HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

BÀI GIẢNG

MẠNG MÁY TÍNH

(Dành cho sinh viên hệ Đại học chính qui
chuyên ngành Công nghệ thông tin)

Người biên soạn: ThS. Nguyễn Xuân Anh

HÀ NỘI – 11/2012

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU

1.1 Mạng máy tính và mạng Internet

Mạng máy tính là một hệ thống bao gồm các máy tính được nối kết với nhau để trao đổi thông tin. Việc kết nối các máy tính với nhau nhằm mục đích sau:

- Chia sẻ phần cứng: người sử dụng có thể dùng chung các thiết bị phần cứng như máy in, máy vẽ. Cao hơn nữa, người dùng có thể tận dụng năng lực xử lý của các máy tính khác.
- Chia sẻ dữ liệu: Dữ liệu được quản lý tập trung, như vậy sẽ đảm bảo an toàn và toàn vẹn dữ liệu.
- Trao đổi thông tin: việc trao đổi thông tin như thư điện tử, đăng tin lên các trang thông tin điện tử một cách dễ dàng, nhanh chóng và tiện lợi.

Nếu nhiều mạng máy tính với nhau gọi là kết nối liên mạng (internet), việc kết liên mạng trên phạm vi toàn cầu đã hình thành nên mạng Internet. Sự phát triển của mạng Internet đã vượt xa những dự đoán của những người sáng lập, nó đã làm thay đổi lối sống của nhân loại.

1.2 Phân loại mạng máy tính theo phạm vi địa lý

Mạng máy tính có thể được đặt trên một khu vực nhất định, ví dụ: trong một căn phòng, một tòa nhà, một quốc gia hay trên phạm vi toàn cầu. Dựa vào phạm vi phân bổ của các máy tính trong mạng người ta có thể phân ra các loại mạng, trong đó khái niệm mạng cục bộ *(LAN - Local Area Network)* và mạng diện rộng (*WAN - Wide Area Network*) thường hay được nhắc tới. Mạng cục bộ kết nối các máy tính trong một khu vực bán kính hẹp, thông thường dưới 1 Km, băng thông tương đối lớn, thường được sử dụng trong nội bộ một gia đình, cơ quan…. Mạng diện rộng kết nối các máy tính có phạm vi lớn hơn 1 Km và thường được lắp đặt dựa trên nền tảng mạng viễn thông. Mạng cục bộ thường có băng thông lớn được thiết kế để kết nối các máy tính trong một khu vực địa lý nhỏ như ở một tầng của toà nhà, hoặc trong một toà nhà.... Mạng cục bộ cho phép dùng chung những thiết bị ngoại vi như máy in, máy chiếu… và thậm chí có thể chia sẻ các tài nguyên trên mỗi máy tính như ổ đĩa, phần mềm, tài nguyên dữ liệu.

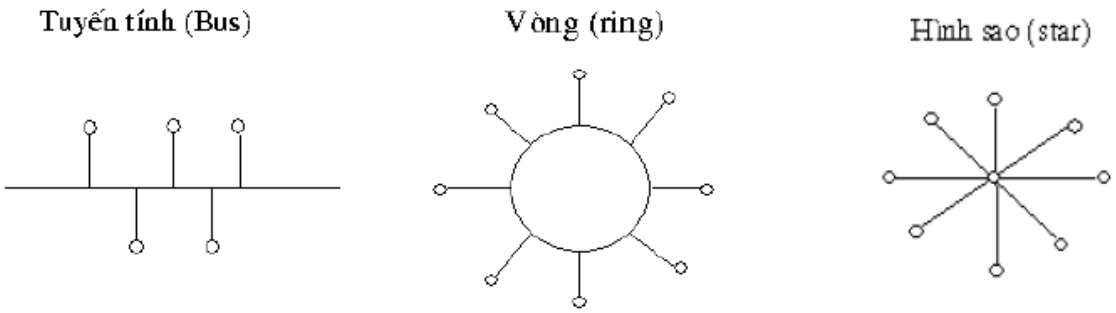
1.3 Hình trạng mạng

Hình trạng mạng (*Network Topology)* là cấu trúc máy tính liên kết các máy tính với nhau, cần phải phân biệt hình trạng vật lý và hình trạng logic. Hình trạng vật lý trả lời cho câu hỏi các máy tính được nối với nhau như thế nào (phản ánh cấu trúc hình học của mạng). Hình trạng logic trả lời cho câu hỏi các máy tính trao đổi thông tin với nhau như thế nào (mạng vận hành theo nguyên tắc nào).

1.3.1 Hình trạng vật lý

Hình trạng vật lý có 3 dạng cấu trúc cơ bản là: dạng tuyến (*Bus Topology*), dạng vòng (*Ring Topology*) và dạng hình sao (*Star Topology*). Từ ba dạng cấu

trúc cơ bản trên sẽ tạo lập các hình thạng mạng khác như: dạng hình cây (*Tree*), dạng hỗn hợp (*Mesh*),v.v....



Hình 1.1 Các hình trạng vật lý

Dạng tuyến:

Theo cách bố trí hành lang, tất cả các máy tính trong mạng đều được nối với nhau trên một trục đường dây cáp chính. Hai đầu mút của dây cáp được bịt bởi một thiết bị gọi là kết cuối (Terminator), kết cuối có tác dụng giữ cho các tín hiệu di chuyển trên dây giảm được suy hao. Loại dạng này dùng ít dây cáp nhất, dễ lắp đặt. Tuy nhiên, khi có sự hỏng hóc ở đoạn nào đó thì rất khó phát hiện, chỉ cần một điểm trên đường dây bị đứt sẽ ngừng hoạt động của toàn bộ mạng.

Dạng vòng:

Đường dây cáp được thiết kế làm thành một vòng khép kín, tín hiệu chạy quanh theo một chiều nào đó. Mạng dạng vòng tiết kiệm dây dẫn nhưng đường dây phải khép kín, nếu bị ngắt ở một nơi nào đó thì toàn bộ mạng sẽ ngừng hoạt động.

Dạng hình sao:

Dạng hình sao bao gồm một điểm trung tâm, các máy tính trao đổi thông tin với nhau đều phải chuyển qua trung tâm này. Dạng hình sao có ưu điểm sau:

- 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.
- Dễ dang mở rộng qui mô mạng.
- Nhược điểm của mạng hình sao:
- 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 thường dưới 100m.

1.3.2 Hình trạng logic

Hình trạng logic được phân thành hai loại: Quảng bá và thẻ bài.

Quảng bá:

Trong hình trạng quảng bá, kênh truyền được chia sẻ cho tất cả các máy tính. Khi một máy tính gởi tin, tất cả các máy tính còn lại sẽ nhận được tin đó. Tại một thời điểm chỉ cho phép một máy tính được phép sử dụng đường truyền.

Thẻ bài:

Hình trạng này không có nút điều phối, một bản tin đặc biệt được gọi là thẻ bài (token) được trao đổi giữa các nút theo một thứ tự định trước. Ví dụ, nút thứ nhất gửi thẻ bài tới nút thứ hai, nút thứ hai gửi thẻ bài tới nút thứ ba. . . nút thứ N gửi thẻ bài tới nút thứ nhất. Khi nút nhận được thẻ bài, nó chỉ giữ thẻ bài khi có dữ liệu cần truyền, nếu không nó sẽ ngay lập tức chuyển thẻ bài tới nút kế tiếp. Nếu nút có dữ liệu cần truyền, khi nhận được thẻ bài, nó gửi đi lượng dữ liệu được phép và sau đó chuyển thẻ bài tới nút kế tiếp.

1.3.3 Kết nối với mạng Internet

Cùng với sự phát triển của mạng Internet, người ta đã có nhiều cách thức để kết nối vào Internet. Mỗi cách có ưu điểm và nhược điểm riêng, tuỳ thuộc vào phần cứng, phần mềm và chi phí phải trả. Thực tế, chúng ta có thể gộp chung thành 3 loại hình dịch vụ kết nối cơ bản sau:

- Kết nối trực tiếp, cố định
- Kết nối trực tiếp, không cố định
- Kết nối gián tiếp

Kết nối trực tiếp, cố định:

Đây là các loại kết nối mà người sử dụng có thể truy cập vào Internet vào bất cứ lúc nào mình muốn. Máy tính sẽ được cung cấp cho một địa chỉ tĩnh và không bị không thay đổi trong một thời gian dài. Tốc độ là ưu điểm lớn nhất của loại hình này vì máy tính được kết nối sử dụng băng thông rộng. Chúng ta có thể thấy kết nối qua modem cáp (cable modem), ISDN ... là những ví dụ điển hình về loại kết nối này. Thông thường đây là loại hình kết nối đắt tiền, cả về giá cước cũng như thiết bị để kết nối.

Kết nối trực tiếp, không cố định:

Mỗi lần kết nối, máy tính sẽ được cấp cho một địa chỉ để phục vụ cho phiên làm việc, địa chỉ này chỉ tồn tại trong thời gian kết nối. Loại kết nối này thường dùng trong mạng điện thoại công cộng. Ưu điểm là giá thành tương đối thấp, tuy nhiên hạn chế của loại kết nối này là tốc độ, đơn giản vì dữ liệu được truyền chung với tín hiệu thoại trên cáp đồng. Nếu kết nối qua đường dây điện thoại thì tốc độ hạn chế ở 56 Kbps.

Kết nối gián tiếp, không cố định:

Đây là kết nối Internet mà máy tính của người dùng không kết nối một cách trực tiếp vào mạng, mà nó được kết nối vào một máy tính khác đang thực sự nối Internet. Cách này thường thấy ở các phòng dịch vụ Internet công cộng. Tốc độ cũng tuỳ thuộc vào loại kết nối Internet mà máy chủ đang có cũng như số máy tính khách đang kết nối vào máy chủ. Hơn nữa, loại hình này có thể không cung

cấp đầy đủ các chức năng cho máy khách, tất cả đều tuỳ thuộc vào chính sách bảo mật được thiết lập trên máy chủ.

Căn cứ vào nhu cầu sử dụng, người dùng có thể lựa chọn các phương tiện kết nối sau:

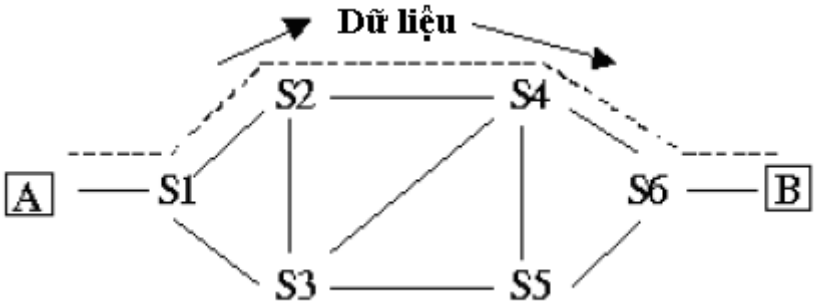
- **Kết nói qua mạng điện thoại công cộng (PSTN)**: Dùng modem quay số, tốc độ chậm, chất lượng không tốt, tuy nhiên đây là mạng bao phủ rộng lớn, kể cả các vùng hẻo lánh.
- **Đường thuê bao (leased line)**. Thuê đường dây riêng của công ty viễn thông.
- **Mạng dịch vụ tích hợp số (ISDN - Integrated Service Digital Nework).** Sử dụng đường điện thoại số thay vì đường tương tự.
- **Frame relay:** Phù hợp với các dịch vụ truyền số liệu.
- **Chế độ truyền không đồng bộ (ATM - Asynchoronous Trangfer Mode)**" ATM thích hợp các dịch vụ đòi hỏi băng thông rộng.
- **Đường vệ tinh (satellite links):** Giá thành đắt, chỉ phù hợp với những khu vực khó triển khai kênh truyền băng thông rộng bằng các đường dây hữu tuyến
- **Điện thoại di động:** Hình thức này đang ngày càng phổ biến, đặc biệt khi các công ty thông tin di động triển khai mạng 3G và 4G.

CHƯƠNG 2: KIẾN TRÚC VÀ HIỆU NĂNG MẠNG

2.1 Chuyển mạch kênh và chuyển mạch gói

Chuyển mạch kênh:

Khi có hai đối tượng cần trao đổi thông tin với nhau thì giữa chúng sẽ thiết lập một kênh cố định và được duy trì cho đến khi một trong hai bên ngắt liên lạc, dữ liệu chỉ được truyền theo con đường cố định đó. Chuyển mạch kênh hoạt động theo mô hình của hệ thống điện thoại công cộng. Ví dụ, hai máy A và B cần phải trao đổi thông tin với nhau. Để có thể giao tiếp với máy B, máy A phải thực hiện một cuộc gọi. Nếu máy B chấp nhận cuộc gọi thì một kênh truyền được thiết lập dành riêng cho việc trao đổi thông tin giữa máy A và máy B. Tất cả các tài nguyên được cấp cho cuộc gọi này như băng thông đường truyền, khả năng của các bộ chuyển đổi thông tin đều được dành riêng cho cuộc gọi, không chia sẻ cho bất kỳ cuộc gọi nào khác ngay cả khi máy A và B không gửi thông tin cho nhau.



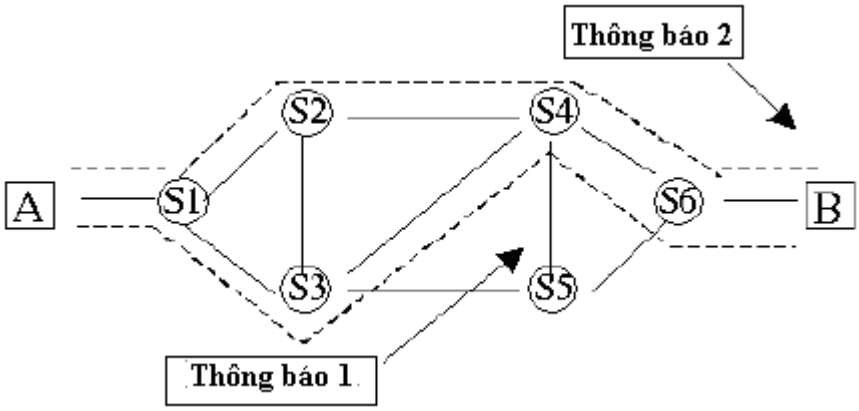
Hình 2.1 Mạng chuyển mạch kênh

Băng thông sẽ được chia thành nhiều phần bằng nhau và sẽ gán cho các cuộc gọi. Khi một cuộc gọi sở hữu phần băng thông đó, mặc dù không sử dụng đến hoặc không sử dụng hết nó cũng không chia sẻ băng thông này cho các cuộc gọi khác. Việc phân chia băng thông của kênh truyền có thể thực hiện kỹ thuật phân chia theo tần số (FDMA-Frequency Division Multi Access) hay phân chia theo thời gian (TDMA- Time Division Multi Access).

Chuyển mạch kênh có hiệu suất không cao do phải mất thời gian để thiết lập kênh truyền, hiệu suất sử dụng kênh truyền thấp vì có những thời điểm kênh truyền đã được thiết lập nhưng lại không được sử dụng hoặc sử dụng rất ít, trong khi đó các thực thể khác có nhu cầu truyền dữ liệu vẫn phải nằm trong hàng đợi.

Chuyển mạch thông báo:

Thông báo là một đơn vị thông tin với khuôn dạng nhất định, mỗi thông báo gồm hai phần: Phần thông tin điều khiển và phần nội dung cần chuyển, phần thông tin điều khiển phải chỉ định rõ đích đến của thông báo. Căn cứ vào thông tin điều khiển, mỗi nút trung gian sẽ quyết định chuyển thông báo tới nút kế tiếp, đường đi của các thông báo sẽ không cố định.

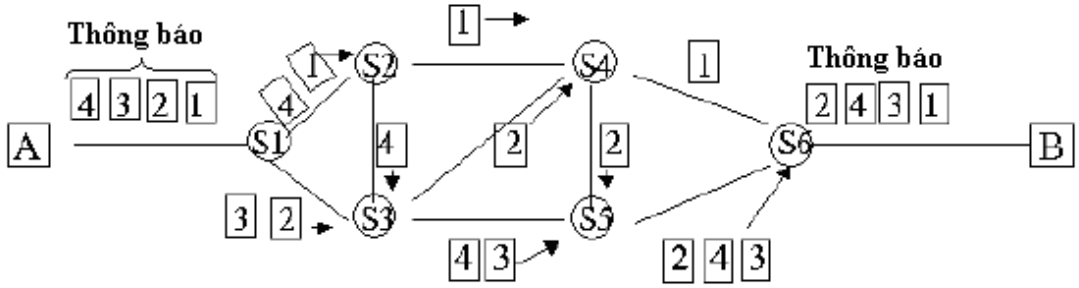


Hình 2.2 Chuyển mạch thông báo

So với chuyển mạch kênh, hiệu suất sử dụng đường truyền của chuyển mạch gói cao hơn, cơ chế truyền tin linh hoạt hơn vì có thể đặt ưu tiên cho từng thông báo. Tuy nhiên, do không qui định độ lớn của mỗi bản tin nên khó qui định thống nhất thời gian đáp ứng của mỗi thông báo, khi có lỗi xảy ra thì phải truyền lại toàn bộ thông báo đó.

Chuyển mạch gói:

Trong phương pháp chuyển mạch gói, thông tin trao đổi giữa hai máy tính được phân thành những gói tin có kích thước tối đa xác định. Gói tin của những người dùng khác nhau sẽ chia sẻ nhau băng thông của kênh truyền. Nếu lượng thông tin cần truyền đi vượt quá khả năng đáp ứng của kênh truyền thì sẽ xảy ra trường hợp mỗi gói tin chiếm dụng toàn bộ băng thông của kênh truyền. Trong trường hợp này, các thiết bị định tuyến sẽ lưu lại các gói tin chưa gửi vào hàng đợi chờ cho đến khi kênh truyền rỗi sẽ lần lượt gửi đi.



Hình 2.3 Chuyển mạch gói

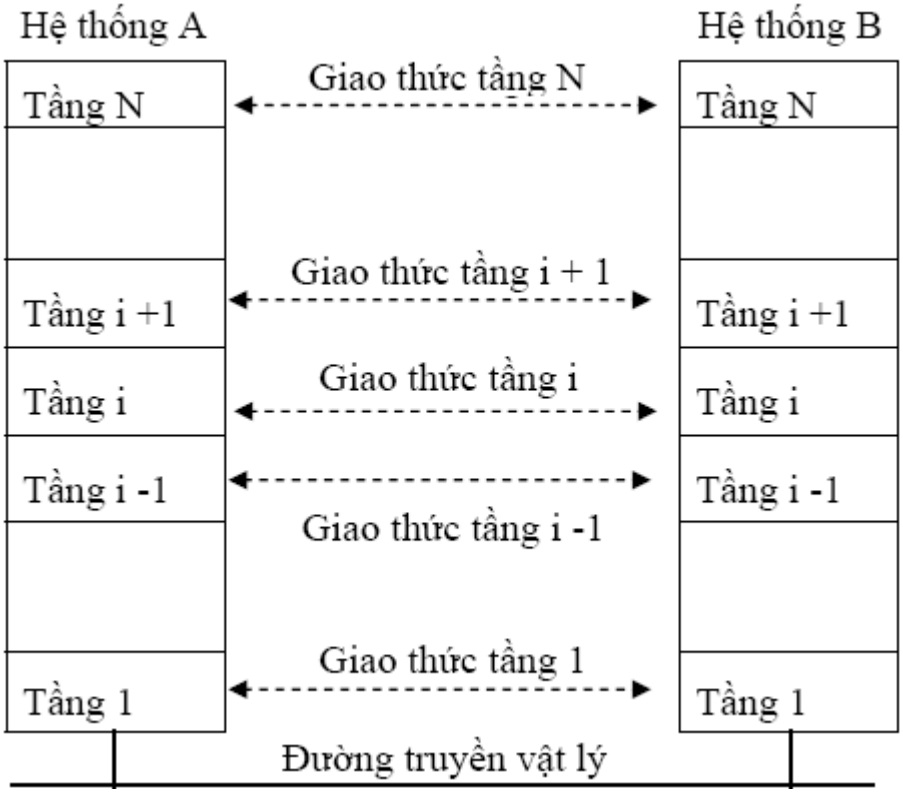
Phương pháp chuyển mạch gói cho phép tận dụng kênh truyền tốt hơn, do đó có thể đáp ứng nhiều người sử dụng hơn mà không cần phải nâng cấp hệ thống phần cứng. Ví dụ: giả sử một đường truyền có tốc độ 2 Mbps, mỗi người dùng được cấp băng thông 100 Kbps và chỉ hoạt động tối đa 10% tổng thời gian. Nếu dùng phương pháp chuyển mạch kênh sẽ chỉ đáp ứng tối đa 20 người sử dụng, trong khi đó nếu dùng phương pháp chuyển mạch gói thì có thể đáp ứng cho khoảng 200 người sử dụng. Chuyển gói thích hợp cho dịch vụ truyền dữ liệu lớn nhưng cần phải có cơ chế điều khiển tắt nghẽn và mất mát dữ liệu. Thông

lượng phụ thuộc vào số lượng người dùng đồng thời nên một số ứng dụng về âm thanh và hình ảnh sẽ có chất lượng không ổn định.

2.2 Phân tầng và chức năng của các tầng

2.2.1 Kiến trúc phân tầng

Để giảm độ phức tạp của việc thiết kế và cài đặt mạng, người ta thường lựa chọn cách thiết kế theo quan điểm phân tầng. Mỗi hệ thống thành phần của mạng được xem như một cấu trúc đa tầng, trong đó mỗi tầng được xây dựng trên tầng dưới nó. Số lượng các tầng cũng như tên và chức năng của mỗi tầng tuỳ thuộc vào ý tưởng của người thiết kế. Mục đích của việc phân tầng là để chuyên môn hóa các chức năng dịch vụ. Mỗi tầng khi sử dụng dịch vụ không cần quan tâm đến cách thực hiện của các tầng dưới.



