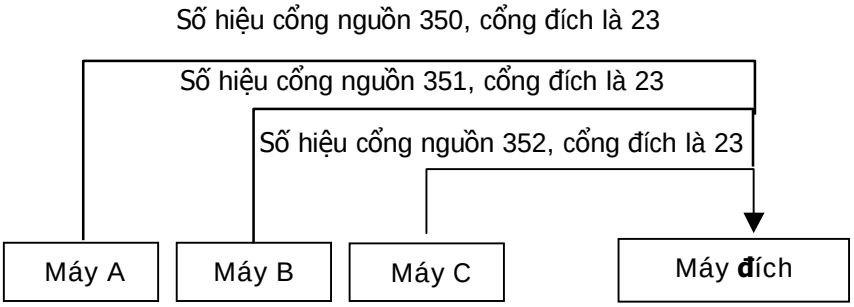
Raw socket chỉ dành cho các người sử dụng muốn phát triển các giao thức giao tiếp mới hoặc muốn truy xuất sâu vào các tiện ích bí mật của giao thức đó.

Một socket có ba thành phần chính :

- Giao diện được liên kết : là địa chỉ IP của máy.
- Port : số hiệu của cổng dịch vụ để truyền hoặc nhận dữ liệu,

- Loại socket : Stream socket hoặc Datagram socket.

Một liên kết giữa hai máy trên với nhau được xác định bởi một cặp socket : Socket (Host1, Port1) và Socket (Host2, Port2). Số **Socket** là duy nhất cho phép một tiến trình có thể giao tiếp với một tiến trình khác trên mạng.



Hình 7-7. Nhiều máy nguồn nối với một máy đích.

Một liên kết có thể được thiết lập theo một trong hai cách : chủ động (active) hoặc bị động. Các thực thể tầng trên sử dụng TCP thông qua bằng cách gọi các hàm dịch vụ nguyên thủy. Dịch vụ TCP được thiết lập 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. Vì các khung tin được đưa qua cổng đều có đầy đủ các thông tin về socket (với địa chỉ IP), cho nên không có xung đột dữ liệu xảy ra.

7.5 Mô hình giao tiếp Client/Server

TCP/IP phụ thuộc vào khái niệm máy khách (Client) và máy chủ (Server). Thuật ngữ Server dùng để chỉ những chương trình cung cấp các dịch vụ thông qua mạng. Các Server nhận đảm nhiệm chức năng đáp ứng các yêu cầu của máy khách, thực hiện việc phục vụ và trả lại kết quả. Thuật ngữ Client dùng để chỉ các chương trình ứng dụng gởi các yêu cầu đến Server và chờ kết quả trả về.

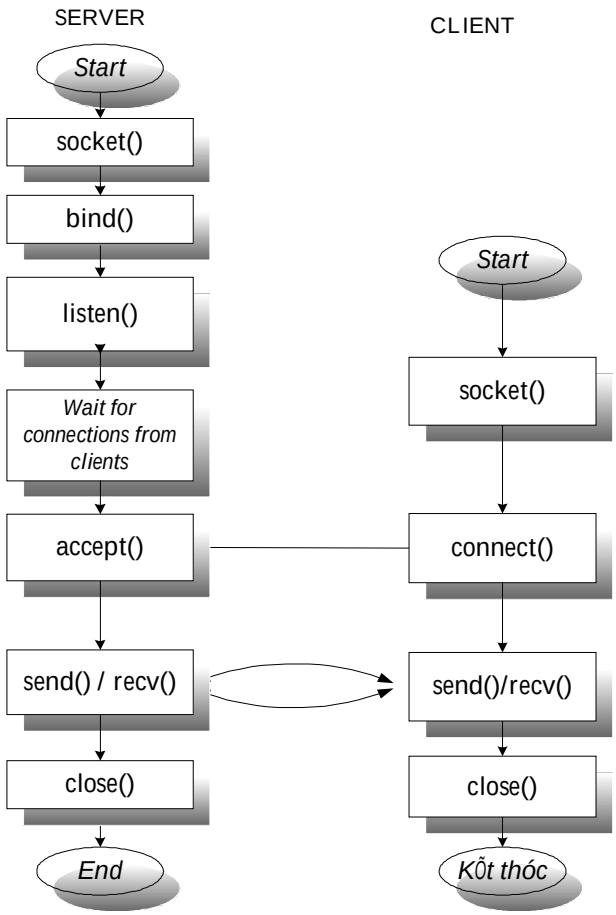
Các chương trình Client và Server thường thực thi trên các máy khác nhau. Mỗi chương trình Server có thể cùng đáp ứng cho nhiều chương trình Client trên nhiều máy tính khác nhau cùng một lúc.

7.5.1 Quá trình trao đổi dữ liệu dùng Stream Socket

Stream socket dựa trên nền giao thức TCP đòi hỏi phải tạo một kết nối trước khi hai bên có thể truyền hoặc nhận dữ liệu cho nhau. Stream Socket cung cấp một dòng các byte dữ liệu không có phân cách có thể truyền hai chiều. Các dòng dữ liệu có thể tin cậy được phân phát tuần tự, dữ liệu không trùng lắp, nghĩa là các gói dữ liệu được phân phát theo thứ tự được phát, và mỗi lần chỉ có một gói riêng biệt được truyền.

Dạng socket này rất thích hợp với mô hình Client/Server. Server sẽ tạo một socket, gán cho nó một tên (cung cấp một địa IP của máy và một port để giao tiếp), và đợi client nối kết đến socket. Bên client cũng tạo một socket và nối kết đến tên socket trên server. Khi server phát hiện có yêu cầu kết nối từ client, nó sẽ tạo một socket mới và sử dụng socket mới đó để giao tiếp với client. Socket cũ tiếp tục đợi kết nối từ các client khác.

Sơ đồ trao đổi dữ liệu giữa Client/Server bằng cách dùng Socket được biểu diễn như sau :



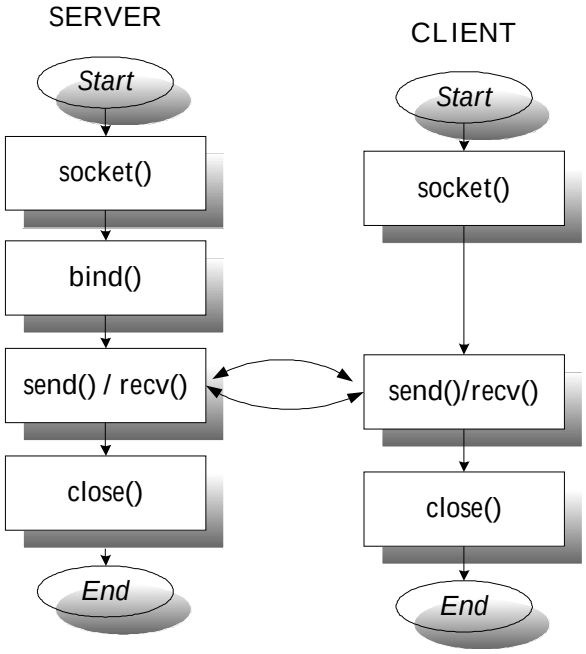
Hình 7-8. Sơ đồ trao đổi dữ liệu giữa Client/Server bằng StreamSocket.

7.5.2 Quá trình trao đổi dữ liệu dùng Datagram Socket

Datagram Socket dựa trên giao thức UDP không đòi hỏi phải thiết lập một kết nối trước khi truyền và nhận dữ liệu. Dữ liệu chỉ là một gói đơn, vì vậy dạng socket này thường dùng để truyền các mẫu tin, không cần nhiều các header lớp ứng dụng. Dạng socket này cung cấp luồng dữ liệu không bảo đảm theo thứ tự hoặc không bị trùng lắp, không bảo đảm dữ liệu sẽ đến được nơi nhận. Dữ liệu có thể đến không theo thứ tự được phát và có khả năng bị trùng lặp. Nhưng sự phân cách giữa các

mẫu tin thì được duy trì. Trong mạng LAN datagram có khả năng tin cậy tương đối tốt, nhưng trong mạng WAN, như mạng Internet thì không được đảm bảo.

§ Lưu đồ client/server sử dụng giao thức UDP



Hình 7-9. Sơ đồ trao đổi dữ liệu giữa Client/Server bằng DatagramSocket.

7.5.3 Ví dụ chương trình client/server

Trong ví dụ dưới đây chương trình server thực hiện các bước thiết lập cho việc chờ đợi một kết nối từ chương trình client. Sau khi thiết lập kết nối với client, cả hai thực hiện một số thao tác truyền và nhận thông tin rồi kết thúc chương trình.

7.5.3.1 Mã lệnh chương trình Server

- Tạo ra một socket với hàm *socket()*.
- Ràng buộc socket với một địa chỉ bằng hàm *bind ()*.
- Dùng hàm *listen()* để chờ đợi một kết nối.
- Nhận bất kỳ thông tin nào yêu cầu kết nối bằng hàm *accept()*.
- Nhận các thông báo gửi đến bằng hàm *read()* và gửi thông báo đến client bằng hàm *write()*.
-

```
/* mksock.c make and bind to a socket - userver*/
#include<stdio.h>
#include<sys/socket.h>
#include<sys/un.h>
#include<unistd.h>

void die(char * message);
void copyData(int from, int to);
```

```
int main(void) {
  struct sockaddr_un address;
  int sock,conn;
  size_t addrLength;
  if ((sock=socket(PF_UNIX,SOCK_STREAM,0))<0)
   die("socket");
/*unlik("./sample_socket");*/
  address.sun_family=AF_UNIX;
  strcpy(address.sun_path,"./sample_socket");

addrLength=sizeof(address.sun_family)+strlen(address.sun_path);
if(bind(sock,(struct sockaddr *)&address,addrLength))
  die("bind");
if(listen(sock,5))
  die("listen");
while((conn=accept(sock,(struct sockaddr*)&address,&addrLength))>=0) {
  printf("---getting data\n");
  copyData(conn,1);
  printf("---done\n");
  close(conn);
}
if (conn<0) die("accept");
close(sock);
return 0;
}
void die(char * message){
  perror(message);
  exit(1);
}
void copyData(int from,int to){
  char buf[1024];
  int amount;
  while ((amount=read(from,buf,sizeof(buf)))>0){
          if(write(to,buf,amount)!=amount){
                  die ("write");
                  return;
  }
}
if (amount<0) die("read");
}
```

7.5.3.2 Mã lệnh chương trình client

Từ chương trình client , để thực hiện được một kết nối đến server và truyền nhận thông tin chỉ cần thực hiện 2 bước cơ bản như sau:

- Tạo một *socket()* tương ứng với chương trình *server* cụ thể .
- Yêu cầu đến server thực hiện kết nối bằng cách gọi hàm *connect()*.

Nếu một kết nối được tạo ra, client có thể gửi yêu cầu bằng hàm *write()* và nhận các đáp ứng phản hồi bằng hàm *read()*.

```
/* sockconn.c - connect to a socket - uclient*/
#include<sys/socket.h>
#include<sys/un.h>
#include<unistd.h>

void die (char * message);
```

```
void copyData(int from, int to);

int main(void){
  struct sockaddr_un address;
  int sock;
  size_t addrLength;

  if ((sock=socket(PF_UNIX,SOCK_STREAM,0))<0)        die("socket");
  address.sun_family=AF_UNIX;
  strcpy(address.sun_path,"./sample_socket");

  addrLength=sizeof (address.sun_family) + strlen(address.sun_path);
  if(connect(sock,(struct sockaddr *)& address,addrLength))   die("connect");
  copyData(0,sock);
  close(sock);
  return 0;
}
void die(char * message){
  perror(message);
  exit(1);
}
void copyData(int from, int to){
  char buf[1024];
  int amount;
  while ((amount=read(from,buf,sizeof(buf)))>0){
    if(write(to,buf,amount)!=amount) {
                    die("write");
                    return;
    }
  }
if (amount<0)      die("read");
}
```

BÀI TẬP

1. Tìm hiểu các mô tả Socket và cấu trúc dữ liệu của socket mà hệ điều hành cấp phát để lưu trữ các thông tin cần thiết cho kết nối mạng.
2. Tìm hiểu các thư viện lập trình WinSock trên hệ điều hành Windows.
3. Viết các chương trình giao tiếp Client/Server theo mô hình giao tiếp TCP/IP hoặc UDP/IP.

Chương 8

TẦNG PHIÊN

Tầng phiên (Session Layer) làm nhiệm vụ tổ chức và đồng bộ sự chuyển đổi dữ liệu giữa các tiến trình ứng dụng khác nhau. Tầng Phiên làm việc với tầng ứng dụng để cung cấp các tập dữ liệu, được gọi là các điểm đồng bộ, các điểm này cho phép một ứng dụng biết quá trình truyền và nhận dữ liệu được thực hiện như thế nào.

Tầng phiên chịu trách nhiệm thiết lập và duy trì một phiên truyền thông giữa hai trạm hoặc nút mạng. Một phiên truyền thông qua một mạng hoạt động có phần giống với một cuộc gọi qua các đường dây điện thoại. Tầng Phiên cố gắng thiết lập một phiên truyền thông giữa hai nút trên một mạng. Cả hai nút đều thừa nhận phiên truyền thông này thường sẽ được gán một số hiệu nhận diện. Mỗi nút có thể ngắt phiên truyền thông giữa hai nút trên một mạng được gọi là *một cổng luận lý* (Socket). Khi một phiên truyền thông được thiết lập, một cổng luận lý sẽ được mở ra. Một phiên truyền thông được kết thúc được gọi là *một cổng luận lý bị đóng* (Close Socket).

Mục tiêu của tầng phiên là có khả năng cung cấp cho người sử dụng các chức năng cần thiết để quản lý các phiên ứng dụng cụ thể như:

- Điều phối việc trao đổi dữ liệu giữa các ứng dụng bằng cách thiết lập và giải phóng (một cách ***logic***) các phiên (hay gọi là các hội thoại ***dialogues***).
- Cung cấp các điểm đồng bộ hóa để kiểm soát việc trao đổi dữ liệu.
- áp đặt các qui tắc cho các tương tác giữa các ứng dụng của người sử dụng.
- Cung cấp cơ chế lấy lượt (nắm quyền) trong các quá trình trao đổi dữ liệu.

Trong tầng phiên thì vấn đề đồng bộ hóa được thực hiện tương tự như một cơ chế kiểm tra / phục hồi (*check point/reset*).Trong một hệ quản trị tập tin, dịch vụ này cho phép người sử dụng xác định các điểm đồng bộ hóa trong dòng dữ liệu và có thể khôi phục lại việc hội thoại bắt đầu từ một trong các điểm đó.

8.1 Dịch vụ OSI cho tầng Phiên

Tầng phiên làm việc quản lý các cuộc thoại giữa hai máy tính bằng cách thiết lặp, quản lý, và kết thúc các phiên truyền thông.

8.1.1 Cung cấp cho người sử dụng dịch vụ tầng phiên (SS-user)

- Thiết lập một liên kết với một người sử dụng dịch vụ tầng phiên khác, trao đổi dữ liệu với người sử dụng đó một cách đồng bộ và hủy bỏ liên kết một cách có trật tự khi không dùng đến nữa.

- Thương lượng về việc dùng các thẻ bài (TOKEN) để trao đổi dữ liệu,đồng bộ hóa và hủy bỏ liên kết ,sắp xếp phương thức trao đổi dữ liệu (half-duplex hoặc full-duplex).
- Thiết lập các điểm đồng bộ hóa trong các hội thoại và khi xảy ra sự cố thì có thể khôi phục lại việc hội thoại bắt đầu từ một điểm đồng bộ hóa đã thỏa thuận.
- Ngắt hội thoại và khôi phục lại hội thoại sau đó từ một điểm xác định trước.

Các dịch vụ xác định điểm đồng bộ hóa là nhằm vào hai mục đích :

1) Các điểm đồng bộ hóa có thể phân tách các phần của một hội thoại.
2) Các điểm đồng bộ hóa có thể dùng để phục hồi lỗi.

Các điểm đồng bộ hóa chính dùng để cấu trúc quá trình trao đổi dữ liệu thành một chuỗi các đơn vị hội thoại (dialogue), mỗi điểm này phải được xác nhận và người sử dụng sẽ bị hạn chế trong một số dịch vụ nhất định cho tới khi nhận được một sự xác nhận mới. Một điểm đồng bộ hóa chính được dùng để tách biệt các hai đơn vị hội thoại liên tiếp.

Các điểm đồng bộ hóa phụ được dùng để cấu trúc quá trình trao đổi dữ liệu ở trong một đơn vị hội thoại, và các điểm này không cần phải được xác định trước.Việc dùng các điểm đồng hóa phụ trong quá trình truyền tập nó sẽ ngăn chặn việc truyền lại dữ liệu với một khối lượng lớn

Một đơn vị hội thoại là một Activity (hành động) nguyên tử trong đó mọi hành động truyền thông không có liên quan gì đến bất kỳ một hoạt động truyền thông nào trước và sau đó. Một hành động bao gồm nhiều đơn vị hội thoại, và đây cũng chính là một tập hợp logic các nhiệm vụ liên quan với nhau; ở một thời điểm thì chỉ có một activity trên một liên kết phiên nhưng một activity thì có thể diễn ra trên nhiều liên kết phiên, nó có thể bị ngắt và sau đó có thể khôi phục lại trong một liên kết phiên khác, một vòng đời của một liên kết phiên thì có thể có nhiều Activity liên tiếp.

8.1.2 Điều khiển trao đổi dữ liệu

Việc trao đổi dữ liệu xảy như sau để thực hiện một trong ba phương thức như sau : hai chiều đồng thời (*full-duplex*), hai chiều luân phiên (*haft -duplex*), một chiều (*simplex*).

8.1.2.1 Trao đổi dữ liệu một chiều

Liên quan đến các đợt chuyển giao dữ liệu một chiều. Báo cháy là một ví dụ, nó gửi một thông điệp báo động đến trạm chống cháy, nhưng không thể (và không cần) nhận các thông điệp từ trạm chống cháy.

Với phương thức một chiều thì ít xảy ra: chẳng hạn như dữ liệu được gửi đến một đối tượng tạm thời không làm việc, thì chỉ có một chương trình nhận với một nhiệm vụ duy nhất là tiếp nhận dữ liệu đến và giữ lại.

8.1.2.2 Trao đổi dữ liệu hai chiều luân phiên

Liên quan đến các đợt chuyển giao dữ liệu hai chiều, ở đó các luồng dữ liệu mỗi lần đi theo mỗi hướng. Khi một thiết bị hoàn tất một phiên truyền, nó phải " trả lại " vật tải cho thiết bị kia để đến phiên thiết bị đó được truyền.

Với phương thức luân phiên hai chiều thì nảy sinh các vấn đề như sau :

- Các đối tượng sử dụng phiên phải “lấy lượt” để truyền dữ liệu (điển hình của phương thức này là dùng cho các ứng dụng hỏi đáp).
- Thực thể tầng phiên *(session entity)* duy trì tương tác luân phiên bằng cách báo cho các đối tượng khi đến lược họ sẽ truyền dữ liệu.

8.1.2.3 Trao đổi dữ liệu hai chiều đồng thời.

Cho phép tiến hành các đợt chuyển giao dữ liệu hai chiều đồng thời bằng cách cung cấp cho mỗi thiết bị một kênh truyền thông riêng biệt. Điện thoại tiếng là những thiết bị song công đầy đủ, và một trong hai bên của một cuộc đàm thoại có thể nói bất kỳ lúc nào. Hầu hết các môđem máy tính đều có thể hoạt động theo chế độ song công đầy đủ.

Chế độ truyền thông bán song công có thể dẫn đến tình trạng băng thông bị lãng phí trong quãng thời gian mà đợt truyền thông đang quay trả. Trong khi đó, chế độ truyền thông song công đầy đủ thường yêu cầu một ban thông lớn hơn so với chế độ truyền thông bán song công

Với phương thức hai chiều đồng thời thì cả hai bên cùng đồng thời gửi dữ liệu cùng một lúc, một khi phương thức này đã được thỏa thuận thì không đòi hỏi phải có nhiệm vụ quản trị tương tác đặt biệt đây cũng là một phương thức phổ biến nhất.

8.1.3 Điều hành phiên làm việc

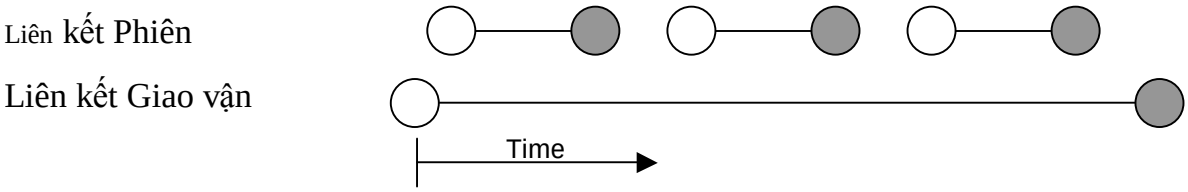
Phiên làm việc (session) là một cuộc thoại chính thức giữa một bên yêu cầu dich vụ và một bên cung cấp dịch vụ. Các phiên bản làm việc thường có ít nhất ba giai đoạn :

- *Thiết lập tuyến liên kết :* Bên yêu cầu dịch vụ sẽ yêu cầu khởi phát một dịch vụ. Trong quá trình xác lập, phiên truyền thông được thiết lập và các quy tắc được thoả thuận.
- *Chuyển giao dữ liệu* : Do các quy tắc được thoả thuận trong khi xác lập, nên mỗi bên của cuộc thoại sẽ biết nội dung mong đợi. Phiên truyền thông sẽ hữu hiệu và các lỗi cũng dễ phát hiện.
- *Giải phóng các kết nối* : Khi hoàn tất phiên làm việc, cuộc thoại kết thúc trong trật tự.

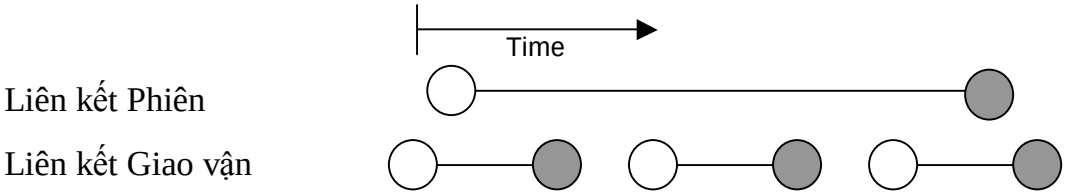
8.1.4 Liên kết phiên

Tầng Phiên thực hiện đặt tương ứng liên kết phiên với các liên kết giao vận. Trong một quá trình liên kết có thể xảy ra 2 trường hợp :

1. Một liên kết giao vận thiết lập với nhiều liên kết phiên liên tiếp :



2. Nhiều liên kết giao vận sử dụng cùng một liên kết phi[a]n:



Ký hiệu : : Thiết lập liên kết

: Giải phóng liên kết

8.2 Giao thức chuẩn tầng phiên

Giao thức chuẩn tầng phiên sử dụng tới 34 loại đơn vị dữ liệu (SPDU) khác nhau,và có khuôn dạng tổng quát như sau :

Trong đó :

- SI: Định danh của loại SPDU (một trong 34 loại)

- LI(length indicator): Chỉ độ dài của vùng tham số(parameters)
- PARAMETERS: vùng khai báo các tham số SPDU, mỗi loại SPDU có danh sách tham số riêng. Mỗi tham số được khai báo dưới dạng tổng quát gồm 3 vùng con : parameter identifier, length indecation, parameter value và chúng được gọi theo đơn vị pi hoặc PGI (mỗi đơn vị PGI gồm có 3 vùng con: PGI, LENGTH INDICATION, PARAMETER VALUE).
- User data: chứa dữ liệu của người sử dụng.

8.2.1 Các loại SPDU, các tham số và chức năng

SPDU	PARAMENTERS	FUNCTION
CONNECT	Connection ID, Protocol Options, Version Number, Serial Number, Token setting, Maximum TSDU size, Requirements, Calling SSAP, Called SSAP, User Data.	Initiate session Connection
ACCEPT	Same as CONNECT SPDU.	Etablist SESSION CONNECTION
REFUSE	Connection ID, Transport disconnect, Requirements, Version number, Season.	Reject connection request
FINISH	Transport Disconnect, User Data.	Initiate Orderly Release
DISC0NNECT	User Data.	Acknowledge orderly Release
NOT FINISHED	User Data.	Reject Orderly Release
ABORT	Transport disconnect, Protocol Error Code, User Data.	Abnormal connection Release
ABORT ACCEPT	Transport disconnect, Protocol Error Code, User Data.	Acknowledge Abort
DATA TRANSFER	Enclosure item,User Data.	Transfer normal Data
EXPEDITED	User data.	Transfer typed data
CAPABILITY DATA ACK	User Data.	Acknowledge Capability data
GIVE TOKENS	Tokens.	Transfer tokens
PLEASE TOKENS	Tokens , User Data.	Request token Assignment
GIVE TOKENS CONFIRM	-	Transfer all tokens
GIVE TOKENS ACK	-	Acknowledge all tokens
MONOR SYNC POINT	Confirm required flag, Serial number, User data.	Define minor sync point
MINOR SYNC ACK	Serial number, User Data.	Acknowledge minor sync point
MAJOR SYNC POINT	End of activity flag, Serial number, User Data.	Define major sync point
MAJOR SYNC ACK	Serial number,User data.	Acknowledge major sync point
RESYNCHRONIZED	Tokens sittings, resync type, serial number, user data.	Resynchorize
RESYNCHRONIZED ACK	Tokens settings, Serial number, User Data.	Acknowledge resynchronize

PREPERE	Type.	Notify type SPDU is coming
EXCEPTION REDORT	SPDU bit pattem.	Protocol Error detected
EXCEPTION DATA	Reason, User Data.	Put protocol in Error state
ACTIVITY START	Activity ID, User data.	Signal beginning of activity
ACTIVITY RESUME	Connect ID, Old activity ID, New Activity ID, User data.	Signal resumption of activity
ACTIVITY INTERRUPT	Reason.	Interrupt activity
ACTIVITY INTERRUPT ACK	-	Acknowledge interrupt
ACTIVITY DISCARD	Reason.	Cancel activity
ACTIVITY DISCARD ACK	-	Acknowledge cancellation
ACTIVITY END	Serial number/User data.	Signal activity end
ACTIVITY END ACK	Serial number/User data.	Acknowledge activity end

Tầng Phiên đóng một vai trò quan trọng trong việc trao đổi thông tin giữa các máy Client với máy Server. Nhưng thông tin mà chúng ta cần truyền tải thì được chia nhỏ ra thành các khung (hay gói) trước khi chúng được truyền tải qua một mạng. Mỗi tầng của mô hình 7 tầng OSI đều có thể bổ sung thêm các thông tin vào đoạn đầu và đoạn cuối của một khung dữ liệu vvà sau đó các thông tin này sẽ được đọc bởi tầng tương đương ở máy trạm tiếp nhận. Và một số tầng khác có thể bổ sung thêm phần đầu(header) và cả một phần đuôi(trailer) vào khung dữ liệu có sẵn. Sau đó, khung dữ liệu này truyền chuyển tới tầng tương đương trên trạm tiếp nhận.

Chương 9

TẦNG TRÌNH DIỄN

Tầng Trình diễn có nhiệm vụ phân cách giữa các tầng cao hơn và các tầng thấp hơn từ định dạng dữ liệu của tầng ứng dụng, chuyển đổi định dạng dữ liệu từ định dạng của tầng ứng dụng thành định dạng thông thường, gọi là "trình diễn hợp với quy tắc". Tầng Trình diễn xử lý dữ liệu không phụ thuộc vào máy tính từ tầng ứng dụng thành dữ liệu có định dạng phụ thuộc vào máy tính để chuyển cho các tầng thấp hơn.

Tầng trình diễn xử lý cú pháp, hoặc các quy tắc văn phạm, cần thiết cho phiên truyền thông giữa hai máy tính, bảo đảm cho các hệ thống cuối truyền th6ng có kết quả khi chúng sử dụng các dạng biểu diễn dữ liệu khác nhau. Tầng này trình bày một dạng thức dữ liệu đồng dạng cho tầng ứng dụng.

9.1 Vai trò và chức năng

Mục đích của tầng trình diễn là đảm bảo cho các hệ thống cuối có thể truyền thông có kết quả ngay cả khi chúng sử dụng các biểu diễn dữ liệu khác nhau .Để đạt được điều đó nó cung cấp một biểu diễn chung để dùng trong truyền thông và cho phép chuyển đổi từ biểu diễn cục bộ sang biểu diễn chung đó.

Tồn tại 3 dạng cú pháp thông tin được trao đổi giữa các thực thể ứng dụng :

- Cú pháp dùng bởi thực thể ứng dụng nguồn.
- Cú pháp dùng bởi thực thể ứng dụng đích.
- Cú pháp dùng bởi giữa các thực thể trình diễn ,loại cú pháp này gọi là cú pháp truyền (transfer syntax).

Tầng trình diễn đảm nhận việc chuyển đổi biểu diễn thông tin giữa cú pháp truyền và mỗi một cú pháp kia khi có yêu cầu

Chú ý rằng không tồn tại một cú pháp truyền xác định trước duy nhất cho mọi hoạt động trao đổi dữ liệu. Cú pháp truyền được duy nhất cho mọi hoạt động trao đổi dữ liệu. Cú pháp truyền được sử dụng trên một liên kết cụ thể của tầng trình diễn phải được thương lượng giữa các thực thể trình diễn tương ứng. Mỗi bên lựa chọn một cú pháp truyền sao cho có thể sẵn sàng được chuyển đổi sang cú pháp người sử dụng và ngược lại. Ngoài ra cú pháp truyền được chọn phải phản ánh các yêu cầu dịch vụ khác chẳng hạn nhu cầu nén dữ liệu .việc thương lượng cú pháp truyền sử dụng có thể được thay đổi trong vòng đời liên kết đó .Tầng trình diễn chỉ liên quan đến cú pháp truyền vì thế trong giao thức sẽ không quan tâm đến các cú pháp sử dụng bởi thực thể ứng dụng. Tuy nhiên mỗi thực thể trình diễn phải chịu trách nhiệm chuyển đổi giữa cú pháp người sử dụng và cú pháp truyền.

Các khái niệm liên quan đến bối cảnh của tầng trình diễn : Khi qua ranh giới giữa hai tầng trình diễn và tầng phiên có một sự thay đổi quan trọng trong cách nhìn dữ liệu. Đối với tầng phiên trở xuống tham số User Data trong các service primitives được đặc tả dưới dạng nhị phân (một chuỗi các byte). Giá trị này có thể được đưa vào trực tiếp trong các SDU (Service Data Unit) để chuyển giữa các tầng trong một hệ thống và trong các PDU (Protocol Data Unit) để chuyển giữa các tầng đồng mức ở hệ thống kết nối với nhau. Tuy nhiên tầng ứng dụng lại liên quan chặt chẽ với cách nhìn dữ liệu của người sử dụng nói chung cách nhìn đó là một tập thông tin có cấu trúc nào đó như là văn bản (text) trong một tài liệu một tệp về nhân sự hoặc một cơ sở dữ liệu .... Người sử dụng chỉ quan tâm đến ngữ nghĩa (semantics) của dữ liệu. Do đó tầng trình diễn ở giữa chỉ có nhiệm vụ cung cấp phương thức biểu diễn dữ liệu và chuyển đổi thành các giá trị nhị phân dùng cho các tầng dưới nghĩa là tất cả những gì liên quan đến cú pháp của dữ liệu

Tuy nhiên trong thực tế không thể tách bạch hoàn toàn giữa cú pháp và ngữ nghĩa và ngữ nghĩa dữ liệu. Nếu tầng ứng dụng không biết gì về cú pháp thì tầng trình diễn không biết gì về nhữ nghĩa thì không thể nào hoàn tất được việc kết hợp ngữ nghĩa với cú pháp dùng để tạo ra một biểu diễn cụ thể các giá trị dữ liệu cho dịch vụ phiên.

ở tầng ứng dụng thông tin được biểu diễn dưới dạng cú pháp trừu tượng (abstract syntax) liên quan đến các kiểu dữ liệu (data values) cú pháp trừu tượng này đặc tả một cách nhìn hình thức dữ liệu độc lập với mọi biểu diễn cụ thể.

Do vậy một cú pháp trừu tượng có nhiều đặc điểm giống kiểu dữ liệu như các ngôn ngữ lập trình Pascal, C .... Các ngữ nghĩa như là BNF. Các giao thức tầng ứng dụng mô tả các PDU của chúng bằng một cú pháp trừu tượng. Tầng trình diễn tương tác với tầng ứng dụng cũng dựa trên cú pháp trừu tượng này, tầng trình diễn có nhiệm vụ dịch thuật cú pháp trừu tượng của tầng ứng dụng và cú pháp truyền (transfer syntax) mô tả các giá trị dữ liệu dưới dạng nhị phân thích hợp cho việc tương tác với dịch vụ phiên việc dịch thuật này được thực hiện nhờ qui tắc mã hoá chỉ rõ biểu diễn của mỗi giá trị dữ liệu thuộc một kiểu nào đó .

Trước khi sử dụng liên kết của một tầng trình diễn để trao đổi dữ liệu thì hai thực thể trình diễn ở hai đầu phải thoả thuận về cú pháp truyền được xem như là bối cảnh trình diễn (presentation context) được dùng để trao đổi dữ liệu

Cú pháp truyền phải yểm trợ cú pháp trừu tượng tương ứng. Ngoài ra cú pháp truyền có thể có các thuộc tính khác không liên quan gì đến cú pháp trừu tượng mà nó yểm trợ ví dụ một cú pháp trừu tượng có thể yểm trợ bởi bất kì một cú pháp truyền về cơ bản thì giống nhau chỉ khác nhau ở chỗ một cung cấp khả năng mật mã, một chỗ cung cấp cả hai và một không cung cấp khả năng nào.

9.1.1 Phiên dịch dữ liệu

Một mục tiêu quan trọng cần giải quyết khi thiết kế các mạng đó là cho phép kiểu máy tính khác nhau trao đổi dữ liệu. Tuy mục tiêu này ít khi được giải quyết toàn vẹn, nhưng việc vận dụng hiệu quả các kỹ thuật phiên dịch dữ liệu có thể giúp nhiều kiểu máy tính truyền thông với nhau. Có bốn dạng phiên dịch dữ liệu, thứ tự bít, thứ tự byte, mã ký tự, và cú pháp tập tin như sau :

- Thứ tự bit : Khi số nhị phân được truyền qua một mạng, chúng gởi đi theo từng bit, thứ tự byte, mã ký tự, và cú pháp tập tin.
- Phiên dịch thứ tự Byte : Các giá trị phức tạp thường phải được biểu thị bằng nhiều byte, nhưng các máy tính khác nhau thường dùng quy ước khác nhau về việc sẽ truyền byte nào trước. Các bộ vi xử lý Intel bắt đầu bằng byte ít quan trọng nhất. Do chúng bắt đầu tại đầu nhỏ, nên được gọi là kết đầu nhỏ. Các bộ vi xử lý Motorola bắt đầu bằng byte quan trọng nhất. Để hoà hợp những khác biệt này, ta cần phải có tính năng phiên dịch thứ tự byte.
- Phiên dịch mã ký tự : Hầu hết các máy tính đều dùng một trong các bảng mã đánh số nhị phân dưới đây để biểu thị các bộ ký tự : Bảng mã ASCII được dùng để biểu thị các ký tự tiếng Anh trên tất cả máy tính và hầu hết các máy tính mini. EBCDIC (Extended Binary Coded Decimal Interchange Code = Mã hoán đổi thập phân mã hoá nhị phân mở rộng) được dùng để biểu thị cho các ký tự tiếng Anh trên máy tính lớn nhất.
- Phiên dịch cú pháp tập tin : Khi các dạng thức tập tin khác nhau giữa các máy tính, các dạng đó đòi hỏi phải phiên dịch.

9.2 Dịch vụ OSI cho tầng trình diễn

Dịch vụ OSI cho tầng trình diễn có 2 loại : một loại bao gồm các dịch vụ liên quan đến biểu diễn của dữ liệu người sử dụng để đảm bảo cho hai thực thể ứng dụng có thể trao đổi dữ liệu thành công ngay khi chúng dùng các biểu diễn cục bộ khác nhau cho dữ liệu đó, loại thứ hai bao gồm các dịch vụ cho phép các thực thể ứng dụng có thể sử dụng các dịch vụ tầng phiên để quản lý hội thoại.

Để cung cấp loại dịch vụ thứ nhất tầng trình diễn thực hiện hai nhiệm vụ sau :

- Thương lượng về cú pháp truyền : với mỗi kiểu dữ liệu người sử dụng cho trước một cú pháp truyền được thương lượng.
- Chuyển đổi : dữ liệu cung cấp bởi người sử dụng được chuyển đổi thành biểu diễn theo cú pháp truyền để truyền đi , ngược lại dữ liệu nhận được để giao cho người sử dụng sẽ chuyển đổi từ biểu diễn theo cú pháp truyền sang biểu diễn của người sử dụng.

ở thời điểm bất kì trong vòng đời của một liên kết trình diễn dịch vụ trình diễn dịch vụ trình diễn có liên quan đến một hoặc nhiều bối cảnh trình diễn (presentation context). Mối bối cảnh chỉ rõ cú pháp trừu tượng của dữ liệu đó. Có hai loại bối cảnh được sử dụng :

- Defined context set : bao gồm các bối cảnh đã được xác định thông qua sự thoả thuận giữa người sử dụng dịch vụ trình diễn (presentation service user) vaf người cung cấp dịch vụ trình diễn (presentation service provider).
- Default context : là một bối cảnh trình diễn mà người cung cấp dịch vụ trình diễn luôn luôn biết rõ và người sử dụng khi vắng mặt

Ở tầng phiên do kiến trúc phân tầng của ISO các thực thể ứng dụng không thể truy cập trực tiếp tới các dịch vụ tầng phiên, do vậy các yêu cầu dịch vụ liên quan đến tầng phiên phải được chuyển qua tầng trình diễn đến các dịch vụ tầng phiên.

9.3 Giao thức chuẩn tầng trình diễn

Giao thức chuẩn của ISO/CCITT cho tầng Trình diễn đặc tả những nội dung chính sau đây:

- Cấu trúc và mã hoá các đơn vị dữ liệu của giao thức trình diễn (PPDU) dùng để truyền dữ liệu và thông tin điều khiển .
- Các thủ tục để truyền dữ liệu và thông tin điều khiển giữa các thực thể trình diễn của hai hệ thống mở.
- Liên kết giữa giao thức trình diễn với dịch vụ trình diễn và với dịch vụ phiên .

Cũng như các PDU ở các tầng khác ,các PPDU cũng có khuôn dạng tổng quát bao gồm một phần đầu (header) chứa các thông tin điều khiển và có thể thêm mmột phần chứa dữ liệu được truyền từ trên xuống hoặc được truyền lên cho tầng trên. Giao thức trình diễn sử dụng 14 PPDU được liệt kê trong bảng 2-17 cùng với các tham số của chúng .

Qua bảng trên ta thấy số lượng PPDU không nhiều như số lượng SPDU (ở tầng Phiên) và nhiều tham số (có đánh dấu *) là giống với các tham số của các SPDU. Như vậy cả về phương diện dịch vụ và giao thức, tầng trình diễn và tầng Phiên có một mối liên kết rất chặt chẽ .

Qua xem xét các tầng dưới từ tầng phiên trở xuống, chúng ta thấy có 2 nguyên lý sau đây luôn được tuân thủ :

- Mỗi dịch vụ tầng n được cài đặt nhờ trao đổi các nPDU;
- Mỗi nPDU trở thành User data và được “nhét” vào trong một (n-1) PDU;

Tuy nhiên ở tầng trình diễn (và cả ở tầng ứng dụng mà ta sẽ thấy), các nguyên lý đó không còn luôn luôn được áp dụng .Thực tế là không phải mọi dịch vụ trình diến đều yêu cầu các PPDU và một số tham số của một số PPDU không được chuyển thành User data trong một SPDU. Để giải thích động cơ củ a sự khác biệt đó, ta xem xét hai dịch vụ trình diễn : thiết lập liên kết (connection etablishment) và chuyển thẻ bài (token passing).

Khi phát triển các giao thức cho 3 tầng cao của Mô hình óI,người ta thấy rõ ràng nên thương lượng và thiết lập đồng thời các liên kết Phiên ,trình diễn và ứng dụng,mặc dù điều đó đòi hỏi một quan hệ 1-1 chặt chẽ (không có dồn kênh) với cùng vòng đời cho cả ba loại liên kết .Quá trình thiết lập đồng thời các liên kết đó được gọi là quá trình nhúng (embeding), vì các PDU CONNECT.request và CONNECT.response cho cả ba tầng cao đó ,cái này được nhúng vào trong cái kia.

Khuôn dạng của các PDDU header được đặc tả theo cú pháp trừu tượng chuẩn.

9.3.1 Các chuẩn khác cho tầng trình diễn

Ngoài các chuẩn về dịch vụ và giao thức cho tầng Trình diễn như đã trình bày ở trên, ISO và CCITT đã phát triển các chuẩn liên quan đến cú pháp trừu tượng (Abstract Syntas) và quy tắc mã hoá (Encoding Rules) mà chúng ta đã nói đến khi trình bày vai trò và chức năng của tầng Trình diễn

Các chuẩn của ISO gồm có :

- ISO 8824:Abstract Syntax Notation One(viết tắt là ASNI.1)
- ISO 8825:Báic Encoding Rules(Viết tắt là BER)
- Tương ứng CCITT có các khuyến nghị X208 (ANSI.1) và X.209 (BER).

Khái niệm cú pháp trừu tượng mà ISO và CCITT định nghĩa được dựa trên khái niệm kiểu dữ liệu (data type) mà chúng ta đã quen thuộc trong các ngôn ngữ lập trình phổ biến. Thông thường các ngôn ngữ này định nghĩa trước các kiểu dữ liệu đơn giản như integer và boolean, cùng với các phương thức tổ hợp các kiểu đơn giản đó để có các cấu trúc dữ liệu phức tạp hơn. Hơn nữa, các phương pháp tổ hợp có thể thực hiện một cách đệ quy cho phép tạo ra các kiểu phức tạp tuỳ ý.

Chương 10

TẦNG ỨNG DING

Tầng ứng dụng giao tiếp trực tiếp với người sử dụng. Nhiệm vụ của tầng ứng dụng là hiển thị các thông tin nhận được và gửi các thông tin mới của người sử dụng cho các tầng thấp hơn.

Tầng ứng dụng liên quan đến tiến trình cung cấp các dịch vụ trên mạng, các dịch vụ này bao gồm : dịch vụ tập tin, dịch vụ in, dịch vụ cơ sở dữ liệu, và các dịch vụ khác.

Chúng ta sẽ xem xét các vấn đề trước khi bắt đầu với các ứng dụng. Đó là sự an toàn mạng, dịch vụ tên miền DNS dùng để điều khiển đặt tên trong Internet, giao thức hỗ trợ quản trị mạng, phần còn lại là các ứng dụng thực như thư điện tử, UserNet, FTP, Telnet, WWW ...

10.1 An toàn thông tin trên mạng

Việc kết nối mạng máy tính nhằm sử dụng và chia sẻ tài nguyên của các đối tượng trong hệ thống mạng cho dù họ có thể cách xa nhau về mặt địa lý. Tài nguyên hệ thống ở đây chủ yếu là là thông tin. Tuy nhiên đây là loại tài nguyên dễ bị xâm phạm, bị đánh cắp, bị tráo đổi nhất, đặc biệt là nó đang được trong lưu giữ trong môi trường mạng đầy phức tạp và phải chia xẻ cho nhiều người dùng khác nhau ở những vị trí khác nhau.

Vấn đề an toàn thông tin trên mạng đòi hỏi phải sử dụng nhiều biện pháp khác nhau từ cơ bản đến phức tạp, tuỳ theo lượng thông tin cần bảo vệ và khả năng cho phép của từng hệ thống cụ thể.

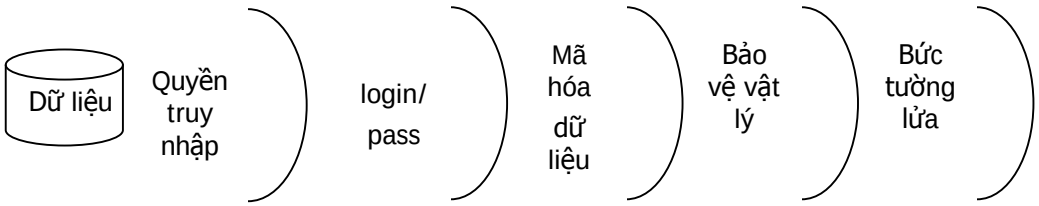
10.1.1 Các chiến lược an toàn hệ thống

1. Quyền hạn tối thiểu : Đây là chiến lược nền tảng nhất. Theo nguyên tắc này bất kì đối tượng nào cũng chỉ có những quyền hạn nhất định đối với những tài nguyên mạng nhất định khi thâm nhập vào mạng.
2. Bảo vệ theo chiều sâu : Tạo nhiều cơ chế an toàn cho hệ thống để chúng hỗ trợ cho nhau.
3. Cơ chế nút thắt : Tạo ra một “cửa khẩu” hẹp và chỉ cho phép thông tin đi vào hệ thông của mình bằng duy nhất con đường này. Đồng thời phải tổ chức một cơ chế kiểm soát và điều khiển các luồng thông tin đi qua cửa khẩu này.
4. Tính toàn cục : Các hệ thống an toàn đòi hỏi phải có tính toàn cục của các hệ thống cục bộ. Nếu có kẻ nào đó có thể bẻ gãy một cơ chế an toàn thì chúng có thể thành công bằng cách tấn công hệ thống nội bộ từ bên trong.

5. Tính đa dạng của việc bảo vệ : Cần phải sử dụng nhiều biện pháp khác nhau cho những hệ thống khác nhau. Nếu không, kẻ nào đó tấn công được hệ thống này thì cũng có thể tấn công vào hệ thống khác.

- Các mức bảo vệ thông tin trên mạng:

Vì không có một giải pháp bảo vệ nào an toàn tuyệt đối nên người ta thường sử dụng nhiều mức bảo vệ khác nhau tạo thành nhiều lớp rào chắn cho hệ thống. Mô hình như sau :



Hình 10-1. Các mức bảo vệ thông tin trên mạng.

10.1.2 An toàn thông tin bằng mã hóa

Để bảo vệ thông tin trên đường truyền, người ta chuyển đổi nó từ dạng nhận thức được sang dạng không nhận thức được trước khi truyền đi trên mạng nhằm bảo đảm tính bí mật cần thiết. Quá trình này diễn ra ở trạm phát được gọi là mã hoá thông tin (encrypting), ở trạm nhận phải thực hiện quá trình ngược lại, tức là biến đổi thông tin từ dạng không nhận thức được (đã mã hoá) sang dạng nhận thức được (dạng gốc), quá trình này gọi là giải mã (decrypting). Đây là một lớp bảo vệ thông tin rất quan trọng và được ứng dụng trong hầu hết các hệ thống mạng.

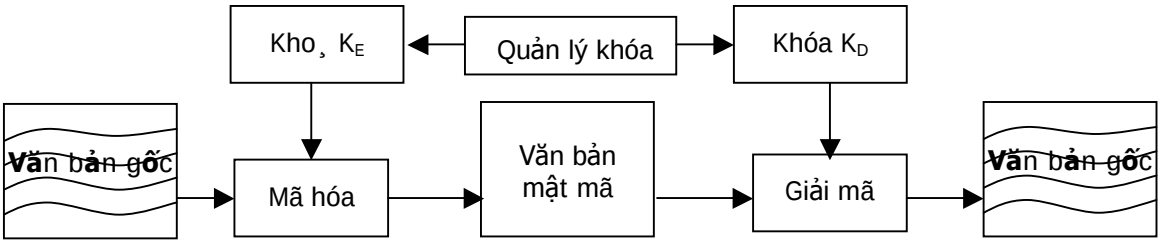
Để bảo vệ thông tin bằng mật mã, người ta thường tiếp cận theo hai hướng:

- Từ nút đến nút (end_to_end)
- Theo đường truyền (link_oriented security)

Theo cách thứ nhất, thông tin được mã hoá để bảo vệ trên đường truyền giữa hai nút mà không quan tâm đến nguồn và đích của thông tin đó. Ớ đây ta chú ý rằng thông tin chỉ được bảo vệ trên đường truyền, tức là ở mỗi nút đều có quá trình giải mã để sau đó thông tin được chuyển đi tiếp, do đó các nút cần được bảo vệ tốt.

Ngược lại theo cách thứ hai, thông tin trên mạng được bảo vệ trên toàn đường truyền từ nguồn đến đích. Thông tin sẽ được mã hoá ngay sau khi mới tạo ra và chỉ được giải mã khi đã về đến đích. Cách này có nhược điểm là chỉ có dữ kiệu người dùng mới được mã hoá còn các thông tin điều khiển thì phải giữ nguyên để có thể xử lí tại các nút.

Quá trình mã hoá và giải mã được mô tả như sau :



Hình 10-2. Sơ đồ quá trình mã hóa.

+ Văn bản gốc (plaintext) là văn bản chưa được mã hoá.