Hình 2.4 Kiến trúc phân tầng

Việc thiết kế phân tầng phải bảo đảm các nguyên tắc sau:

- Trong một mạng số lượng tầng và chức năng/nhiệm vụ của mỗi tầng phải như nhau.
- Dữ liệu không được truyền trực tiếp từ tầng thứ i của hệ thống này sang tầng thứ i của hệ thống kia, ngoại trừ đối với tầng thấp nhất. Bên gửi phải chuyển dữ liệu đến tầng dưới nó, đến tầng thấp nhất sẽ chuyển cho tầng thấp nhất của bên nhận và dữ liệu lại được chuyển tiếp cho tầng cao hơn.
- Liên kết giữa hai tầng thấp nhất gọi là liên kết vật lý, liên kết của tất cả các tầng cao hơn gọi là liên kết logic.

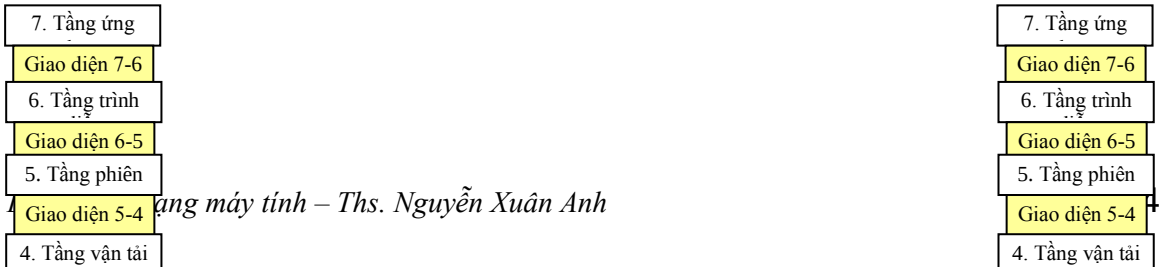
- Giao tiếp giữa hai tầng liền kề gọi là giao diện, chúng trao đổi dữ liệu với nhau qua các điểm truy nhập dịch vụ.
- Các tầng tương ứng giao tiếp với nhau dựa trên các qui tắc nhất định gọi là giao thức, mỗi tầng có thể có nhiều giao thức.
- Giao thức (*Protocol*) là tập các tiêu chuẩn để trao đổi thông tin giữa hai hệ thống máy tính hoặc hai thiết bị máy tính với nhau. Mỗi giao thức phải qui định đơn vị dữ liệu của giao thức (PDU – Protocol Data Unit) và tập các qui tắc để trao đổi thông tin: Gửi/nhận, qui định tốc độ, phương pháp truyền (một hướng, hai hướng hay hai hướng luân phiên).
- Dữ liệu của tầng trên khi chuyển qua tầng dưới có thể sẽ được tách thành những đơn vị dữ liệu nhỏ hơn và đồng thời được thêm các thông tin điều khiển để phù hợp với giao thức truyền tin của tầng dưới. Bên nhận, mỗi tầng tương ứng sẽ bóc tách thông tin điều khiển và tập hợp các đơn vị dữ liệu để chuyển lên tầng cao hơn.

2.2.2 Mô hình OSI

ISO là tên viết tắt của Tổ chức Quốc tế về tiêu chuẩn hoá (International Organization for Standardization), được thành lập năm 1946 và chính thức hoạt động vào ngày 23/2/1947, nhằm mục đích xây dựng các tiêu chuẩn về sản xuất, thương mại và thông tin. Một trong các chuẩn ISO về truyền thông mạng là mô hình liên kết giữa các hệ thống mở (OSI - Open Systems lnterconnection Model). Đây là mô hình để các hệ thống khác nhau có thể trao đổi thông tin với nhau mà không cần quan tâm đến kiến trúc bên trong của chúng. Ban đầu, mỗi hãng tự thiết kế các giao thức riêng nhằm tạo thế độc quyền cho các sản phẩm mạng của mình. Mô hình OSI ra đời nhằm mục đích cho phép hai hệ thống bất kì trao đổi thông tin với nhau mà không cần thay đổi bất cứ phần cứng hoặc phần mềm nào của mỗi hãng sản xuất.

Mô hình OSI được phân tầng với mục đích thiết kế các hệ thống mạng cho phép việc truyền thông thực hiện được qua tất cả các kiểu hệ thống máy tính khác nhau. Các tầng được thiết kế riêng biệt nhưng liên quan chặt chẽ với nhau, mỗi tầng định nghĩa một phần của quá trình truyền thông tin trên mạng. Nắm vững những quy tắc cơ bản của mô hình OSI là tiền đề vững chắc để thiết kế và phát triển các hệ thống thông tin trên mạng. Mô hình OSI gồm 7 tầng sau:

- Tầng ứng dụng. (Application layer).
- Tầng trình diễn (Presentation layer)
- Tầng phiên (Session layer)
- Tầng vận tải (Transport layer)
- Tầng mạng (Network layer)
- Tầng liên kết dữ liệu (Datalink layer)
- Tầng vật lý (Physical layer)



Hình 2.5 Mô hình phân tầng OSI

Hình 2.5 minh họa mối quan hệ giữa các tầng khi thông tin của người sử dụng được gửi từ máy tính PC1 đến máy tính PC2. Người sử dụng trên máy tính PC1 gửi tin cho người sử dụng tại máy tính PC2, thông tin sẽ lần lượt được chuyển từ tầng ứng dụng xuống tầng vật lý của máy tính PC1, sau đó đi qua nhiều nút trung gian khác (gọi là thiết bị mạng, những nút trung gian này thường chỉ liên quan đến tầng vật lý, tầng liên kết dữ liệu và tầng mạng của mô hình OSI) trước khi đến máy tính PC2. Tại máy tính PC2, thông tin sẽ lại lần lượt được chuyển tiếp từ lớp vật lý đến lớp ứng dụng.

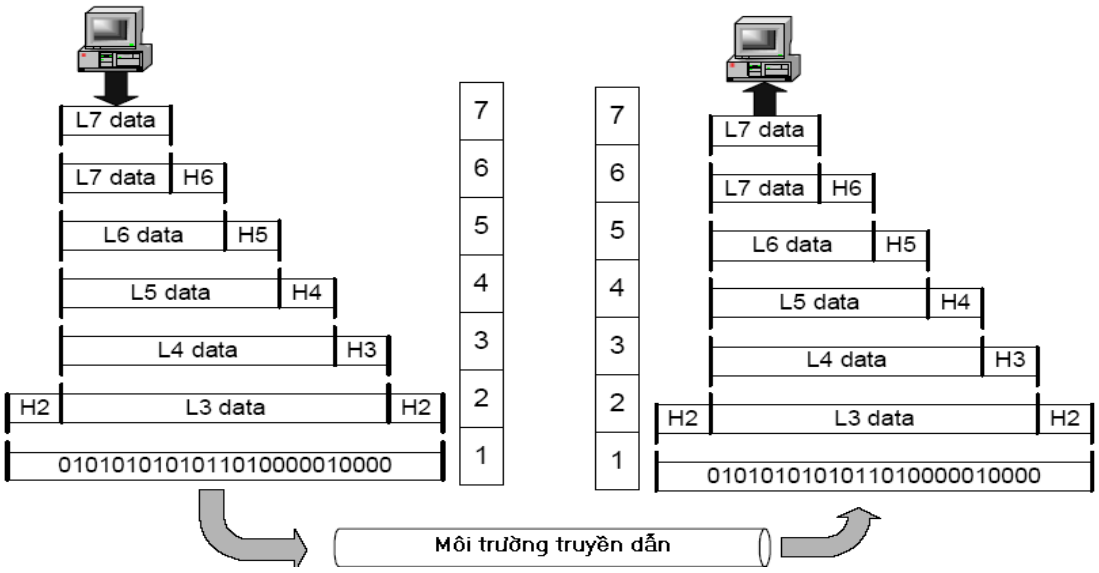
Khi xây dựng mô hình, các nhà thiết kế đã phân tích quá trình truyền dữ liệu thành những chức năng cơ bản nhất. Những chức năng nào có mục đích sử dụng liên quan đến nhau được gộp thành từng nhóm và gọi là tầng trong mô hình tham chiếu OSI. Như vậy, mỗi tầng được xác định chức năng và nhiệm vụ riêng biệt. Với cách thiết kế như vậy, mô hình tham chiếu OSI khá toàn diện và linh hoạt, đồng thời đảm bảo tính trong suốt giữa các hệ thống.

2.2.2.1 Các tiến trình ngang hàng

Trong mỗi máy, mỗi tầng sử dụng các dịch vụ do tầng bên dưới cung cấp. Ví dụ, tầng 3 sử dụng các dịch vụ do tầng 2 cung cấp và đến lượt mình lại cung cấp dịch vụ cho tầng 4. Giữa các máy tính, tầng x trên một thiết bị giao tiếp với tầng x trên thiết bị khác. Việc giao tiếp này được tiến hành theo các quy tắc và quy ước đã được thỏa thuận trước - gọi là giao thức.

Tại tầng vật lý, việc truyền thông là trực tiếp: Máy PC1 gửi một luồng bit đến máy tính PC2. Tại các tầng cao hơn trên máy PC1, dữ liệu được chuyển gần xuống các tầng bên dưới, đến máy PC2 và tiếp tục đi lên các tầng cao hơn (trong máy PC2). Mỗi tầng trong máy gửi dữ liệu đi (máy PC1) thêm các thông tin của tầng đó vào bản tin nhận được từ phía trên rồi sau đó chuyển toàn bộ gói dữ liệu xuống tầng phía dưới. Các thông tin được thêm vào này gọi là thông tin điều khiển, nếu thêm vào trước gọi là tiêu đề chèn trước (header), các thông tin điều

khiển được thêm vào cuối gọi là thông tin chèn sau (Trailer). Thông tin chèn trước được thêm vào bản tin tại các tầng trình diễn, vận tải, phiên, mạng. Tầng liên kết dữ liệu thêm cả thông tin trước lẫn sau bản tin.



Hình 2.6 Thêm và tách thông tin điều khiển tại các tầng

Tại tầng vật lý, toàn bộ dữ liệu được chuyển thành dạng phù hợp với môi trường truyền dẫn nhằm bảo đảm thông tin có thể đến được máy nhận. Tại bên nhận, các tiêu đề được bóc tách dần dần khi chuyển dữ liệu từ tầng thấp lên tầng cao. Ví dụ, tầng liên kết dữ liệu loại bỏ các thông tin điều khiển của nó, kết quả sau khi bóc tách sẽ là dữ liệu của tầng mạng và được chuyển lên tầng trên (tầng mạng). Tương tự như vậy tầng mạng loại bỏ thông tin điều khiển sẽ được dữ liệu của tầng vận tải và dữ liệu đó sẽ lại chuyển cho tầng trên (tầng vận tải), quá trình tương tự được thực hiện cho đến tầng ứng dụng.

2.2.2.2 Giao diện giữa các tầng

Trên cùng một máy tính, hai tầng kề nhau trao đổi dữ liệu với nhau qua các giao diện, tầng trên yêu cầu dịch vụ của tầng dưới thông qua giao diện gọi là điểm truy cập dịch vụ (SAP- Service Access Point). Tại mỗi điểm truy nhập dịch vụ người ta qui định phương pháp và khuôn dạng dữ liệu trao đổi giữa hai tầng kề nhau trên cùng một thiết bị. Định nghĩa giao diện giữa các tầng một cách rõ ràng sẽ cho phép thay đổi nghiệp vụ tại một tầng mà không ảnh hưởng đến các tầng khác.

2.2.2.3 Tổ chức các tầng

Có thể chia bảy tầng có thể thành ba nhóm: Nhóm hỗ trợ mạng, nhóm hỗ trợ người sử dụng, nhóm trung gian. Nhóm hỗ trợ mạng bao gồm ba tầng thấp của mô hình OSI (Tầng vật lý, liên kết dữ liệu và mạng), chúng đảm nhiệm về các vấn đề liên quan đến mặt vật lý khi truyền dữ liệu từ một thiết bị này đến một thiết bị khác (ví dụ: những đặc tả về điện, các kết nối vật lý, định địa chỉ vật lý, định thời gian truyền…). Nhóm hỗ trợ người sử dụng bao gồm ba tầng cao nhất của mô hình OSI (Tầng phiên, trình diễn, ứng dụng), chúng cung cấp

các tính năng tương tác giữa các hệ thống phần mềm tách biệt. Nhóm trung gian gồm các tầng nằm giữa hai nhóm trênm cụ thể trong mô hình OSI đó là tầng vận tảim nó đảm bảo việc chuyển dữ liệu tin cậy giữa các thiết bị đầu cuối. Nói chung, các tầng trên của mô hình OSI thường được thực hiện bởi phần mềm trong khi nhóm các tầng dưới được được triển khai dưới sự kết hợp của cả phần cứng và phần mềm, tầng vật lý hầu như được triển khai bởi phần cứng. Hình 2.5 mô tả tổng quan các tầng trong mô hình OSI, mỗi tầng đều định nghĩa đơn vị dữ liệu (PDU - Protocol Data Unit) của giao thức trong tầng đó. Quá trình trao đổi thông tin được bắt đầu tại tầng ứng dụng, sau đó chuyển xuống các tầng dưới. Tại mỗi tầng, ngoại trừ tầng ứng dụng và tầng vật lý, thông tin điều khiển sẽ được thêm vào đơn vị dữ liệu. Khi đơn vị dữ liệu chuyển đến tầng vật lý, chúng được chuyển thành tín hiệu điện từ và truyền đi trên đường truyền vật lý. Đến trạm nhận, tín hiệu điện từ đi đến tầng vật lý và được chuyển ngược lại thành chuỗi các bít. Các đơn vị dữ liệu sau đó sẽ được chuyển từ tầng vật lý lên các tầng trên trong mô hình OSI. Khi đi qua mỗi tầng, các thông tin điều khiển sẽ bị loại bỏ và đến tầng ứng dụng sẽ được bản tin giống như bản tin gốc tại tầng ứng dụng của bên gửi.

2.2.3 Chức năng các tầng trong mô hình OSI

2.2.3.1 Tầng vật lý

Tầng vật lý thực hiện các chức năng cần thiết để truyền luồng bit dữ liệu đi qua các môi trường truyền đẫn, nó giải quyết các vấn đề liên quan đến đặc điểm kỹ thuật về cơ và điện của giữa giao diện của thiết bị mạng với môi trường truyền dẫn. Để thực hiện vai trò này, lớp vật lý cần phải xác định các thủ tục và các chức năng mà các thiết bị vật lý và thiết bị giao tiếp cần phải tuân thủ. Tầng vật lý thực hiện các chức năng sau:

- **Đảm bảo giao tiếp với môi trường truyền dẫn:** Tầng vật lý xác định các đặc tính giao diện giữa các thiết bị mạng và môi trường truyền dẫn.
- **Biểu diễn dữ liệu dưới dạng bit:** dữ liệu tầng vật lý là luồng bit liên tục 0 và l. Để truyền đi, các bit phải được mã hóa thành các tín hiệu điện, quang hoặc tần số.
- **Tốc độ truyền dẫn**: Qui định số lượng bit được gửi đi trong một đơn vị thời gian và khoảng thời gian để truyền đi một bit.
- **Đồng bộ:** Máy gửi và nhận phải được đồng bộ hóa ở mức bit.
- **Quản lý kênh truyền**: Tầng vật lý liên quan đến việc kết nối các thiết bị vào môi trường truyền thông. Trong cấu hình điểm-điểm, hai thiết bị được nối với nhau qua một đường truyền dành riêng. Trong cấu hình điểm-nhiều điểm, một đường truyền được nhiều thiết bị dùng chung.
- **Hình trạng vật lý**: Hình trạng vật lý xác định cách nối các thiết bị với nhau để tạo thành mạng. Có ba hình trạng cơ bản: dạng bus, dạng vòng và dạng sao.
- **Chế độ truyền dẫn**: Tầng vật lý cũng xác định hướng truyền dữ liệu giữa hai thiết bị: đơn công (simplex), bán song công (haif-duplex) hay song công

(full-duplex). Trong chế độ đơn công, một thiết bị chỉ có thể gửi hoặc nhận dữ liệu. Chế độ đơn công là truyền thông một chiều. Trong chế độ bán song công, một thiết bị có thể gửi và nhận dữ liệu, nhưng không phải tại cùng một thời điểm. Trong chế độ song công, một thiết bị có thể nhận và gửi dữ liệu tại cùng một thời điểm.

2.2.3.2 Tầng liên kết dữ liệu

Tầng liên kết dữ liệu đảm bảo truyền tin tin cậy giữa hai thiết bị vật lý kết nối trực tiếp với nhau, dữ liệu tại tầng này gọi là khung (Frame). Tầng liên kết dữ liệu đảm nhiệm các chức năng sau:

- **Tạo khung dữ liệu**: Tầng liên kết dữ liệu chia gói tin nhận được từ tầng mạng thành các đơn vị dữ liệu gọi là các khung dữ liệu.
- **Quản lý địa chỉ vật lý**: Tầng liên kết dữ liệu phải xác định, gói tin cần chuyển có đích là thiết bị trong mạng nội bộ hay mạng khác. Nếu gói dữ liệu được chuyển đến thiết bị khác trong mạng nội bộ, nó thêm địa chỉ vật lý của thiết bị đích vào khung dữ liệu. Nếu gói tin cần chuyển ra ngoài mạng nội bộ, nó thêm địa chỉ vật lý của cổng mặc định.
- **Kiểm soát lưu lượng**: Nếu tốc độ nhận dữ liệu nhỏ hơn tốc độ gửi dữ liệu, tầng liên kết dữ liệu phải thực hiện một kỹ thuật kiểm soát lưu lượng để ngăn ngừa tình trạng quá tải tại nơi nhận.
- **Kiểm soát lỗi**: Tầng liên kết dữ liệu làm tăng tính tin cậy cho tầng vật lý bằng cách sử dụng một kỹ thuật phát hiện và truyền lại các khung bị lỗi hoặc bị mất. Nó cũng sử dụng kỹ thuật ngăn ngừa hiện tượng lặp khung. Kiểm soát lỗi thường được thực hiện bằng cách thêm một thông tin điều khiển vào phần cuối của khung, thông thường người ta sử dụng kỹ thuật kiểm tra vòng (CRC – Cyclic Redandunce Check).
- **Kiểm soát truy cập**: Khi nhiều thiết bị được nối với cùng một đường truyền, các giao thức ở tầng liên kết dữ liệu cần xác định xem thiết bị nào được quyền sử dụng đường truyền tại một thời điểm xác định.

2.2.3.3 Tầng mạng

Tầng mạng chịu trách nhiệm chuyển dữ liệu (dữ liệu của tầng mạng gọi là gói tin – packet) giữa các thiết bị đầu cuối của người sử dụng. Nếu như tầng vận tải đảm bảo liên kết đầu cuối tới mức tiến trình thì tầng mạng chỉ đảm bảo liên kết ở mức đầu cuối của người sử dụng. Theo định nghĩa ban đầu, tầng mạng giải quyết các vấn đề dẫn các gói tin qua một mạng. Một số ví dụ về các giao thức như vậy là X.25, và giao thức Host/IMP của mạng ARPANET. Với sự xuất hiện của khái niệm liên mạng, các chức năng mới đã được bổ sung cho tầng này, đó là chức năng dẫn đường cho dữ liệu từ mạng nguồn đến mạng đích. Nhiệm vụ này thường đòi hỏi việc định tuyến cho gói tin quan một mạng lưới của các mạng máy tính, đó là liên mạng. Tầng mạng đảm nhiệm các chức năng sau:

- **Quản lý địa chỉ logic**: Địa chỉ vật lý của tầng liên kết dữ liệu đã đảm bảo tính duy nhất trong toàn mạng, tuy nhiên nó chỉ giải quyết được vấn đề định địa chỉ cục bộ. Nếu gói dữ liệu được chuyển đến một mạng khác, cần phải có

một hệ thống địa chỉ khác nhằm phân biệt được hệ thống gửi và hệ thống nhận. Tầng mạng bổ sung thêm thông tin điều khiển vào mỗi gói dữ liệu gửi đi, trong đó chứa địa chỉ logic của thiết bị nhận và thiết bị gửi.

- **Định tuyến**: Khi các mạng hoặc các nút riêng rẽ được nối với nhau tạo thành một liên mạng (mạng của các mạng), các thiết bị. Các thiết bị kết nối trung gian (thiết bị định tuyến - router) phải xác định tuyến đường cho các gói dữ liệu để chúng đến được nơi nhận cuối cùng.

2.2.3.4 Tầng vận tải

Tầng vận tải chịu trách nhiệm chuyển toàn bộ bản tin từ nơi gửi đến nơi nhận một cách toàn vẹn. Nói cách khác, tầng vận tải đảm bảo liên kết giữa các tiến trình trên các máy tính khác nhau trên môi trường mạng. Có hai loại liên kết: Liên kết có hướng (Connection Oriented) và liên kết vô hướng (Connectionless). Đối với liên kết có hướng, tầng vận tải tạo ra một kết nối logic giữa hai cổng đầu cuối: tất cả dữ liệu của cùng một bản tin được truyền theo đường kết nối đó. Kết nối có hướng gồm ba giai đoạn: thiết lập liên kết, truyền dữ liệu, giải phóng liên kết. Do phải truyền tất cả các dữ liệu trên một kết nối, tầng vận tải còn phải kiểm soát thứ tự truyền, lưu lượng, phát hiện và sửa lỗi. Tầng vận tải đảm nhiệm các chức năng sau:

- **Thiết lập liên kết logic giữa các tiến trình trên thiết bị đầu cuối của người dùng:** Mỗi máy tính thường chạy nhiều chương trình tại cùng một thời điểm, việc chuyển bản tin không chỉ đơn thuần là truyền dữ liệu từ một máy tính này sang máy tính khác mà phải chuyển bản tin từ một tiến trình trên máy tính này đến tiến trình tương ứng trên một máy tính khác. Để đảm nhiệm chức năng này, một loại thông tin điều khiển được thêm vào tầng vận tải gọi là cổng (port), mỗi cổng sẽ tương ứng với một tiến trình tại tầng phiên.
- **Phân đoạn và tái hợp**: dữ liệu tại tầng ứng dụng thường có dung lượng lớn, để vận chuyển được hiệu quả, máy tính phải chia mỗi bản tin thành các đoạn (segment) nhỏ hơn và chúng được truyền độc lập với nhau. Mỗi đoạn tin được gán một số thứ tự, số thứ tự này giúp cho tầng vận tải phía nhận tái hợp các đoạn lại thành bản tin hoàn chỉnh.
- **Kiểm soát kết nối**: Tại tầng vận tải người ta sử dụng hai kỹ thuật truyền số liệu: kết nối có hướng hoặc kết nối vô hướng. Kết nối có hướng nối gửi yêu cầu kết nối đến tầng vận tải của máy nhận, nếu được chấp thuận thì mới chuyển các đoạn dữ liệu, sau khi truyền xong dữ liệu phải gửi tiếp yêu cầu hủy kết nối. Kết nối vô hướng không phải gửi yêu cầu kết nối trước khi truyền dữ liệu (datagram), do đó kết nối vô hướng sẽ không tin cậy bằng kết nối có hướng, tính tin cậy trong truyền dữ liệu vô hướng do các tiến trình tầng trên đảm nhiệm.
- **Kiểm soát lưu lượng**: Tầng vận tải chịu trách nhiệm kiểm soát lưu lượng giữa hai máy tính đầu cuối của người sử dụng. Để thực hiện chức năng này, phần thông tin điều khiển của đoạn tin phải có thành phần kiểm soát lượng dữ liệu được phép gửi đi.