+ Khoá (key) : gồm một số hữu hạn các bit thường được biểu thị dưới dạng các xâu kí tự chữ số, số thập phân hoặc thập lục phân. Trong thực tế thường dùng các khoá có 8 kí tự.

Nếu gọi : M là văn bản gốc

C là văn bản mật mã (Ciphertext)

E là hàm mã hoá (Encryption Function)

D là hàm giải mã (Descryption Function)

Ta có hàm biểu diễn sự phụ thuộc giữa văn bản gốc và văn bản mã như sau:

$C = E(M)$

$M = D(C) = D(E(M))$

Khoá KE được dùng để mã hoá, khoá KD được dùng để giải mã .

Có rất nhiều phương pháp mã hoá nhưng tất cả đều qui về 2 phương pháp chung tuỳ theo việc sử dụng cặp khoá KD và KE:

- Khoá KD trùng với khoá KE : phương pháp này gọi là mã hoá khoá đối xứng, với phương pháp này yêu cầu khoá phải được giữ bí mật tuyệt đối, vì khoá dùng để mã hoá cũng được dùng giải mã.
- Khoá KD khác với khoá KE : phương pháp này gọi là mã hoá khoá công khai. Trong đó, có thể chuyển đổi vai trò giữa 2 khoá và rất khó để suy ra khoá này từ khoá kia. Khoá mã hoá (KE) có thể đưa ra công khai nhưng khoá dùng giải mã (KD) phải được giữ bí mật tuyệt đối.

Người ta còn phân biệt 2 loại khóa:

- Các khoá dùng trong thời gian dài gọi là khoá chính (primary) hay khoá mã hoá (key encryption).
- Các khoá được dùng trong khuôn khổ một cuộc truyền thông gọi là khoá làm việc (working) hay khoá mã hoá dữ liệu (data encryption).

10.2 CáC phương pháp mã hóa dữ liệu

10.2.1 Phương pháp hoán vị

Phương pháp này sắp xếp lại các kí tự trong văn bản gốc để tạo ra văn bản mật mã. Phương pháp này có một số kỹ thuật sau :

1. Đảo ngược toàn bộ văn bản gốc

Từ văn bản gốc, ta mã hoá bằng cách viết theo thứ tự ngược lại. Ví dụ DHKTDN được mã hoá thành NDTKHD. Đây là một trong những phương pháp mã hoá đơn giản nhất và chỉ mang tính tham khảo vì không an toàn.

2. *Mã hoá theo mẫu hình học*

Sắp xếp lại văn bản gốc theo mẫu hình học nào đó (thường là ma trận 2 chiều) để tạo văn bản mật mã.

Ví dụ : ĐAIHOCĐANANG được viết thành ma trận 3 x 4:

Đ	A	I	H
O	C	Đ	A
N	A	N	G

Nếu ta lấy các kí tự ra theo thứ tự các hàng là 3,1,2 ta sẽ có văn bản mật mã là N A N G O C Đ A Đ A I H. Phương pháp cũng kém an toàn, có thể dựa vào tần số xuất hiện của các kí tự trong bản mã để suy ra văn bản gốc.

3. *Đổi chổ cột*

Sắp xếp lại văn bản gốc thành dạng hình chữ nhật theo các cột, sau đó các cột được sắp xếp lại và lấy các kí tự theo chiều ngang.

Ví dụ : văn bản TRUONGDAIHOCKYTHUATDANANG được viết thành ma trận 5 x 5 :

Cột	1	2	3	4	5
Văn bản	T	R	U	O	N
	G	D	A	I	H
	O	C	K	I	T
	H	U	A	T	D
	A	N	A	N	G

Vì có 5 cột nên có thể sắp xếp lại theo 5!= 120 cách khác nhau. Nếu ta chuyển vị các cột theo thứ tự 2,3,4,1,5 rồi lấy các kí tự theo hàng ta sẽ có văn bản mã như sau: RUOTN DAIGH CKYOT UATHD NANAG.

Ta thấy rằng, với một văn bản càng lớn (nhiều kí tự) số cách sắp xếp có thể sẽ rất lớn làm tăng khả năng an toàn. Hạn chế của phương pháp này là toàn bộ ma trận kí tự phải được sinh để mã hoá và giải mã và cũng dễ nhầm lẫn trong việc giải mã.

4. Hoán vị các kí tự của văn bản gốc theo chu kì cố định T

Cho hàm f là hoán vị của một khối gồm T kí tự thì khoá mã hoá được biểu diễn bởi hàm K(T,f). Do vậy, văn bản gốc :

$M = m_1\ m_2\ md\ md_{+1} \ldots m_{2d}$

Trong đó m_i là các kí tự riêng lẻ sẽ được mã hoá thành :

$Ek(M) = mf_{(1)}\ mf_{(2)} \ldots mf_{(d)}\ md_{+f(1)} \ldots md_{+f(d)}$

Với $mf_{(1)}\ mf_{(2)} \ldots mf_{(d)}$ là một hoán vị của $m_1\ m_2 \ldots md$

Ví dụ : giả sử T=7 và f hoán vị dãy i = 12345 thành f(i)=23415, chẳng hạn từ gốc STUDY được biểu diễn như sau :

Vị trí đầu	*Vị trí hoán vị*	*từ*	*Mã hoá*
1	2	S	T
2	3	T	U
3	4	U	D
4	1	D	S
5	5	Y	Y

Bằng cách đó văn bản gốc TRUONGDAIHOCKYTHUATDANANG được mã hoá thành RUOTN DAIGH CKYOT UATHD NANAG

10.2.2 Phương pháp thay thế

Phương pháp này mã hoá văn bản bằng cách thay thế mỗi kí tự trongvăn bản bằng một kí tự khác nào đó (có thể là chữ cái, chữ số hoặc kí hiệu), có thể dùng một trong các phương pháp thay thế sau :

1. Thay thế đơn giản

Mỗi kí tự trong văn bản gốc được thay thế bằng một kí tự tương ứng trong văn bản mật mã. Một ánh xạ 1 – 1 được dùng để mã hoá và giải mã thông điệp.

2. Thay thế đồng âm

Mỗi kí tự trong văn bản gốc được mã hoá với một số kí tự của văn bản mật mã (ánh xạ 1 - n). Ngoài ra còn một số phương pháp thay thế khác như thay thế đa mẫu tự, thay thế theo sơ đồ...

Một trong những mật mã thay thế đơn giản được biết đến nhiều nhất là mã Morse, trong đó các chữ cái được thay thế bằng các kí tự gạch và chấm. Bảng mã ASCII ta thường dùng cũng là một dạng mật mã thay thế đơn giản. Trong đó, chữ A

được biểu diễn bằng chuỗi số nhị phân 1000001 hoặc số thập phân 65, chữ B được biểu diễn bởi 1000010 hoặc 66,v.v...

Một dạng mật mã khác cũng được biết đến nhiều đó là bảng chữ cái dịch chuyển. ở đây, các chữ cái trong bảng được dịch chuyển sang phải k vị trí, k gọi là khoá. Ta có hàm dịch chuyển : f(a) = (a + k) mod n với a là một chữ cái trong bảng mã, n là số chữ cái (n=26 nếu là bảng chữ tiếng Anh chuẩn).

Ví dụ : với k = 5 ta có :

Bảng chữ gốc : A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Bảng chữ mật mã : F G H I J K LMN O P Q R S T U V W X Y Z A B C D E

Để giải mã, người nhận phải biết khoá k và phục hồi lại văn bản gốc bằng cách biến đổi ngược.

Như vậy nếu văn bản gốc là TRUONG DAI HOC KY THUAT DA NANG thì văn bản mã sẽ là YWZTSL IFN MTH PN YMZFY IF SFSL. Phương pháp này kém an toàn vì chỉ cần thử lần lượt 26 khả năng là ta đã giải mã được.

Một phương pháp thay thế khác tốt hơn là dùng từ khoá theo sơ đồ Vignère. Theo sơ đồ này, từ khoá được cộng vào liên tiếp theo từng kí tự một cho văn bản gốc, mỗi kí tự được biểu diễn bởi một vị trí của nó trong bảng kí tự và phép cộng được thực hiện theo môđun 26. Ví dụ, giả sử ta có bằng :

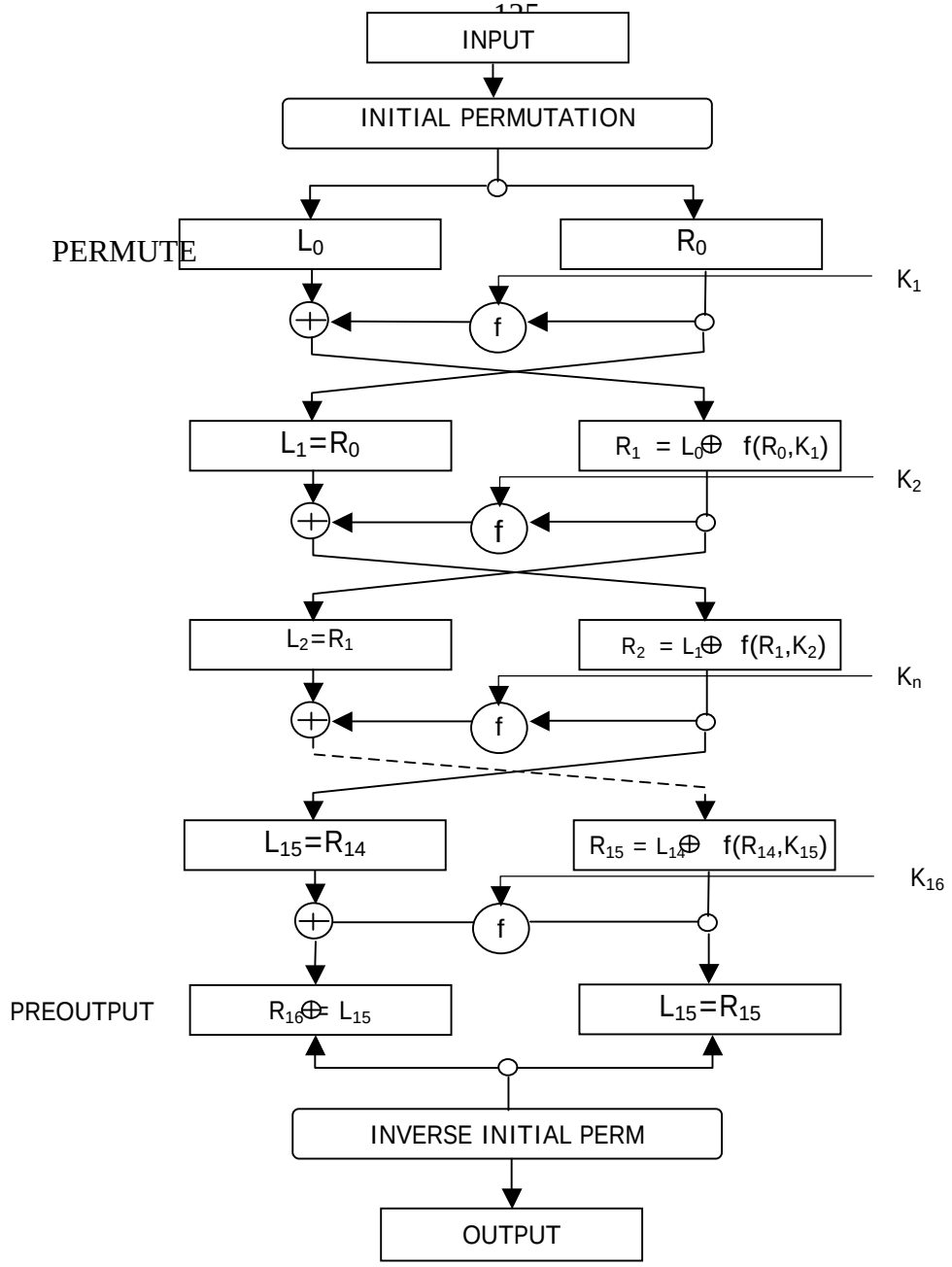
Vị trí	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Ký tự	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S

19	20	21	22	23	24	25
T	U	V	W	X	Y	Z

Cho từ khoá là ABCDEF, văn bản DAIHOC được mã hoá thành DBKKSH. Lưu ý rằng ở trên các từ được viết tách ra cho dễ đọc, trên thực tế dữ liệu được truyền thành từng dòng liên tục để sau đó bí mật tính chu kì.

10.2.3 Phương pháp mã hóa chuẩn DES

Những thuật toán hiện đại hơn dùng kết hợp cả hai kỹ thuật thay thế và đổi chỗ. Trong đó phương pháp được biết đến nhiều nhất là chuẩn mã hoá dữ liệu DES (Data Encryption Standard) được Mỹ và hãng IBM phát triển trong những năm 70. Lưu đồ mã hoá DES được mô tả như hình sau.

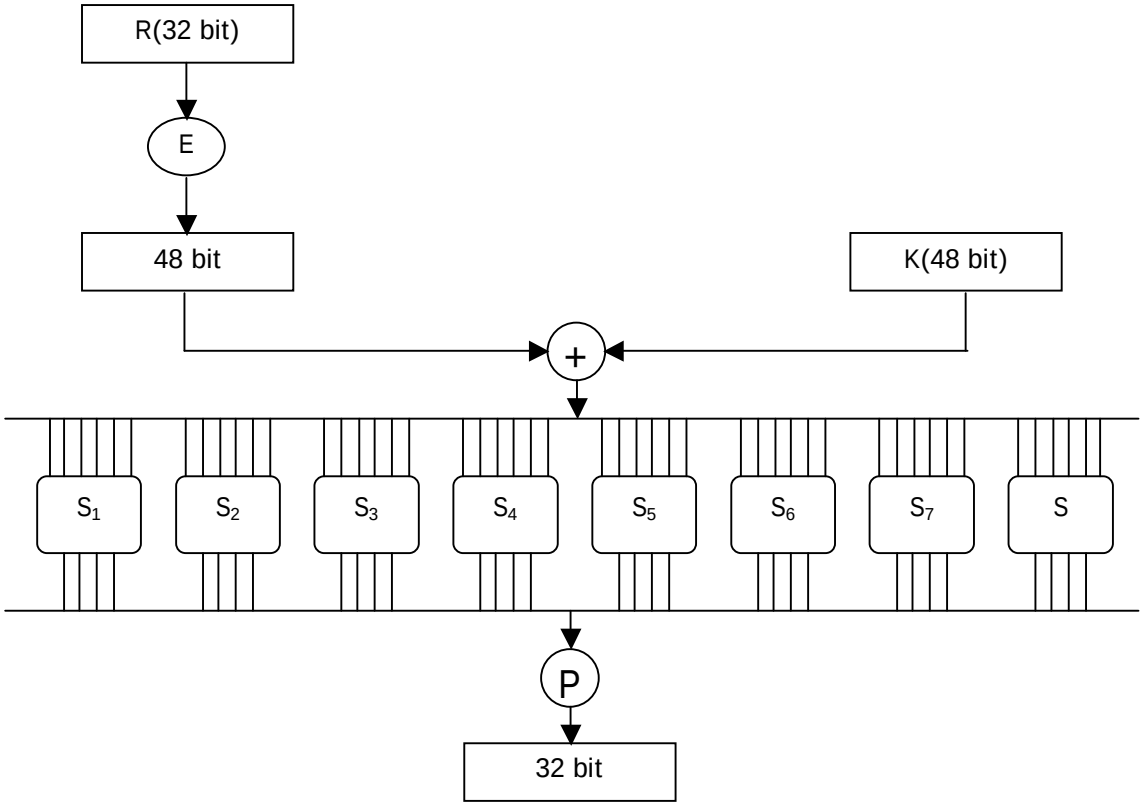


Hình 10-3. Sơ đồ mã hoá DES.

Đầu vào là một dãy 64 bit biểu diễn một khối các kí tự trong văn bản gốc và đầu ra là một dãy 64 bit biểu diễn văn bản mã. Quá trình mã hoá được chia làm 3 giai đoạn :

Đầu tiên văn bản gốc được chuyển qua bộ hoán vị khởi đầu (initial permutation-IP) để tạo ra 64 bit đã hoán vị . Sau đó thực hiện 16 phép lặp của một hàm chữ số (cipher function), kí hiệu là f(R,K) là tổ hợp cả kĩ thuật hoán vị lẫn kĩ thuật thay thế. Trong đó R là dãy con phải (32 bit) của văn bản gốc, khoá K có độ dài 56 bit. 64 bit đầu ra được làm đầu vào cho hoán vị ngược với hoán vị khởi đầu IP^{-1} để tạo ra 64 bit văn bản gốc.

Chi tiết của hàm f(R,K) được mô tả như sau :



Hình 10-4. Hàm f(R,K).

Phép toán của f(R,K) :

Giả sử, bit đầu tiên trong kết quả hoán vị là bit 58 trong dãy ban đầu, bit thứ 2 trong kết quả là bit thứ 50 trong dãy ban đầu,v.v... Dãy hoán vị được chia làm 2 dãy con 32 bit : dãy con trái, kí hiệu là L_0 trong sơ đồ, và dãy con phải kí hiệu là R_0. Hàm f(R,K) dùng các phép toán thay thế và một khoá K_1 để chuyển R_0 thành một dãy 32 bit mới, kí hiệu $f(R_0,K_1)$. Dãy bit này được cộng vào L_0 từng bit một theo mođun 2(phép toán cộng loại trừ) để tạo ra dãy con phải ở giai đoạn tiếp theo. Dãy R_0 ban đầu trở thành dãy con trái L_1.

Phép hoán vị ban đầu IP được cho như bảng dưới đây :

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Chuỗi các phép toán được thực hiện 16 lần với 16 khoá khác nhau K_1, K_2,..., K_{16}, ngoại trừ một điều là không có "phép chuyển qua" ở giai đoạn cuối cùng. Những phép toán này tạo ra dãy 64 bit $R_{16}L_{16}$, được đánh dấu PREOUTPUT trong sơ đồ. Phép toán ngược IP^{-1} của phép hoán vị IP được dùng để biến đổi dãy PREOUTPUT để tạo ra bản mã cuối cùng.

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Dãy con phải được kí hiệu bởi R trước hết được mở rộng thành một dãy số 48 bit dùng bảng chọn bit E sau đây :

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Như vậy, khối 6 bit đầu tiên gồm các bit 32,1,2,3,4,5 của R; khối thứ hai gồm các bit 4,5,6,7,8,9, ... Sau đó một phép toán thay thế được áp dụng cho dãy 48 bit này bằng cách cộng nó (theo phép cộng loại trừ) với khoá 48 bit. Một phép thay thế khác được sử dụng cho các khối 6 bit để tạo ra các khối 4 bit để kết quả cuối cùng là dãy 32 bit. Ví dụ bảng thay thế cho S_1 là :

S_1																
Số hàng	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	5	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	12	11	15	12	9	7	3	10	5	0
3	15	12	8	2	14	9	1	7	5	11	3	14	10	0	6	13

Để minh hoạ cách sử dụng, giả sử rằng khối 6 bit đầu tiên là 101000. Số nhị phân 10 tạo bởi bit đầu tiên và bit cuối cùng xác định một hàng trong bảng, cụ thể là hàng 2, 4 bit giữa 0100 xác định cột trong bảng, cụ thể là cột 4. Biểu diễn nhị phân 4 bit 1101 của phần tử 13 ở hàng 2 cột 4 trong bảng là giá trị thay thế cho 6 bit này. các phép toán tương tự S_2, S_3,..., S_8 được dùng để chuyển đổi cho các khối 6 bit khác.

Phép hoán vị cuối cùng P được áp dụng cho dãy 32 bit để tạo ra f(R,K):

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Mười sáu khoá khác nhau dùng trong DES được lấy ra theo một qui định chặt chẽ từ một khoá 64 bit duy nhất. Như vậy người dùng chỉ cần giữ một khoá để mã hoá và giải mã hơn là giữ 16 khoá khác nhau. Thuật toán giải mã cũng tương tự như khi mã hoá, chỉ khác một diều là 16 khoá được dùng theo thứ tự ngược lại.

Việc giải mã được thực hiện ngược lại với 64 bit văn bản mã làm đầu vào cho hoán vị ngược với hoán vị khởi đầu IP^{-1} để tạo ra 64 bit văn bản gốc.

Phương pháp DES được Uỷ ban tiêu chuẩn quốc gia (National Bureau of Standards) Hoa Kỳ đề nghị như là một sơ đồ mã hoá “chuẩn “. Tuy nhiên, người ta còn đang tranh luận liệu khoá 48 bit có đủ dài hay chưa và các phép toán thay thế có đủ độ bảo mật cần thiết hay chưa.

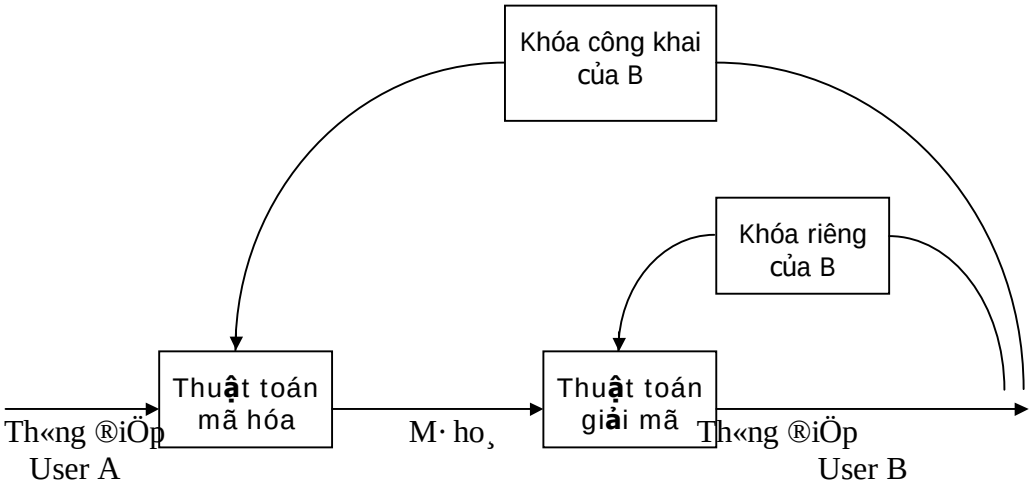
10.2.4 Phương pháp mã hoá khoá công khai

10.2.4.1 Nguyên lí mã hóa công khai

Trong khi thuật toán mã hoá cổ điển dùng một khoá chung cho mã hoá và giải mã thì phương pháp mã hoá bằng khoá công khai sử dụng hai khoá có quan hệ với nhau trong thuật toán để ứng dụng trong mã hoá/giải mã. Các thuật toán này có đặc trưng quan trọng là khó có thể tính toán bằng máy để tìm ra được khoá giải mã nếu chỉ biết được khoá mã hoá và phương pháp mã hoá.

Một số các thuật toán mã hóa công khai (như RSA chẳng hạn) còn có một đặc trưng nữa là khả năng hoán đổi vai trò giữa cặp khoá. Có nghĩa là khi khoá này dùng để mã hoá thì khoá kia dùng để giải mã và ngược lại.

Hình sau mô tả nguyên lí quá trình mã hoá/giải mã bằng khoá công khai :



Hình 10-5. Quá trình mã hoá/giải mã bằng khoá công khai.

Quá trình mã hoá/giải mã như sau:

b. Mỗi hệ thống cuối trong một mạng tạo ra một cặp khoá dùng để mã hoá và giải mã thông tin khi nhận được chúng.

c. Mỗi hệ thống phải có 2 khóa, khóa công khai và khóa bí mật, khoá công khai được công bố lên mạng tại nơi cho phép đăng kí công cộng hoặc đưa vào file. Khoá còn lại phải được giữ bí mật tuyệt đối.

d. Nếu A muốn gửi thông điệp cho B, A sẽ dùng khoá công khai của B trên mạng để mã hoá nó rồi gởi.

e. Khi B nhận được thông điệp của A, B sẽ dùng khoá riêng của mình để giải mã thông điệp nhận được. Không ai có thể giải mã thông điệp được vì chỉ có một mình B biết khoá giải mã.

Thông tin về khoá phải được giữ an toàn tuyệt đối và có thể cập nhật hoặc thay đổi lại khoá cũ. Việc tạo ra các hệ thống bảo vệ và quản lí khoá cũng cần hết sức chặt chẽ.

10.2.4.2 Phương pháp mã hóa RSA

Bản thuyết trình đầu tiên của Diffe và Hellman đưa ra năm 1976 tại hội nghị MIT và gần như ngay lập tức, sự thách thức về vấn đề mã hoá đã tìm được câu trả lời bởi hệ thống mã hoá công khai. Một trong những câu trả lời đầu tiên đưa ra vào năm 1977 bởi Ron Rivest, Adi Shamir và Len Adltôian được công bố vào năm 1978

(gọi tắt là rsa). ý tưởng RSA trở thành gần như độc tôn và được sử dụng rộng rãi trong phương pháp mã hoá bằng khoá công khai.

Giả sử ta có :

Văn bản gốc : $M = M_1 M_2 ... Mk$

Văn bản mã hóa : $C = C_1 C_2 ... Ck$, trong đó $Ci = M_i^E \bmod n$, n là tích 2 số nguyên tố bất kì p và q.

Thuật toán RSA dùng thuyết số để phát triển phương pháp phát sinh một cặp các số nguyên tố - các khoá, thuật toán dựa trên nhận xét: *Có thể dễ dàng sinh ra 2 số nguyên tố lớn và khi nhân chúng với nhau thì rất khó khi muốn phân tích tích của chúng thành thừa số và khó có thể tìm được số còn lại từ số kia*.

Theo một hệ quả của định lí Euler đưa ra: *Cho 2 số nguyên tố p và q và hai số nguyên n và m để n=p.q và 0<m<n, tồn tại một số nguyên duy nhất k sao cho:*

$$(mk^{\phi(n)+1} = mk^{(p-1)(q-1)+1}) \bmod m = n$$

trong đó $\phi(n)$ là hàm Euler với giá trị số nhỏ hơn n và có quan hệ nguyên tố với n, $\phi(n)=(p-1)(q-1)$.

Do đó ta có thể đạt được kết quả mong muốn nếu: $ED = k\phi(n) + 1$

Điều này tương đương với: $ED \bmod \phi(n) = 1$.

Thuật toán RSA được mô tả như sau:

1. Chọn 2 số nguyên tố p, q.
2. Tính tích n = p*q
3. Tính $\phi(n) = (p-1)(q-1)$
4. Chọn E thỏa $USCLN(\phi(n), E) = 1$; *với* $1 < E < \phi(n)$
5. Tìm D thỏa $DE \bmod \phi(n) = 1$.

Khoá công khai là KE = {E,n}, khoá riêng là KD = {D,n} hoặc ngược lại.

Giả sử rằng user A công bố khoá công khai KE lên mạng và user B muốn gửi thông điệp cho user A :

- B sẽ dùng khóa công khai của user A để mã hoá thông điệp của mình bằng công thức $C = ME \bmod n$, rồi gởi nó đi.
- User A sẽ nhận được thông điệp đã mã hoá và giải mã nó bằng khoá riêng của mình bằng công thức $M = CD \bmod n$

Ví dụ: Chọn p =7, q =17

Tính n = p*q = 7*17 = 119

$\phi(n) = (p-1)*(q-1) = 96$

Chọn E thỏa : USCLN(E, 96) = 1. Ta chọn E = 5.

Tìm D thỏa : D*E mod 96 = 1 và D < 96 , suy ra D =77.

Ta được KE = {5,119} , KD = {77,119}.

Giả sử M = 19. Quá trình mã hoá: $C = 19^5 \bmod 119 = 66$.

Quá trình giải mã: $M = 66^{77} \bmod 119 = 19$.

10.2.4.3 Các vấn đề nảy sinh trong thuật toán

1. Vấn đề phức tạp trong tính toán.

Trong quá trình mã hoá và giải mã, thuật toán RSA phát sinh ra các số nguyên rất lớn, cho dù có phép chia modulo n. Rivest, Shamir và Adltôian đề nghị rằng các số p và q phải có độ dài trên 100 chữ số để đảm bảo an toàn gần như tuyệt đối. Như vậy sự luỹ thừa quá lớn và sau đó cho dù có chia modulo n thì kết quả trung gian cũng sẽ khổng lồ và rất dễ dẫn đến tràn số. Ta có thể ứng dụng tính chất của phép chia modulo sau:

((a mod n) * (b mod n)) mod n = (a * b) mod n

Do đó, chúng ta có thể làm giảm kết quả trung gian trong phép chia này đi. Điều này làm cho các phép toán trở nên khả thi hơn.

2. Vấn đề bẻ khoá

Với thuật toán thay thế và hoán vị, về mặt lí thuyết khi độ dài của khoá càng lớn thì mức độ an toàn càng cao, nhưng những người giải mã giàu kinh nghiệm vẫn có thể phân tích tần số xuất hiện của một số kí tự xác định hay tổ hợp của chúng để từ đó suy ra khoá và thực hiện giải mã. Trong thuật toán RSA khoá KE(E,n) là khoá công khai nên ta không cần giữ bí mật, ta chỉ giữ bí mật cho khoá riêng KD(D,n). Vì vậy, để bẻ khóa phải xác định được D từ các giá trị E và n. Theo như cách chọn các số E và D, điều này có thể làm được nếu có thể phân tích n thành tích của hai số nguyên tố. Như vậy tính an toàn của thuật toán RSA phụ thuộc vào sự khó khăn của việc xác định các thừa số nguyên tố của một số nguyên tố lớn. Hiện nay nếu sử dụng thuật toán phân tích thừa số nhanh nhất của Schroeppel thì cũng cần đến : $S = \exp[(\ln n)\ln(\ln n)]^{1/2}$ bước tính toán để phân tích n thành p và q.

Bảng dưới đây hiển thị các thời gian dự đoán của các nhà phân tích, giả sử rằng mỗi phép toán được thực hiện trong 1 micro giây :

Độ dài của khóa	Thời gian
50	4 giờ

75	104 ngày
100	74 năm
200	4.000.000 năm
300	5×10^{15} năm
500	4×10^{25} năm

Phương pháp mã hoá với khoá công khai xem như được bảo đảm vì hiện nay vẫn chưa tìm ra một thuật toán phân tích thừa số nguyên tố có hiệu quả.

10.2.4.4 ứng dụng của mã hoá dữ liệu

Mã hoá dữ liệu có các ưu điểm là an toàn vì ít phụ thuộc vào cấu trúc hệ thống mạng. Ngoài ra mã hoá dữ liệu có tính bảo mật do dữ liệu được mã hoá rồi thì chỉ có những người có quyền mới có thể giải mã để nhận lại được dữ liệu ban đầu. Các phương pháp mã hoá trên có thể áp dụng trong những tình huống sau :

- Phương pháp mã hoá thay thế kết hợp với phương pháp mã hoá hoán vị dùng tạo ra phương pháp mã hoá DES.
- Các dịch vụ e-mail trên mạng Internet hay các mạng cục bộ có thể sử dụng thuật toán RSA để tạo ra một mặt nạ nhận dạng (authentication mask) các thông điệp giữa các cá nhân với nhau. Có nghĩa là chỉ những người nhận được thư gởi cho mình bằng khoá mã hoá của mình thì mới giải mã được thông điệp đó và hoàn toàn không thể (nói theo nguyên tắc) đọc được các thư không phải gởi cho mình.
- Kỹ thuật mã hoá chữ kí số (digital signature) có thể dùng để tạo ra một chữ kí mã hoá dùng để xác định, nhận dạng một đối tượng trong các dịch vụ thương mại, ví dụ như các thẻ tín dụng hoặc các loại visa, cardphone chẳng hạn....
- Thư điện tử e-mail cũng có thể kết hợp thuật toán này với các thuật toán mã hoá khác như DES theo mô hình có thể là:

- Nội dung thư được mã hoá bằng phương pháp DES
- Tạo một chữ ký số và mã hoá bằng khoá RSA
- Khoá DES dùng để giải mã có thể được mã hoá bằng RSA và gửi kèm trong thư luôn mà không cần phải bí mật. Người nhận sẽ dùng khóa riêng của mình để giải mã khoá DES, sau đó giải mã thư nhận được.

10.3 Cơ chế bảo vệ bằng firewall

Vấn đề quan trọng trong việc quản lý các tài nguyên thông tin là cơ chế bảo vệ chống việc truy cập bất hợp pháp trong khi vẫn cho phép người được ủy nhiệm sử dụng những nguồn thông tin mà họ được cấp quyền, và phương pháp chống thất

thoát thông tin được truyền tải trên các mạng truyền dữ liệu công cộng (Public Data Communication Network). Đó chính là yêu cầu của một giải pháp hoặc hệ thống an ninh cho hệ thống mạng hay còn gọi là hệ thống an ninh dữ liệu (Data Security System).

Nhu cầu an ninh hệ thống ngày càng trở nên quan trọng vì nhiều nguyên nhân như các đối thủ luôn tìm cách để nắm được mọi thông tin liên quan, ngày càng nhiều hacker truy cập thông tin từ các mạng nội bộ theo nhiều mục đích khác nhau.

Một giải pháp an ninh cho hệ thống mạng được ứng dụng nhiều đó là bức tường lửa (firewall). Thuật ngữ firewall có nguồn gốc từ một kỹ thuật thiết kế trong xây dựng để ngăn chặn, hạn chế hỏa hoạn. Trong công nghệ mạng thông tin, firewall là một kỹ thuật được tích hợp vào hệ thống mạng để chống lại việc truy cập trái phép nhằm bảo vệ các nguồn thông tin nội bộ cũng như hạn chế sự xâm nhập vào hệ thống của một số thông tin khác không mong muốn.

Về mặt chức năng hệ thống, firewall là một thành phần được đặt giữa hai mạng để kiểm soát tất cả các việc lưu thông và truy cập giữa chúng với nhau, bao gồm:

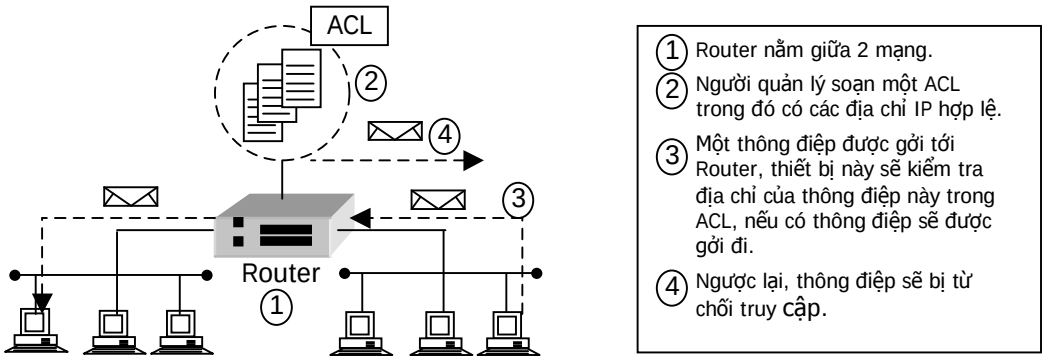
1. Tất cả các trao đổi dữ liệu từ trong ra ngoài và ngược lại phải thực hiện thông qua firewall.
2. Chỉ có những trao đổi nào được phép bởi chế độ an ninh của hệ thống mạng nội bộ (trusted network) mới được quyền lưu thông qua firewall.

Về mặt vật lý, firewall bao gồm:

1. Một hoặc nhiều hệ thống máy chủ kết nối với các bộ định tuyến (router) hoặc có chức năng router.
2. Các phần mềm quản lý an ninh chạy trên các hệ thống máy chủ. Thông thường là các hệ quản trị xác thực (Authentication), cấp quyền (Authorization) và kế toán (Accounting).

Firewall bao gồm phần cứng và/hoặc phần mềm nằm giữa 2 mạng (như mạng nội bộ và mạng Internet), bảo vệ mạng nội bộ bằng cách cấm các người sử dụng truy cập trái phép đến và đồng thời ngăn chặn những thông điệp không được phép gửi đi cho người nhận bên ngoài mạng. Firewall có thể nằm trên bộ dẫn đường hay trên Server. Cơ chế làm việc của Firewall dựa trên việc kiểm tra các gói dữ liệu IP lưu chuyển giữa hai mạng tùy thuộc vào các qui tắc mà người quản trị hệ thống đã xác lập.

Khái quát phương thức làm việc của Firewall như trong hình vẽ sau:



Hình 10-6. Cơ chế hoạt động của Firewall.

10.3.1 Các loại firewall và cơ chế hoạt động

Khi nói đến việc lưu thông dữ liệu giữa các mạng với nhau thông qua firewall thì điều đó có nghĩa rằng firewall hoạt động kết hợp chặt chẽ với giao thức TCP/IP. Vì giao thức này làm việc theo thuật toán chia nhỏ các dữ liệu nhận được từ các ứng dụng trên mạng, hay chính xác hơn là các dịch vụ chạy trên các giao thức (Telnet, SMTP, DSN, SMNP, NFS,...) thành các gói dữ liệu rồi gán cho các gói này những địa chỉ để có thể nhận dạng tái lập lại ở đích cần gửi đến. Do đó các loại firewall cũng liên quan rất nhiều đến các packet và các địa chỉ của chúng.

10.3.1.1 Bộ lọc packet (Packet filtering)

Loại firewall này thực hiện việc kiểm tra số nhận dạng địa chỉ của các packet để cho phép chúng có thể lưu thông qua lại hay không. Các thông số có thể lọc được của một packet như sau:

1. Địa chỉ IP nơi xuất phát (source IP address).
2. Địa chỉ IP nơi nhận (destination IP address).
3. Cổng TCP nơi xuất phát (TCP source port).
4. Cổng TCP nơi nhận (TCP destination port).

Nhờ đó firewall có thể ngăn cản được các kết nối vào những máy chủ hoặc mạng nào đó được xác định, hoặc khóa việc truy cập vào hệ thống nội bộ từ những địa chỉ không cho phép.

Hơn nữa việc kiểm soát các cổng làm cho firewall có khả năng chỉ cho phép một số loại kết nối nhất định vào máy chủ nào đó, hoặc chỉ có những dịch vụ nào đó (Telnet, SMTP, FTP,...) được phép mới chạy được trên hệ thống mạng nội bộ.

10.3.1.2 Cổng ứng dụng (Application gateway)

Đây là một loại firewall được thiết kế để tăng cường chức năng kiểm soát các loại dịch vụ, giao thức được cho phép truy cập vào hệ thống mạng. Cơ chế hoạt

động của nó dựa trên cách thức gọi là Proxy Service (dịch vụ đại diện): một ứng dụng nào đó được quy chiếu đến (hay đại diện bởi) một Proxy Service trong khi các Proxy Service chạy trên các hệ thống máy chủ thì được quy chiếu đến application gateway của firewall. Cơ chế lọc của packet filtering phối hợp kiểm soát với cơ chế "đại diện" của application gateway cung cấp một khả năng an toàn và uyển chuyển hơn.

Ví dụ một hệ thống mạng có chức năng lọc các gói tin ngăn các kết nối bằng Telnet vào hệ thống chỉ trừ một chủ duy nhất -Telnet application gateway là được phép. Một người sử dụng dịch vụ Telnet muốn kết nối vào hệ thống phải thực hiện các bước sau :

1. Thực hiện dịch vụ TELNET đến Telnet application gateway rồi cho biết tên của máy chủ bên trong cần truy cập.
2. Gateway kiểm tra địa chỉ IP nơi xuất phát của người truy cập rồi cho phép hoặc từ chối tùy theo chế độ an ninh của hệ thống.
3. Người truy cập phải vượt qua được hệ thống kiểm tra xác thực.
4. Proxy Service tạo một kết nối Telnet giữa gateway và máy chủ cần truy cập.
5. Proxy Service liên kết lưu thông giữa người truy cập và máy chủ.

Cơ chế hoạt động này có ý nghĩa quan trọng trong việc thiết kế an ninh hệ thống ví dụ như:

1. Che giấu các thông tin: người dùng chỉ có thể nhìn thấy trực tiếp các gateway được phép.
2. Tăng cường kiểm tra truy cập bằng các dịch vụ xác thực (Authentication).
3. Giảm đáng kể giá thành cho việc phát triển các hệ quản trị xác thực vì các hệ thống này được thiết kế chỉ quy chiếu đến application gateway.
4. Giảm thiểu các quy tắc kiểm soát của bộ lọc (Packet filtering). Điều này làm tăng tốc độ hoạt động của firewall.

10.3.1.3 Bộ lọc session thông minh (Smart session filtering)

Cơ chế hoạt động phối hợp giữa bộ lọc packet và cổng ứng dụng như trên cung cấp một chế độ an ninh cao tuy nhiên nó cũng bị vài hạn chế. Vấn đề chính hiện nay là làm sao để cung cấp đủ Proxy Service cho rất nhiều ứng dụng khác nhau đang phát triển ồ ạt. Điều này có nghĩa là nguy cơ, áp lực đối với việc đánh lừa firewall gia tăng lên rất lớn nếu các proxy không kịp đáp ứng.

Trong khi giám sát các packet ở những mức phía trên, nếu như lớp network đòi hỏi nhiều công sức hơn đối với việc lọc các packet đơn giản, thì việc giám sát

các giao dịch lưu thông ở mức mạng (Session) đòi hỏi ít công việc hơn. Cách này cũng loại bỏ được các dịch vụ đặc thù cho từng loại ứng dụng khác nhau.

Nếu kết hợp khả năng ghi nhận thông tin về các session và sử dụng nó để tạo các quy tắc cho bộ lọc thì sẽ có được một bộ lọc thông minh hơn. Đó chính là cơ chế hoạt động của bộ lọc session thông minh.

Vì một session ở mức network được tạo bởi 2 packet lưu thông theo 2 chiều, cho nên nếu thiết kế 2 quy tắc lọc cho 2 chiều này: một để kiểm soát các packet lưu thông từ host phát sinh ra nó đến máy chủ cần tới, một để kiểm soát packet trở về từ máy chủ phát sinh. Một bộ lọc thông minh sẽ nhận biết được rằng packet trở về theo chiều ngược lại nên quy tắc thứ 2 là không cần thiết. Do vậy, cách để tiếp nhận các packet không mong muốn sinh ra từ bên ngoài firewall sẽ khác biệt rất rõ với cách tiếp cận cho các packet do những kết nối được phép (ra bên ngoài). Và như vậy dễ dàng nhận dạng các packet "bất hợp pháp".

10.3.1.4 Firewall hỗn hợp (Hybrid firewall)

Trong thực tế các firewall được sử dụng là sự kết hợp của nhiều kỹ thuật để tạo ra hiệu quả an ninh tối đa. Ví dụ việc để lọt lưới tại các kiểm soát của bộ lọc packet có thể được thực hiện tại bộ lọc session thông minh ở mức ứng dụng. Các giám sát của bộ lọc lại được bọc lót chặt chẽ bởi các dịch vụ proxy của application gateway.

10.3.1.5 Một vài ứng dụng của Firewall

Từ các chế độ hoạt động trên, firewall được ứng dụng nhiều vào hệ thống an ninh dữ liệu. Có 3 yêu cầu chính cho vấn đề an ninh hệ thống theo tiêu chuẩn ISO cho mô hình mạng OSI :

- Quản lý xác thực (Authentication)
- Quản lý cấp quyền (Authorization)
- Quản lý kế toán (Accounting management)
-

f. Ưu điểm của Firewall

Firewall là điểm kiểm tra các kết nối giữa mạng nội bộ và mạng Internet bên ngoài, mọi kết nối đều phải đi qua cửa khẩu này. Đây chính là một bộ lọc an toàn bởi vì có rất nhiều dịch vụ đang hoạt động trên Internet, nếu chúng ta không có một cơ chế kiểm soát chặt chẽ thì các dịch vụ này sẽ tự do mang thông tin tràn vào mạng của chúng ta và ngược lại.

Firewall có thể được sử dụng để ghi nhận lại các hoạt động kết nối với Internet. Bởi vì, mọi hoạt động như vậy đều phải thông qua Firewall nên nó có thể cung cấp thêm chức năng thu thập mọi thông tin về các kết nối xảy ra giữa mạng nội bộ và mạng Internet bên ngoài.

Ta cũng có thể sử dụng Firewall để bảo vệ một máy đơn của người sử dụng.

g. Hạn chế của Firewall

Bên cạnh những mặt tích cực của Firewall kể trên, nó còn có những hạn chế và những việc mà nó không thể thực hiện được như sau:

1. Bên cạnh việc ngăn chặn các người dùng trong mạng nội bộ kết nối ra ngoài khi không được phép thì nó cũng ngăn cản các việc làm tốt của họ.
2. Firewall không thể chống lại các mối nguy hiểm mới, bởi vì chúng nằm ngoài sự kiểm soát của Firewall.
3. Do không kiểm tra trên nội dung của các gói tin, nên Firewall không sử dụng để ngăn ngừa các thông tin xấu trên một dịch vụ đã được cho phép và cũng không thể nhận biết các đoạm mã virus trong các tập tin truyền đi.

10.4 Hệ thống tên miền DNS (Domain Name System)

Địa chỉ Internet 32 bit thỏa mãn yêu cầu kỹ thuật, nhưng phức tạp và khó nhớ đối với người dùng. Giải pháp đưa ra ở đây là dùng những tên gợi nhớ thay cho địa chỉ số là tự nhiên và dễ nhớ đối với người sử dụng. Hơn nữa, dùng tên tin cậy hơn địa chỉ số vì địa chỉ số có thể thay đổi những tên luôn luôn dùng lại được. Do đó nảy sinh vấn đề cách đặt tên và ánh xạ địa chỉ IP với tên.

Trước đây trung tâm thông tin Internet NIC chịu trách nhiệm cấp phát và quản lý tên. Người ta dùng một file có tên host.txt trên Windows hoặc /etc/hosts trên Unix, tập tin này chứa tên của tất cả các mạng, router, host và địa chỉ IP tương ứng với chúng. Các tên được cấp phát không có mối liên hệ gì với nhau. Khi Internet phát triển, giải pháp này trở nên phức tạp không chấp nhận được về mặt quản lý.

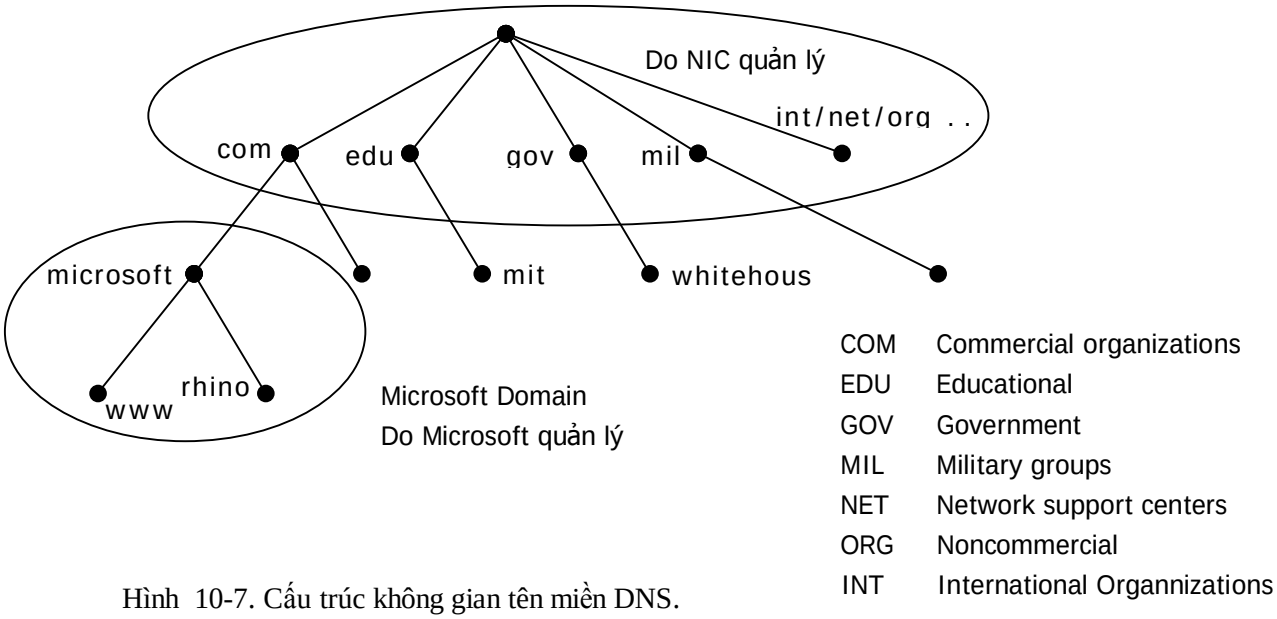
Theo Paul Mockepetris, người thiết kế chính DNS, mục tiêu thiết kế bắt đầu của DNS là để thay thế các tập tin host phức tạp bằng một cơ sở dữ liệu phân tán nhẹ hơn có khả năng cung cấp một *không gian tên thứ bậc, sự quản lý phân tán, có bộ đệm cục bộ (caching), các kiểu dữ liệu mở rộng, kích thước cơ sở dữ liệu không giới hạn và có hiệu năng.*

DNS tương ứng với tầng 7 của mô hình OSI và dùng giao thức UDP hay TCP ở tầng dưới. Việc truy cập DNS thực hiện theo mô hình Client/Server. Hầu hết các hệ thống kết nối Internet đều hỗ trợ DNS. Các đặc tả chính của DNS được định

nghĩa trong các tài liệu RFC 974, 1034, 1035. Dịch vụ cài đặt giao thức DNS phổ biến nhất là BIND (Berkeley Internet Name Domain), được phát triển đầu tiên tại Berkeyley cho hệ điều hành Unix.