- **Kiểm soát lỗi**: Tầng vận tải chịu trách nhiệm kiểm soát lỗi tại các thiết bị đầu cuối của người sử dụng. Tất cả các đoạn tin gửi đi phải được đảm bảm đến đích chính xác, nếu có lỗi thì phải truyền lại.

2.2.3.5 Tầng phiên

Tầng phiên đóng vai trò kiểm soát viên hội thoại giữa các tiến trình trên lớp ứng dụng qua mạng, nó đảm bảo nhiệm vụ thiết lập, duy trì và đồng bộ hóa tính tương tác giữa các tiến trình đồng cấp trên các máy tính khác nhau. Tầng phiên đảm nhiệm các chức năng sau:

- **Kiểm soát hội thoại**: Tầng phiên cho phép hai tiến trình cùng tham gia vào một cuộc hội thoại. Nó cho phép truyền thông giữa hai tiến trình được thực hiện hoặc theo kiểu bán song công hoặc theo kiểu song công. Ví dụ, hội thoại giữa một thiết bị đầu cuối với một máy chủ có thể theo kiểu bán song công.
- **Đồng bộ hóa**: Tầng phiên cho phép một tiến trình thêm các mốc gọi là điểm đồng bộ (***synchronization point)*** vào luồng dữ liệu. Ví dụ, nếu hệ thống cần gửi đi một tập tin lớn gồm N trang, cứ sau M trang nên chèn thêm các điểm đồng bộ để đảm bảo rằng việc nhận từng cụm M trang được thực hiện độc lập. Trong trường hợp này nếu như có lỗi khi đang truyền đi trang thứ P x M+1, việc truyền lại sẽ được bắt đầu từ trang P x M+1, không cần phải truyền lại các trang từ 1 đến trang P x M.

2.2.3.6 Tầng trình diễn

Tầng trình diễn thực hiện các nhiệm vụ liên quan đến cú pháp và ngữ nghĩa của các thông tin được trao đổi giữa hai hệ thống. Tầng trình diễn có nhiệm vụ:

- **Mã hóa/Giải mã dữ liệu (Encode/Decode):** Các tiến trình trên hai thiết bị trao đổi các thông tin dưới nhiều dạng khác nhau (xâu kí tự, số, âm thanh, hình ảnh…), các thông tin này sau đó được chuyển sang dạng bit để truyền đi. Do các hệ thống máy tính khác nhau sử dụng các chẩn mã hóa khác nhau, tầng trình diễn chịu trách nhiệm đảm bảo tính trong suốt đối với người sử dụng trên các hệ thống sử dụng cách mã hóa khác nhau đó. Tầng trình diễn tại phía gửi chuyển thông tin theo khuôn dạng của mình thành thông tin theo khuôn dạng chung. Tầng trình diễn tại máy nhận sẽ chuyển thông tin trong khuôn dạng chung thành thông tin theo khuôn dạng của máy nhận.
- **Nén/Giải nén:** Nén dữ liệu là quá trình làm làm giảm số lượng bit cần thiết phải chuyển trên đường truyền vật lý, từ đó nâng cao hiệu suất truyền tin. Nén dữ liệu ngày càng trở nên quan trọng, đặc biệt trong việc truyền các dữ liệu đa phương tiện âm thanh, hình ảnh.
- **Mã hóa/Giải mã bảo mật (Encrypt/Decrypt)**: Hệ thống phải có khả năng đảm bảo tính bí mật khi chuyển những thông tin quan trọng. Do vậy phía gửi sẽ biến đổi thông tin ban đầu (bản rõ) thành một dạng khác (bản mã hóa) và gửi nó đến phía nhận - đây là tiến trình mã hóa. Phía nhận thực hiện quá trình ngược lại bằng cách chuyển bản tin nhận được (bản mã hóa) thành nguyên dạng ban đầu (bản rõ), quá trình này được gọi là giải mã.

2.2.3.7 Tầng ứng dụng

Tầng ứng dụng cung cấp các tiện ích để người dùng truy cập vào mạng như: các dịch vụ như gửi thư điện tử, truy cập và chuyển file từ xa…. Tầng ứng dụng cũng cung cấp các phương thức cho các ứng dụng khác (ví dụ truy nhập cơ sở dữ liệu mô hình khách/chủ…). Tầng ứng dụng là tầng cao nhất trong mô hình OSI, do đó nó tạo ra dữ liệu thực sự chứ không có các thông tin điều khiển. Tầng ứng dụng cung cấp các dịch vụ sau:

- **Thiết bị đầu cuối ảo của mạng**: một thiết bị đầu cuối ảo của mạng là phiên bản phần mềm của một thiết bị đầu cuối vật lý, cho phép người đùng đăng nhập vào một máy từ xa.
- **Quản lý, truy cập và chuyển tập tin**: ứng dụng này cho phép người dùng truy cập tập tin, quản lý các tập trên một máy tính khác.
- **Các dịch vụ khác**: Hai dịch vụ phổ biến nhất là thư điện tử và truy nhập web. Dịch vụ thư điện tử cho phép hai hoặc nhiều người người trao đổi thư với nhau qua mạng (gọi là thư điện tử). Dịch vụ truy nhập web cho phép người dùng đọc tin tức trên các trang thông tin điện tử. Nói chung các dịch vụ loại này rất nhiều và ngày càng đa dạng.

2.2.4 Mô hình TCP/IP

Mô hình OSI là mô hình tham chiếu được ISO xây dựng nhằm tạo một chuẩn phục vụ việc nối kết các hệ thống mở. Tuy nhiên mô hình này chỉ dừng lại ở mức độ lý thuyết, trong thực tế mô hình TCP/IP đang được sử dụng rộng rãi nhất hiện nay (mạng Internet đang sử dụng mô hình này), hầu hết tất cả các hệ điều hành đều có cài đặt bộ giao thức TCP/IP. Bộ giao thức này được đặt tên theo hai giao thức chính của nó là giao thức điều khiển vận tải (TCP - Transmission Control Protocol) và giao thức liên mạng (IP - Internet protocol).

Hình 2.7 Các tầng trong bộ giao thức TCP/IP

Về mặt lịch sử, mô hình TCP/IP ra đời trước khi có mô hình OSI. Giống như mô hình OSI, mô hình cũng được phân thành bốn tầng, mỗi tầng gồm bộ giao thức đảm nhiệm các chức năng riêng biệt. Tuy số lượng tầng ít hơn, nhưng mô hình TCP/IP vẫn phải đảm nhiệm đầy đủ các chức năng đã nêu trong mô

hình OSI. Mô hình TCP/IP chia theo 4 tầng: truy nhập mạng, mạng, vận tải và ứng dụng.

2.2.4.1 Tầng truy nhập mạng

Đây là tầng thấp nhất của mô hình TCP/IP, chịu trách nhiệm nhận các gói tin của tầng trên Internet và việc truyền phát chúng trên một mạng xác định. Theo quan điểm hiện nay mô hình TCP/IP không còn bao gồm các đặc tả vật lý, nói cách khác tầng liên kết cũng không còn bao gồm vấn đề về phần cứng hay việc truyền tín hiệu vật lý nữa. Tuy nhiên trong các trường hợp triển khai cụ thể, tầng truy nhập mạng sẽ được chia thành hai tầng con thực hiện các chức năng của tầng liên kết dữ liệu và tầng vật lý của mô hình OSI.

Đối với truy nhập mạng qua modem quay số, các gói IP thường được truyền bằng cách sử dụng giao thức PPP. Đối với truy nhập Internet băng thông rộng (broadband) như ADSL hay modem cáp, giao thức PPPoE thường được sử dụng. Mạng dây cục bộ (local wired network) thường sử dụng Ethernet, còn mạng không dây cục bộ thường dùng chuẩn IEEE 802.11. Đối với các mạng diện rộng (wide-area network), các giao thức thường được sử dụng là PPP đối với các đường T-carrier hoặc E-carrier, Frame relay, ATM (Asynchronous Transfer Mode), hoặc giao thức packet over SONET/SDH (POS). Tầng truy nhập mạng kết hợp của các thành phần vật lý thực sự như các bộ lặp, cáp mạng và các thiết bị nối khác.

2.2.4.2 Tầng Internet

Tầng Internet tương ứng với tầng mạng trong mô hình OSI, nó đảm bảo liên kết logic giữa hai thiết bị đầu cuối của người sử dụng. Các giao thức trong tầng này nhận dữ liệu từ tầng vận tải cùng với một địa chỉ của máy đích mà gói tin sẽ được gửi tới đóng gói dữ liệu và thực hiện nhiệm vụ chọn đường để chuyển tiếp gói tin đến địa chỉ đích. Trong bộ giao thức liên mạng, giao thức IP thực hiện nhiệm vụ cơ bản dẫn đường dữ liệu từ nguồn tới đích. Giao thức IP có thể chuyển dữ liệu theo yêu cầu của nhiều giao thức tầng trên khác nhau; mỗi giao thức trong đó được định danh bởi một số hiệu giao thức duy nhất: giao thức ICMP (Internet Control Message Protocol) là giao thức 1 và giao thức IGMP (Internet Group Management Protocol) là giao thức 2.

Một số giao thức truyền bởi giao thức IP, chẳng hạn ICMP (dùng để gửi thông tin chẩn đoán về truyền dữ liệu bằng IP) và IGMP (dùng để quản lý dữ liệu đa truyền (multicast)), được đặt lên trên IP nhưng thực hiện các chức năng của tầng liên mạng, điều này minh họa một sự bất tương thích giữa liên mạng và chồng TCP/IP và mô hình OSI. Tất cả các giao thức định tuyến, chẳng hạn giao thức BGP (Border Gateway Protocol), giao thức OSPF, và giao thức RIP (Routing information protocol|), đều thực sự là một phần của tầng mạng, mặc dù chúng có thể có vẻ thuộc về phần trên của chồng giao thức.

2.2.4.3 Tầng vận tải

Nhiệm vụ trước tiên của tầng vận tải là đảm bảo liên kết giữa các tiến trình trên các thiết bị đầu cuối của người sử dụng. Tầng vận tải cũng có thể điều

chỉnh lưu lượng luồng thông tin. Nó cũng cung cấp một sự vận chuyển tin cậy, đảm bảo rằng dữ liệu đến mà không bị lỗi. Để làm như vậy, phần mềm giao thức hỗ trợ để bên nhận có thể gửi lại các thông báo xác nhận về việc thu dữ liệu và bên gửi có thể truyền lại các đoạn tin bị mất hoặc bị lỗi.

Nhiệm vụ của tầng vận tải là kết hợp các khả năng truyền bản tin từ đầu cuối đến đầu cuối mà không phụ thuộc vào mạng bên dưới, kiểm soát lỗi (error control), phân mảnh dữ liệu và điều khiển lưu lượng. Việc truyền bản tin giữa các tiến trình trên các thiết bị đầu cuối của người sử dụng tại tầng vận tải gồm hai loại:

- Kết nối có hướng (connection-oriented), ví dụ giao thức TCP
- Kết nối vô hướng (connectionless), ví dụ giao thức UDP

Tầng vận tải có thể được xem như một cơ chế vận chuyển thông thường, nghĩa là trách nhiệm của một phương tiện vận tải là đảm bảo rằng hàng hóa/hành khách của nó đến đích an toàn và đầy đủ. Tầng vận tải cung cấp dịch vụ kết nối các ứng dụng với nhau thông qua việc sử dụng các cổng TCP và UDP. Do IP chỉ cung cấp dịch vụ phát chuyển nỗ lực tối đa (best effort delivery), tầng vận tải là tầng đầu tiên giải quyết vấn đề độ tin cậy. TCP là một giao thức kết nối có hướng, nó giải quyết nhiều vấn đề độ tin cậy để cung cấp một dòng dữ liệu đáng tin cậy:

- Dữ liệu đến đích đúng thứ tự
- Sửa lỗi dữ liệu ở mức độ tối thiểu
- Loại bỏ dữ liệu trùng lặp
- Gửi lại các gói tin bị thất lạc hoặc bị lỗi
- Kiểm soát lưu lượng truyền tin

UDP là một giao thức kết nối vô hướng. Giống như giao thức IP, nó là một giao thức nỗ lực tối đa phân phát dữ liệu và không tin cậy. Giao thức này thường được dùng cho các dịch vụ yêu không đòi hỏi độ chính xác cao. RTP (Real-time Transport Protocol - giao thức vận tải thời gian thực) là một giao thức được thiết kế cho dữ liệu thời gian thực, đó là giao thức tầng phiên sử dụng định dạng gói tin UDP. Tuy nhiên, nó vẫn được xếp vào giao thức thuộc tầng vận tải.

2.2.4.4 Tầng ứng dụng

Đây là tầng cao nhất trong cấu trúc phân lớp của TCP/IP, nó bao gồm ba tầng trên của mô hình OSI. Tầng này bao gồm tất cả các chương trình ứng dụng sử dụng các dịch vụ sẵn có thông qua một chồng giao thức TCP/IP. Các chương trình ứng dụng tương tác với một trong các giao thức của tầng vận tải để truyền hoặc nhận dữ liệu. Mỗi chương trình ứng dụng lựa chọn một kiểu giao thức thích hợp cho công việc của nó.

Tầng ứng dụng là nơi các chương trình mạng thường dùng nhất làm việc nhằm liên lạc giữa các nút trong một mạng. Giao tiếp xảy ra trong tầng này là tùy theo các ứng dụng cụ thể và dữ liệu được truyền từ chương trình, trong định

dạng được sử dụng nội bộ bởi ứng dụng này, và được đóng gói theo một giao thức tầng vận tải.

Trong mô hình TCP/IP, không có tầng nào nằm giữa ứng dụng và các tầng vận tải, tầng ứng dụng trong bộ TCP/IP phải bao gồm các giao thức hoạt động như các giao thức tại tầng trình diễn và tầng phiên của mô hình OSI. Việc này thường được thực hiện qua các thư viện lập trình.

Dữ liệu thực để gửi qua mạng được truyền cho tầng ứng dụng, nơi nó được đóng gói theo giao thức tầng ứng dụng. Từ đó, dữ liệu được truyền xuống giao thức tầng thấp tại tầng vận tải. Hai giao thức tầng thấp thông dụng nhất là TCP và UDP. Mỗi ứng dụng sử dụng dịch vụ của một trong hai giao thức trên đều cần có cổng. Hầu hết các ứng dụng phổ biến đều có các cổng riêng biệt (HTTP - Giao thức truyền siêu văn bản dùng cổng 80; FTP - Giao thức truyền tệp dùng cổng 21, v.v..)

2.2.5 So sánh mô hình OSI và mô hình TCP/IP

Bộ giao thức trong mô hình TCP/IP đã được sử dụng trước khi mô hình OSI được công bố. Trong khi mô hình TCP/IP đã được triển khai thực tế trong các hệ thống mạng, việc sử dụng mô hình thường để diễn tả chức năng và hoạt động của mạng. Hai mô hình này có liên quan với nhau, nhưng không phải là hoàn toàn giống nhau. Điểm khác biệt đầu tiên dễ thấy nhất là số lượng của các tầng cấp. Mô hình của Bộ Quốc Phòng Mỹ (DoD model) với bộ giao thức IP chỉ có bốn hoặc năm tầng (tầng liên kết có thể được coi như là một tầng riêng biệt, song cũng có thể được phân tách ra thành hai tầng, tầng vật lý và tầng liên kết dữ liệu) trong khi đó mô hình OSI lại dùng bảy tầng. So sánh tên của chúng một cách chặt chẽ cho chúng ta thấy rằng, hai tầng tầng trình diễn và tầng phiên đã gộp lại vào tầng ứng dụng.

Các tầng của mô hình OSI không có nhiều chức năng đủ để phản ánh hoạt động của mô hình TCP/IP. Chẳng hạn, cần phải có một tầng nằm giữa tầng mạng và tầng vận tải để chỉ ra nơi tồn tại của giao thức ICMP (Internet Control Message Protocol) và IGMP (Internet Group Management Protocol). Tương tự như vậy cũng cần phải có một tầng ở giữa tầng mạng và tầng liên kết dữ liệu dành cho giao thức ARP (Address Resolution Protocol) và giao thức RARP (Reverse Address Resolution Protocol).

Bảng sau tóm tắt một số giao thức và vị trí của chúng trong mô hình OSI. Để thuận tiện, các phần tiếp theo sẽ trình bày các giao thức theo mô hình TCP/IP (vì đây là mô hình đang được áp dụng trên mạng Internet), đôi khi cũng sẽ dùng mô hình OSI để giải thích nguyên lý làm việc của các thiết bị trên mạng.

Tầng	Mô hình OSI	Mô hình TCP/IP	Giao thức
7	Tầng ứng dụng	Tầng ứng dụng	HTTP, SMTP, SNMP, FTP, Telnet, ECHO, SIP, SSH, NFS, RTSP, XMPP, Whois, ENRP
6	Tầng trình diễn		XDR, ASN.1, SMB, AFP, NCP

5	Tầng phiên		ASAP, TLS, SSH, ISO 8327 / CCITT X.225, RPC, NetBIOS, ASP
4	Tầng vận tải	Tầng vận tải	TCP, UDP, RTP, SCTP, SPX, ATP, IL
3	Tầng mạng	Tầng Internet	IP, ICMP, IGMP, IPX, BGP, OSPF, RIP, IGRP, EIGRP, ARP, RARP, X.25
2	Tầng liên kết dữ liệu	Tầng truy nhập mạng	Ethernet 802.2, Token ring, HDLC, Frame relay, ISDN, ATM, 802.11 WiFi, FDDI, PPP
1	Tầng vật lý		Ethernet 802.3 (10BASE-T, 100BASE-T, 1000BASE-T), SONET/SDH, T-carrier/E-carrier, 802.11

2.3 Tên miền và địa chỉ

Mỗi con người có thể được xác định theo nhiều cách như: nhận biết qua tên trong giấy khai sinh, số chứng minh thư, số hộ chiếu…. Dù có nhiều cách nhận biết để phân biệt mọi người nhưng phương thức nhận biết nào phụ thuộc vào hoàn cảnh. Ví dụ công an sử dụng số chứng minh thư nhân dân chứ không sử dụng tên. Bình thường mọi người thích nhớ tên nhau hơn là chứng minh thư.

Giống như con người, máy tính trên mạng cũng có thể được xác định bằng nhiều cách: Tên máy tính hoặc tên miền. Những tên đó tương đối dễ nhớ đối với con người, nhưng cung cấp ít thông tin về vị trí trên mạng (tên miền www.ptit.edu.vn chỉ cho chúng ta biết máy tính đó thuộc Việt Nam). Hơn nữa tên máy tính bao gồm nhiều ký tự - cả chữ cái và chữ số - có độ dài thay đổi nên thiết bị định tuyến khó có thể xử lý được. Vì vậy, máy tính được xác định thông qua địa chỉ logic gọi là địa chỉ IP. Địa chỉ IP gồm có 32 bit (phiên bản 4) hoặc 128 bit (phiên bản 6) và có cấu trúc phân cấp, để đơn giản chúng ta sẽ chỉ nói đến địa chỉ IP phiên bản 4. Địa chỉ IP phân cấp vì khi duyệt địa chỉ từ trái qua phải, chúng ta nhận được thêm nhiều thông tin xác đinh về vị trí của máy tính trên mạng (Vị trí ở trong mạng của các mạng, trong một mạng . . .). Điều này tương tự khi xét địa chỉ bưu điện từ dưới lên, chúng ta nhận được nhiều thông tin về địa chỉ đó.

2.3.1 Các dịch vụ tên miền

Có hai cách để xác định một máy tính: dựa vào tên máy tính hoặc địa chỉ IP. Con người thích sử dụng tên máy dễ nhớ, trong khi thiết bị định tuyến lại sử dụng địa chỉ IP có cấu trúc phân cấp và độ dài cố định. Để dung hoà giữa hai cách cần phải có một dịch vụ chỉ dẫn để chuyển đổi tên máy tính sang địa chỉ IP và đây chính là nhiệm vụ của hệ thống tên miền trên mạng Intemet (DNS). DNS là một cơ sở dữ liệu phân tán được đặt trên một hệ thống phân cấp các máy chủ tên miền và cung cấp dịch vụ thuộc tầng ứng dụng cho phép máy tính và máy chủ tên trao đổi thông tin phục vụ mục đích xác định địa chỉ IP. Giao thức trao đổi tên miền DNS thuộc tầng ứng dung và chạy trên nền giao thức UDP với số hiệu cổng là 53.