DNS gồm 3 thành phần : ***Namespace, các NameServer và Resolver.***

10.4.1 Không gian tên miền DNS



Hình 10-7. Cấu trúc không gian tên miền DNS.

DNS tổ chức không gian tên miền theo cấu trúc cây, trên cùng là gốc, rồi đến các nút cha, nút con... và cuối cùng là các nút lá.

Một máy tính trong mạng sẽ ứng với một nút của cây. Như ở cây trên, máy ở lá www sẽ có địa chỉ hoàn chỉnh là www.microsoft.com. Mỗi nút trên cây biểu diễn một miền (domain) trong hệ thống DNS; mỗi miền lại có một hay nhiều miền con. Tại mỗi miền này đều phải có máy chủ DNS tương ứng quản lý hệ thống tên trong miền đó.

Nút trên cây : Mỗi nút có một tên tương ứng dài từ - đến 63 ký tự dưới 128 trong bảng mã ASCII. Các nút kề nhau không được có cùng tên. Mỗi nút có một tập (có thể rỗng) các *bản ghi tài nguyên* (Resource Record - RR) chứa thông tin đi kèm nút đó. Nhãn rỗng dành riêng cho nút gốc, ký hiệu bằng dấu chấm (.).

Miền con : Được tạo thành từ mỗi nút của không gian tên và các nút bên dưới có thể đi đến được các nút đó.

Vùng : là một phần cây con của cây DNS được quản lý như một thực thể riêng. Vùng có thể bao gồm một miền hay một miền với một số miền con. Các miền con mức thấp hơn của một vùng lại có thể chia thành các vùng rời nhau.

Tên miền của một nút : là dãy các nhãn từ một nút trên cây đến gốc của cây. Các nhãn trong tên miền cách nhau bằng dấu chấm (.). *Tên miền tuyệt đối* kết thúc bằng dấu chấm. Ví dụ "poneria.ISI.EDU.". *Tên miền tương đối* không kết thúc bằng dấu chấm và sẽ được phần mềm cục bộ ghép đầy đủ khi xử lý. Để đơn giản việc cài đặt, độ dài tên miền được giới hạn dưới 255. Một miền là miền con của miền khác nếu tên miền đó chứa tên miền kia. Ví dụ A.B.C.D là miền con của các miền con của các miền B.C.D, C.D, D và miền gốc.

Tên miền đầy đủ là tên các nút từ gốc đến lá của cây nối với nhau và phân cách bằng dấu chấm. Ví dụ : mrp2.widgets.mfg.universal.co.uk

Các miền mức đỉnh : Miền gốc và các miền mức đỉnh của cây DNS do NIC quản lý. Các tên miền mức đỉnh có thể chia ba loại :

- Các miền tổ chức (tên 3 ký tự) : com, edu, gov, . . .
- Các miền địa lý (các mã quốc gia, 2 ký tự) : uk, vn, ca, fr, . . .
- Miền in-addr-arpa : miền đặc biệt dùng để ánh xạ địa chỉ thành tên.

Trách nhiệm quản lý không gian tên DNS dưới mức đỉnh được NIC ủy nhiệm cho các tổ chức khác. Các tổ chức này lại chia không gian tên phía dưới và uỷ nhiệm xuống. Mô hình quản lý phân tán này cho phép DNS được quản lý tự trị bởi các tổ chức tham gia. Cách đặt tên như vậy có tác dụng phân cấp quản lý vùng tên. Các tổ chức có thể tự tạo và quản lý không gian tên riêng của mình trong mạng, không phụ thuộc vào sự cho phép của NIC.

Vấn đề tên và vùng còn được nhiều hãng lớn bổ sung và làm phong phú thêm bằng những giải pháp của riêng họ. Ví dụ Microsoft có WINS - Windows Internet Naming Service, IBM có DDNS - Dynamic Domain Name System.

10.4.1.1 Cú pháp tên miền

Cú pháp cho tên miền sau đây cho phép phù hợp với nhiều ứng dụng như mail, telnet, . . .

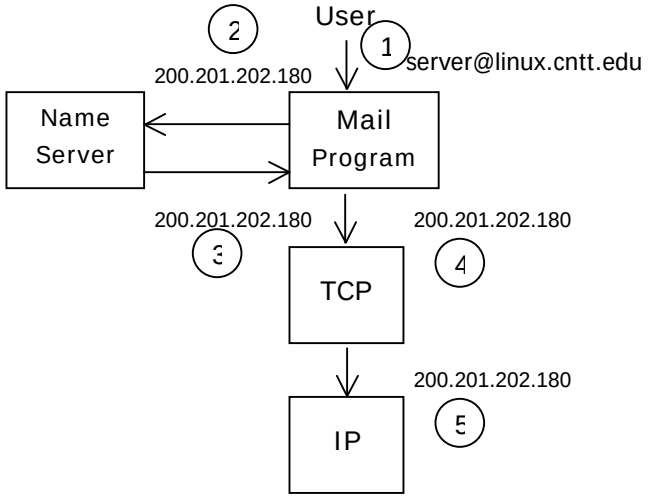
<domain>::=<subdomain>| " "
<subdomain>::=<label>|<subdomain> "." <label>
<label>::=<letter>[[<ldh-str>]<let-dig>]
<ldh-str>::=<let-dig-hyp>|<let-hyp><ldh-str>
<let-dig-hyp>::=<let-dig>| "-"
<let-dig>::=<letter>|<digit>
<letter>::= ký tự từ A-Z, a-z
<digit>::= chữ số 0-9

10.4.2 Máy chủ quản lý tên

Máy chủ quản lý tên (Name Server) là hệ thống chương trình quản lý cấu trúc cây của miền và các tập thông tin đi kèm. Máy chủ tên có thông tin đầy đủ về một số tập con gọi là vùng của không gian tên và các con trỏ đến các nameserver khác để lấy tin về một miền bất kỳ của cây miền. Các máy chủ tên có thông tin đầy đủ về một số phần của cây miền được gọi là có thẩm quyền (authoritative) về các phần đó. Một *vùng* (zone) là một đơn vị thông tin có thẩm quyền của cơ sở dữ liệu DNS. Trong thực tế, các máy chủ tên thường lưu tạm thời trong bộ đệm cấu trúc và thông tin các vùng và thông tin về các vùng khác để tăng hiệu năng. Các máy chủ quản lý tên trong vùng trao đổi thông tin với nhau bằng Zone Transfer Protocol.

10.4.3 Chương trình phân giải tên

Chương trình phân giải tên (Resolver) là các thường trình hệ thống lấy thông tin từ namserver để trả lời yêu cầu của những ứng dụng khách (client). Resolver phải có khả năng truy cập đến ít nhất một nameserver và dùng thông tin từ nameserver đó để trực tiếp trả lời câu hỏi hay để hỏi tiếp đến các nameserver khác. Chương trình người sử dụng có thể truy cập trực tiếp đến resolver, do đó không cần có một giao thức giữa resolver và chương trình người dïng.



Hình 10-8. Quá trình phân giải tên trong thực tế .

10.5 Hệ quản trị mạng

Hệ thống quản trị mạng (Network Management) còn gọi là mô hình Manager/Agent bao gồm các thành phần như sau :

- Hệ quản trị - Manager
- Hệ bị quản trị - Managed system
- Một cơ sở dữ liệu chứa thông tin quản trị và giao thức quản trị mạng.
- Hệ quản trị - Manager

Thực hiện cung cấp giao diện giữa người quản trị mạng và các thiết bị mạng được quản trị, bao gồm các thông tin thể hiện dưới dạng đồ họa, đồ thị, số liệu thống kê, báo cáo. Ví dụ như hiển thị dạng đồ họa bản đồ về topology liên mạng thể hiện các vị trí của các LAN segments, từ đó có thể chọn xem trạng thái hoạt động hiện hành của nó.

10.5.1 Hệ bị quản trị

- Bao gồm tiến trình Agent và các đối thượng quản trị (manager objects).
- Tiến trình Agent thực hiện các thao tác quản trị mạng như đặt các tham số cấu hình và các thống kê hoạt động hiện hành của các router trên một segments cho trước.
- Các đối tượng quản trị bao gồm các trạm làm việc, máy server, hub, kênh truyền.

10.5.2 Cơ sở dữ liệu chứa thông tin quản trị mạng

Được gọi là *cơ sở thông tin quản trị* (Management Information Base - MIB) được lưu trữ tại Server và Client. MIB được tổ chức thành một cấu trúc cây, gọi là SMI (Structure of Management Information). SMI bắt đầu từ gốc root, tiếp theo là các nhánh chứa các đối tượng quản trị được phân loại lôgic.

Kiến trúc quản trị mạng ISO như sau :

1. Quản trị sự cố (Fault Management) : phát hiện, cô lập và khắc phục sự cố.
2. Quản trị kế toán (Accounting Management) : kiểm soát va 2đánh giá việc sử dụng tài nguyên trong mạng
3. Quản trị cấu hình (Configuration Management)
4. Quản trị hiệu năng (Performance Management)
5. Quản trị an toàn (Security Management)

Simple Network Management Protocol (SNMP) được tạo ra ban đầu với mục đích cung cấp phương tiện để điều khiển các router trên mạng. SNMP, mặc dù là một phần trong gia đình giao thức TCP/IP, không phụ thuộc vào IP. SMNP được thiết kế độc lập với giao thức truyền, tuy nhiên phần lớn các hãng đều sản xuất SNMP chạy trên IP.

SNMP thực chất là gồm 3 giao thức cấu tạo thành, tất cả đều được thiết kế để làm việc với mục đích điều hành:

- Management Information Base (MIB): Một cơ sở dữ liệu chứa các thông tin trạng thái.

- Structure and Identification of Management Information (SMI): Một tiêu chuẩn định nghĩa các đầu mục của một MIB.
- Simple Network Management Protocol (SNMP): Phương thức trao đổi thông tin giữa các thiết bị và Server.

10.6 Dịch vụ thư điện tử

Electronic Mail (viết gọn là e-Mail, thư điện tử) là một trong những dịch vụ thông tin phổ biến nhất trên Internet. Dịch vụ e-Mail giúp mọi người có thể trao đổi thông tin với nhau trên mạng Internet. Liên lạc bằng thư điện tử nhanh hơn, thuận tiện hơn và chi phí thấp hơn rất nhiều so với trao đổi thư từ qua đường bưu điện bình thường. Ngoài ra còn cho phép họ gửi cho nhau cả các loại tài liệu như: các văn bản, các báo cáo, các chương trình máy tính, . .. và nhiều thông tin khác nữa.

Mỗi người sử dụng đều có một thư mục lưu trữ thư trên máy Server gọi là Mailbox. Tất cả các địa chỉ mail bao gồm hai phần được ngăn cách nhau bằng 1 ký tự @ (ampersand). Ví dụ : . Tên miền có thể được chia nhiều phần cách nhau bởi dấu chấm (.). Một địa chỉ mail tiêu biểu có các thành phần như sau :

Username @ ServerName. Type of Organization . Country

Cấu trúc của một E-Mail bao gồm các phần như sau :

- ***Phần tiêu đề thư***

Phần này do các MTA (Message Transfer Agent) tạo ra và sử dụng, nó chứa các thông tin để chuyển nhận e-Mail như địa chỉ của nơi nhận, địa chỉ của nơi gửi. Các hệ thống e-Mail cần những thông tin này để chuyển dữ liệu từ máy tính này sang máy tính khác. Cấu tạo phần này gồm nhiều trường (field), mỗi trường là một dòng văn bản ASCII chuẩn 7 bit như sau: <tên trường >: <nội dung của trường>.

Sau đây là một số trường thông tin thông dông:

Trường	Chức năng
DATE	Chỉ ngày giờ nhận mail.
FROM	Chỉ địa chỉ người gởi.
TO	Chỉ địa chỉ người nhận.
CC	Chỉ địa chỉ những người nhận bản copy của mail. Các người nhận thấy được địa chỉ của những người cùng nhận trong nhóm.
BCC	Chỉ địa chỉ những người nhận bản sao chép của bức mail, nhưng từng người không biết những người nào sẽ nhận bức thư này.
REPLY-TO	Chứa các thông tin để người nhận có thể trả lời lại, thường nó chính là địa chỉ người gởi.
MESSAGE-ID	Định danh duy nhất, được sử dụng bởi hệ điều hành.
SUBJECT	Chủ đề của nội dung thư.

Các trường trên là các trường chuẩn do giao thức SMTP quy định, ngoài ra trong phần header cũng có thể có thêm một số trường khác do chương trình e-Mail tạo ra nhằm quản lý các e-Mail riêng. Các trường này được bắt đầu bằng ký tự X- và thông tin theo sau là cũng giống như ta thấy trên một trường chuẩn.

- ***Phần nội dung***

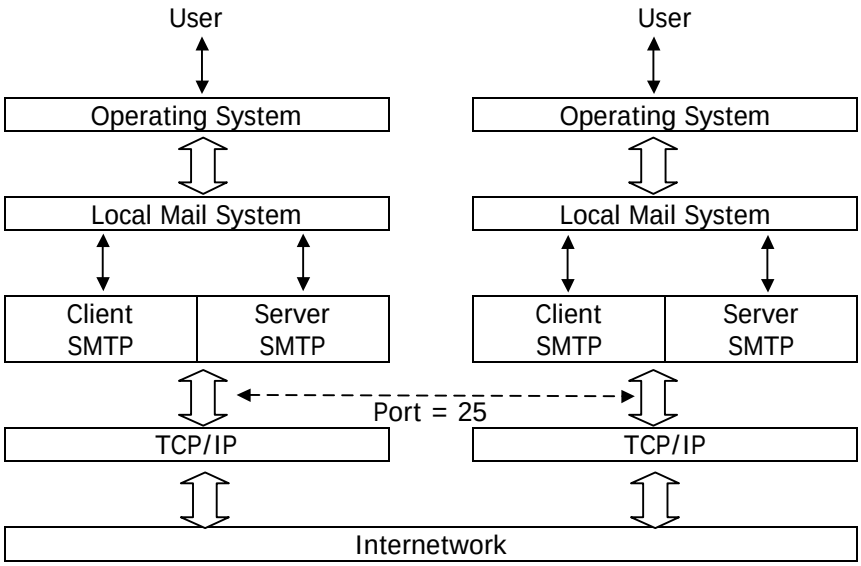
Để phân biệt phần tiêu đề và phần nội dung của e-Mail, người ta qui ước đặt ranh giới là một dòng trắng (chuỗi ký tự "\r\n"). Kết thúc của phần nội dung là chuỗi ký tự "\r\n.\r\n".

Như vậy nội dung bức thư nằm trong khoảng giữa dòng trắng đầu tiên và ký tự kết thúc thư, và trong phần nội dung của bức thư không được phép tồn tại chuỗi ký tự kết thúc thư. Mặt khác do môi trường truyền thông là mạng Internet nên các ký tự cấu thành phần thân của bức thư phải là các ký tự ASCII chuẩn.

10.6.1 Giao thức SMTP

SMTP (Simple Mail Transfer Protocol) là giao thức qui định việc truyền mail chủ yếu dùng trong mạng Internet.

Mối quan hệ giữa SMTP và hệ thống Mail cục bộ như sau:



Hình 10-9. Quan hệ giữa SMTP và hệ thống Mail cục bộ.

Client liên quan đến thư đi, Server liên quan đến nhận thư. Hệ thống thư cục bộ hộp thư (mailbox) cho mỗi user. Mail box có 2 phần: phần cục bộ và phần toàn cục.

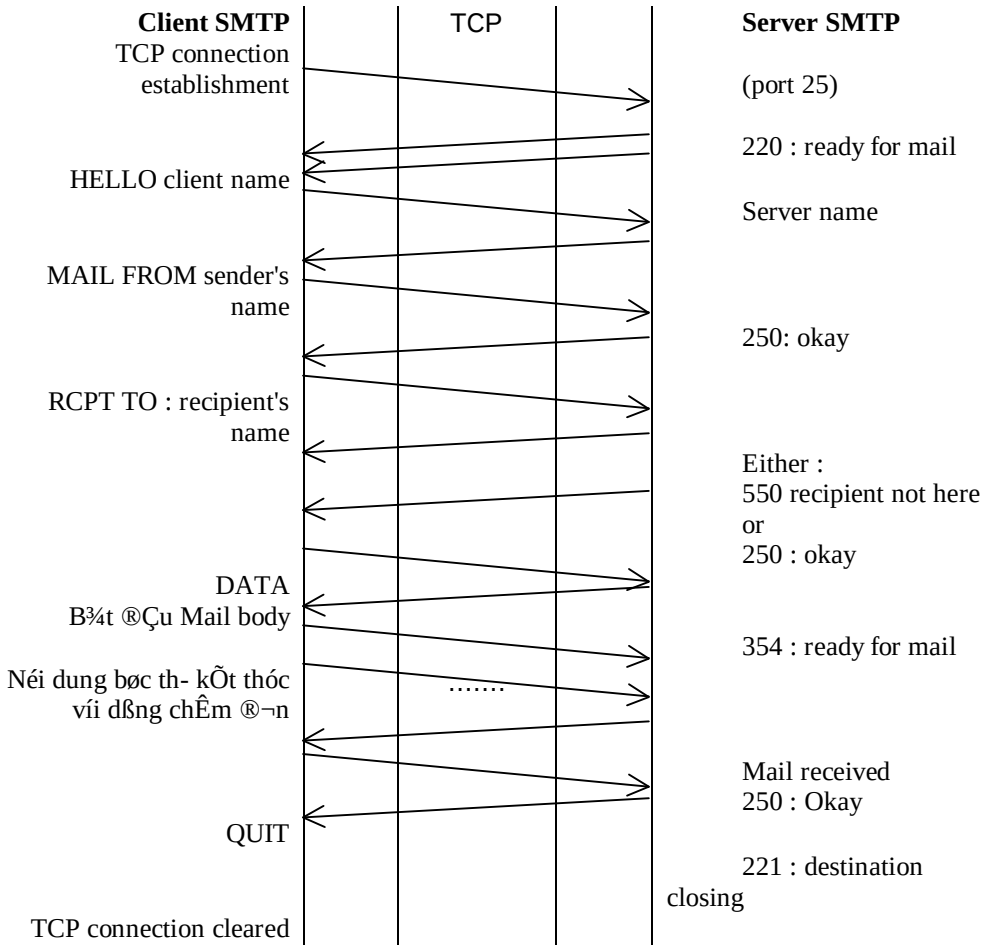
Sau khi tháo bức thư trong khuôn dạng chuẩn, hệ thống mail cục bộ xác định tên người nhận ở hộp thư cục bộ hay phải gửi ra ngoài. để gửi bức thư Client SMTP

phải biết địa chỉ IP của nơi nhận qua DNS và gửi qua cổng địa chỉ SMTP (25) để bắt đầu thiết lập kết nối server SMTP nơi nhận. Khi mối nối đã được thiết lập, Client bắt đầu chuyển bức thư đến Server bởi các lệnh của SMTP. SMTP dùng từ khóa như các lệnh để thực hiện thao tác chuyển giao mail. Một số lệnh chính của SMTP trong phiên làm việc giữa Client MTA và Server MTA như sau :

Lệnh	Tác dụng
HELLO	Xưng danh với SMTP bên nhận, báo cho bên nhận biết bên gởi là ai. SMTP bên gởi gửi lệnh này đầu tiên cho SMTP bên nhận.
MAIL	Khởi động một cuộc giao dịch mail mà mục đích cuối cùng là chuyển giao các mail tới một hay nhiều Mailbox (nơi chứa Mail nhận được) khác nhau.
RCPT	Nói rõ người nhận mail là ai.
DATA	Các dòng sau lệnh DATA là dữ liệu của Mail. Đối với SMTP, chuỗi ký tự "CRLF.CRLF" báo nhận biết kết thúc nội dung bức Mail.
RSET	Bỏ (Reset) cuộc giao dịch hiện tại.
NOOP	Yêu cầu SMTP bên nhận không làm gì ngoài việc trả về câu trả lời OK (dùng để kiểm tra).
QUIT	Yêu cầu SMTP nhận trả lời OK và kết thúc phiên giao dịch hiện tại.
VRFY	Yêu cầu SMTP bên nhận kiểm tra người nhận là đúng, xác nhận các tham số gởi theo dòng lệnh.
SEND	Khởi động một cuộc giao dịch mà mail sẽ được gởi tới một hay nhiều thiết bị đầu cuối chứ không phải mailbox.
SOML	Khởi động một cuộc giao dịch mà mail sẽ được gởi tới một hay nhiều thiết bị đầu cuối hay mailbox.
SAML	Khởi động một cuộc giao dịch mà mail sẽ được gởi tới một hay nhiều thiết bị đầu cuối và mailbox.
HELP	Yêu cầu SMTP bên nhận gửi thông tin giúp đỡ cho SMTP bên phát.
EXPN	Yêu cầu SMTP bên nhận gởi về danh sách những người nhận Mail để có thể mở rộng việc chuyển mail cho các user khác.
TURN	Yêu cầu SMTP bên nhận gưi OK và đổi vai trò trở tránh SMTP gởi.

Bảng 10-1. Các lệnh của giao thức SMTP.

SMTP (trong RFC 821) ban đầu được thiết kế để cho phép các mail server chuyển đổi các mail message. Cơ chế chính được dùng để chuyển đổi các mail là phân đường các message quanh Internet. SMTP hoạt động trên mô hình lưu và truyền trong đó client nắm các message cần để truyền đến server và gửi các lệnh đến server để báo cho server cách xử lí các message. Mail client có thể là một mail server khác, nó có một hay nhiều message phải truyền đến một server khác. Hầu hết các Internet mail client sử dụng SMTP để gửi các message.



Hình 10-10. Cơ chế trao đổi SMTP.

10.6.1.1 Quy tắc làm việc với SMTP

1. Mỗi câu lệnh phân cách tham số theo sau bằng khoảng trắng và kết thúc bằng ký tự CRLF. Mail đi từ SMTP gởi đến một SMTP nhận và đến lượt SMTP nhận trở thành SMTP gởi để gửi mail đi tiếp cho đến khi chúng được giao vào Mailbox của người nhận.
2. Các lệnh SMTP phải diễn ra một cách tuần tự.
3. Việc đánh địa chỉ phải theo cách đánh địa chỉ Internet.

Giao thức SMTP qui định các Server MTA (ở đây là SMTP bên nhận) phải gởi tín hiệu phản hồi ACK sau mỗi lệnh mà nó nhận được từ Client MTA. Mỗi câu trả lời của bên nhận đều mở đầu với một mã số theo sau mới là thông tin dạng text. Mỗi số mở đầu trong mã số có một ý nghĩa khác nhau, nó chỉ ra rằng kết quả thực hiện thao tác là tốt (số 2), thất bại (số 5) hay chưa hoàn thành (số 3).

10.6.1.2 Một số mã phản hồi thông dụng của SMTP

220 Dịch vụ đã sẵn sàng.

221 Đóng kết nối đã được thiết lập.

250 Thao tác do Client MTA yêu cầu đã được hoàn thành.

354 Sẵn sàng nhận nội dung của mail.

550 Thao tác yêu cầu không thực hiện được do không có mailbox trên máy.

.v.v...

10.6.1.3 Phiên giao dịch SMTP

Để hiểu cách dùng một số lệnh chúng ta xem xét qua ví dụ sau: Bên gởi tên Thuận ở máy Sample1 muốn gởi cho Tín , Thức ở máy Sample2, giả sử Thức không có Mailbox tại Sample2.

Bên gởi thực hiện một kết nối đến SMTP Server.

RECEIVER : 220 sample2 Simple Mail Transfer Service Ready
Khi được kết nối qua giao thức TCP/IP, máy nhận trả lời với mã 220
để báo cho máy gởi biết dịch vụ SMTP đã sẵn sàng.
SENDER : HELO sample1
Bên nhận đã sẵn sàng, bên gởi gởi HELLO và xưng tên người gởi.
RECEIVER : 250 sample2
Trả với mã 250 báo cho biết bên nhận đã sẵn sàng.
SENDER : MAIL FROM: <>
Bên gởi dùng lệnh MAIL để khởi động phiên giao dịch. Cú pháp trên cho bên nhận biết
địa chỉ bên gởi (mailbox của bên gởi) để bên nhận gởi thông báo lỗi nếu có về bên gởi.
RECEIVER : 250 OK
Trả lời với mã 250 cho biết đã chấp nhận.
SENDER: RCPT TO: <>
Bên gởi cho biết e-Mail đích
RECEIVER: 250 OK
Trả lời với mã 250 cho biết đã chấp nhận
SENDER : RCPT TO: <>
Muốn gởi cho bao nhiêu người dùng bấy nhiêu lệnh RCPT kèm theo địa chỉ nhận,
bên nhận nếu đúng sẽ trả về mã 250 kèm theo OK.
RECEIVER : 550 No such user here
Báo kèm theo mã 550 cho biết không có mailbox trên địa chỉ trên đối với nơi nhận.
SENDER : DATA
Báo cho bên nhận biết dữ liệu bắt đầu từ sau từ DATA.
RECEIVER : 354 Start mail input; end with <CRLF>.<CRLF>
Mã 354 báo cho biết đã sẵn sàng nhận mail, kết thúc mail với ký tự "CRLF.CRLF".
SENDER : Bắt đầu thân của mail
SENDER : . . .
SENDER : (đến khi kết thúc gởi CRLF.CRLF)
RECEIVER : 250 OK
E-Mail đã được chấp nhận.

SENDER : QUIT

Phát lệnh báo kết thúc phiên giao dịch.

RECEIVER : 221 sample2 Service closing transmission channel

Mã 221 đóng kết nối đã thiết lập

10.6.1.4 Giao thức mở rộng ESMTP

SMTP có một hạn chế gây khó khăn lớn trong việc truyền nhận mail là giới hạn tối đa kích thước nội dung một bức mail chỉ là 128KB. Do vậy người ta đã cải tiến chuẩn SMTP thành một chuẩn mở rộng mới gọi là ESMTP, cho phép tăng giới hạn kích thước của mail lên trên 1MB.

Để biết xem Server MTA có theo chuẩn ESMTP hay không, thay vì dùng lệnh HELO ở đầu một cuộc giao dịch, Client MTA dùng lệnh mới EHLO, nếu Server MTA có trang bị, nó sẽ trả về mã thành công là 250. Ngày nay chuẩn ESMTP đã thay thế chuẩn SMTP ở đa số các hệ thống.

Chẳng hạn để khởi động cuộc giao dịch với kích thước mail lên tới 1MB, sử dụng dòng lệnh sau:

MAIL FROM :<thuan@sample1> SIZE=1000000

10.6.2 MIME

Từ khi MIME (Multipurpose Internet Mail Extention) được đưa ra, kiểu dữ liệu mà user có thể gửi thông qua e-Mail được mở rộng. Ban đầu dữ liệu chỉ ở dạng text. Ngày nay, ta có thể gửi các tài liệu (file *.doc), các file ảnh hay các file âm thanh.

Để có thể phân phát các kiểu dữ liệu này, khuôn dạng các message trên Internet nên được mở rộng. MIME được phát triển cho mục đích này.

10.6.2.1 Cấu trúc message của MIME

MIME không phải cho các ứng dụng e-Mail mới, nhưng cho phép mở rộng khả năng e-Mail trên Internet trong khi vẫn giữ các ứng dụng giao vận và nền tảng hiện tại. Khuôn dạng MIME duy trì các cấu trúc message cơ bản với các phần Header và phần body (tham khảo RFC 822). Ví dụ về khuôn dạng của một tài liệu MIME như sau :

{Dòng này xác định MIME message}

MIME-Version: 1.0

To:

Subject: Book CD

{Dòng này xác định đây là một kiểu messsage hỗn hợp và các phần được phân tách nhau bởi dấu biên}

Content-Type: multipart/mixed; boundary="--------6B9767D111AE"

X-Mozilla-Status: 0001

{Kết thúc phần header}
{Biên đầu tiên, thể hiện phần đầu của message}

--------6B9767D111AE
{Đây là đoạn text, thể hiện các kí tự dạng US-ASCII}
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
{Kết thúc phần header}

Davis,
I am .....
Thanhs,
Davis
{Phần sau là phần đánh dấu biên}

--------6B9767D111AE
{Phần tiếp sau là một file nhị phân}
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="Sublic2.doc"
{Phần dưới đây là nội dung file}

0M8.....
{Phần sau đây là biên kết thúc file}

--------6B9767D111AE

10.6.2.2 MIME version header

MIME version header định danh một message như một message MIME, và xác định version của MIME chuẩn để dịch message. Nếu không tìm thấy header, client sẽ đối xử với message theo khuôn dạng chuẩn trong RFC. Phiên bản hiện tại của MIME là 1.0. Cú pháp của MIME header version như sau:

MIME-Version: 1.0

1. Content Type header

Content Type header xác định khuôn dạng file được gán vào trong một đối tượng. Header báo cho MIME cách hiển thị hay thao tác trên thân của message. Content Type Header bao gồm tên của header, theo sau bởi kiểu MIME. Kiểu MIME theo sau hai tên và được cách biệt nhau bởi kí tự slash (/). Tên đầu tiên là tên kiểu và tên thứ hai là một tên phụ. Sau đây là các ví dụ của Content type header:

Content-Type: image/jpeg
Content-Type: image/gif
Content-Type: image/bmp
Content-Type: image/mpeg

Content-Type: application/octet-stream

Ba ví dụ đầu tiên trong phần này, đối tượng là kiểu ảnh (cũng là kiểu nhị phân), kiểu con của nó là jpeg, gif, và bmp. Các file ảnh này được nhúng vào trong các message. Dòng thứ tư trong các ví dụ này đó là một file chương trình.

Các kiểu và kiểu con có thể được thiết lập bởi các tham số. Mỗi tham số bao gồm một tên tham số, theo sau bởi dấu bằng (=) và tiếp theo là giá trị tham số. Các tham số này được tách biệt giữa kiểu và kiểu con, cũng như các tham số khác và được tách biệt nhau bởi dấu chấm phẩy. Ví dụ sau đây thể hiện một tập các tham số:

Content-Type: text/plain; charset=us-ascii

Kiểu đối tượng này báo cho người đọc message rằng các phần theo sau là dạng text và sử dụng các kí tự theo kiểu text.

Header này có thể hoàn toàn tuỳ chọn. Nếu nó không được cung cấp thì message được đối xử như một chuỗi các kí tự ASCII.

2. *Content Transfer Encoding Header*

Content Transfer Encoding Header xác định mô hình mã hoá được sử dụng để nhúng đối tượng vào trong thân của message. Để nhúng một đối tượng nhị phân vào trong một thư điện tử, cần phải chuyển nó sang kiểu dạng ASCII, do vậy nó được biên dịch theo khuôn dạng RFC 822. Ví dụ một cú pháp header dùng để mã hoá nội dung khi truyền là Content-Transfer-Encoding Base64.

Tài liệu MIME định nghĩa 5 kiểu mã hoá, nhưng 3 kiểu mã hoá thể hiện đối tượng không được mã hoá. Mã hoá 7 bit thường được dùng cho các vùng text theo khuôn dạng MIME. Hai kiểu kia mã hoá theo kiểu 8 bit và nhị phân, chỉ được sử dụng khi chuyển thư không phải SMTP, do SMTP chỉ cho phép các kí tự ASCII theo kiểu mã hoá 7 bit. Hai mô hình mã hoá còn lại đó là quoted-printable và base64 để chuyển các các đối tượng từ dạng nhị phân sang kiểu ASCII.

10.6.2.3 Cấu trúc message MIME đa phần

Một trong số các khả năng phổ biến của MIME đó là có một message đa phần. Bằng cách sử dụng message đa phần, ta có thể nhúng cả hình ảnh và âm thanh vào các message text hay xây dựng một ứng dụng về một đối tượng hoạt hình, nó bao gồm một số file cần thiết để chạy ứng dụng.

Cấu trúc message đa phần bao gồm nhiều message kết hợp vào trong thân của một message, mỗi message với thông tin header của nó thể hiện kiểu nội dung mà mô hình mã hoá. Các phần này được tách biệt bởi các dấu biên mà message chính định ra. Để hiểu chi tiết về cấu trúc của một message đa phần, xem RFC 1521.

10.6.2.4 Mã hóa BASE64

Thuật toán mã hoá Base64 được thiết kế để mô tả một chuỗi tuỳ ý các giá trị 8bit mà con người không có khả năng đọc được thành các kí tự ASCII. Thuật toán mã hoá và giải mã đơn giản nhưng dữ liệu mã hoá sẽ lớn hơn dữ liệu nguồn 33%.

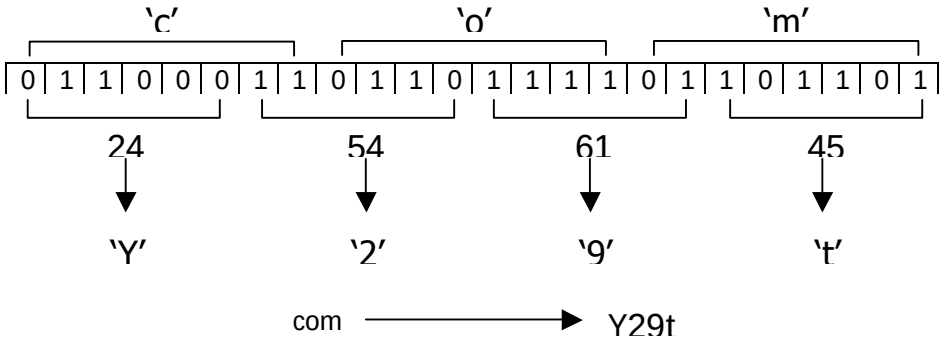
Một tập 65 kí tự US-ASCII được dùng, cho phép 6bits biểu diễn cho các kí tự có thể in được. (Kí tự thứ 65, "=", là một kí tự xử lý đặc biệt)

Tiến trình mã hoá biểu diễn nhóm 24 bits dữ liệu nhập thành 4 kí tự mã hoá ở đầu ra. Tiến trình thực hiện từ trái sang phải, một nhóm 24 bit nhập được kết hợp từ nhóm 3 kí tự 8bits. 24 bits đó được chia làm 4 nhóm kí tự 6bits, mỗi nhóm được dịch thành một kí tự đơn dựa vào bảng mã Base64.

Bảng mã Base64

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

Ví dụ sau mô tả tiến trình mã hoá 3 kí tự nhập là "com":



Luồng dữ liệu được mã hoá đầu ra phải được biểu diễn bằng các dòng có độ dài không lớn hơn 76 kí tự. Tất cả các kí tự xuống dòng hay các kí tự khác không có trong bảng mã Base64 đều được phần mềm giải mã bỏ qua.

Khi nhóm bit dòng nhập ít hơn 24 bits (nghĩa là đến cuối của dữ liệu cần mã hoá) thì cần có xử lý đặc biệt. Khi có ít hơn 24 bits dòng nhập thì các bits 0 được thêm vào phía bên phải nhóm bit để được đủ số 24 bits. Khi dòng nhập đã đủ 24bits thì có các khả năng có thể xảy ra:

1. Phần cuối cùng của dữ liệu cần mã hoá là 24 bits thì dữ liệu đầu ra cuối cùng sẽ là 4 kí tự đã mã hoá mà không có kí tự đệm "=".
2. Phần cuối cùng của dữ liệu cần mã hoá chính xác là 8 bits thì dữ liệu đầu ra cuối cùng sẽ là 2 kí tự đã mã hoá kèm theo với 2 kí tự đệm “=” ở cuối.

- Nếu phần cuối cùng của dữ liệu cần mã hoá chính xác là 16 bits thì dữ liệu đầu ra cuối cùng sẽ gồm 3 kí tự đã mã hoá kèm theo với 1 kí tự đệm “=” ở cuối.

Bởi vì các kí tự đệm chỉ được thêm vào cuối của dữ liệu nên khi gặp bất kì một kí tự “=” nào thì hiển nhiên là đã đến vị trí kết thúc của dữ liệu.

10.6.3 Giao thức POP

Người sử dụng có thể gửi thư bằng cách sử dụng SMTP, và có thể nhúng bất kỳ đối tượng nào vào trong message thông qua việc sử dụng khuôn dạng MIME. Tuy nhiên, với SMTP, server để nhận được các message thư phải nối đến client và gửi tất cả các message được phân phát cho client. Do đó, người sử dụng phải đăng ký tên máy dưới dạng tên địa chỉ Internet của người nhận.

SMTP được thiết kế trong trường hợp nhiều user sử dụng tất cả thời gian của họ kết nối đến một vài host và chạy một phiên đầu cuối. Giao thức không được thiết kế cho các tình huống thông dụng hiện nay, trong đó, hầu hết tất cả các user sử dụng e-mail kết nối hạn chế đến mail server đang giữ hộp thư. Người sử dụng phải duy trì các message thư trên server và chuyển nó đến cho client khi client yêu cầu. Đây là một mục đích trong thiết kế của POP.

POP (Post office Protocol) được thiết kế để bù đắp cho SMTP trong phần nhận các message. Những người thiết kế POP không gộp các chức năng gửi message và cho rằng SMTP tiếp tục được sử dụng để thực hiện các chức năng đó. Với giao thức POP, máy tính nhận khởi tạo kết nối. Máy nhận kết nối đến mail server, login và nhận bất kỳ một message nào đang chờ. Do vậy mà máy gửi không cần biết gì về máy nhận trừ khi nó sử dụng login và password để đăng nhập. Ngày nay, hầu hết tất cả các mail client trên Internet mà bạn có thể sử dụng để kết hợp cả SMTP và POP.

10.6.3.1 Mô hình thông tin POP

Trong mô hình lưu và phát, server mail cục bộ lưu các message đến khi các client nhận nó. POP client kết nối với server trên cổng 110 của TCP. Để đăng nhập vào server, user sử dụng định danh (ID) và password. Sau khi đăng nhập thành công vào server, client có thể yêu cầu server về các message mới đang sẵn sàng, lấy bất kỳ message nào mà server đang gửi hay xoá đi một message nào đó trên server.

Mô hình thông tin POP sử dụng 3 trạng thái giao tác để cung cấp chức năng này đến POP client:

- Trạng thái đặc quyền : Server kiểm tra quyền truy nhập của client (ID và password).
- Trạng thái giao tác : Client có thể nhận hay xoá các message.
- Trạng thái cập nhật : Trạng thái này được chuyển đến ngay sau khi client tạo ra lệnh QUIT.

Trạng thái cập nhật là trạng thái cho phép thao tác trên các message. Khi client đang ở trên trạng thái giao tác, bạn có thể tạo ra lệnh reset để huỷ bỏ tất cả các thao tác xóa trước đó (undo).

10.6.3.2 Chuẩn POP3

Giao thức POP3 được cải tiến từ giao thức POP. Nhiệm vụ của giao thức POP3 là lấy mail từ mailbox về khi nào người nhận muốn.

Đặc điểm của hệ thống dùng POP là cho phép người sử dụng login vào POP Server và nhận các mail từ mailbox của mình mà không cần phải login vào mạng mặc dù các mailbox thường nằm ở các Mail Server nằm trong mạng (thông thường muốn thâm nhập mạng ta phải có một account trên mạng và phải cung cấp Password khi đăng nhập vào mạng). Người sử dụng có thể truy xuất POP Server từ bất cứ một hệ thống nào trên mạng Internet, từ bất cứ UA nào dùng giao thức POP.

POP3 định nghĩa 3 giai đoạn tạo thành POP Session : Giai đoạn 1 là giai đoạn xác định tính hợp pháp của người nhận mail (Authorization); giai đoạn 2 là giai đoạn giao dịch giữa PC và POP Server (Transaction) và giai đoạn 3 là giai đoạn cập nhật thông tin (Update).

Sau khi thiết lập kết nối với Server, giai đoạn đầu Client sẽ cho Server biết nó là ai. Nếu Client hợp pháp POP Server sẽ mở Mailbox và bắt đầu chuyển sang giai đoạn giao dịch. Giai đoạn giao dịch, chương trình Client sẽ yêu cầu POP3 Server cung cấp các thông tin như danh sách mail..v..v..hay yêu cầu gởi về cho nó một bức mail xác định nào đó. Giai đoạn cuối cùng sẽ cập nhật và đóng kết hiện hành.

Các lệnh thông dụng của giao thức POP3 :

Lệnh	ý nghĩa
User	Cho biết tên của user cho POP Server
Pass	Yêu cầu một Password cho người sử dụng trên Server
Quit	Đóng kết nối TCT đã được thiết lập trước đó
Stat	POP Server trả về số lượng Mail có trong mailbox của người sử dụng cùng kích thước chúng
List	Trả về các ID và size của các Message
Retr	Nhận một Message từ Mailbox (yêu cầu tham số là ID của mail cần nhận)
Dele	Đánh dấu một Message để xóa (yêu cầu tham số là ID của mail cần xóa)
Noop	POP Server trả về +OK nhưng không làm gì cả
Last	Yêu cầu POP Server trả về số Message đã truy nhập
Top	Liệt kê Header của Mail
Rset	Hủy đánh dấu trên Message bị đánh dấu để xóa

POP3 chỉ định nghĩa 2 loại trả lời cho mỗi câu lệnh là : +OK để chỉ thao tác hoàn thành tốt và - ERR để báo có lỗi. Ví dụ cách dùng một số lệnh của POP3 như sau (các hàng sau dấu chấm phẩy để chú thích lệnh).

Giai đoạn 1 : Nhận dạng user

```
CLIENT : USER user01     ; cho biết tên user là user01
SERVER : +OK             ; báo thành công
CLIENT : PASS abc        ; cho biết password là abc
SERVER : +OK user01's    ; maildrop has 2 messages ( 520 octets)
```

Giai đoạn 2 : Trao đổi

```
CLIENT : STAT            ; số mail có trong mailbox
SERVER : +OK 2 520       ; Có 2 mail với tổng kích thước là 520
CLIENT : LIST            ; Liệt kê các ID và kích thước các mail
SERVER : +OK 2 message ( 520 octets )
SERVER : 1 110           ; mail thứ 1 kích thước 110
SERVER : 2 410           ; mail thứ 2 kích thước 410
CLIENT : LIST 1  ; Cho thông tin về mail có ID là 1
SERVER : +OK 1 110
CLIENT : LIST 4
SERVER : -ERR            no such message, only 2 message in maildrop
. . .v. . .v . . .
```

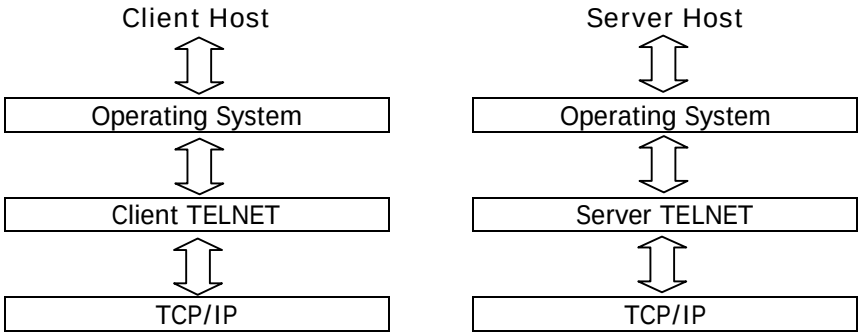
Giai đoạn 3 : Kết thúc

```
CLIENT : QUIT ; đóng kết nối TCP hiện hành
SERVER : +OK dhbk POP3 server signing off
```

Chú ý rằng các message bị đánh dấu để xóa bằng lệnh DELE thực sự chưa bị xóa ngay để nếu sau đó ta có thể dùng lệnh phục hồi không xóa bằng lệnh RSET,

chúng chỉ thực sự bị xóa bỏ khỏi maildrop khi bước vào giai đoạn Update (khi gởi lệnh QUIT).

10.7 Dịch vụ truy cập từ xa - TELNET



Hình 10-11. Phương thức truy nhập từ xa Telnet.

Chương trình Telnet (TELecommunication NETwork) cho phép truy cập từ xa hoặc có các thiết bị ảo thông qua mạng (điều này có nghĩa là bình thường thì bạn không thể có được thiết bị này nhưng nay nhờ có dịch vụ Telnet, bạn có thể truy cập và dùng được các thiết bị đầu cuối do đó gọi là các thiết bị đầu cuối ảo). Nói cách khác, một user A có thể truy cập vào một máy B ở bất cứ nơi nào trong mạng và làm việc với máy đó giống như đang ngồi trước máy đó. Dịch vụ Telnet được cung cấp qua cổng số 23 của TCP/IP. Khái niệm Telnet để chỉ cả *dịch vụ và giao thức* cung cấp các dịch vụ truy cập từ xa này.

Giao thức Telnet dùng một khái niệm Network ***Virtual Terminal*** (NVT), để định nghĩa kết nối Telnet cho cả hai phía. Mỗi đầu của kết nối (mỗi NVT) có một bàn phím và một máy in logic. Máy in logic có thể hiển thị các kí tự và bàn phím logic có thể tạo các kí tự. Máy in logic thường là một màn hình của thiết bị đầu cuối, trong khi đó bàn phím logic thường là bàn phím của người dùng

Khi một kết nối Telnet được thiết lập, Telnetd (hay bất kỳ một chương trình nào khác mà làm việc như là Telnet server) bắt đầu quá trình chạy một số các ứng dụng. Mỗi phím được ấn sẽ phải qua Telnet, Telnetd, và các ứng dụng được dùng trong quá trình thực hiện một phiên làm việc của kết nối Telnet.

Người sử dụng đưa vào lệnh và số liệu, chương trình Telnet ở máy khách (client Telnet) sẽ chuyển lệnh và số liệu đến chương trình Telnet trên máy chủ (server telnet) tương ứng. Server telnet xử lý và gửi kết quả trở lại cho Client Telnet.

10.7.1.1 Các lệnh của Telnet

Hai hệ thống Telnet Client/Server liên lạc với nhau bằng những lệnh gồm những *kí tự đơn* hay một *chuỗi kí tự*, nó được mã hoá trong dạng chuẩn NVT (Network Virtual Terminal - Mạng đầu cuối ảo).

Khi một kết nối Telnet được thiết lập, một số dịch vụ có thể sẵn sàng để lựa chọn. Giá trị của chúng có thể thay đổi trong một phiên làm việc Telnet (*Telnet Session*) nếu cả hai phía của kết nối đồng ý sự thay đổi đó. (Có thể xảy ra trường hợp một đầu của kết nối Telnet không thể cho phép hay không cho phép một dịch vụ trong quá trình kết nối Telnet diễn ra do sự cho quyền của nhà quản lý hoặc các thiết lập nguồn (Source settings)). Có bốn giao thức Telnet được dùng để Để nghị (offer), Từ chối (refuse), Yêu cầu (request) và Ngăn chặn (prevent) các dịch vụ, đó là các động từ: WILL, WON'T, DO và DON'T. Các động từ trên được thiết kế đi với nhau theo từng cặp (WILL/WON'T và DO/DON'T).

Lệnh	**Mã thập phân**	**ý nghĩa**
IAC	255	Nhận biết byte tiếp theo là lệnh
NOP	241	Không điều khiển
EC	247	Xóa ký tự (Erase character)
EL	248	Xóa dòng (Erase line)
GA	249	Về đầu (Go ahead)
AYT	246	Are you there
IP	244	Quá trình ngắt (Interrupt process)
AO	245	Xóa bỏ đầu ra (Abort output)
BRK	243	Dừng (break output)
DMARK	242	Phục hồi đầu ra (Resume output)
SB	250	Bắt đầu trao đổi (Start potion request)
SE	240	Kết thúc (End)
WILL	251	Thỏa thuận/Yêu cầu (Agreement/request option)
WONT	252	Từ chối (Refuse option request)
DO	253	Tiếp nhận yêu cầu (Accept request option)
DON'T	254	Từ chối tiếp nhận yêu cầu

- Các hàm chức năng khác :

Tên	**Mã**	**ý nghĩa**
Transmit binary	0	Yêu cầu/T.nhận trao đổi số nhị phân 8 bit
Echo	1	Ký tự phản hồi (Echo character receiving back to sender)
Status	5	Trạng thái (Request/reply status of receiving TELNET)
Timing mark	6	Đánh dấu thời gian.
Terminal type	24	Loại yêu cầu/trả lời của thiết bị đầu cuối.
Line mode	34	Gửi dòng ký tự

Ví dụ các dòng lệnh tiêu biểu như sau :

IAC, SB, WILL, 'O', SE : Yêu cầu bên nhận nhận số nhị phân 8 bit
IAC, SB, DO, 'O', SE : Hệ truy nhập từ xa nhận trả lời tiếp nhận
IAC, SB, DON'T, 'O', SE : Hoặc từ chối
IAC, SB, DO, 'O', SE : Bên nhận yêu cầu
IAC, SB, WILL, 'O', SE : Bên gửi thỏa thuận
IAC, SB, WON'T, 'O', SE : Hoặc từ chối

- Làm việc với Telnet

- Truy nhập vào mạng TCP/IP từ máy trạm
- Gõ lệnh : telnet <ĐịachỉIP hoặc tênmáyServer>
- Thao tác trên màn hình Telnet.