Thông thường DNS được các giao thức tầng ứng dụng khác như HTTP, SMTP và FTP sử dụng để xác định địa chỉ IP từ tên máy tính do người dùng đưa vào. Điều gì xảy ra khi người sử dụng muốn truy nhập vào trang www.ptit.edu.vn? Để gửi được bản tin HTTP yêu cầu tới máy chủ web thì máy tính của người sử dụng phải xác định được địa chỉ IP của www.ptit.edu.vn. Điều này được thực hiện như sau: máy tính của người sử dụng chạy ứng dụng DNS. Trình duyệt sẽ lấy ra tên máy tính (www.ptit.edu.vn) từ địa chỉ URL trên trình duyệt và chuyển nó cho tiến trình DNS trên máy trạm. Tiến trình DNS máy trạm gửi một yêu cầu chứa tên miền www.ptit.edu.vn tới máy chủ DNS đã được đăng ký trong cấu hình địa chỉ. Nhận được yêu cầu này, máy chủ DNS sẽ tìm kiếm trong cơ sươr dữ liệu của mình, nếu không tìm thấy sẽ gửi chuyển tiếp yêu cầu đó đến máy chủ tên miền cấp cao hơn. Kết quả là tiến trình DNS máy trạm sẽ nhận được một bản tin trả lời từ máy chủ DNS chứa địa chi IP cần xác định. Sau đó trình duyệt sẽ mở một kết nối TCP tới tiến trình HTTP máy chủ trên máy tính có địa chỉ IP vừa được xác định. Rõ ràng các ứng dụng Internet sử dụng DNS hoạt động chậm đi. Tuy nhiên, địa chỉ IP đã được xác định thường được ghi tạm trong một máy chủ DNS trong một thời gian nhất định và như vậy làm giảm tải cho hệ thống DNS cũng như độ trễ của ứng dụng. Bên cạnh dịch vụ xác định địa chỉ IP từ tên máy, DNS cung cấp một số dịch vụ quan trọng sau:

2.3.1.1 Dịch vụ đặt bí danh cho máy tính

Máy tinh có thể có một hoặc nhiều bí danh, ví dụ tên máy chủ trang web Server1.www.ptit.edu.vn có thể có hai bí đanh là www.ptit.edu.vn và ptit.edu.vn. Tên bí danh thường dễ nhớ hơn tên thật. Một ứng dụng có thể yêu cầu DNS xác định tên thật cũng như địa chỉ IP của một tên bí danh.

2.3.1.2 Dịch vụ đặt bí danh cho máy chủ

Địa chỉ của thư điện tử cần dễ nhớ. Ví dụ nếu trên máy chủ của học viện Công nghệ bưu chính viễn thông (trang thông tin điện tử là www.ptit.edu.vn) thì địa chỉ hộp thư điện tử của các thành viên sẽ là XXX@ptit.edu.vn. Ứng dụng có thể sử dụng DNS để xác định tên đầy đủ của một bí danh cũng như địa chỉ IP của máy tính đó. Trên thực tế, DNS cho phép đặt tên miền cho các dịch vụ của một máy chủ, ví dụ Học viện Công nghệ Bưu chính viễn thông có tên miền là ptit.edu.vn thì các dịch vụ cơ bản như trang web có tên miền là www.ptit.edu.vn, thư điện tử là mail.ptit.edu.vn…

2.3.1.3 Phân tán tải

DNS thực hiện việc phân tán tải cho các máy chủ, đặc biệt là các máy chủ web nhân bản (các máy chủ có nội dung giống hệt nhau). Những trang có nhiều người truy cập như yahoo.com được đặt trên nhiều máy chủ giống hệt nhau. Mỗi máy chủ là một hệ thống đầu cuối khác nhau, có địa chỉ IP khác nhau. Đối với các máy chủ giống hệt nhau như vậy, một nhóm địa chỉ IP sẽ gắn với tên đầy đủ của một máy nào đó. Cơ sở dữ liệu DNS chứa toàn bộ nhóm địa chỉ IP đó. Khi máy khách gửi truy vấn DNS để xác định địa chỉ IP thì máy chủ sẽ gửi toàn bộ nhóm địa chỉ IP đó nhưng máy chủ thay đổi thứ tự các địa chỉ IP trong nhóm. Thông thường máy khách gửi bản tin HTTP tới máy tính có địa chỉ

IP được liệt kê đầu tiên trong nhóm. sự hoán chuyển vị trí các địa chỉ IP mà DNS thực hiện đã phân tải cho các máy chủ. Việc hoán vị của DNS cũng được ở dụng cho email khi nhiều máy chủ thư điện tử có chung bí danh. DNS được đặc tả trong RFC 1034 và RFC 1035 và cập nhật trong một số RFC khác. DNS là hệ thống phức tạp và chúng ta chỉ nghiên cứu một vài khía cạnh của nó. Chi tiết về DNS có thể đọc trong [Abitz 1993].

Giống như các giao thức HTTP, FPT hay SMTP, giao thức DNS nằm ở tầng ứng dụng vì nó hoạt động giữa hai thực thể truyền thông đầu cuối sử dụng mô hình khách/ chủ, sử dụng một giao thức ở tầng vận tải để trao đổi bản tin DNS giữa hai đầu cuối. Tuy nhiên vai trò của DNS khác các ứng dụng Web, FTP hay Email nhiều. DNS không phải ứng dụng được người dùng trực tiếp sử dụng mà DNS chỉ cung cấp một dịch vụ Internet thiết yếu cho các ứng dụng: chuyển đổi tên máy tính sang địa chỉ IP.

2.3.2 Cơ chế hoạt động của dịch vụ tên miền

Máy trạm gửi bản tin truy vấn tên miền đến máy chủ DNS đã được đăng ký trong phần cấu hình địa chỉ IP, trong bản tin chứa tên miền cần xác định địa chỉ IP. Sau một khoảng thời gian nào đó – từ vài phần nghìn giây đến vài chục giây, máy trạm nhận được bản tin trả lời của DNS chứa địa chỉ IP cần xác định. Vì vậy, với máy khách thì DNS là một dịch vụ xác định IP đơn giản và dễ hiểu. Nhưng triển khai dịch vụ đó thực sự rất phức tạp, bao gồm nhiều máy chủ tên miền đặt khắp nơi trên thế giới và một giao thức ở tầng ứng dụng xác định cách thức trao đổi thông tin giữa các máy chủ tên miền.

Để triển khai DNS, người ta có thể đưa ra một kiến trúc đơn giản sau: có một máy chủ chứa tất cả các ánh xạ tên và địa chỉ IP. Theo thiết kế tập trung này, máy khách chỉ cần gửi tất cả các truy vấn tới máy chủ duy nhất và máy chủ này sẽ trực tiếp trả lời mọi truy vấn. Mặc dù tính đơn giản của thiết kế này rất hấp dẫn nhưng nó hoàn toàn không thích hợp cho mạng Intemet với số lượng lớn và ngày càng tăng các máy tính. Thiết kể tập trung như vậy nẩy sinh một số vấn đề sau:

- **Điểm hỏng duy nhất:** nếu máy chủ tên miền duy nhất ngừng làm việc cũng có nghĩa là toàn bộ mạng Internet ngừng hoạt động.
- **Khối lượng xử lý lớn:** một máy chủ tên miền duy nhất phải xử lý tất cả các truy vấn DNS (cho tất cả các bản tin yêu cầu từ hàng tỉ máy tính trên toàn cầu).
- **Cơ sở dữ liệu tập trung ở xa:** máy chủ tên miền duy nhất không thể gần tất cả các máy khách. Nếu máy chủ tên miền đặt ở Hoa Kỳ thì tất cả truy vấn từ các nước khác phải chuyển tới phía bên kia trái đất và có thể qua một đường kết nối chậm và tắc nghẽn. Hậu quả là các ứng dụng phải chịu độ trễ rất lớn.
- **Bảo trì:** máy chủ tên miền phải ghi nhớ thông tin về tất cả các tên miền trên mạng Intemet. Khi đó cơ sở dữ liệu sẽ rất lớn và máy chủ tên miền phải cập nhật thường xuyên thông tin cho mọi tên miền mới, đồng thời phải giải quyết

các vấn đề kiểm chứng và xác nhận khi người dùng sử dụng cơ sở dữ liệu tập trung.

Như vậy, một cơ sở dữ liệu tập trung trên một máy chủ tên miền duy nhất không phù hợp khi quy mô hệ thống lớn. Do đó, hệ thống máy chủ têm miền được thiết kế phân tán, đó là một ví dụ điển hình về triển khai cơ sở dữ liệu phân tán trên mạng Internet. Để giải quyết vấn đề quy mô mạng , DNS sử dụng nhiều máy chủ tên miền tổ chức phân cấp và phân tán trên toàn cầu. Không có máy chủ tên miền nào chứa tất cả tên và địa chỉ IP các tên miền trên mạng Internet, những thông tin này được phân tán trên nhiều máy chủ tên miền. Có ba loại máy chủ tên miền: máy chủ tên miền cục bộ, máy chủ tên miền gốc và máy chủ tên miền ủy quyền. Các máy chủ tên miền đó trao đổi thông tin với nhau và với các máy tính khác.

Máy chủ tên miền cục bộ:

Mỗi nhà cung cấp dịch vụ Internet (ISP) đều có máy chủ tên miền cục bộ (còn được gọi là máy chủ tên miền mặc định). Khi máy tính trong cơ quan tạo ra một bản tin truy vấn DNS thì đầu tiên bản tin đó được gửi tới máy chủ tên miền đó. Địa chỉ IP của máy chủ tên miền cục bộ phải được cấu hình trong máy tính của người sử dụng (trong máy tính chạy hệ điều hành windows, gõ lệnh ipconfig /all). Loại máy chủ tên miền thường gần với máy trạm, trong trường hợp tại cơ quan của một tổ chức, nó có thể ở trên cùng mạng nội bộ. Với các ISP thì khoảng cách giữa máy chủ tên miền và các máy tính của người sử dụng chỉ là vài thiết bị định tuyến. Nếu máy tính yêu cầu xác định địa chỉ của một máy tính khác trong cùng một ISP thì máy chủ tên miền cục bộ có thể ngay lập tức xác định được địa chi IP mà không phải liên hệ với bất kỳ máy chủ tên miền nào khác.

Máy chủ tên miền gốc:

Trên mạng Intemet có 13 máy chủ tên miền gốc, hầu hết đặt tại Bắc Mỹ. Khi máy chủ tên miền cục bộ không có thông tin về tên miền được yêu cầu thì máy chủ tên miền cục bộ sẽ đóng vai trò máy khách DNS và gửi câu hỏi truy vấn tới một trong số các máy chủ tên miền gốc. Nếu máy chủ tên miền gốc có thông tin của tên miền được hỏi, nó sẽ gửi một bản tin trả lời đến máy chủ tên miền cục bộ và sau đó thông tin này được máy chủ tên miền cục bộ gửi trả lời cho máy trạm đã yêu cầu. Nếu máy chủ tên miền gốc không có thông tin tên miền đó, nó sẽ tìm kiếm thông tin về máy chủ tên miền quản lý tên miền đã yêu cầu.

Máy chủ tên miền ủy quyền:

Mỗi máy tính phải đăng ký tới một máy chủ tên miền ủy quyền. Thông thường máy chủ tên miền ủy quyền một máy tính là một máy chủ tên miền trong miền ISP của máy tính đó (thực tế mỗi máy tính phải có ít nhất hai máy chủ tên miền ủy quyền để đề phòng trường hợp một máy chủ tên miền bị hỏng). Có thể định nghĩa, máy chủ tên miền ủy quyền của một máy tính là máy chủ tên miền luôn lưu trữ bản ghi DNS cho phép xác định địa chỉ IP của máy tính đó từ tên. Khi máy chủ tên miền ủy quyền nhận được truy vấn từ máy chủ tên miền gốc,

nó sẽ gửi một bản tin DNS trả lời chứa ánh xạ được yêu cầu. Sau đó, máy chủ tên miền gốc gửi ánh xạ đó tới máy chủ và máy chủ tên miền cục bộ lại tiếp tục gửi ánh xạ đó tới máy tính yêu cầu. Nhiều máy chủ tên miền vừa là máy chủ tên miền cục bộ vừa là máy chủ tên miền ủy quyền.

Xét ví dụ đơn giản sau. Giả sử trạm muốn có địa chỉ IP của máy tính tên miền là www.yahoo.com, giả sử máy chủ gốc của miền là opendns.com và máy chủ tên miền ủy quyền của www.yahoo.com là dns.yahoo.com. Đầu tiên máy trạm gửi một bản tin truy vấn tới máy chủ tên miền cục bộ dns.vnn.vn. Bản tin đó chứa tên miền www.yahoo.com cần xác định địa chỉ IP. Máy chủ tên miền cục bộ không chứa bản ghi tên miền www.yahoo.com, do đó nó gửi bản tin tới máy chủ tên miền gốc, nhưng nó phân tích phần đuôi của tên miền là .com do đó nó gửi tới máy chủ tên miền gốc chuyên quản lý các tên miền có phần đuôi là .com, trong trường hợp này nó gửi đến máy chủ opendns.com. Máy chủ tên miền opendns.com không chứa bản ghi về tên miền www.yahoo.com nhưng lại chứa địa chỉ IP của máy chủ tên miền ủy quyền dns.yahoo.com của tên miền www.yahoo.com, vì vậy nó trả về địa chỉ IP của máy chủ dns.yahoo.com cho máy chủ tên miền cục bộ dns.vnn.vn. Nhận được địa chỉ IP này, máy chủ tên miền dns.vnn.vn gửi tiếp bản tin đến máy chủ dns.yahoo.com để yêu cầu cung cấp địa chỉ IP của tên miền www.yahoo.com, tại đây máy chủ tên miền dns.yahoo.com có chứa địa chỉ IP của máy chủ www.yahoo.com, nó gửi kết quả cho máy chủ tên miền cục bộ dns.vnn.vn, máy chủ tên miền dns.vnn.vn sẽ chuyển tiếp cho máy trạm và đồng thời lưu địa chỉ này trong cơ sở dữ liệu tạm của nó.

Một đặc tính quan trọng của DNS là lưu trữ tạm thời các bản ghi DNS (DNS caching). Trên thực tế, DNS lưu trữ tạm thời để làm giảm độ trễ cũng như làm giảm số bản tin DNS trao đổi trên mạng. Ý tưởng này rất đơn giản: Khi nhận được ánh xạ DNS của máy tính nào đó, bên cạnh việc gửi tiếp bản tin, máy chủ tên miền sẽ lưu ánh xạ này vào bộ nhớ cục bộ (ổ địa cứng hay RAM). Với ánh xạ tên máy - địa chỉ IP được lưu trữ, nếu có một truy vấn khác yêu cầu địa chỉ IP của cùng tên máy mà máy chủ tên miền vừa lưu trữ, máy chủ tên miền sẽ xác định được địa chỉ áp mong muốn, ngay cả khi nó không phải là máy chủ tên miền ủy quyền cho máy tính đó. Để tránh bị lạc hậu, thông tin lưu trữ tạm thời sẽ bị xoá bỏ sau một khoảng thời gian nhất định (thường là 48 giờ).

2.3.3 Bản ghi dịch vụ tên miền

Máy chủ tên miền cũng triển khai cơ sở dữ liệu phân tán, ghi lại các bản ghi tài nguyên cho các ánh xạ tên máy một địa chỉ IP. Mỗi bản tin trả lời DNS chứa một hay nhiều bản ghi tài nguyên, chi tiết trong RFC 1034, RFC 1035.

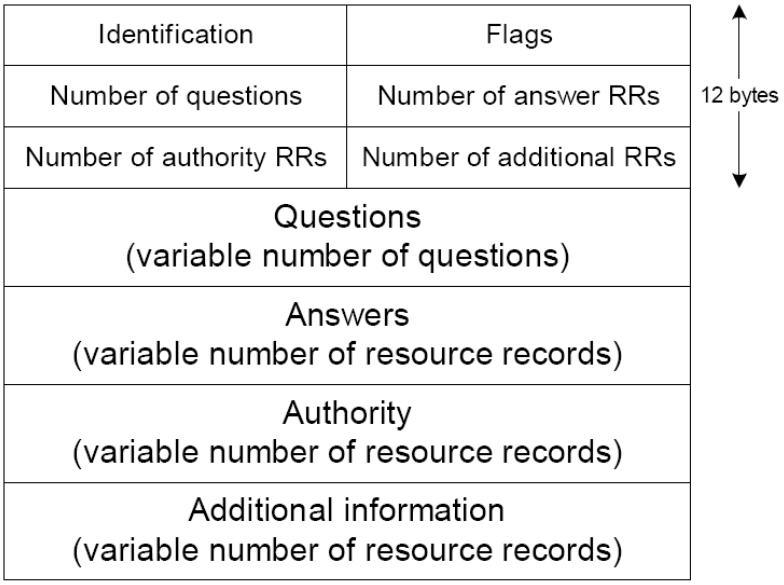
Bản ghi tài nguyên gồm 4 trường sau: (Name, Value, Type, TTL). TTL là thời gian tồn tại của bản ghi tài nguyên, dùng để xác định thời điểm có thể xoá bản ghi tài nguyên khỏi bộ nhớ lưu trữ. Ý nghĩa của trường Name và Value phụ thuộc vào trường Type:

Nếu Type = A thì Name là tên máy và Value là địa chỉ IP của máy đó. Bản ghi kiểu A là ánh xạ Tên máy - Địa chỉ IP chuẩn.

Nếu Type = NS thì Name là một miền và Value là tên máy của máy chủ tên miền ủy quyền của các máy tính trong miền đó. Bản ghi này thường được sử dụng để gửi tiếp các truy vấn DNS.

Nếu Type = CNAME thì Value là tên đầy đủ của máy có tên bí danh đặt trong Name. Bản ghi kiểu này cho phép xác định tên đầy đủ của một máy tính từ tên bí danh.

Nếu Type : MX thì Value là tên máy của máy chủ thư điện tử có tên bí danh đặt trong Name.



Hình 2.8 Khuôn dạng bản tin DNS

Nếu một máy chủ tên miền là máy chủ tên miền ủy quyền cho một máy tính nào đó thì máy chủ tên miền sẽ chứa bản ghi kiểu A của máy tính đó (ngay cả nếu máy chủ tên miền đó không là máy chủ tên miền ủy quyền thì có thể nó chứa bản ghi kiểu A trong bộ nhớ tạm của nó). Nếu máy chủ tên miền không là máy chủ tên miền ủy quyền của máy tính được hỏi thì nó sẽ chứa một bản ghi kiểu NS cho miền của máy tính này và nó cũng có một bản ghi kiểu A xác định địa chỉ IP của máy chủ tên miền của tên miền này đặt trong trường Value của bản ghi NS. Có hai loại bản tin DNS: bản tin yêu cầu và bản tin trả lời, cả hai kiểu bản tin này có chung khuôn dạng minh họa trên hình 2.8.

Mười hai byte đầu tiên là phần tiêu đề gồm một số trường sau: Trường đầu tiên là một định danh 16 bit cho mỗi bản tin yêu cầu, 16 bít định danh này được ghi lại vào bản tin trả lời, cho phép máy khách xác định được đó là câu trả lời cho bản tin yêu cầu nào. Có nhiều cờ trong trường Flag (mối cờ ứng với một bit), cờ truy vấn (query/reply flag) xác định bản tin là yêu cầu (0) hay là trả lời (1). Cờ authoritiative được đặt trong bản tin trả lời khi máy chủ tên miền là máy chủ tên miền ủy quyền của tên máy tính cần xác định địa chỉ IP, cờ mong muốn đệ quy truy vấn (recursive-desired query) được đặt khi máy khách (máy trạm hay máy chủ tên miền) mong muốn máy chủ tên miền thực hiện truy vấn đệ quy

khi nó không có bản ghi đó, cờ chấp nhận đệ quy (recursion-available flag) được đặt trong bản tin trả lời nếu máy chủ tên miền đó hỗ trợ đệ quy. Trong phần tiêu đề cũng có 4 trường số lượng, các trường này xác định số lượng các bản ghi trong 4 phần dữ liệu sau phần tiêu đề.

Phần câu hỏi (Question session) chứa thông tin về câu hỏi được tạo ra. Nó bao gồm trường tên chứa tên đang được hỏi và trường kiểu xác định kiểu câu hỏi cho tên máy tính đó (Kiểu A cho tên máy tính, kiểu MX cho máy chủ thư điện tử). Trong bản tin trả lời từ máy chủ tên miền, phần trả lời (answer section) chứa các bản ghi tài nguyên cho tên được yêu cầu trước đó. Mỗi bản ghi tài nguyên có 4 trường: Type (A, NS, CNAME, MX), Name, Value, TTL. Bản tin trả lời có thể có nhiều bản ghi tài nguyên vì tên máy tính có thể ứng với nhiều địa chỉ IP. Phần ủy quyền (authonty section) chứa các bản ghi của các máy chủ ủy quyền. Phần phụ trợ (additional section) chứa các bản ghi khác. Ví dụ trường trả lời trong bản tin trả lời một truy vấn MX sẽ chứa tên đầy đủ của máy chủ thư điện tử có tên bí danh đặt ở trong Name. Phần phụ trợ có thể có một bản ghi kiểu A cung cấp địa chỉ IP cho chính máy chủ thư điện tử đó.

Các phần trên mô tả cách thức lấy dữ liệu trong cơ sở dữ liệu DNS. vậy làm thế nào để đưa được dữ liệu vào cơ sở dữ liệu? Cho tới gần đây, nội dung của máy chủ DNS được cấu hình tĩnh, ví dụ, thông qua file cấu hình được người quản trị hệ thống tạo ra gần đây, lựa chọn UPDATE được đưa vào giao thức DNS cho phép dữ liệu được tự động thêm vào hay xoá bỏ khỏi cơ sở dữ liệu thông qua bản tin DNS. RFC 2136 đặc tả quá trình cập nhật động của DNS.