10.7.2 Dịch vụ truyền tập tin FTP

Giao thức truyền tập tin FTP (File Transfer Protocol) cho phép truyền các tập tin giữa hai máy tính, quản lý các thư mục và truy cập vào thư tín điện tử. FTP không được thiết kế để truy cập vào một máy khác và chạy các chương trình ở máy đó. FTP giúp người sử dụng truy cập file và thư mục trên một máy chủ ở xa và thực hiện những thao tác trên thư mục như sau :

- Liệt kê các file trên một thư mục cục bộ hay ở xa.
- Đổi tên và xóa tập tin (nếu có quyền).
- Truyền file đi hay về từ trạm và máy ở xa (download/upload).

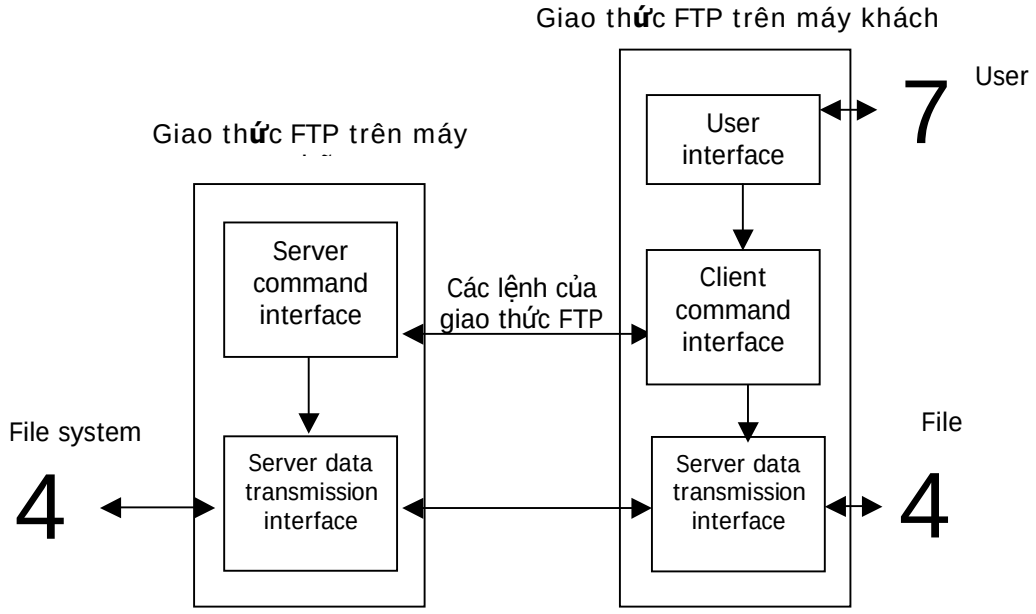
FTP dùng hai kênh TCP, với số hiệu cổng 20 là ***kênh dữ liệu***, và số hiệu cổng 21 là ***kênh lệnh*** (*command channel*). FTP khác các ứng dụng khác của TCP/IP ở là FTP quản lý tất cả việc truyền các tập tin bằng foreground thay vì background. Nói cách khác, FTP không dùng các hàng đợi hay các tiến trình kiểu ống (spooler) do đó bạn có thể quan sát quá trình truyền tập tin trong thời gian thực. Bằng cách dùng TCP, FTP loại trừ được việc quản lý kết nối và độ tin cậy, bởi vì FTP có thể dựa trên TCP để thực hiện các chức năng này một cách chính xác.

Kết nối đầu tiên, kênh lệnh, được khởi tạo thông qua FTP client. Client kết nối với server dựa trên cổng 21 của TCP, cung cấp cho server tên (login) và password và sau đó tiến đến các phiên FTP. Nếu client tạo ra một lệnh yêu cầu một dòng trả lời từ server, kênh lệnh sẽ truyền trả lời này.

Khi client gởi một yêu cầu có nhiều hơn một trả lời để gửi hay nhận dữ liệu, kênh thứ hai được đặt vào hoạt động. Để thiết lập kết nối thứ hai, bạn có 3 tuỳ chọn. Mặc định, server khởi tạo kết nối thứ 2 thông qua cổng 20 của TCP và kết nối đến một socket thứ hai trên client, sử dụng cùng một địa chỉ và cổng như trong kết nối thứ nhất trên client. Tuy nhiên, client có thể chỉ định một địa chỉ khác hay một cổng khác để truyền dữ liệu, trong trường hợp này, server cố gắng kết nối đến client

thông qua việc sử dụng một địa chỉ mới. Tuỳ chọn thứ 3 là client khởi tạo một kết nối truyền dữ liệu là báo cho server chuyển sang chế độ thụ động, server trả lời một địa chỉ và số hiệu cổng để truyền dữ liệu.

Ngay sau khi truyền dữ liệu kết thúc, kết nối để truyền dữ liệu được đóng lại. Kết nối này được mở lại khi client tạo ra một lệnh yêu cầu truyền dữ liÖu.



Hình 10-12. Mô hình giao tiếp FTP.

- FTP hoạt động theo mô hình Client/Server bao gồm thành phần chính :

  + Đơn vị trao đổi dữ liệu (Data Transmission interface)/

  + Đơn vị nhận biết lệnh (Command interface)

10.7.2.1 Chế độ truyền dẫn

Có 3 chế độ được dùng để truyền dữ liệu giữa hai hệ thống. Chế độ đầu tiên là ngầm định nhưng 2 chế độ kia truyền hiệu quả hơn và có thể phục hồi.

- ***Truyền theo dòng:*** đây là chế độ truyền ngầm định, gửi một file dưới dạng một chuỗi các byte; FTP server và client không định dạng file đó. File nguồn không có cách gì để báo hết nội dung truyền, do vậy vấn đề kết thúc file được qui định bằng đóng kết nối dữ liệu.
- ***Truyền theo khối:*** chia file thành các khối, và mỗi khối có thêm các byte điều khiển (header). Trong header có một trường xác định số lượng byte trong khối, trường mô tả mã, nó có thể định đó là khối đặc biệt, kết thúc trong quá trình truyền. Chế độ truyền này cho phép phục hồi khi bị ngắt trong quá trình truyền file thông qua việc báo truyền lại một khối chỉ định trong trường count của header.

- ***Chế độ truyền nén:*** nén file để truyền thông qua việc sử dụng thuật toán mã hoá mã run-length. Thuật toán nhằm làm giảm các byte lặp lại vào trong hai byte kế tiếp. Byte đầu tiên cho biết byte theo sau là nén và số lần nó được lặp lại. Để thể hiện nén, bit đầu tiên của byte điều khiển được thiết lập 1. Nếu bit này là 0, nó cho biết byte theo sau không phải là byte nén. Phần còn lại của byte điều khiển xác định số lượng các byte không nén theo sau. Do vậy, hiệu quả khi nén các kí tự lặp lại đó là không làm mất đi các kí tự không nén.

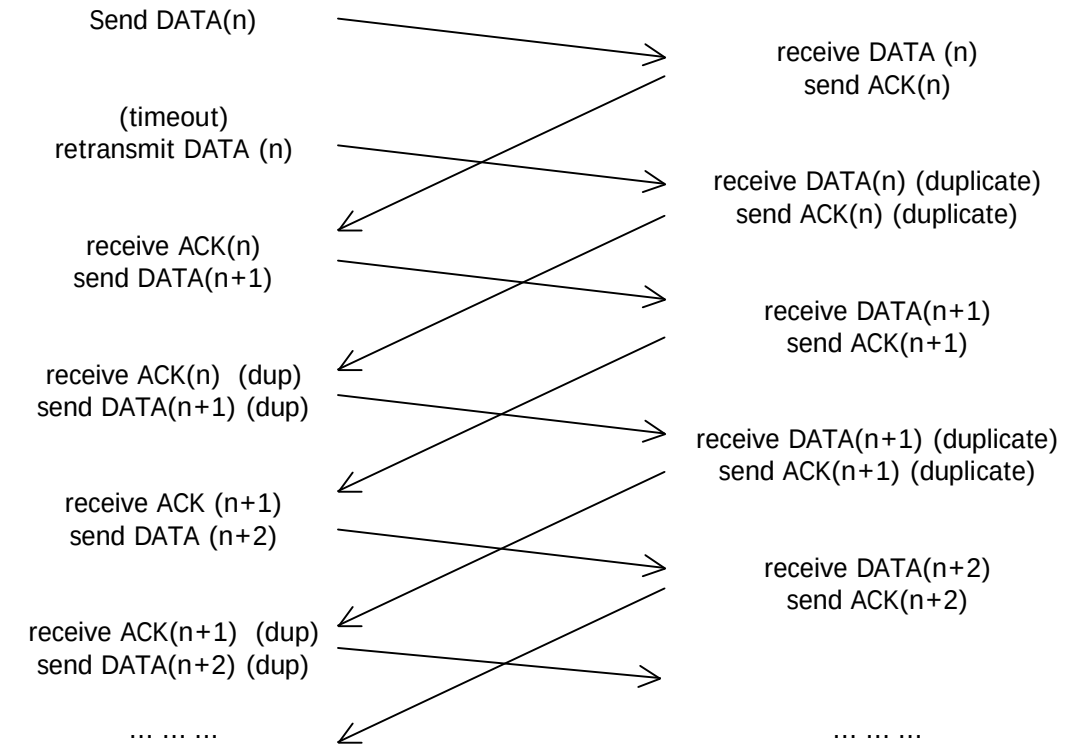
Để bắt đầu, client gửi yêu cầu **read** hay **write**. Gói tin trao đổi có độ dài đến 512 bytes. Mỗi block số liệu có đánh số và phải được biên nhận để gửi tiếp hay phát lại. Để tránh phát trùng lặp khi hết thời hạn, phát lại bản tin vừa phát và khi nhận ACK (n) trùng lặp thì không phát gì .

10.7.2.2 Dạng bản tin FTP

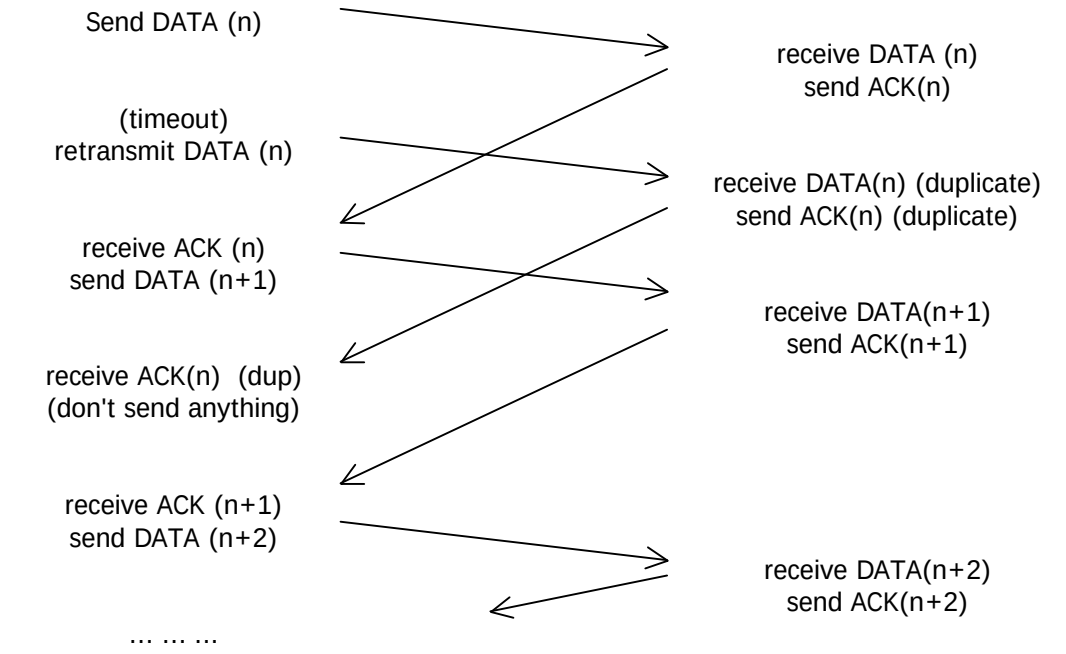
Read request (RRQ)	opcode	String	EOs	String	EOs
	01	File name	0	mode	0
	2 bytes	n bytes	1 byte	n bytes	1 byte
Write request (WRQ)	opcode	String	EOs	String	EOs
	02	File name	0	mode	0
	2 bytes	n bytes	1 byte	n bytes	1 byte
DATA	opcode	Block#	Data		
	03				
	2 bytes	2 bytes	n bytes, 0≤n≤512		
Acknowled-gement (ACK)	opcode	Block#			
	04				
	2 bytes	2 bytes			
Read request (RRQ)	opcode		String	EOs	
	05	Errorcode	Err String	0	(EOs : End of String)
	2 bytes	2 bytes	n bytes	1 byte	

Hình 10-13. Khuôn dạng bản tin FTP.

Ví dụ: Quá trình phát lặp :



+ Tr¸nh ph¸t lÆp



Hình 10-14. Quá trình phát lặp bản tin FTP.

10.7.2.3 Quá trình làm việc FTP

1. Truy nhập vào mạng TCP/IP từ máy trạm.
2. Gõ lệnh : ftp <Địa_chỉ_máy_Server>.
3. Làm việc với FTP.

Khi một kết nối FTP được thiết lập, thực hiện các bước như sau:

- Duyệt tên và mật khẩu (ID) của người dùng.
- Xác định thư mục bắt đầu làm việc.
- Định nghĩa chế độ truyền tập tin.
- Cho phép các lệnh của người dùng.
- Huỷ kết nối.

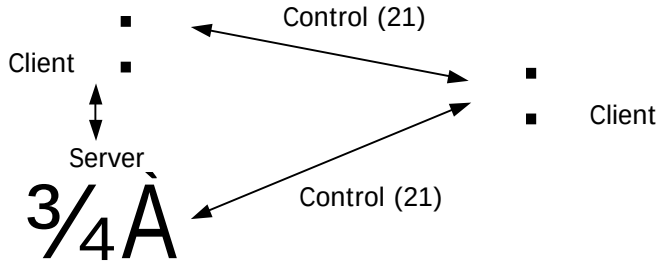
Lệnh FTP	**Mô tả**
ascii	Chuyển sang chế độ truyền ascii
bell	âm thanh của chương trình sau khi truyền mỗi tập tin
binary	Chuyển sang chế độ truyền nhị phân
cd *directory*	Chuyển đổi thư mục hiện hành trên server
cdup	Lùi thư mục hiện hành về một cấp trước đó
close	Huỷ kết nối
delete *filename*	Xoá một tập tin trên server
dir *directory*	Hiển thị thư mục *directory* của server
get *filename*	Truyền tập tin trên server về máy cục bộ
hash	Hiển thị/làm mất dấu # cho mỗi khối các ký tự đã truyền được
help	Hiển thị các trợ giúp
lcd *directory*	Chuyển đổi thư mục hiện hành trên máy cục bộ
ls *directory*	Xem danh sách các tập tin trong thư mục directory trên Server
mdelete *files*	Xóa nhiều tập tin trên máy Server
mdir *directories*	Liệt kê các tập tin trong nhiều thư mục trên máy Server
mget *files*	Lấy một số file trên Server về thư mục hiện hành của máy cục bộ
mkdir *directory*	Tạo thư mục directory trên máy Server
mput *files*	Gửi một số tập tin từ máy cục bộ lên máy Server
open *host*	Kết nối với Server host từ xa
put *filename*	Truyền tập tin từ máy cục bộ lên máy Server
pwd	Hiển thị thư mục hiện thời của server
status	Hiển thị trạng thái của ftp
rename *file1 file2*	Đổi tên *file1* trên máy Server thành *file2*
quote	Cung cấp một lệnh FTP một cách trực tiếp
quit	Chấm dứt kết nối và thoát khỏi ftp
?	Hiển thị danh sách lệnh

Đề truyền một tập tin từ *thư mục hiện hành* trên máy Client đến máy Server bạn dùng lệnh *put*, ngược lại, muốn tải tập tin từ máy Server về máy Client, bạn dùng lệnh *get*. Cú pháp như sau :

```
ftp>put   local_file   remote_file
ftp>get   remote_file   local_file
```

Khi truy cập vào hệ thống, nếu chưa có account, người sử dụng có thể sử dụng một login name đặc biệt là *anonymous* để truy cập vào hệ thống. Acount này không có mật khẩu.

FTP cho phép truyền các tập tin thông qua máy thứ 3, máy này nằm giữa client và server. Thủ tục này được gọi là truyền tay ba điều này cần thiết để có được có được sự cho phép chính xác để truy cập vào máy ở xa. Hình sau mô tả sơ đồ của thủ tục này :



Hình 10-15. Truyền các tập tin thông qua máy thứ 3.

10.7.2.4 Khuôn dạng dữ liệu

Khi truyền dữ liệu giữa hai hệ thống, có thể sử dụng 4 kiểu dữ liệu để truyền. Trong số các kiểu dữ liệu này thì có 2 kiểu dữ liệu hay được sử dụng nhất hiện nay, hai kiểu khác vẫn được hỗ trợ nhưng ít được sử dụng. Các hệ thống ở cả hai đầu trong quá trình đàm thoại FTP phải hỗ trợ tất cả các kiểu dữ liệu sau đây:

- Kiểu ASCII, đây là kiểu mặc định được dùng trong các phiên FTP. Nó được dùng để truyền các file text. Nếu bạn cố truyền các file nhị phân mà bạn không thay đổi mode thì bạn cũng nhận được kết quả ở dạng text, do vậy nội dung của file đã bị thay đổi.
- EBCDIC được sử dụng để truyền các file giữa giữa các host, sử dụng EBCDIC như một tập các kí tự bên trong của nó. Về mặt kỹ thuật thì kiểu dữ liệu ASCII và EBCDIC là giống nhau, chỉ khác một điều là tập các kí tự mà nó sử dụng
- Kiểu nhị phân là kiểu được sử dụng để truyền các file nhị phân như các file ảnh và các file chương trình (các file ZIP và các file DOC). Việc truyền các file này dưới dạng một chuỗi các byte, kiểu dữ liệu này không quan tâm đến môi trường của máy đích và cấu trúc từ. Tất cả các cài đặt FTP nên hỗ trợ kiểu truyền dữ liệu này cũng như kiểu ASCII.
- Kiểu dữ liệu cục bộ. Kiểu dữ liệu này dựa trên byte, xác định cho các host cục bộ. Khuôn dạng phải khả dụng với các hệ thống khác để cấu trúc lại dữ liệu dựa vào dựa trên khuôn dạng ban đầu.

Kiểu dữ liệu ASCII và EBCDIC có thể có tham số tuỳ chọn thứ hai để xác định các ràng buộc dữ liệu. Khi được sử dụng, tham số này là một tuỳ chọn được thêm vào để xác định kiểu dữ liệu. Các ràng buộc định dạng phụ thuộc vào việc sử dụng của file được truyền. Liệu một file có thể được in, xem, hay được xử lí như một đầu vào. Việc định dạng một file có thể khác nhau ở mỗi đích. Các khuôn dạng dữ liệu sau được ít sử dụng hơn kiển dữ liệu ngầm định:

- Khuôn dạng không in: là kiểu dữ liệu ngầm định ASCII và EBCDIC. Khuôn dạng file này không có thông tin định dạng. Chú ý rằng, định dạng sử các dạng chuẩn cho kí tự cách và phân lề.
- Định dạng Telnet được sử dụng cho các file một thiết bị đầu cuối dùng để hiển thị. Định dạng này gồm các kí tự điều khiển, kí tự xuống dòng, tab.
- Kiểm soát di chuyển bao gồm các kí tự điều khiển định dạng in. Theo khuôn dạng này, kí tự đầu tiên của mỗi dòng không được in ra. Thay vào đó, kí tự này xác định sự di chuyển theo trục đứng so với mép giấy trước khi một bản ghi hay một dòng nào đó được in ra.

10.7.2.5 Các cấu trúc dữ liệu

Giao thức FTP cho phép truyền các file có cấu trúc với 3 cấu trúc file khác nhau. Các cấu trúc tập tin này chủ yếu dùng để truyền các tập tin giữa các hệ thống có cấu trúc lưu trữ khác nhau. Có các dạng như sau :

- Cấu trúc theo kiểu file, xem file một chuỗi các byte dữ liệu nối tiếp nhau mà không được cấu trúc bên trong.
- Cấu trúc bản ghi được sử dụng để truyền các file là một chuỗi các bản ghi. Cấu trúc này được sử dụng cho các Host IBM nhưng hiện nay ít sử dụng.
- Cấu trúc trang được sử dụng cho các file được chia thành các đối tượng với kích thước khác nhau, có thể có các thông tin khác được thêm vào trong đó. Cấu trúc trang có một cấu trúc header để định nghĩa kích thước của trang, theo sau là nội dung của trang. Header của mỗi trang còn chứa số hiệu trang logic của các trang dữ liệu nhưng số hiệu trang đó không cần thiết khi truyền.

10.7.3 UserNEWS

Biểu tượng	Ý nghĩa	Biểu tượng	Ý nghĩa
:-)	Tôi hạnh phúc	=):=)	ABC Lincol
:-(	Tôi buồn/ tức giận	=):=)	Bác Sorn
:-|	Tôi thờ ơ	*<:-)	ông già Noel
;-)	Tôi nháy mắt	<:-(	Người tối dạ
;-(0)	Tôi kêu la	(-:	Người Uực
:-(*)	Người nôn (mửa)	:-)x	Man with bowtic
:+)	Cằm chẻ	# -)	Tóc mướt
:-))	Cằm chẻ	8 -)	Mang kính
:-{)	Ria	C:-)	Mão lớn

Khi mà có nhiều người thuê bao USENET, nhu cầu về những newsgroup mới, chuyên biệt hơn luôn được đòi hỏi. Kết quả là một thủ tục để tạo ra newsgroup mới, chuyên biệt hơn luôn được đòi hỏi. Kết quả là một thủ tục để tạo ra những

Newsgroup mới được. Trên Newsgroup, người ta có thể thảo luận, bầu cử, trao đổi với nhau.

10.7.4 WORLD-WIDE-WEB

World Wide Web (WWW) là một hệ thống quản lý thông tin phi cấu trúc. Bao gồm các Server cung cấp thông tin theo định dạng siêu văn bản (Hypertext) và các client (Browser, trình duyệt) nhận thông tin từ người sử dụng và đồng thời hiển thị thông tin mà các Server cung cấp theo định dạng được chỉ định bởi người sử dụng.

Thông tin trên WWW được biểu diễn trong các trang Web. Mỗi trang Web có thể là một chỉ mục hoặc một tài liệu chứa văn bản, hình ảnh, âm thanh, các liên kết... Người sử dụng có thể truy cập thông tin cần thiết trên WWW thông qua các đối tượng đã được đánh dấu trong tài liệu.

Các lệnh được dùng với WWW đã được định nghĩa trong giao thức HTTP (HyperText Transfer Protocol). Đây là giao thức chuẩn để liên lạc giữa Client và Server. Yêu cầu được gửi tới Server thông qua Client. Server xử lý các yêu cầu và gửi kết quả về cho Client yêu cầu. Kết quả sẽ được trình bày dưới dạng thích hợp cho người sử dụng.

- Phía máy chủ

Mỗi web Site có một máy chủ đảm nhận việc “lắng nghe” TCP tại cổng 80 cho những kết nối đến từ các máy khách (thường là các trình duyệt). Sau khi một kết nối được thiết lập, máy khách gửi yêu cầu và máy chủ trả lời đáp lại, kết nối chấm dứt. Giao thức HTTP định nghĩa cho các yêu cầu và trả lời hợp lệ.

Ví dụ người dùng kích lên một mẫu văn bản hoặc có thể là biểu tượng trỏ đến trang có tên (tức là URL hay địa chỉ tới máy trạm Internet). Một URL có 3 phần sau: tên của giao thức (http), tên của máy nơi có chứa trang web, và tên của tập tin chứa trang đó (hypetext/WWW/TheProject.html). Từ khi người dùng nhấp chuột cho đến khi trang web được hiện ra trên màn hình đã xảy ra các sự kiện sau :

1. Trình duyệt kiểm tra URL (xem xét đối tượng được chọn là gì).
2. Trình duyệt hỏi DNS về địa chỉ IP của URL.
3. DNS trả lời là 18.23.0.23
4. Trình duyệt tạo một kết nối TCP đến cổng 80 trên địa chỉ 18.23.0.23
5. Trình duyệt gửi lệnh GET /hypetext/WWW/TheProject.html.
6. Máy chủ gửi đến tập tin TheProject.html
7. Giải phóng kết nối TCP.

8. Trình duyệt hiển thị tất cả các văn bản trong tập tin TheProject.html.
9. Trình duyệt tiếp tục lấy về và hiển thị tất cả các hình ảnh có trong TheProject.html.

10.7.4.1 Ngôn ngữ HTML

HTML (HyperText Markup Laguage) là một ngôn ngữ HTML là một ngôn ngữ có cấu trúc, nó bao gồm các thẻ (TAGS) và các thực thể (ENTITY), dùng để cung cấp các chỉ thị định dạng để phục vụ cho việc trình bày văn bản trên Web.