2.4 Nguyên tắc thiết kế Internet

Mạng Internet thực chất là mạng của các mạng được kết nối trên toàn cầu, do đó việc thiết kế mạng Internet phải dựa trên mô hình phân cấp. Như vậy chúng ta có thể phân biệt hai loại thiết kế: thiết kế cho mạng diện rộng và thiết kế cho mạng nội bộ. Dù là loại nào thì việc thiết kế đều phải tuân thủ các bước sau:

Xác định yêu cầu:

Trước khi thiết kế cần phải xác định các yêu cầu đối với mạng bao gồm:

- Yêu cầu kỹ thuật.
- Yêu cầu về hiệu năng.
- Yêu cầu về ứng dụng.
- Yêu cầu về quản lý mạng.
- Yêu cầu về an ninh và an toàn mạng.
- Yêu cầu ràng buộc về tài chính, thời gian thực hiện, yêu cầu về chính trị của dự án, xác định nguồn nhân lực, xác định các tài nguyên đã có và có thể tái sử dụng.

Mục đích của giai đoạn này là nhằm xác định mong muốn của khách hàng trên mạng mà chúng ta sắp xây dựng. Những câu hỏi cần được trả lời trong giai đoạn này là:

- Xây dựng mạng để làm gì? Sử dụng cho mục đích gì?

- Các máy tính nào sẽ được nối mạng?
- Những người nào sẽ được sử dụng mạng, mức độ khai thác sử dụng mạng của từng người / nhóm người ra sao?
- Trong tương lai gần (3 đến 5 năm tới) có nối thêm máy tính vào mạng không, nếu có thì nối ở đâu, số lượng bao nhiêu?

Phương pháp thực hiện của giai đoạn này là phỏng vấn khách hàng, nhân viên các phòng mạng có máy tính sẽ nối mạng. Thông thường các đối tượng mà phỏng vấn không có chuyên môn sâu hoặc không có chuyên môn về mạng, cho nên hạn chế sử dụng những thuật ngữ chuyên môn để trao đổi với họ. Chẳng hạn nên hỏi khách hàng "***Bạn có muốn người trong cơ quan bạn gửi mail được cho nhau không***?", hơn là hỏi "***Bạn có muốn cài đặt Mail server cho mạng không?*** ". Những câu trả lời của khách hàng thường không có cấu trúc, rất lộn xộn, nó xuất phát từ góc nhìn của người sử dụng, không phải là góc nhìn của kỹ sư mạng. Người thực hiện phỏng vấn phải có kỹ năng và kinh nghiệm trong lĩnh vực này, phải biết cách đặt câu hỏi và tổng hợp thông tin.

Một công việc cũng hết sức quan trọng trong giai đoạn này là "**Quan sát địa hình**" để xác định những nơi mạng sẽ đi qua, khoảng cách xa nhất giữa hai máy tính trong mạng, dự kiến đường đi của dây mạng, quan sát hiện trạng công trình kiến trúc nơi mạng sẽ đi qua. Địa hình đóng vai trò quan trọng trong việc chọn công nghệ và ảnh hưởng lớn đến chi phí xây dựng và vận hành mạng. Chú ý đến ràng buộc về mặt thẩm mỹ cho các công trình kiến trúc khi chúng ta triển khai đường dây mạng bên trong nó. Giải pháp để nối kết mạng cho 2 tòa nhà tách rời nhau bằng một khoảng không phải đặc biệt lưu ý.

Sau khi khảo sát địa hình, cần vẽ lại địa hình hoặc yêu cầu khách hàng cung cấp cho chúng ta sơ đồ thiết kế của công trình kiến trúc mà mạng đi qua. Trong quá trình phỏng vấn và khảo sát địa hình, đồng thời ta cũng cần tìm hiểu yêu cầu trao đổi thông tin giữa các phòng ban, bộ phận trong cơ quan khách hàng, mức độ thường xuyên và lượng thông tin trao đổi. Điều này giúp ích ta trong việc chọn băng thông cần thiết cho các nhánh mạng sau này.

Phân tích yêu cầu:

Khi đã có được yêu cầu của khách hàng, bước kế tiếp cần phải phân tích yêu cầu để xây dựng bảng "***Đặc tả yêu cầu hệ thống mạng***", trong đó xác định rõ những vấn đề sau:

- Những dịch vụ mạng nào cần phải có trên mạng ? (Dịch vụ chia sẻ tập tin, chia sẻ máy in, Dịch vụ web, Dịch vụ thư điện tử, Truy cập Internet hay không?, ...)
- Mô hình mạng là gì? (Workgoup hay Client / Server? ...)
- Mức độ yêu cầu an toàn mạng.
- Ràng buộc về băng thông tối thiểu trên mạng.
- Xác định số lượng nút mạng để quyết định phương thức phân cấp, chọn kỹ thuật thiết bị chuyển mạch.
- Dựa vào cơ cấu tổ chức để phân đoạn vật lý đảm bảo hai yêu cầu an ninh và đảm bảo chất lượng dịch vụ.

- Dựa vào mô hình vật lý để lựa chọn môi trường truyền dẫn.
- Dự báo các yêu cầu mở rộng.

Thiết kế giải pháp:

Người thiết kế cần phải đưa ra hình trạng mạng, bao gồm hình trạng vật lý và hình trạng logic, các cônng nghệ cần sử dụng. Bước kế tiếp trong tiến trình xây dựng mạng là thiết kế giải pháp để thỏa mãn những yêu cầu đặt ra trong bảng Đặc tả yêu cầu hệ thống mạng. Việc chọn lựa giải pháp cho một hệ thống mạng phụ thuộc vào nhiều yếu tố, có thể liệt kê như sau:

- Kinh phí dành cho hệ thống mạng.
- Công nghệ phổ biến trên thị trường.
- Thói quen về công nghệ của khách hàng.
- Yêu cầu về tính ổn định và băng thông của hệ thống mạng.
- Ràng buộc về pháp lý.

Tùy thuộc vào mỗi khách hàng cụ thể mà thứ tự ưu tiên, sự chi phối của các yếu tố sẽ khác nhau dẫn đến giải pháp thiết kế sẽ khác nhau. Tuy nhiên các công việc mà giai đoạn thiết kế phải làm thì giống nhau, chúng được mô tả như sau:

Thiết kế sơ đồ mạng ở mức mô hình liên quan đến việc chọn lựa mô hình mạng, giao thức mạng và thiết đặt các cấu hình cho các thành phần nhận dạng mạng. Mô hình mạng được chọn phải hỗ trợ được tất cả các dịch vụ đã được mô tả trong bảng Đặc tả yêu cầu hệ thống mạng. Mô hình mạng có thể chọn là Workgroup hay Domain (Client/Server) đi kèm với giao thức TCP/IP, NETBEUI hay IPX/SPX….

Ví dụ:

- Một hệ thống mạng chỉ cần có dịch vụ chia sẻ máy in và thư mục giữa những người dùng trong mạng cục bộ và không đặt nặng vấn đề an toàn mạng thì ta có thể chọn Mô hình Workgroup.
- Một hệ thống mạng chỉ cần có dịch vụ chia sẻ máy in và thư mục giữa những người dùng trong mạng cục bộ nhưng có yêu cầu quản lý người dùng trên mạng thì phải chọn Mô hình Domain. Nếu hai mạng trên cần có dịch vụ mail hoặc kích thước mạng được mở rộng, số lượng máy tính trong mạng lớn thì cần lưu ý thêm về giao thức sử dụng cho mạng phải là TCP/IP. Mỗi mô hình mạng có yêu cầu thiết đặt cấu hình riêng. Những vấn đề chung nhất khi thiết đặt cấu hình cho mô hình mạng là:
- Định vị các thành phần nhận dạng mạng, bao gồm việc đặt tên cho Domain, Workgroup, máy tính, định địa chỉ IP cho các máy, định cổng cho từng dịch vụ.
- Phân chia mạng con, thực hiện tìm đường đi cho thông tin trên mạng.

Cần phải xây dựng chiến lược khai thác và quản lý tài nguyên mạng ,chiến lược này nhằm xác định ai được quyền làm gì trên hệ thống mạng. Thông thường, người dùng trong mạng được nhóm lại thành từng nhóm và việc phân quyền được thực hiện trên các nhóm người dùng.

Căn cứ vào sơ đồ thiết kế mạng ở mức mô hình, kết hợp với kết quả khảo sát địa hình bước kế tiếp ta tiến hành thiết kế mạng ở mức vật lý. Sơ đồ mạng ở mức vật lý mô tả chi tiết về vị trí đi dây mạng ở địa hình, vị trí của các thiết bị nối kết mạng như Hub, Switch, Router, vị trí các máy chủ và các máy trạm. Từ đó đưa ra được một bảng dự trù các thiết bị mạng cần mua. Trong đó mỗi thiết bị cần nêu rõ: Tên thiết bị, thông số kỹ thuật, đơn vị tính, đơn giá,…

Lựa chọn phần cứng, phần mềm:

Dựa trên các phân tích yêu cầu và kinh phí dự kiến cho việc triển khai để lựa chọn thiết bị của các nhà cung cấp thiết bị mạng, phần mềm hệ điều hành và các phần mềm ứng dụng. Một mô hình mạng có thể được cài đặt dưới nhiều hệ điều hành khác nhau. Chẳng hạn với mô hình Domain, ta có nhiều lựa chọn như: Windows NT, Windows 2003, Netware, Unix, Linux,... Tương tự, các giao thức thông dụng như TCP/IP, NETBEUI, IPX/SPX cũng được hỗ trợ trong hầu hết các hệ điều hành. Chính vì thế ta có một phạm vi chọn lựa rất lớn. Quyết định chọn lựa hệ điều hành mạng thông thường dựa vào các yếu tố như:

- Giá thành phần mềm của giải pháp.
- Sự quen thuộc của khách hàng đối với phần mềm.
- Sự quen thuộc của người xây dựng mạng đối với phần mềm.

Hệ điều hành là nền tảng để cho các phần mềm sau đó vận hành trên nó. Giá thành phần mềm của giải pháp không phải chỉ có giá thành của hệ điều hành được chọn mà nó còn bao gồm cả giá thành của các phầm mềm ứng dụng chạy trên nó. Hiện nay có 2 xu hướng chọn lựa hệ điều hành mạng: các hệ điều hành mạng của Microsoft Windows hoặc các phiên bản của Unix, Linux. Sau khi đã chọn hệ điều hành mạng, bước kế tiếp là tiến hành chọn các phần mềm ứng dụng cho từng dịch vụ. Các phần mềm này phải tương thích với hệ điều hành đã chọn.

Tính toán giá thành:

Tính toán giá thành để đảm bảo các chỉ tiêu kỹ thuật, các yêu cầu của ứng dụng, tính khả
mở của hệ thống. Việc tính toán giá thành cần phải xem xét tới yếu tố đầu tư ban đầu và chi phí phải trả trong quá trình vận hành hệ thống.

Triển khai mẫu thử nghiệm:

Triển khai ở quy mô nhỏ nhưng vẫn minh họa được toàn bộ các yêu cầu về kỹ thuật, yêu cầu về ứng dụng làm cơ sở cho việc đánh giá khả năng và giá thành của mạng trước khi triển khai trên diện rộng. Khi bản thiết kế đã được thẩm định, bước kế tiếp là tiến hành lắp đặt phần cứng và cài đặt phần mềm mạng theo thiết kế. Cài đặt phần cứng liên quan đến việc đi dây mạng và lắp đặt các thiết bị nối kết mạng (Hub, Switch, Router) vào đúng vị trí như trong thiết kế mạng ở mức vật lý đã mô tả. Tiến trình cài đặt phần mềm bao gồm:

- Cài đặt hệ điều hành mạng cho các máy chủ, các máy trạm
- Cài đặt và cấu hình các dịch vụ mạng.

- Tạo người dùng, phân quyền sử dụng mạng cho người dùng.

Tiến trình cài đặt và cấu hình phần mềm phải tuân thủ theo sơ đồ thiết kế mạng mức mô hình đã mô tả. Việc phân quyền cho người dùng pheo theo đúng chiến lược khai thác và quản lý tài nguyên mạng. Nếu trong mạng có sử dụng router hay phân nhánh mạng con thì cần thiết phải thực hiện bước xây dựng bảng định tuyến trên các router và trên các máy tính.

Kiểm thử và đánh giá:

Sau khi đã cài đặt xong phần cứng và các máy tính đã được nối vào mạng, bước kế tiếp là kiểm tra sự vận hành của mạng. Trước tiên, kiểm tra sự nối kết giữa các máy tính với nhau. Sau đó, kiểm tra hoạt động của các dịch vụ, khả năng truy cập của người dùng vào các dịch vụ và mức độ an toàn của hệ thống. Nội dung kiểm thử dựa vào bảng đặc tả yêu cầu mạng đã được xác định lúc đầu. Đối chiếu với các yêu cầu xác định ban đầu, nếu không đáp ứng thì phải lặp lại các bước trên để tìm ra giải pháp tối ưu nhất có thể. Đánh giá bản thiết kế mạng cần phải dựa trên các tiêu chí sau:

- Đáp ứng yêu cầu người sử dụng
- Giá thành thấp
- Dễ cài đặt
- Dễ mở rộng
- Dễ cô lập trong trường hợp xảy ra lỗi.

Đối với thiết kế cho mạng diện rộng (thường là nhiệm vụ của các nhà cung cấp dịch vụ Internet), người ta chia thành ba lớp sau:

Lớp lõi:

Lớp lõi là trục xương sống của mạng thường dùng các bộ chuyển mạch có tốc độ cao, nó yêu cầu độ tin cậy cao, có công suất dư thừa, có khả năng tự chịu lỗi, có khả năng thích nghi cao, đáp ứng nhanh, dễ quản lý, có khả năng lọc gói, hay lọc các tiến trình đang truyền trong mạng.

Lớp phân phối:

Lớp phân phối là gianh giới giữa lớp truy nhập và lớp lõi của mạng. lớp phân tán thực hiện các chức năng đảm bảo gửi dữ liệu đến từng phân đoạn mạng, đảm bảo an ninh-an toàn, phân đoạn mạng theo nhóm công tác, chia miền, định tuyến giữa các mạng, chuyển môi trường truyền dẫn, định tuyến giữa các miền, tạo biên giới giữa các miền trong định tuyến tĩnh và động, thực hiện các bộ lọc gói tin, thực hiện các cơ chế đảm bảo chất lượng dịch vụ.

Lớp truy nhập:

Lớp truy nhập cung cấp các khả năng truy nhập cho người dùng cục bộ hay từ xa truy nhập vào mạng, thường được thực hiện bằng các bộ chuyển mạch trong một khu vực nhỏ.

2.5 Các yếu tố tạo nên hiệu năng mạng

Hiệu năng mạng là khái niệm cho biết hiệu suất hoạt động của hệ thống mạng. Hiệu năng chủ yếu được xác định bởi sự kết hợp của nhiều yếu tố, có những yếu như: Băng thông, thông lượng, thời gian đáp ứng, độ trễ, độ tin cậy, tỉ lệ lỗi, tốc độ xử lý của phần mềm ứng dụng, tính sẵn sàng của hệ thống. Tuỳ theo mục đích nghiên cứu cụ thể, hiệu năng có thể chỉ bao gồm một nhân tố nào đó hoặc là sự kết hợp một số trong các nhân tố nêu trên.

2.5.1 Các yếu tố đánh giá hiệu năng mạng

Có thể phân các yếu tố đánh giá hiệu năng thành hai loại: các yếu tố hướng tới người sử dụng và các yếu tố hướng tới hệ thống. Đối với người sử dụng, thời gian đáp ứng là yếu tố quan trọng nhất, đặc biệt trong các hệ thống thời gian thực hoặc các môi trường hệ thống tương tác. Đó là khoảng thời gian từ khi gửi một yêu cầu cho đến khi nhận được kết quả chính xác. Trong các hệ thống tương tác, đôi khi người ta sử dụng yếu tố thời gian phản ứng của hệ thống thay cho thời gian đáp ứng, yếu tố này được tính bằng khoảng thời gian từ khi dữ liệu yêu cầu đến được hệ thống cho đến khi yêu cầu chứa trong dữ liệu đó nhận được khe thời gian phục vụ đầu tiên. Đây là yếu tố thể hiện mức độ hiệu dụng của bộ lập lịch của hệ thống trong việc nhanh chóng cung cấp dịch vụ cho một yêu cầu mới đến. Trong các hệ thống mạng máy tính, các đại lượng thời gian đáp ứng, thời gian phản ứng của hệ thống đều được xem là các biến ngẫu nhiên, vì vậy người ta thường nói về phân bố, kỳ vọng, phương sai... của chúng.

Các yếu tố hướng tới hệ thống điển hình là băng thông, thông lượng và thời gian trễ. Băng thông được định nghĩa là lượng thông tin tối đa có thể chuyển tải trên mạng trong một đơn vị thời gian, trong khi đó thông lượng được định nghĩa là lượng thông tin thực tế được vận chuyển qua mạng trong một đơn vị thời gian. Đơn vị thông tin ở đây có thể là bit, byte hay gói số liệu... Nếu các đơn vị thông tin đi vào mạng theo một cơ chế độc lập với trạng thái của mạng, thì thông lượng cũng chính bằng tốc độ đến trung bình nếu mạng vẫn còn có khả năng vận chuyển, không dẫn đến trạng thái bị tắc nghẽn. Một số trường hợp người ta sử dụng đại lượng hệ số sử dụng đường truyền, đó là tỉ số của thông lượng trên băng thông. Thời gian trễ là thời gian trung bình để vận chuyển một gói số liệu qua mạng từ nguồn tới đích. Cũng có trường hợp người ta sử dụng đại lượng thời gian trễ chuẩn hoá, đó là tỉ số của thời gian trễ trên một tham số thời gian nào đó, thí dụ thời gian cần thiết để truyền một gói tin.

2.5.2 Vai trò của việc đánh giá hiệu năng mạng máy tính

Trong suốt lịch sử phát triển của mạng máy tính, vấn đề đánh giá và dự đoán hiệu năng mạng luôn thu hút sự quan tâm của những người nghiên cứu và thiết kế mạng, mục đích chính là để nắm được và cải thiện đặc trưng chi phí và hiệu năng. Yêu cầu đánh giá và dự đoán hiệu năng mạng đặt ra ngay từ khi người ta thiết kế kiến trúc của hệ thống cho đến khi mạng đã được lắp đặt và đưa vào hoạt động. Trong giai đoạn đầu của quá trình thiết kế, người ta thường phải dự đoán hai điều: Thứ nhất, các ứng dụng sẽ chạy trên mạng và các yêu cầu dịch vụ mà các ứng dụng này đòi hỏi hệ thống mạng phải đáp ứng, thứ hai lựa chọn

kiến trúc dựa trên các công nghệ phần cứng và phần mềm sẽ được phát triển và đưa ra thị trường trong tương lai khi hệ thống mạng bước vào giai đoạn triển khai thực hiện.

Trong giai đoạn thiết kế chi tiết, việc dự đoán và đánh giá hiệu năng sẽ trở nên cụ thể hơn. Thí dụ sẽ chọn đường truyền vật lý như thế nào, các đặc tính của đường truyền được chọn sẽ ảnh hưởng thế nào đến hiệu năng của mạng. Các kỹ thuật được dùng để dự đoán và đánh giá hiệu năng mạng trong giai đoạn thiết kế và triển khai thực hiện có khi chỉ là các tính toán bằng tay, nhưng cũng có khi là các mô phỏng rất tinh vi. Việc so sánh hiệu năng dự đoán với hiệu năng thực tế đạt được thường giúp cho nhà nghiên cứu thấy được các khiếm khuyết chính trong thiết kế hoặc các lỗi trong việc lập trình hệ thống. Ngày nay, việc dự đoán và đánh giá hiệu năng thường được người ta coi là một phần không thể thiếu được của công việc thiết kế và triển khai thực hiện hệ thống. Sau khi triển khai hệ thống mạng, hai công việc quan trọng cần phải thực hiện: Cấu hình mạng và tinh chỉnh mạng.

Cấu hình mạng:

Sau khi mạng đã được triển khai thực hiện, việc dự đoán và đánh giá hiệu năng mạng đối với các ứng dụng cụ thể cũng có ý nghĩa quan trọng. Nhằm đạt được sự tối ưu hoá, nhà sản xuất phải chỉ ra được các cách kết hợp và tổ chức phần cứng và phần mềm mạng để đem lại một giải pháp tốt nhất cho các yêu cầu của khách hàng, việc này thường được gọi là định cấu hình mạng. Mặc dù có thể vẫn sử dụng các công cụ và phương pháp đã được sử dụng trong giai đoạn phát triển hệ thống, nhưng cần phải bổ sung thêm một số yếu tố nữa. Đặc điểm môi trường của người sử dụng sản phẩm mạng cần được biểu diễn bằng các tham số định lượng và đưa vào mô hình mô phỏng hiệu năng.

Điều chỉnh hiệu suất hoạt động của hệ thống:

Sau khi hệ thống mạng đã được lắp đặt và đi vào vận hành, người quản trị hệ thống cần phải làm sao cho hệ thống đạt được hiệu năng hoạt động tốt nhất, việc này được gọi là điều chỉnh hiệu suất hoạt động của hệ thống. Đối với các hệ thống mạng, việc tìm ra được điểm làm việc tối ưu và ổn định trên toàn mạng là rất khó, nó phụ thuộc vào nhiều yếu tố khác nhau và đòi hỏi kinh nghiệm thực tiễn của người quản trị hệ thống.

2.5.3 Các phương pháp đánh giá hiệu năng mạng

Có nhiều phương pháp đánh giá hiệu năng mạng máy tính, có thể chia chúng làm ba loại: mô hình giải tích, mô phỏng và phương pháp đo lường.

Mô hình giải tích:

Trong các mạng chuyển mạch gói, gói tin là các khối dữ liệu có chiều dài thay đổi được, được truyền qua mạng từ nguồn tới đích theo một con đường nào đó do hệ thống mạng quyết định. Các tài nguyên mạng sẽ được chia sẻ giữa các gói số liệu khi chúng đi qua mạng. Số lượng và chiều dài các gói số liệu đi vào hoặc đi qua mạng tại mọi thời điểm, thời gian kéo dài các cuộc kết nối v.v., tất cả các tham số này nói chung thay đổi theo thời gian và hiện trạng của hệ thống mạng. Vì vậy, để nêu ra các tiêu chuẩn đo lường định lượng về hiệu năng, cần

phải sử dụng các khái niệm về xác suất để nghiên cứu sự tương tác của chúng với mạng. Lý thuyết hàng đợi đóng vai trò then chốt trong việc phân tích mạng, bởi vì đó là công cụ toán học thích hợp nhất để phát biểu và giải các bài toán về hiệu năng. Theo phương pháp này, chúng ta viết ra các mối quan hệ hàm giữa các tiêu chuẩn hiệu năng cần quan tâm và các tham số của hệ thống mạng bằng các phương trình có thể giải được bằng giải tích.

Mô phỏng:

Mô phỏng là sự bắt chước một hay nhiều khía cạnh của sự vật có thực, bằng một cách nào đó càng giống càng tốt. Trong vấn đề đánh giá hiệu năng mạng, mô phỏng được hiểu là một kỹ thuật sử dụng máy tính điện tử số để làm các thí nghiệm về mạng có liên quan đến thời gian. Mô hình mô phỏng giả lập hành vi hoạt động của mạng, ngay cả khi người nghiên cứu chỉ quan tâm đến giá trị trung bình của một số độ đo trong trạng thái dừng. Cấu trúc và độ phức tạp của bộ mô phỏng phụ thuộc vào phạm vi của thí nghiệm mô phỏng, nó thường được xây dựng có cấu trúc, cho phép mô-đun hoá chương trình mô phỏng thành tập các chương trình con, sao cho việc sửa đổi hoặc bổ sung các chương trình con được dễ dàng. Ngoài ra, chương trình mô phỏng cũng phải được xây dựng sao cho đạt được tốc độ cao nhằm làm giảm thời gian chạy mô phỏng càng nhiều càng tốt.

Phương pháp đo lường:

Đây là phương pháp xác định hiệu năng dựa trên việc đo các tham số mạng cấu thành độ đo hiệu năng cần quan tâm trên mạng thực, việc đo hiệu năng nhằm thực hiện các nhiệm vụ sau:

- Giám sát hiệu năng của mạng.
- Thu thập số liệu để lập mô hình dữ liệu vào cho các phương pháp đánh giá hiệu năng bằng giải tích hoặc mô phỏng.
- Kiểm chứng các mô hình khác dựa trên các số liệu đo được.

Đo hiệu năng không chỉ quan trọng trong các giai đoạn triển khai thực hiện và tích hợp hệ thống mà còn cả trong các giai đoạn lắp đặt và vận hành hệ thống. Bởi vì sau khi lắp đặt và đưa vào sử dụng, mỗi một hệ thống cụ thể sẽ có một tải hệ thống và các độ đo hiệu năng được quan tâm riêng của nó, cho nên sau khi lắp đặt, người ta thường phải điều chỉnh cấu hình cho phù hợp. Các tham số cấu hình sẽ được chọn sau khi các phép đo hiệu năng cho thấy các tham số cấu hình này làm cho hệ thống đạt được hiệu năng tốt nhất. Trong thực tế, mọi người đều thừa nhận tầm quan trọng của việc đo và đánh giá hiệu năng. Chúng ta có thể thấy rõ điều này qua việc, hầu như tất cả các hệ thống mạng đều tích hợp bên trong nó các công cụ đo và đánh giá hiệu năng; nhờ đó có thể đo hiệu năng bất cứ lúc nào trong suốt thời gian tồn tại của hệ thống.

So sánh các phương pháp đánh giá hiệu năng:

Phương pháp mô hình Giải tích: Sử dụng mô hình giải tích có thể thay đổi các tham số hệ thống và cấu hình mạng trong một miền rộng với chi phí thấp mà vẫn có thể đạt được các kết quả mong muốn. Tuy nhiên, các mô hình giải tích mà chúng ta xây dựng thường là không thể giải được nếu không được đơn

giản hoá nhờ các giả thiết, hoặc được phân rã thành các mô hình nhiều cấp. Các mô hình giải được thường rất đơn giản hoặc khác xa thực tế, cho nên phương pháp này thường chỉ được sử dụng ngay trong giai đoạn đầu của việc thiết kế mạng, giúp cho người thiết kế dự đoán được các giá trị giới hạn của hiệu năng. Ngoài ra, các kết quả của phương pháp này bắt buộc phải được kiểm nghiệm bằng kết quả của các phương pháp khác, như mô phỏng hoặc đo.

Phương pháp mô phỏng: Trong những trường hợp mô hình giải tích mà chúng ta nhận được, dù đã được đơn giản hoá, hoặc phân rã nhưng vẫn không thể giải được bằng toán học, khi đó chúng ta sẽ chỉ còn một phương pháp là mô phỏng. Phương pháp mô phỏng có thể được sử dụng ngay trong giai đoạn đầu của việc thiết kế hệ thống mạng cho đến giai đoạn triển khai thực hiện và tích hợp hệ thống. Phương pháp này nói chung, đòi hỏi một chi phí rất cao cho việc xây dựng bộ mô phỏng cũng như kiểm chứng tính đúng đắn của nó. Tuy nhiên, sau khi đã xây dựng xong bộ mô phỏng, người nghiên cứu có thể tiến hành chạy chương trình mô phỏng bao nhiêu lần tuỳ ý, với độ chính xác theo yêu cầu và chi phí cho mỗi lần chạy thường là rất thấp. Các kết quả mô phỏng nói chung vẫn cần được kiểm chứng, bằng phương pháp giải tích hoặc đo lường. Phương pháp mô hình giải tích và mô hình mô phỏng đóng vai trò rất quan trọng trong việc thiết kế và triển khai thực hiện hệ thống, đặc biệt là ở giai đoạn đầu của dự án.

Phương pháp đo lường: Phương pháp đo chỉ có thể thực hiện được trên đang hoạt động, nó đòi hỏi chi phí cho các công cụ đo và cho việc tiến hành đo. Việc đo cần được tiến hành tại nhiều điểm trên mạng thực ở những thời điểm khác nhau và cần lặp đi lặp lại trong một khoảng thời gian đủ dài, thậm chí có thể dài đến hàng tháng. Ngoài ra, người nghiên cứu phải có kiến thức về lý thuyết thống kê thì mới có thể rút ra được các kết luận hữu ích từ các số liệu thu thập được. Mặc dầu vậy, phương pháp đo lường vẫn có thể không phát hiện hoặc dự đoán được các hiện tượng đặc biệt của mạng.

CHƯƠNG 3: TẦNG ỨNG DỤNG

3.1 Các khái niệm và cài đặt các giao thức tầng ứng dụng

Ứng dụng mạng là động lực phát triển của mạng máy tính, đã có nhiều phát minh đột phá trong việc phát triển các ứng dụng mạng. Bắt đầu từ thập niên 60, những ứng dụng đơn giản tương tác với người dùng qua chế độ dòng lệnh (text-based) đã trở nên phổ biến như truy cập máy tính từ xa (telnet), thư điện tử (email), truyền tập tin (ftp), nhóm thông tin (newsgroup), và trò chuyện từ xa (chat). Hiện nay, những ứng dụng đa phương tiện phức tạp hơn như World Wide Web, điện thoại trực tuyến, hội thảo từ xa, chia sẻ tập tin ngày càng trở nên quen thuộc.

Mặc dù chương trình ứng dụng mạng có nhiều loại khác nhau, có thể có nhiều thành phần tương tác với nhau, nhưng lõi của chúng là phần mềm. Phần mềm ứng dụng mạng được cài đặt phân tán trên các thiết bị đầu cuối của người sử dụng như máy tính, điện thoại di động… Ví dụ, với việc truy nhập trang tin điện tử có hai phần mềm tương tác với nhau: phần mềm trình duyệt trong máy tính của người dùng và phần mềm cho phép truy nhập vào nội dung được cài đặt trên máy chủ web.

Việc kết nối được thực hiện giữa các tiến trình chứ không phải giữa các chương trình phần mềm, tiến trình là một chương trình chạy trên thiết bị đầu cuối. Khi các tiến trình chạy trên cùng một thiết bị, chúng sẽ trao đổi dữ liệu với nhau thông qua cơ chế truyền thông liên tiến trình, hệ điều hành của thiết bị đầu cuối người sử dụng sẽ kiểm soát cơ chế này. Việc kết nối như vậy sẽ được thực hiện bằng cách trao đổi bản tin qua mạng máy tính. Tiến trình gửi sẽ tạo và gửi bản tin qua mạng, tiến trình nhận sẽ nhận bản tin và có thể phản hồi lại bằng cách gửi một bản tin trả lời. Ứng dụng mạng có các giao thức định nghĩa khuôn dạng, thứ tự trao đổi các bản tin cũng như hành vi của mỗi bên khi nhận được bản tin.

3.1.1 Mô hình dịch vụ của tầng ứng dụng

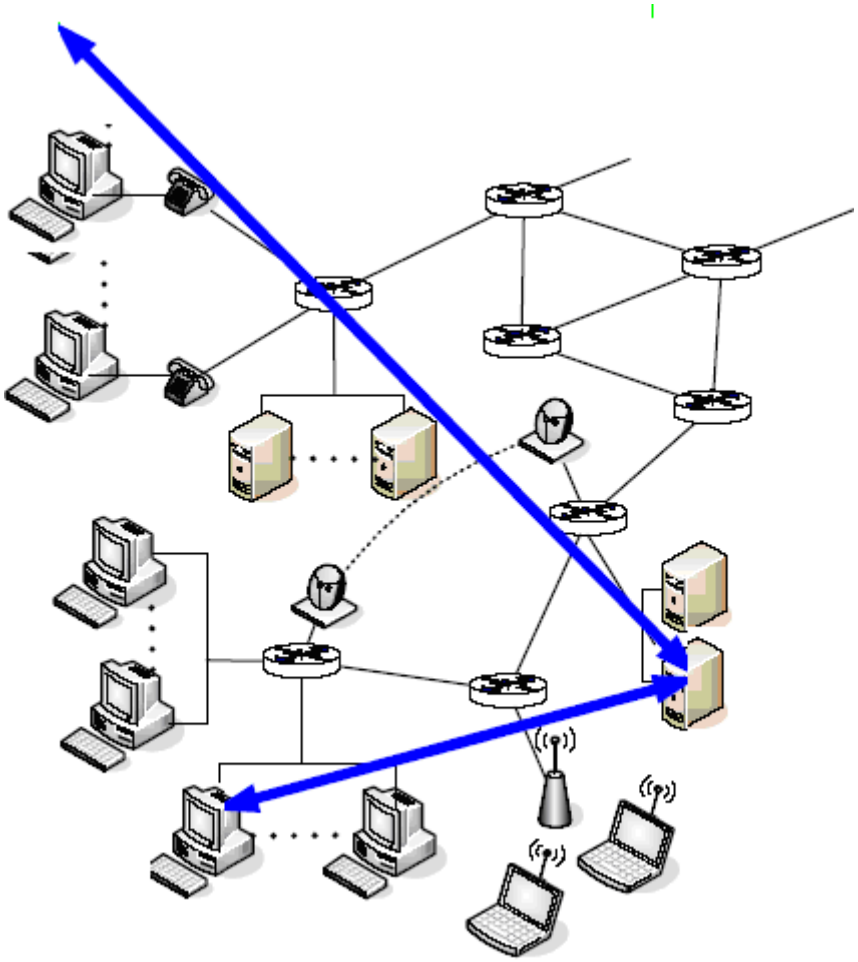
Cần phân biệt ứng dụng mạng và giao thức tầng ứng dụng, giao thức tầng ứng dụng chỉ và một phần (cho dù là phần quan trọng) của ứng dụng mạng. Ví dụ Web là ứng dụng mạng cho phép người dùng lấy các thông tin từ máy chủ web nhưng để làm được điều đó thì ứng dụng mạng phải sử dụng nhiều giao thức khác nhau của tầng ứng dụng. Ứng dụng mạng bao gồm nhiều thành phần, như tiêu chuẩn định dạng văn bản (HTML), trình duyệt Web (Netscape Navigator hay Microsoft Internet Explorer), phần mềm máy chủ web (Apache, Microsoft, và Netscape), và giao thức tầng ứng dụng. Giao thức tầng ứng dụng của Web - HTTP (Hypertext Transfer Protocol, RFC 2616), định nghĩa cách thức chuyển bản tin giữa trình duyệt và máy chủ web, như vậy HTTP chỉ là một phần của ứng dụng Web.

Một ví dụ khác là ứng dụng thư điện tử, nó cũng bao gồm nhiều thành phần: máy chủ thư điện tử có chức năng như một hòm thư, máy chủ đọc thư điện tử cho phép người dùng đọc và gửi thư, chuẩn định nghĩa cấu trúc của thư

điện tử và giao thức tầng ứng dụng định nghĩa cách thức chuyển bản tin giữa máy chủ thư điện tử và máy chủ đọc thư điện tử, cũng như ý nghĩa của một số trường trong thư (ví dụ các tiêu đề thư : người nhận, người gửi…). Giao thức tầng ứng dụng cho thư điện tử là SMTP (Simple Mail Transfer Protocol, RFC 821). Do đó, giao thức SMTP chỉ là một phần của ứng dụng thư điện tử.

Giao thức tầng ứng dụng định nghĩa cách thức truyền bản tin giữa các tiến trình ứng dụng chạy trên các thiết bị khác nhau, nó xác định:

- Kiểu bản tin trao đổi, ví dụ như bản tin yêu cầu hay bản tin trả lời.
- Cú pháp của bản tin, ví dụ các trường trong bản tin cũng như cách xác định chúng.
- Ý nghĩa của các trường.
- Qui tắc xác định tiến trình gửi và trả lời bản tin khi nào và như thế nào.



Hình 3.1 Các ứng dụng trên mạng

Nhiều giao thức tầng ứng dụng được đặc tả trong các, ví dụ: đặc tả của HTTP là HTTP RFC 2616. Nếu người thiết kế trình duyệt tuân theo các qui tắc của HTTP RFC 2616, trình duyệt sẽ có thể lấy được các trang WEB từ bất kỳ máy chủ web nào tuân theo các qui tắc của RFC 2616.

3.1.2 Mô hình khách chủ

Trong mô hình khách/chủ (Client/Server), một hay một số máy tính được thiết lập để cung cấp các dịch vụ như file server, mail server, Web server, Printer server, … Các máy tính được thiết lập để cung cấp các dịch vụ được gọi là Server, còn các máy tính truy cập và sử dụng dịch vụ thì được gọi là Client. Giao thức ứng dụng mạng thường chia ra hai: máy khách và máy chủ, phần máy khách trong thiết bị liên lạc với phần máy chủ trong một thiết bị khác. Ví dụ: trình duyệt Web thuộc máy khách và phần cài đặt trang web thuộc máy chủ. Trong nhiều ứng dụng, máy tính sẽ thực hiện cả phần máy khách và phần máy chủ của ứng dụng.

3.1.3 Mô hình ngang hàng

Trong mô hình ngang hàng (*peer-to-peer*)các máy tính trong mạng có thể hoạt động vừa như một Client vừa như một Server. Ví dụ xét một phiên làm việc Telnet giữa máy A và máy B. Nếu máy A bắt đầu trước (có nghĩa là người dùng ở máy A đăng nhập vào máy B), khi đó máy A chạy phía máy khách và máy B chạy phía máy chủ của ứng dụng. Mặt khác, nếu máy B bắt đầu trước thì máy B chạy phía máy khách của ứng dụng. FTP - được dùng để truyền tập tin giữa hai máy là ví dụ khác. Sau khi thiết lập phiên làm việc FTP giữa hai máy tính, mỗi máy đều có thể truyền tập tin tới máy kia trong suốt phiên làm việc. Tuy nhiên giống như hầu hết các ứng dụng mạng, máy nào bắt đầu trước được coi là máy khách. hơn nữa, máy tính có thể chạy cả phía máy khách và máy chủ tại cùng một thời điểm. Ví dụ: máy chủ thư điện tử đóng vai trò máy khách khi gửi thư và đóng vai trò máy chủ của khi nhận thư.

3.2 Các giao thức thường dùng tại lớp ứng dụng

Các ứng dụng mạng ngày càng đa dạng và phong phú, phần này sẽ trình bày một số giao thức dùng trong các ứng dụng phổ biến: Web, truyền tập tin, thư điện tử và dịch vụ tên miền. Web là ứng dụng đầu tiên và vì Web được sử dụng phổ biến và giao thức tầng ứng dụng của nó - HTTP, tương đối đơn giản và minh họa nhiều đặc trưng cơ bản của giao thức. Tiếp theo là ứng dụng truyền tập tin, bởi vì ứng dụng này có nhiều đặc điểm trái ngược với HTTP. Chúng ta cũng sẽ nghiên cứu thư điện tử, một trong những ứng dụng đầu tiên và thông dụng nhất của Internet. Thư điện tử ngày nay sử dụng nhiều giao thức tầng ứng dụng. Web, truyền tập tin, và thư điện tử đều yêu cầu một dịch vụ truyền đáng tin cậy, không có yêu cầu thời gian và yêu cầu về băng thông. Do vậy ba ứng dụng này sử dụng TCP ở tầng vận tải. Ứng dụng thứ tư là DNS (Domain Nam System) cung cấp dịch vụ chuyển đổi tên miền thành địa chỉ IP. Người dùng không tương tác trực tiếp với DNS mà yêu cầu dịch vụ DNS gián tiếp thông qua các ứng dụng khác (ví dụ Web, truyền file và thư điện tử). DNS minh họa rõ việc triển khai một cơ sở dữ liệu phân tán trên mạng như thế nào.

3.2.1 Giao thức truy nhập trang web HTTP

Cho đến những năm 1990, Internet chỉ được sử dụng trong các cơ quan nghiên cứu, các trường đại học với các dịch vụ đơn giản như truy cập từ xa,

truyền tập tin, nhận và gửi thư điện tử. Mặc dù các ứng dụng này đã phổ biến - nhưng Internet về cơ bản vẫn chỉ được biết tới trong cộng đồng nghiên cứu. Vào đầu thập niên 90, ứng dụng quan trọng nhất của Internet - World Wide Web xuất hiện, và nhanh chóng được mọi người chấp nhận. Nó thay đổi cách thức tương tác giữa con người và môi trường làm việc. Chính điều này đã giúp đưa Intemet từ một trong rất nhiều mạng thông tin (ví dụ mạng trực tuyến Prodigy, American Onlie hay Compuserve, hệ thống thông tin quốc gia : Minitee/Tranpac ở Pháp, Private X25, Frame Relay) thành một mạng lớn nhất toàn cầu.

Lịch sử phát triển của ngành công nghệ viễn thông ảnh hưởng lớn đến xã hội loài người. Công nghệ đầu tiên là điện thoại - được phát minh vào năm 1870. Điện thoại cho phép hai người nói chuyện trực tiếp mà không cần ở trong cùng một vùng. Nó có những ảnh hưởng cả tốt lẫn xấu đến xã hội. Công nghệ tiếp theo là truyền thanh, truyền hình - ra đời vào những năm 1920-1930, giúp con người thu nhận một tượng thông tin rất lớn bằng âm thanh và hình ảnh, và tác động lớn đến xã hội. Có lẽ công nghệ thứ ba làm thay đổi cuộc sống và công việc của con người là Web. Sức lôi cuốn của Web đối với con người là ở chỗ Web hoạt động theo yêu cầu, nghĩa là có thể nhận được thông tin cần thiết vào các thời điểm cần thiết. Điều này khác so với công nghệ quảng bá (truyền thanh, truyền hình) chỉ phát đi những nội dung có sẵn tại những thời điểm định trước. Ngoài ra Web có nhiều đặc điểm lý thú khác. Ai cũng có thể dễ dàng trở thành các nhà xuất bản; các siêu liên kết và các công cụ tìm kiếm giúp ta tìm kiếm qua nhiều trang web. Các hình ảnh đồ họa và hoạt hình khuấy động thị giác. Các thành phần khác như: Form, Java applet, Active X cho phép tương tác tới các website khác.

3.2.1.1 Tổng quan về giao thức HTTP

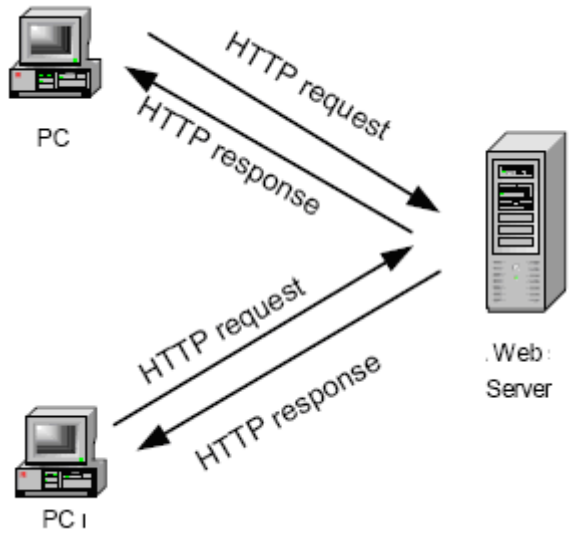
Hyper Text Transfer Protocol (HTTP) - giao thức tầng ứng dụng của Web - là trái tim của Web. HTTP được triển khai trên cả hai phía máy khách và máy chủ. Các tiến trình máy khách và máy chủ trên các hệ thống đầu cuối khác nhau giao tiếp với nhau thông qua việc trao đổi các bản tin HTTP. HTTP quy định cấu trúc bản tin cũng như cách thức trao đổi bản tin giữa máy khách và máy chủ. Trước khi nói về HTTP, chúng ta hãy nói lại các thuật ngữ về web.

Trang Web (webpage - hay còn gọi là một tập tin) chứa các đối tượng (Object). Đối tượng đơn giản chỉ là một file như file HTML, file ảnh JPEG, file ảnh GIF file java applet, một đoạn âm thanh… Đối tượng được xác định qua địa chỉ URL. Trang Web chứa một file HTML cơ sở và tham chiếu đến các đối tượng khác. Ví dụ một trang web chứa một tập tin HTML văn bản và 5 đối tượng ảnh JPEG khi đó trang web cỏ 6 đối tượng: 1 file văn bản HTML và 5 file ảnh. File HTML cơ sở này tham chiếu đến các đối tượng khác thông qua địa chỉ URL. Mỗi địa chỉ URL có hai thành phần là: tên của máy chủ và đường dẫn của đối tượng. Đây là một địa chỉ URL www.ptit.edu.vn/Portals/0/ptitlogo72.gif

www.ptit.edu.vn là tên máy chủ và Portals/0/ptitlogo72.gif là đường đẫn đối tượng.