Một tập tin HTML là một tập tin văn bản trong đó một số xâu ký tự được coi là các thẻ đánh dấu các vùng tài liệu và ấn định các ý nghĩa đặc biệt cho chúng. Các thẻ là các xâu ký tự được bắt đầu là dấu nhỏ hơn (<) và kết thúc bằng dấu lớn hơn (>). Các thẻ có thể được phân làm nhiều loại tùy theo nội dung, chức năng, kiểu tác động của chúng như: Thẻ mô tả định dạng, thẻ mô tả cấu trúc, thẻ rỗng, thẻ chứa...

Cấu trúc tổng quát của một tài liệu HTML như sau :

<HTML> *Thông báo cho trình duyệt đây là một văn bản tài liệu HTML*
<HEAD> *Thông báo bắt đầu phần đầu của tài liệu*
<TITLE> *Tiêu đề của tài liệu* </TITLE>
Phần đầu của tài liệu đặt tại đây
</HEAD> *Kết thúc phần đầu*
<BODY> *Thông báo bắt đầu phần thân tài liệu*
.....
Nội dung tài liệu HTML được đặt tại đây
</BODY> *Kết thúc phần thân tài liệu*
</HTML> *Kết thúc tài liệu HTML*

Phần đầu đề của tài liệu HTML thường chứa tiêu đề của tài liệu, tên tác giả, lời chú thích, tóm tắt... Đây là phần giúp ích cho việc tìm kiếm thông tin trên WEB hoặc cho các dịch vụ tìm kiếm có thể đánh chỉ mục, tiến hành tìm kiếm một cách dễ dàng. Một số các thẻ phục vụ trong phần đầu như: Title, Meta, Isindex...

Phần thân là phần chính của tài liệu HTML, nằm giữa cặp thẻ <BODY> và </BODY>, nó định nghĩa, hiển thị toàn bộ nội dung bên trong của tài liệu. Trong phần thân ta có thể sử dụng các thẻ để định dạng văn bản, chèn các hình ảnh, bảng biểu, liên kết...

Người sử dụng có thể tạo một tài liệu HTML bằng cách sử dụng các trình soạn thảo Web chuyên dụng như Microsoft Front Page 2000, hoặc Microsoft Word, Notepad ...

Một số thẻ HTML quan trọng :

1. *Thẻ <!-- (chú thích) -->:*

Dùng để thêm những dòng chú thích trong file HTML, người ta dùng thẻ nầy. Nội dung văn bản nằm giữa <!-- và --> sẽ được chương trình Browse bỏ qua. Cho phép có khoảnh trắng giữa -- và >, nhưng không được có khoảng trắng giữa <! và -- .

Thí dụ:

<HEAD> <TITLE>The HTML Reference</TITLE>

<!-- Created by Nguyen Tan Khoi, April 1996 --> </HEAD>

2. *Thẻ <A>*

Dùng để tạo các siêu liên kết (HyperLink). WWW cho phép kết nối và giao tiếp giữa các tài nguyên một cách dễ dàng nhờ định nghĩa các loại liên kết sau:

1. Liên kết giữa các thành phần khác nhau trong một tài liệu HTML.
2. Liên kết giữa các tài liệu HTML khác nhau.
3. Liên kết với các dạng tài liệu Multimedia.
4. Truy cập tới các dịch vụ thông tin khác trên mạng Intranet/Internet

Các thuộc tính của thẻ <A> như sau:

a. Liên kết đến điểm neo trong trang HTML

- NAME: Thuộc tính NAME xác định một vị trí để những thành phần khác trong tài liệu hoặc trong tài liệu khác có thể tham trỏ đến (gọi là điểm neo trong tài liệu HTML). Thí dụ :

  <A NAME="coffee"> Coffee</A>

  Các tài liệu khác có thể liên kết với tài liệu này ngay tại vị trí xác định.

b. Liên kết đến một trang HTML

<A HREF = "URL_HTML[#Name_Anchor]"> *Nội dung thông báo* </A>

Trong đó URL_HTML là địa chỉ để tham chiếu tới tài liệu HTML.

Nếu chỉ ra Name_Anchor thì có nghĩa ta định nghĩa một điểm neo dùng để chuyển đến một vị trí được quy định sẵn trong tài liệu HTML này. Thí dụ:

The <A HREF="document.html#glossary"> GLOSSARY </A>

Trong thí dụ trên, nếu kích vào "GLOSSARY" sẽ được chuyển đến tài liệu document.html, ngay tại vị trí điểm neo có tên glossary trong tài liệu này.

c. *Liên kết với các kiểu dữ liệu khác nhau*

Để liên kết giữa tài liệu hiện thời với các kiểu dữ liệu khác nhau như: hình ảnh, âm thanh, video...

<A HREF ="URL_DATA"> ...</A>

Trong đó URL_DATA là địa chỉ tới kiểu dữ liệu cần liên kết. Ví dụ:

<A HREF ="car.jpg">

<IMG SRC ="carla.gif" WIDTH=87 HEIGHT=60> </A>

d. *Liên kết với các dịch vụ thông tin khác trên mạng*

<A HREF = "URL_Service" > ... </A>

Trong đó URL_Service là một địa chỉ đến các dịch vụ trên internet.

<A HREF="http://..."> Liên kết với 1 Web Site.

<A HREF="ftp://..."> Với 1 Ftp Site.

<A HREF="gopher://..."> Với 1 Gopher server.

<A HREF="news:..."> Liên kết với 1 nhóm Tin.

<A HREF="mailto:..."> liên kết tới 1 địa chỉ gởi Mail. Liên kết nầy sẽ kích hoạt chương trình Mail và tự động điền địa chỉ vào mục To dùm bạn. Bạn có thể khai báo luôn cả chủ đề thư (?subject).

Thí dụ: <A HREF="mailto:cmlehunt@swan.ac.uk?

subject=The HTMLib is fantastic">link text</A>

- TARGET: Chương trình Browser có thể nạp đối tượng liên kết vào 1 cửa sổ chỉ định bằng thẻ nầy. Nếu cửa sổ nầy cha có, trình Browse sẽ mở 1 cửa sổ mới. Chủ yếu thẻ nầy dùng cho frames.

  Dạng chung:

  <A HREF="url.html" TARGET="window_name">Link text</A>

  Trong đó window_name là tên đặt cho Frame.

  Khi kích chuột vào dòng "Link text", trang "url.html" sẽ được nạp vào frame có tên chỉ định.

Ngoài ra ta còn có thể chèn thêm các Script sau vào thẻ <A> dựa vào các phương thức như sau :

Phương thức	Giải thích
OnMouseOver	Khi bạn di chuyển Mouse đến liên kết, sẽ có 1 dòng văn bản mô tả xuất hiện trong thanh trạng thái của trình Browse. Thí dụ: <A HREF="index.html" OnMouseOver="self.status=('Back to the main page')">Link text</A> Dòng chữ "Back to the main page" sẽ hiện trong thanh trạng thái khi dời Mouse đến chữ "Link text".
OnMouseOut	Tương tự như trên nhưng dòng chữ nầy lại xuất hiện khi kéo Mouse ra khỏi liên kết. Thí dụ: <A HREF="index.html" OnMouseOut="alert('Oh please go to this document')">Link text</A>
OnClick	Khi bấm Mouse lên liên kết, sẽ xuất hiện hộp thoại yêu cầu xác nhận. Thí dụ: <A HREF="http://www.netscape.com/" OnClick="confirm('Are you want to go to the Netscape site?')">Link text</A>

3. *Thẻ <INPUT>*

Dùng để tạo một field để nhận tác động của người sử dụng.

<INPUT TYPE = "*Kiểu*" NAMQ = "*TênĐT*"
SIZQ = "*KíchThước*"
VALUQ = "*Giatri*" MAXLQNGTH = "n" . . . >

Các thuộc tính:

Thuộc tính	Giải thích
ALIGN	So hàng cho field.
CHECKED	Kiểm tra người dùng đã đánh dấu cho checkbox hay radio button chưa.
MAXLENGTH	Chỉ định độ dài ký tự có thể nhập vào text field, độ dài nầy có thể lớn hơn kích thước Text field. Mặc định là không giới hạn.

NAME	Tên của Field.
SIZE	Khai báo kích thước hay số lượng ký tự cho field.

- TYPE: Chỉ định kiểu của Field:

Giá trị	Giải thích
BUTTON	Chèn một nút bấm vào tài liệu. Giá trị VALUE dùng chỉ định Text sẽ hiện trong nút nầy. Thí dụ: <input type="button" value="hello" name="btnhello">
HIDDEN	Với thuộc tính nầy, field sẽ không hiển thị ra nhưng nội dung của field vẫn có giá trị. Dùng trao đổi thông tin ngầm giữa Client/Server.
PASSWORD	Giống như Text, nhưng ký tự nhập vào sẽ không hiển thị ra.
CHECKBOX	Chèn 1 chechbox vào tài liệu. Thí dụ : So thich <input type="checkbox" name="C1" value="ĂN">The thao <input type="checkbox" name="C2" value="ĂN">Xem phim
RADIO	Chèn 1 field có dạng nút Radio. Ví dụ : Gioi tinh <input type="radio" checked value="V1" name="R1">Nam <input type="radio" name="R2" value="V2">Nu
RESET	Chèn 1 nút bấm dùng phục hồi lại tình trạng cũ cho các field. Đặt tên của nút nầy qua thuộc tính Values.
SUBMIT	Một dạng nút bấm giống RỌSỌT. Có tác dụng giống nh xác nhận đồng ý. Thí dụ: <input type="submit" value="Submit" name="B1"> <input type="reset" value="Reset" name="B2"> Chèn 1 nút có tên "SUBMIT" và sẽ hiển thị thông báo "Xin chao các bạn" khi người sử dụng Mouse vào nút nầy : <INPUT TYPE="SUBMIT" OnClick="Xin chao các bạn'">
TEXT	Nhập 1 dòng text vào fields. Dùng thuộc tính SIZỌ và MAXLENGTH để quy định kích thước. Trong trường hợp cần nhập

	nhiều dòng, phải dùng thẻ <TEXTAREA>.
VALUE	Chỉ định Text sẽ hiển thị trên các nút bấm.
IMAGE	Chèn field chứa hình ảnh để người dùng bấm Mouse khi chọn. <INPUT TYPE="IMAGE" SRC="../ iexplore.gif" ALIGN="middle">

4. *Thẻ TEXTAREA*

Cho phép nhập nhiều dòng văn bản vào một hộp Text.

Thí dụ:

```
<TEXTAREA
NAME="descr"
COLS="30"          ROWS="3"
OnBlur="count_char(document.egForm.descr.value)">Enter a short description here
 </TEXTAREA>
```

Ví dụ : <textarea name="Ghichu" rows="2" cols="20"></textarea>

5. *Thẻ FORM*

Forms là một thiết lập nhỏ trong HTML, nó cho phép người sử dụng đưa vào các thông tin. Giao diện Forms tạo nên sự thuận lợi trong việc tương tác giữa người sử dụng và các dịch vụ. Trên Form ta có thể tạo các thành phần như các nút lệnh, các trường văn bản (Text) hay các danh sách lựa chọn ... Khi forms được hoàn thành bởi người sử dụng, Client sẽ gửi thông tin đến Server, Server sẽ thực thi các chương trình kết hợp với form và các tham số là các thông tin nhận từ Form.

Thông thường các Form sử dụng cho hai mục đích chính:

- Dùng để thu thập thông tin từ người sử dụng.
- Là trung gian để tương tác qua lại giữa người sử dụng và hệ thống.

Cú pháp : <Form ACTION = “Action” METHOD="PhuongThuc">

Action: là một URL hoặc một Script mà khi nút ***Submit*** được nhấn nó sẽ thực thi.

Method=GET/POST : Xác định kiểu yêu cầu mà trình duyệt gởi đến cho Server.

- METHOD = GET: trình duyệt sẽ bổ sung dữ liệu đầu vào dưới dạng một biến môi trường là CGI_QueryString.
- METHOD=POST: Forrn dữ liệu đầu vào sẽ đợi từ các thiết bị nhập của Server cùng với một số dữ liệu được lưu trữ trong biến môi trường CGI_ContentLength.

EncType: cung cấp kiểu Mime của tập được dùng như đầu vào trong các biểu mẫu.

Ví dụ : <Form ACTION = METHOD="GET">

6. *Thẻ TABLE*

- Dùng để tạo ra một bảng. Bảng được tạo thành từ các hàng, trên mỗi hàng có các ô (cell).

```
<TABLE>
        <TABLE   BORDER = “n” . . . >
        <TR>
                <TD>  . . . </TD> <TD>  . . . </TD> <TD>  . . . </TD>
        </TR>
        . . . .
        <TR>
                <TD>  . . . </TD> <TD>  . . . </TD> <TD>  . . . </TD>
        </TR>
</TABLE>
```

7. *Thẻ SELECT*

- Hiển thị hộp ComboBox cho phép chọn lựa một trong nhiều giá trị :

```
<SELECT  NAME =”TenĐT”>
        <OPTION SELECTED VALUE =”Gia trị 1”> Nội dung 1
        <OPTION SELECT VALUE =”Gia trị 2”> Nội dung 2
                . . .
</SELECT>
```

Ví dụ :

```
<p>Que quan <select size="1" name="cboQuequan">
        <option selected>Da Nang</option>
        <option>Hue</option>
        <option>Ha Noi</option>
        </select></p>
```

8. *Thẻ <APPLET>*

Dùng để chèn Applet Java vào trang Web. Có dạng tổng quát sau:

```
<APPLET
    [CODEBASE = URL] [CODE = appletFile]
    [NAME = appletInstanceName]:
    [ARCHIVE = compressed file] [ALT = alternateText]
    [WIDTH = pixels]  [HEIGHT = pixels]  [ALIGN = alignment]
    [VSPACE = pixels] [HSPACE = pixels]
    [ARCHIVE = URL to archive]
 </APPLET>
```

Trong đó :

Tham số	Giải thích
CODEBASE=URL	Chỉ định địa chỉ tuyệt đối của Applet.
CODE=appletFile	Chỉ định địa chỉ tương đối của Applet.
ALT=alternateText	Chỉ định dòng text sẽ hiển thị trong trường hợp trình Browse không hiểu Applet.
NAME = appletInstanceName	Đặt tên cho Applet để phục vụ cho việc tìm kiếm.
WIDTH=pixels HEIGHT=pixels	Chỉ định kích thước cho Applet.
ALIGN=alignment	Dùng canh lề, có các giá trị sau: LEFT, RIGHT, TOP, TEXTTOP, MIDDLE, ABSMIDDLE, BASELINE, BOTTOM, ABSBOTTOM.
VSPACE=pixels HSPACE=pixels	Chỉ định khoảng trống bao chung quanh Applet.
ARCHIVE=compressed file	Khai báo các file nén cần thiết của Applet để trình Browse tải về máy cá nhân, phục vụ cho việc đọc lại sau nầy.

Ví dụ:

```
<APPLET CODEBASE=http://200.201.202.180/applets/ NervousText
CODE="NervousText.class"
WIDTH=400      HEIGHT=75
ALIGN=CENTER>
<PARAM NAME="text" VALUE="This is the Applet Viewer.">
</APPLET>
```

Chỉ thị cho trình Browse nạp Applet ở địa chỉ htttp://java.sun.com/JDK-prebeta1/applets/NervousText/NervousText.class". Chỉ định kích thước là 400x75 pixels và canh giữa dòng. Nếu trình Browse hiểu Applet, dòng "This is the Applet Viewer." sẽ hiển thị và Applet tạo hiệu ứng cho dòng chữ nầy. Nếu trình Browse không hiểu Applet, nó sẽ bỏ qua nội dung của <APPLET> cũng như <PARAM> và chỉ hiển thị nội dung của <BLOCKQUOTE>

9. ***Thẻ <IMG>***

Dùng để chèn 1 file hình vào tài liệu HTML

Các thuộc tính :

- ALIGN="left/right/top/texttop/middle/absmiddle/baseline/bottom/absbotto": So hàng hình ảnh với Text.
- ALT="Alternative Text": Cho hiển thị 1 dòng text thay thế cho file hình trong trường hợp trình Browse đang ở trong chế độ không hiển thị hình ảnh. Dòng Text nầy cũng hiển thị theo dạng ToolTip khi dời chuột đến hình.

Ví dụ: <IMG SRC="triangle.gif" ALT="Warning:"> Read these instructions.

- SRC="URL of image": Chỉ định đia chỉ file hình chèn vào trang Web.

Ví dụ : <IMG SRC="warning.gif">Be sure to read these instructions.

- WIDTH=value/ HEIGHT=value: Chỉ định khoảng cách dành sẳn cho hình trong khi trình Browse nạp toàn bộ hình.
- BORDER=value: Chỉ định cho hiển thị đường viền bao quanh hình ảnh. Ta có thể chọn "0" để hiển thị đường viền màu xanh khi có liên kết.
- VSPACE=value HSPACE=value: Quy định khoảng trống giữa hình và Text. VSPACE cho trên và dới hình, HSPACE cho trái và phải hình. Value tính theo pixel.
- LOWSRC: Thuộc tính nầy cho phép hiển thị 2 hình lần lượt trong cùng 1 vị trí. Thường dùng để nạp một hình nhỏ trong khi chờ đợi nạp hình chính có dung luợng file lớn hơn:

Ví dụ: <IMG SRC="hiquality.gif" LOWSRC="lowquality.gif">

Đầu tiên trình Browse sẽ hiển thị file hình "lowquality.gif". Sau khi nạp hoàn tất cả trang, trình duyệt sẽ nạp file hình chính thức vào thay thế.

10.7.4.2 Chỉ định tài nguyên trong URL

Để chỉ định vị trí của tài nguyên HTTP dùng URL (Uniform Resouce Locators) đó là tên quy ước để nhận diện một cách duy nhất vị trí của một thư mục

hoặc một tập tin trên Intranet/Internet. Trong URL cũng chỉ định giao thức kết nối như HTTP, GOPHER... cần thiết cho việc tìm kiếm và lấy tài nguyên. Nếu ta biết URL của một tài nguyên ta có thể truy xuất nó một cách trực tiếp hoặc thông qua các siêu liên kết trong các tài liệu.

URL sử dụng một dòng đơn các ký tự ASCII. Sơ đồ này bao gồm các giao thức trên Intranet/Internet như FTP, Gopher, http... URL là một trong những công cụ cơ sở của WWW và được dùng trong các tài liệu HTML để tham chiếu đến các tài nguyên trên mạng.

Một URL gồm các thông tin sau :

a. Tên các giao thức khi truy cập Server (như HTTP, Gopher, Wais...).

b. Tên miền của Server thực thi, theo bất cứ thông tin về user và password của site trên Intranet/Internet.

c. Số cổng mà server sử dụng. Nếu điều này không được chỉ rõ trình duyệt sẽ dùng số cổng mặc định trong giao thức (cổng 80).

d. Định vị của tài nguyên trong kiến trúc phân cấp của Server.

10.7.4.3 Giao thức HTTP

Giao thức HTTP (Hyper Text Transfer Protocol - Giao thức truyền siêu văn bản) sử dụng cho các dịch vụ truyền thông đa phương tiện WWW, dựa trên mô hình Client/Server. Dịch vụ WWW cho phép NSD kết hợp văn bản, âm thanh, hình ảnh, hoạt hình tạo nên nguồn thông tin tư liệu. Đặc biệt ở đây là thông tin tư liệu trong WWW có dạng HyperText - là dạng tư liệu chuẩn trong WWW. Giao thức cho phép lấy và đọc nhanh các tư liệu đó. HTTP là giao thức truyền thông nhưng có thêm ưu điểm là thông tin tư liệu cần truy cập lại có chứa các liên kết tới cá tư liệu khác nằm khắp nơi trên mạng Internet.

Phần mềm cho WWW Server là một chương trình điều khiển sự thu nhập các tư liệu WWW trên một máy chủ. Để truy cập WWW, cần thiết phải chạy hệ thống ứng dụng WWW là một trình duyệt (browser) trên máy của WWW Client.

HTTP là một giao thức Internet Client/Server, được thiết kế để truyền các dạng dữ liệu siêu văn bản. HTTP là một giao thức không trạng thái, nghĩa là khi Server đáp ứng dữ liệu được yêu cầu bởi Client xong thì server huỷ bỏ kết nối đó không tốn bộ nhớ cho sự kiện. Không trạng thái là yếu tố làm cho tốc độ truyền dẫn giữa HTTP Server và HTTP Client rất nhanh.

Các giao tiếp HTTP truyền dữ liệu dưới dạng các ký tự 8 bit hay một octet. Điều này đảm bảo truyền dẫn an toàn mọi dạng dữ liệu bao gồm hình ảnh, âm thanh, các tài liệu HTML hay các chương trình khả thi.

1. *Các giai đoạn kết nối của HTTP*

Một HTTP Server kết nối thông qua 4 giai đoạn:

- ***Mở kết nối:*** Client tiếp xúc với Server tại địa chỉ internet và số cổng chỉ định trong URL (cổng mặc định là 80)
- ***Tạo yêu cầu :*** Client gửi một thông điệp tới Server yêu cầu dịch vụ. Yêu cầu bao gồm các tiêu đề HTTP, nó định nghĩa phương thức được yêu cầu cho tác vụ và cung cấp thông tin về khả năng của Client (được theo sau dữ liệu gửi tới Server). Các phương thức HTTP điển hình là GET để nhận các đối tượng từ Server hoặc POST để chuyển dữ liệu cho đối tượng (ví dụ như các chương trình GateWay) trên Server.
- ***Gửi đáp ứng :*** Server trả lời cho Client bao gồm các tiêu đề để trả lời trạng mô tả trạng thái của tác vụ (ví dụ thành công, không thành công...) theo sau dữ liệu thật sự.
- ***Đóng kết nối:*** Kết nối được đóng, Server không giữ lại dấu vết của tác vụ đã hoàn thành. Thủ tục này có nghĩa là mỗi kết nối chỉ xử lý một tác vụ và do đó chỉ có thể tải xuống Client chỉ một tệp dữ liệu. Tính chất không trạng thái của tác vụ cũng có nghĩa là mỗi kết nối không hề biết về các kết nối trước đó.

2. *Các phương thức của giao thức HTTP*

Phương thức	Giải thích
GET	Lấy dữ liệu hiển thị trong URL. Dữ liệu cũng có thể gửi trong URL thông qua một chuỗi truy vấn. Đây cũng là nơi dữ liệu gửi từ ISINDEX hoặc Form với thuộc tính METHOD="GET"
HEAD	Lấy thông tin của HTTP, Header chỉ định trong URL.
POST	Gửi dữ liệu đến cho URL nếu URL là tồn tại. Phương thức này được dùng bởi những thành phần của Form trong HTML với giá trị thuộc tính METHOD="POST".
PUT	Là nơi mà dữ liệu gửi bởi Client biểu thị trong URL, nó sẽ thay thế nội dung của URL đã có.
DELETE	Xóa tài nguyên cục bộ tại nơi được chỉ định bởi URL.
LINK	Liên kết một đối tượng đã tồn tại với một đối tượng khác.
UNLINK	Huỷ bỏ một liên kết đã được tạo bởi phương thức LINK.

BÀI TẬP

1. Những nguyên tắc cơ bản giám sát và quản trị hệ thống mạng máy tính
2. Khảo sát cấu trúc và hoạt động dịch vụ DNS
3. Khảo sát cấu trúc và hoạt động của giao thức SNMP
4. Khảo sát cấu trúc và hoạt động của giao thức HTTP
5. Tìm hiểu giao thức DHCP.

- -

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Nguyễn Thúc Hải, ***Mạng máy tính và các hệ thống mở*** , NXB Giáo dục, 1997

[2] Lê Văn Sơn, ***Giáo trình mạng máy tính***, Trường ĐH Bách Khoa Đà Nẵng, 1998

[3] Nguyễn Hồng Sơn, ***Giáo trình hệ thống mạng máy tính CCNA,*** Nhà XB Lao động, 2002

Tiếng Anh

[4] Douglas E.Comer, ***Computer Networks and Internets***, Prentice Hall, 1997

[5] Ed Taylor, ***TCP/IP complete***, McGraw-Hill, 1998

[6] Microsoft Press, ***Networking Essentials***

[7] Stallings W., ***Data and Computer Communications***, Macmillan Publishing, 1995

[8] Tanenbaum Andrew S., ***Computer*** **Networks,** Prentice Hall, 1997

[9] Pujolle, ***Les rĐseaux***, EYROLLES, 2003

@2004, Nguyễn Tấn Khôi

Khoa Công Nghệ Thông Tin - Trường Đại học Bách Khoa Đà Nẵng

----------o & o----------