Trình duyệt (Browser) - chương trình giao tiếp người dùng của ứng dụng Web cho phép hiển thị trang Web. Browser là phía máy khách của giao thức HTTP. Hiện nay có rất nhiều phần mềm trình duyệt nhưng phổ biến nhất là Nestcape Communication và Microsoft Internet Explorer. Máy chủ web lưu giữ các đối tượng web và được xác định qua địa chỉ URL . Phần mềm Máy chủ web là phía máy chủ của giao thức HTTP. một số phần mềm Máy chủ web phổ biến là Apache, Microsoft Internet Information Máy chủ và Nestcape Enterprise Máy chủ.

HTTP xác định cách thức trình duyệt yêu cầu trang web từ web máy chủ cũng như cách thức máy chủ gửi trang web được yêu cầu tới trình duyệt. dưới đây chúng ta sẽ nói rõ hơn về quá trình trao đổi giữa máy khách và máy chủ. Hình 2.6 minh họa quá trình này. Khi người dùng yêu cầu một đối tượng (ví dụ kích vào một siêu liên kết), browser sẽ gửi một bản tin HTTP tới máy chủ yêu cầu đối tượng đó. Máy chủ nhận được yêu cầu và trả lời bằng cách gửi lại một thông điệp trả lời chứa đối tượng được yêu cầu.



Hình 3.2 tương tác máy khách/máy chủ

Cho tới những năm 1997, phần lớn các trình duyệt Web và Máy chủ web tuân thủ phiên bản HTTP 1.0 (đặc tả trong RFC 1945), từ năm 1998 một số trình duyệt và máy chủ web sử dụng phiên bản 1.1 theo khuyến nghị RFC 2616. Phiên bản mới này tương thích với phiên bản 1.0, nghĩa là Web máy chủ dùng phiên bản 1.1 có thể nói chuyện được với trình duyệt sử dụng phiên bản 1.0 và ngược lại cả phiên bản 1.0 và 1.1 đều sử dụng TCP làm giao thức ở tầng vận tải. HTTP máy khách khởi tạo một kết nối TCP tới HTTP máy chủ. Sau khi thiết lập được kết nối, cả tiến trình browser và Máy chủ web đều truy cập tới TCP thông qua socket.

Máy chủ nhận bản tin yêu cầu này và gửi bản tin trả lời qua socket. Sau khi gửi bản tin qua socket thì bản tin nằm ngoài tầm kiểm soát của máy khách và chính thực thể TCP chịu trách nhiệm chuyển nó sang phía bên kia. Giao thức TCP cung cấp dịch vụ truyền tin tin cậy cho HTTP, như vậy bản tin của tiến trình máy trạm sẽ được chuyển tải nguyên vẹn đến máy chủ và ngược lại. Giao

thức HTTP không giải quyết việc mất mát dữ liệu mà việc này là công việc của giao thức TCP.

Một điểm quan trọng là máy chủ gửi các đối tượng được yêu cầu cho máy khách mà không ghi lại bất kỳ một thông tin trạng thái nào của máy khách. Nếu máy khách nào đó yêu cầu lại cùng một đối tượng thì máy chủ sẽ không thể trả lời cho máy khách rằng đối tượng đó vừa được gửi cho máy khách, máy chủ sẽ gửi lại cho máy khách đối tượng đó như thể nó không biết việc gửi lần trước. HTTP máy chủ không nhớ các thông tin về máy khách, vì thế HTTP được gọi là giao thức không trạng thái.

Kết nối liên tục và không liên tục

HTTP hỗ trợ cả hai cách kết nối liên tục và không liên tục. HTTP 1.0 sử dụng kết nối không liên tục. Chế độ mặc định của HTTP 1.1 là kết nối liên tục.

Kết nối không liên tục

Giả sử trang web có chứa một tập tin HTML cơ sở và 10 file ảnh JPEG và đồng thời cả 11 đối tượng này cùng ở trên một máy chủ, địa chỉ của file HTML này là www.ptit.edu.vn/english/index.html, các bước thực hiện như sau:

1. HTTP máy khách khởi tạo một kết nối TCP tới máy chủ có địa chỉ là www.ptit.edu.vn. Cổng 80 là cổng được HTTP máy chủ sử dụng để "Lắng nghe" các yêu cầu lấy trang Web từ máy khách thông qua giao thức HTTP.
2. HTTP máy khách gửi bản tin yêu cầu qua socket tới thực thể TCP đã được kết nối ở bước trước. Thông điệp bao gồm đường dẫn english/index.html.
3. HTTP máy chủ nhận được bản tin yêu cầu từ socket, lấy đối tượng english/index.html trong bộ nhớ của mình (ổ cứng hoặc RAM), đặt đối tượng này vào trong một bản tin trả lời và gửi đi qua socket.
4. HTTP máy chủ yêu cầu thực thể TCP kết thúc kết nối (nhưng nó không đóng lại thực sự cho đến khi máy khách nhận được bản tin).
5. HTTP máy khách nhận được bản tin trả lời, kết nối được đóng lại. Bản tin chỉ ra rằng nó chứa một đối tượng là file HTML. Client sẽ lấy file đó ra từ bản tin trả lời. File HTML tham chiếu đến 10 đối tượng ảnh JPEG.
6. Bốn bước đầu được lặp lại cho mỗi đối tượng ảnh được tham chiếu trong file HTML.

Khi nhận được bản tin trả lời có chứa trang Web, browser sẽ hiển thị trang web. Các browser khác nhau thì có thể có các cách hiển thị khác nhau đối với cùng một trang web. HTTP không ảnh hưởng gì đối với cách hiển thị trang web của máy khách. Các đặc tả trong HTTP chỉ định nghĩa giao thức truyền thông giữa tiến trình máy khách và máy chủ mà thôi.

Các bước ở trên sử dụng cách kết nối không liên tục vì sau khi gửi đi một đối tượng thì máy chủ sẽ đóng kết nối TCP lại, kết nối không được sử dụng để lấy các đối tượng khác. lưu ý rằng mỗi kết nối TCP chuyển duy nhất một bản tin yêu cầu và một bản tin trả lời, như vậy trong ví dụ trên, máy khách yêu cầu toàn bộ đối tượng trên trang web thì sẽ có thể có tới 11 kết nối TCP được thiết lập.

Trong ví dụ trên, chúng ta không hề nói đến việc máy khách nhận được 10 file ảnh JPEG qua 10 liên kết TCP riêng rẽ hay một số file được nhận qua cùng một kết nối. Trên thực tế, người dùng có thể cấu hình cho trình duyệt điều khiển mức độ song song của các kết nối. Chế độ mặc định của trình duyệt thường là từ 5 đến 10 kết nối TCP song song và mỗi kết nối kiểm soát một cặp bản tin yêu cầu/trả lời. Nếu người dùng không thích thì có thể đặt số kết nối song song tối đa là 1, trong trường hợp này 10 kết nối được thiết lập riêng lẻ. Trong chương sau chúng ta sẽ thấy rằng cách kết nối song song làm giảm thời gian trả lời.

Kết nối liên tục:

Có một vài nhược điểm trong kết nối không liên tục: Thứ nhất, khi liên kết mới được tạo ra, phía máy khách và máy chủ phải tạo ra vùng đệm TCP (buffer) cũng như lưu giữ các biến TCP. Điều này chính là gánh nhng cho máy chủ khi có nhiều máy khách cùng yêu cầu một lúc.

Với cách kết nối liên tục, máy chủ không đóng liên kết TCP sau khi gửi bản tin trả lời. Các bản tin yêu cầu và trả lời sau đó (giữa cùng một máy khách và máy chủ) được gửi qua cùng một kết nối. Trong ví dụ trên, toàn bộ đối tượng trong trang Web (một file HTML và 10 file ảnh JPEG) được truyền nối tiếp nhau trên cùng một kết nối TCP. Ngoài ra, có thể các trang web khác trên cùng máy chủ có thể được truyền qua một kết nối TCP. Thông thường thì HTTP máy chủ đóng liên kết khi liên kết không được sử dụng trong một khoảng thời gian nào đó.

Chế độ làm việc mặc định của phiên bản HTTP 1.1 và gửi liên tục. Trong trường hợp này, HTTP máy khách gửi yêu cầu khi nó nhận được một tham chiếu (ví dụ một siêu liên kết, hay tham chiếu đến file ảnh) vì vậy máy khách có thể gửi các yêu cầu liên tiếp. Khi máy chủ nhận được yêu cầu thì nó sẽ gửi các đối tượng nối tiếp nhau.

3.2.1.2 Khuôn dạng của bản tin HTTP

Các đặc tả HTTP 1.0 (RFC 1945) và HTTP 1.1 (RFC 2016) đặc tả khuôn dạng bản tin HTTP. Có hai kiểu khuôn dạng HTTP: bản tin yêu cầu và bản tin trả lời.

Bản tin yêu cầu HTTP:

Một bản tin yêu cầu thường có dạng sau:

GET /english/index.html HTTP/1.l Host: www.ptit.edu.vn

Connection :close

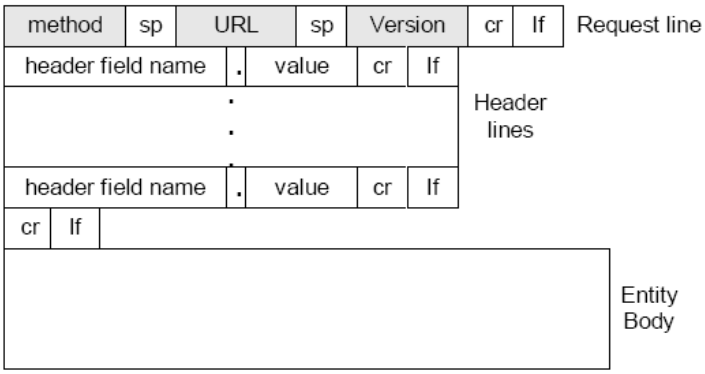
User-agent :Mozzilla /4. 0

Accept- language:Fr

(extra carry return line feed)

Trước hết ta thấy rằng bản tin được viết bằng mã ASCII - vì thế bất kỳ máy tính thông thường nào cũng có thể đọc được. Thứ hai, bản tin gồm 5 dòng và mỗi dòng đều kết thúc bởi lặp ký tự đặc biệt Carriage Return (CR=13h) và Line Feed (LF-10h). Trên thực tế một bản tin có thể có nhiều dòng hơn. Dòng

đầu tiên của bản tin được gọi là dòng yêu cầu (request line), các dòng sau gọi là tiêu đề (header). Dòng yêu cầu có 3 trường: trường method, trường địa chỉ URL và trường phiên bản HTTP. Trường method nhận một trong ba giá trị: GET, POST và HEAD. Phần lớn các yêu cầu sử dụng phương thức GET. Phương thức này được trình duyệt sử dụng để yêu cầu đối tượng có địa chỉ URL. Trong ví dụ trên thì trình duyệt yêu cầu đối tượng somedir/page.html. Trường phiên bản xác định phiên bản giao thức HTTP (trong ví dụ là 1.1).



Hình 3.3 Khuôn dạng chung của bản tin yêu cầu

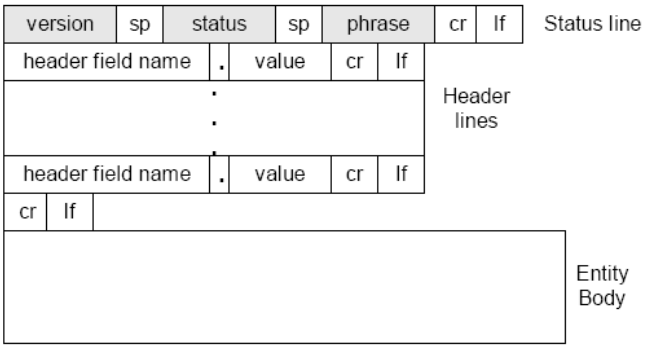
Bây giờ hãy xét các trường trong tiêu đề. Host: www.ptit.edu.vn là địa chỉ của máy tính có chứa đối tượng được yêu cầu. Ý nghĩa của trường Connection: close là trình duyệt yêu cầu máy chủ không sử dụng cách kết nối liên tục và yêu cầu máy chủ đóng kết nối lại sau khi đã gửi đi đói tượng được yêu cầu. mặc đù máy khách sử dụng phiên bản HTTP 1.1 nhưng nó không sử dụng kết nối liên tục. Trường User-agent là phần mềm trình duyệt của người sử dụng. Phần mềm trình duyệt ở đây là Mozzil1a, một sản phẩm của hãng Nestcape. Trường này rất quan trọng vì máy chủ có thể gửi các bản khác nhau của cùng một đối tượng đến các trình duyệt khác nhau (các bản đối tượng này đều được xác định qua cùng một địa chỉ URL duy nhất). Cuối cùng là trường Accept language trong ví đụ này người sử dụng yêu cầu bản tiếng Pháp của đối tượng - nếu máy chủ có bản này. Trong trường hợp không có thì máy chủ gửi đi bản mặc định.

Khuôn dạng tổng quát của bản tin có thêm trường Entity Body sau các dòng tiêu đề. Trường này không được sử dụng trong phương thức GET nhưng được sử dụng trong phương thức POST. HTTP máy khách sử dụng phương thức POST khi người dùng điền vào một form - ví dụ khi muốn tìm kiếm qua một máy tìm kiếm như Google. với phương thức POST người dùng vẫn yêu cầu trang web nhưng nội dung cụ thể phụ thuộc vào nội dung điền trong form.

Nếu giá trị của trường method là POST thì phần entity body sẽ chứa nội dung mà người dùng điền vào form. Phương thức HEAD cũng tương tự như phương thức POST. Khi nhận được yêu cầu với phương thức POST, máy chủ sẽ gửi lại bản tin HTTP trả lời nhưng không gửi đối tượng được yêu cầu. Thường người ta sử dụng phương thức HEAD để gỡ lỗi.

Bản tin trả lời:

Sau đây là một ví dụ về bản tin trả lời, bản tin này có thể là trả lời cho bản tin yêu cầu trên.



Hình 3.4 Khuôn dạng bản tin trả lời

HTTP/1.1 200 OK

Connection :close

Date : Thu, 01 Dec 2011 11:00:15 GMT Máy chủ Apache/1. 3. 0 (unix)

Last modified :Mon, 14 Nov 2011 19:29:04 GMT Connect length : 8611

Connect type :text/html

(data data . . . . .)

Bản tin trên gồm có 3 phần : Dòng đầu tiên là dòng trạng thái (status link), 6 dòng tiêu đề và cuối cùng là phần thân (Entity body) chứa đối tượng được yêu cầu (là phần data data...). Dòng trạng thái có 3 trường : trường phiên bản của giao thức, mã trạng thái và trường trạng thái bản tin trả lời. Trong ví dụ này thì dòng trạng thái cho biết máy chủ sử dụng phiên bản HTTP 1.1 và trạng thái là sẵn sàng (máy chủ đã nhận được yêu cầu và gửi đối tượng được yêu cầu).

Trường Connection: close báo cho máy khách biết máy chủ sẽ đóng kết nối sau khi gửi đi bản tin.

Trường Date: cho biết thời gian khi máy chủ tạo ra bản tin và gửi đi, chú ý rằng đây không phải là thời gian khi đối tượng được tạo ra hay lần cuối cùng đối tượng được cập nhật. Đó là thời điểm mà máy chủ tìm thấy đối tượng trong hệ thống file của mình, chèn đối tượng vào bản tin trả lời và gửi đi.

Trường server cho biết bản tin trả lời này được tạo ra tớ phần mềm Máy chủ web Apache, ý nghĩa của nó giống với trường User agent trong bản tin yêu cầu. Trường Last modified là thời gian cuối cùng đối tượng được cập nhật.

Trường Content lenght: cho biết độ dài của đối tượng được gửi. Content type xác định kiểu của đối tượng là file văn bản HTML (kiểu của đối tượng được đặt ở đây chứ không phải trong phần mở rộng của tên file).

Khi nhận được một bản tin yêu cầu HTTP 1.0, máy chủ cũng sẽ không sử dụng kết nối liên tục ngay cả khi máy chủ dùng phiên bản 1.1. Máy chủ sẽ đóng

kết nối ngay sau khi gửi đối tượng. Điều này cần thiết vì máy khách sử dụng phiên bản HTTP 1.0 sẽ chờ máy chủ đóng kết nối lại.

Khuôn dạng chung của một bản tin trả lời được minh họa trên hình 2.8. Khuôn dạng này tương thích với ví dụ trên. Tuy nhiên cần phải nói thêm về mã trạng thái (status code) và ý nghĩa của chúng. Mã trạng thái cùng với cụm tớ đi sau cho biết kết quả của yêu cầu. Sau đây là một vài giá trị thông dụng và ý nghĩa của chúng:

200 OK: Yêu cầu được đáp ứng và dữ liệu được yêu cầu nằm trong bản tin

301 Moved permanetly: cho biết đối tượng đã được chuyển và địa chỉ URL mới của đối tượng được đặt trong trường Location: của bản tin trả lời, phần mềm tại máy khách sẽ tự động lấy đối tượng tại địa chỉ URL mới (Đây là hiện tượng chuyển hướng thường gặp khi duyệt web).

400 Bad Request: máy chủ không hiểu được yêu cầu từ máy khách

404 Not found: đối tượng không được lưu trên máy chủ

505 HTTP version not support: máy chủ không hỗ trợ giao thức của máy khách.

3.2.1.3 Tương tác người dùng-máy chủ

HTTP máy chủ không lưu giữ trạng thái, điều này đơn giản hoá kiến trúc và làm tăng hiệu suất hoạt động của máy chủ. Tuy nhiên các máy chủ muốn phân biệt người dùng không chỉ vì muốn hạn chế sự truy cập mà còn vì máy chủ muốn phục vụ theo định danh người dùng. HTTP có 2 cơ chế để máy chủ phân biệt người dùng: Authentication và cookies.

Authentication (Kiểm Chứng)

Nhiều máy chủ yêu cầu người dùng phải cung cấp tênvà mật khẩu để có thể truy cập vào tài nguyên trên máy chủ. Yêu cầu này được gọi là kiểm chứng. HTTP có các mã trạng thái và trường để thực hiện quá trình kiểm chứng. Giả sử máy khách yêu cầu một đối tượng từ máy chủ và máy chủ yêu cầu máy khách cung cấp tên và mật khẩu. Đầu tiên máy khách vẫn gửi một bản tin yêu cầu thông thường. Máy chủ sẽ trả lời với bản tin có phần thân rỗng và trường mã trạng thái là 401 Authentication required. Trong bản tin trả lời này có trường www-authenticate: xác định phương thức kiểm chứng mà người đùng phải thực hiện, thông thường là đưa tên và mật khẩu. Nhận được bản tin này, máy khách yêu cầu người dùng cung cấp tên và mật khẩu. Sau đó, máy khách sẽ gửi lại bản tin yêu cầu có trường Authoziration: trong tiêu đề, trường này chứa tên và mật khẩu của người dùng.

Sau khi nhận được đối tượng đầu tiên, máy khách tiếp tục gửi tên và mật khẩu trong các bản tin kế tiếp (thường thì cho đến khi người dùng đóng trình duyệt lại. Khi trình duyệt còn mở, tên và mật khẩu được lưu lại trong cache để người dùng không phải đánh lại nữa). Theo cách này máy chủ có thể phân biệt các người dùng. Trong chương 7 ta sẽ thấy rằng HTTP phân biệt người dùng khá lỏng lẻo và không khó để vượt qua.

Cookies:

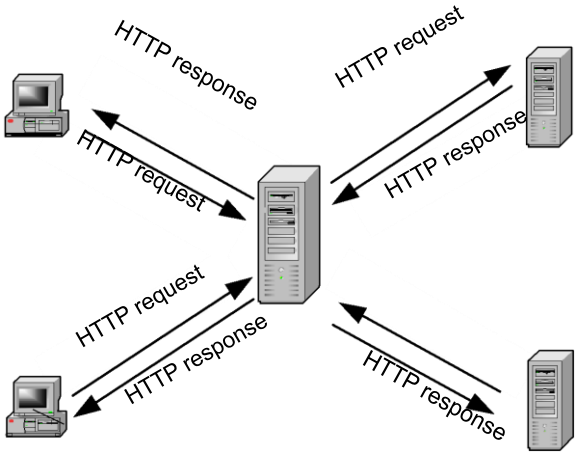
Cookie là kỹ thuật khác được máy chủ sử dụng để ghi lại đấu vết của người truy cập. Nó được đặc tả trong RFC 2109. Ví dụ lần đầu tiên người dùng truy cập vào một máy chủ nào đó có sử dụng cookie. Bản tin trả lời của máy chủ có trường Set-cookies: trong tiêu đề cùng với một chuỗi ký tự do Máy chủ web tạo ra. Khi nhận được bản tin trả lời, máy khách xác định được trường Set-cookies và chuỗi ký tự đi kèm, Trình duyệt sẽ thêm một dòng vào cuối file cookie (và một file đặc biệt trên máy máy khách). Dòng này thường là dòng chứa tên máy chủ và chuỗi ký tự cookie. Giả sử một tuần sau, máy khách gửi thông điệp yêu cầu đến máy chủ, máy khách sẽ tự động chèn trường Cookies: trong tiêu đề của bản tin yêu cầu với giá trị là chuỗi giá trị cookie lưu trong file cookie. Máy chủ web sử dụng cookie cho nhiều mục đích:

- Nếu máy chủ yêu cầu kiểm chứng nhưng không muốn đòi hỏi người dùng đăng nhập qua tên và mật khẩu thì có thể sử dụng cookie cho mỗi lần người dùng truy cập vào máy chủ.
- Máy chủ sử dụng cookie nếu muốn ghi nhớ các hoạt động của người dùng, phục vụ mục đích quảng cáo.
- Nếu người sử dụng mua hàng trên mạng thì máy chủ sử dụng cookie để ghi lại những gì mà người sử dụng đã mua.

Sử dụng cookie gây khó khăn cho người dùng không có máy tính cố định mà truy cập vào máy chủ từ nhiều máy khác nhau. Máy chủ sẽ coi đó là những người dùng khác nhau.

3.2.1.4 GET có điều kiện

Lưu giữ lại các đối tượng đã từng được lấy, web cache có thể làm giảm thời gian chờ từ khi gửi yêu cầu đến khi nhận đối tượng và làm giảm lưu lượng thông tin truyền trên mạng Internet. Web cache được triển khai trên trình duyệt hay các cache máy chủ. Mặc dù web cache làm giảm thời gian chờ nhận đối tượng nhưng vấn đề nảy sinh là bản sao của đối tượng được lưu giữ trên máy khách có thể đã cũ, nói cách khác đối tượng trên máy chủ có thể đã thay đổi từ khi máy khách lấy đối tượng đó về. Tuy nhiên HTTP có cơ chế cho phép sử dụng cache trong khi vẫn đảm bảo đối tượng trong cache chưa bị cũ.



Hình 3.5 Client yêu cầu đối tượng thông qua cache

Cơ chế này chính là GET có điều kiện (conditional GET). một bản tin HTTP được gọi là có điều kiện nếu: (1) thông điệp sử dụng phương thức GET và (2) thông điệp có trường If- modified- since trong tiêu đề. Ví dụ, trình duyệt yêu cầu một đối tượng từ máy chủ mà trong cache của nó chưa có:

GET /english/index.html HTTP/1.0

User-agent :Mozilla/4.0

Sau đó máy chủ gửi bản tin trả lời kèm với đối tượng

HTTP /1.0 200 OK

Date : Thu, 01 Dec 2011 11:00:15

Server : Apache/1.3.0 (Unix)

Last-modified: Mon, 14 Nov 2011 19:29:04

Content-type :image/gif

(data data …….)

Trình duyệt hiển thị đối tượng đồng thời lưu lại đối tượng trong cache cục bộ cùng với thời gian trong trường Last-modined kèm theo đối tượng. Một tuần sau, người sử dụng lại yêu cầu đối tượng này trong khi đối tượng đã được lưu trên cache. Nhưng đối tượng có thể đã bị thay đổi trong thời gian 1 tuần nên trình duyệt phải thực hiện kiểm tra bằng cách gửi một bản tin GET có điều kiện, cụ thể browser gửi đi:

GET /english/index.html HTTP/1.0

User-agent : Mozỉlla /4.0

If-modified-since : Thu, 01 Dec 2011 11:00:15

Giá trị trường If-modified-since: là giá trị của trường Last- modified: trong tiêu đề mà máy chủ đã gửi cho máy khách tuần trước. Bản tin GET có điều kiện yêu cầu máy chủ chỉ gửi đối tượng cho máy khách nếu như đối tượng đó được cập nhật sau thời gian được chỉ ra trên.

Giả sử đối tượng đó không thay đổi gì từ Thu, 01 Dec 2011 11:00:15 thì máy chủ sẽ gửi cho máy khách bản tin:

HTTP /1.0 304 Not modified

Date : Thu, 08 Dec 2011 12:05:17

Server: Apache 11. 3. 0 (Unix) (empty entity body)

Bản tin trả lời này không kèm theo đối tượng. Việc gửi kèm đối tượng chi làm lãng phi đường truyền và làm tăng thời gian máy khách phải chờ để nhận được đối tượng, đặc biệt khi đối tượng có kích thước lớn. Giá trị trường trạng thái là 304 Not modified báo cho máy khách biết đối tượng mà máy khách lưu trong cache giống đối tượng gốc tại máy chủ, do đó máy khách có thể sử dụng lại đối tượng này.

3.2.1.5 Web caches

Web cache (máy chủ đại diện) là thực thể đáp ứng yêu cầu từ máy khách. Máy tính làm nhiệm vụ Web cache có ổ đĩa riêng lưu trữ bản sao các đối tượng đã từng được yêu cầu. Người sử dụng có thể cấu hình cho trình duyệt sao cho tất cả các yêu cầu đều được gửi đến web cache trước, khi đó tất cả yêu cầu của trình duyệt về một đối tượng nào đó sẽ được chuyển đến webcache trước. Giả sử trình duyệt yêu cầu đối tượng là một file ảnh có địa chỉ là http://www.ptit.edu.vn/campus.gif

- Trình duyệt khởi tạo một kết nối TCP tới webcache và gửi yêu cầu tới webeache
- Webcache sẽ kiểm tra và tìm đối tượng, nếu tìm được thì webcache sẽ gửi đối tượng cho máy khách qua kết nối TCP đã được thiết lập.
- Nếu webcache không có đối tượng đó thì nó sẽ khởi tạo một kết nối từ máy chủ thật sự chứa đối tượng, ở đây là www.ptit.edu.vn. Sau đó webcache gửi bản tin yêu cầu tới cho máy chủ này thông qua kết nối TCP vừa khởi tạo. Sau khi nhận được yêu cầu từ webcache, máy chủ sẽ gửi lại đối tượng cho webcache.
- Khi nhận được đối tượng, webcache sẽ lưu lại bản sao của đối tượng và gửi đối tượng trong bản tin HTTP trả lời cho máy máy khách (thông qua kết nối TCP đã được thiết lập trước đó).

Như vậy webcache vừa là máy khách vừa là máy chủ. Webcache đóng vai trò máy chủ khi nhận yêu cầu và trả lời, đóng vai trò máy khách khi gửi yêu cầu và nhận bản tin trả lời. Webcache được sử dụng rộng rãi vì ba nguyên nhân sau: webcache làm giảm thời gian máy khách phải đợi. Trong trường hợp cache có đối tượng được yêu cầu thì đối tượng này sẽ ngay lập tức được chuyển tới máy khách. Thứ hai, webcache làm giảm tải mạng. Bằng cách giảm tải đường truyền ra mạng Intemet, cơ quan không cần phải nâng cấp đường truyền và giảm chi phí.

Webcache làm giảm lượng thông tin Web trao đổi trên Internet, do đó tăng hiệu suất hoạt động của tất cả các ứng dụng. Năm 1998, theo thống kê hơn 75% thông tin được truyền trên mạng là ứng dụng web, vì vậy giảm tải web cải thiện đáng kể hiệu suất toàn bộ Intemet. Thứ ba, mạng Internet với nhiều webcache giúp cho việc nhanh chóng phát tán thông tin - thậm chí ngay cho những nhà cung cấp thông tin có tốc độ máy chủ chậm hay tốc độ kết nối chậm. Nếu một nhà cung cấp có một nội dung cần phổ biến thì nội dung này ngay lập tức được chuyển đến các webcache và yêu cầu của người dùng tớ mọi nơi được đáp ứng nhanh chóng.

Cache cộng tác:

Có thể kết hợp nhiều webcache đặt ở các vị trí khác nhau trên mạng nhằm nâng cao hiệu suất tổng thể. Ví dụ, cache của một cơ quan có thể được cấu hình sao cho các yêu cầu của nó được gửi từ cache của nhà cung cấp dịch vụ Intemet cấp quốc gia. Khi đó nếu cache của cơ quan không có đối tượng được yêu cầu thì nó sẽ gửi bản tin yêu cầu HTTP đến cache của ISP. Cache ở ISP sẽ

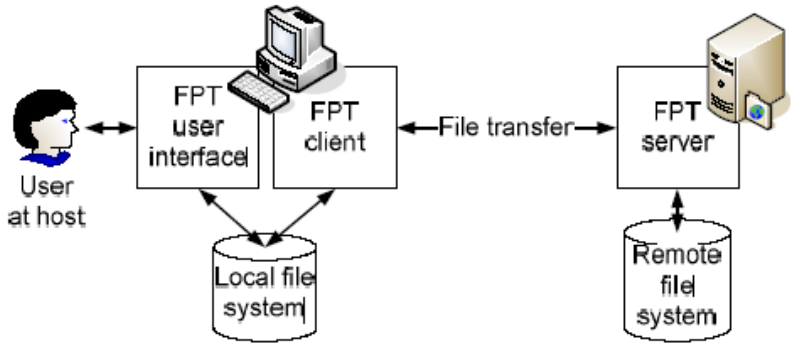
tìm đối tượng trong hệ thống lưu trữ của mình hoặc tại chính máy chủ có lưu giữ đối tượng. Sau đó nó sẽ gửi đối tượng trong bản tin trả lời HTTP tới cache của cơ quan. Cache của cơ quan lại gửi đối tượng từ trình duyệt yêu cầu. Mỗi lần đối tượng khi qua cache đều được sao chép lại trong cache.

Một ví dụ về hệ thống cache cộng tác là hệ thống cache NLANR. Hệ thống này có nhiều máy làm nhiệm vụ webcache ở Mỹ, cung cấp dịch vụ cho các webcache của các tổ chức và các khu vực trên toàn thế giới. Cache này lấy đối tượng từ cache khác bằng cách kết hợp sử dụng giao thức HTTP và ICP (Intemet Caching Protocol). ICP là giao thức ở tầng ứng dụng cho phép một cache nhanh chóng xác định một cache khác có một đối tượng nào đó hay không, và nếu có thì cache có thể sử dụng giao thức HTTP để lấy đối tượng về. ICP được sử dụng rộng rãi trên rất nhiều hệ thống cache liên hợp.

Một kiểu cộng tác khác là cụm cache (cache cluster), thường đặt trên cùng một mạng LAN. Cache được thay thế bởi cụm cache khi một cache duy nhất không đáp ứng hiệu quả khi có quá nhiều yêu cầu hay khi dung lượng thiết bị nhớ hạn chế. Tuy nhiên khi trình duyệt yêu cầu một đối tượng thì vấn đề nẩy sinh là yêu cầu được gửi đến cache nào trong cụm cache này. vấn đề này có thể được giải quyết bằng cách tìm kiếm theo hàm băm. Đơn giản nhất, trình duyệt thực hiện phép băm trên địa chỉ URL, trình duyệt sẽ căn cứ vào kết quả để gửi yêu cầu đến một trong các cache trong cụm. Nếu tất cả trình duyệt dùng cùng một thuật toán băm, đối tượng không bao giờ được lưu trên các cache khác nhau trong cụm. Nếu đối tượng thực sự được lưu trữ trong cụm thì trình duyệt luôn có thể gửi yêu cầu đến cache thích hợp; Tìm kiếm theo hàm băm là cốt lõi của giao thức Cache Array Routing (CARP).

3.2.2 Giao thức truyền tập tin FTP

FTP (File Transfer Protocol) là giao thức truyền tập tin tin cậy giữa hai máy tính. Giao thức này xuất hiện từ những năm 1971 nhưng vẫn còn được sử dụng rộng rãi cho đến tận ngày nay. Các tiêu chuẩn truyền tin của giao thức FTP được mô tả trong RFC 959, hình 3.5 minh họa các dịch vụ của FTP.



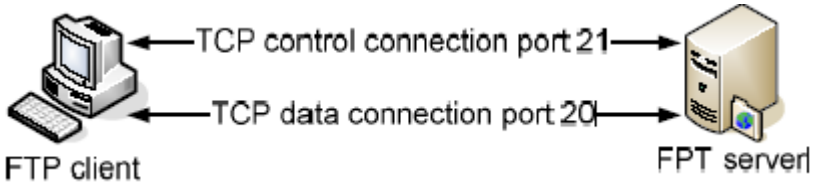
Hình 3.6 FTP cho phép trao đổi file giữa hai máy tính

Trong phiên làm việc của FTP, người dùng làm việc trên máy tính của mình và trao đổi tập tin với một máy tính khác. Để truy cập tới máy tính khác, người dùng phải đăng nhập thông qua việc cung cấp tên người dùng và mật

khẩu. Sau khi những thông tin này được kiểm chứng thì công việc truyền tập tin từ hệ thống tập tin trên máy tính của mình đến hệ thống tập tin ở đầu kia mới có thể thực hiện được.

Như mô tả trên hình 3.6, người dùng tương tác với FTP thông qua chương trình giao tiếp người dùng của FTP. Đầu tiên người dùng đánh tên máy tính cần truyền tập tin. Tiến trình FTP ở máy khách khởi tạo một kết nối TCP tới tiến trình FTP máy chủ sau đó người dùng đưa các thông tin về tên và mật khẩu để máy chủ kiểm chứng. Sau khi được máy chủ xác định, người đùng mới có thể thực hiện việc trao đổi file giữa hai hệ thống file.

HTTP và FTP đều là giao thức truyền file và có rất nhiều đặc điểm chung như cả hai đều sử dụng các dịch vụ của TCP. Tuy vậy hai giao thức này có những điểm khác nhau cơ bản. Điểm khác nhau nổi bật nhất là FTP sử dụng hai kết nối TCP song song, một đường truyền thông tin điều khiển (control connection) và một đường truyền dữ liệu (data connection). Các thông tin điều khiển như thông tin định danh người dùng, mật khẩu truy nhập, lệnh thay đổi thư mục, lệnh "put" hoặc "get" file giữa hai máy tính được trao đổi qua đường truyền thông tin điều khiển. Đường truyền dữ liệu để truyền file dữ liệu thực sự. Vì FTP phân biệt luồng thông tin điều khiển với luồng dữ liệu nên nó dược gọi là gửi thông tin điều khiển out-of-band. Trong chương 6 chúng ta sẽ thấy rằng giao thức RTSP dùng để truyền âm thanh và hình ảnh liên tục cũng sử dụng cách gửi thông tin điều khiển kiểu out-of-band. Như đã nói, HTTP gửi tiêu đề của bản tin và file dữ liệu trên cùng một kết nối TCP. Vì vậy mà HTTP được gọi là gửi thông tin điều khiển in-band. Trong phần tiếp theo ta sẽ thấy rằng SMTP - giao thức gửi thư điện tử cũng sử dụng truyền thông tin điều khiển kiểu in-band. Đường truyền thông tin điều khiển và đường truyền dữ liệu của giao thức FTP được minh họa trong hình 3.7.



Hình 3.7 FTP gồm 2 đường: Kiểm soát và dữ liệu

Khi người dùng bắt đầu một phiên làm việc FTP, đầu tiên FTP sẽ thiết lập một đường kết nối thông tin điều khiển TCP qua cổng 21. Phía máy khách của giao thức FTP truyền thông tin về định danh người dùng và mật khẩu cũng như lệnh thay đổi thư mục qua kết nối này. Khi người dùng có một yêu cầu trao đổi file (truyền từ/đến máy người dùng), FTP mở một kết nối TCP để truyền dữ liệu qua cổng 20. FTP truyền đúng một file qua kết nối này và ngay sau khi truyền xong thì đóng kết nối lại. Nếu trong cùng phiên làm việc người dùng có yêu cầu truyền file thì FTP sẽ mở một kết nối khác. Như vậy với FTP, luồng thông tin điều khiển được mở và tồn tại trong suốt phiên làm việc của người dùng, nhưng

mỗi kết nối dữ liệu được tạo ra cho mỗi một yêu cầu truyền file (kết nối dữ liệu là không liên tục).

Trong suốt phiên làm việc, FTP máy chủ phải giữ lại các thông tin về trạng thái của người dùng, đặc biệt nó phải kết hợp các thông tin điều khiển với tài khoản của người dùng. Máy chủ cũng lưu giữ thư mục hiện thời mà người dùng truy cập cũng như cây thư mục của người dùng. Ghi lại các thông tin trạng thái của mỗi phiên làm việc hạn chế đáng kể tổng số phiên làm việc đồng thời. HTTP không lưu giữ trạng thái nên nó không ghi lại bất kì thông tin nào về trạng thái của người dùng.

Các lệnh FTP

Lệnh (yêu cầu) từ máy khách đến máy chủ và kết quả (trả lời) từ máy chủ tới máy khách được gửi thông qua kết nối điều khiển và được mã hoá bằng bảng mã ASCII 7 bit. Do vậy giống như lệnh HTTP, người ta có thể đọc được lệnh FTP. Trường hợp các lệnh viết liên tục thì lặp ký tự CR (carriage return) và LF (line feed) được sử dụng để phân biệt các lệnh (và trả lời) mỗi câu lệnh chứa 4 kí tự ASCII in hoa, một số lệnh có tham số. Sau đây là một số câu lệnh hay dùng:

USER username: sử dụng để gửi thông tin tên tài khoản của người dùng cho máy chủ

PASS password: dùng để gửi mật khẩu cho máy chủ

LIST: dùng để yêu cầu máy chủ gửi một danh sách các tập tin trong thư mục hiện thời. Danh sách này được gửi thông qua một kết nối dữ liệu TCP

RETR filename: dùng để lấy một tập tin từ thư mục hiện thời (trên máy ở xa)

STOR filename: dùng để tải một tập tin vào thư mục hiện thời (trên máy ở xa)

Thông thường có quan hệ 1-1 giữa lệnh của người dùng và lệnh của FTP. Ứng với mỗi lệnh từ máy khách là một trả lời của máy chủ. Câu trả lời là một mã ba chữ số và có thể có một thông báo kèm theo. Điều này tương tự như trường mã trạng thái trong bản tin trả lời HTTP. dưới đây là một số câu trả lời thường gặp:

331 username OK, password requiered

125 connection already open; Transfer staring

425 canDt open data connection

452 error writing file

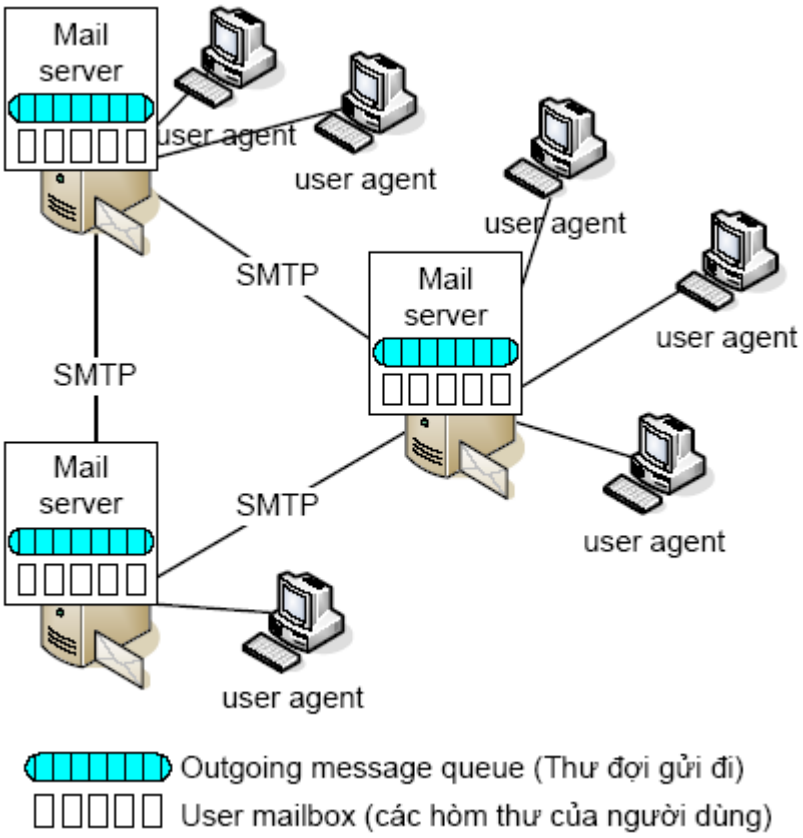
3.2.3 Giao thức chuyển thư điện tử

Cùng với Web, thư điện tử là một trong những ứng dụng Internet thông dụng nhất. gần giống thư tín thông thường, e-mail là dịch vụ không đòi hỏi đồng bộ - nghĩa là mọi người gửi và đọc thư khi thấy thuận tiện, không cần theo kế hoạch trước. Nhưng khác với thư tín thường, e-mail nhanh, dễ gửi và chi phí thấp. hơn nữa những bản tin e-mail ngày nay có thể chứa đựng các hyperlink, văn bản định dạng HTML, hình ảnh, âm thanh và cả video.

Hình 3.8 minh họa hệ thống thư điện tử trên mạng Internet gồm có 3 thành phần phần chính: user agent, mail máy chủ và SMTP (Simple Mail

Transfer Protocol). Để tiện theo dõi, chúng ta sẽ lấy ví dụ người A gửi e-mail cho B để mô tả 3 thành phần trên. Chương trình giao tiếp người dùng cho phép đọc, hồi âm, gửi, lưu giữ và soạn thảo các thư (user agent dành cho e-mail còn được gọi là trình đọc thư). Khi A soạn thảo xong thư, user agent của A sẽ gửi thư tới máy chủ thư điện tử của A, tại đây thư được đặt vào trong hàng đợi để gửi ra ngoài. Khi B muốn đọc thư, user agent của B sẽ lấy thư trên hộp thư của B tại máy chủ thư điện tử. Hiện nay, những phần mềm soạn e-mail thông dụng là Eudora, Microsoft Outlook và Nescape Messenger.

Máy chủ thư điện tử là thành phần cốt lõi trong hệ thống e-mail. Mỗi người có một hộp thư đặt trên máy chủ thư điện tử. Hộp thư của B quản lý, lưu giữ các thư gửi từ B. Thư được tạo ra tại user agent của người gửi, được gửi từ máy chủ thư điện tử của người gửi, rồi tới máy chủ thư điện tử của người nhận - và cuối cùng được chuyển vào hộp thư của người nhận. Khi B muốn truy cập vào hộp thư của mình, máy chủ thư điện tử chứa hộp thư của B sẽ kiểm chứng B thông qua tên và mật khẩu truy nhập. Máy chủ thư điện tử của A cần phải xử lý khi máy chủ thư điện tử của B gặp sự cố. Nếu máy chủ thư điện tử của A không thể gửi thư cho máy chủ thư điện tử của B, nó sẽ giữ những thư đó trong hàng đợi gửi bản tin và sẽ cố gắng gửi lại các bản tin. Quá trình gửi lại được tiến hành thường xuyên 30 phút một lần trong năm ngày. Và sau vài ngày, nếu vẫn không thành công thì máy chủ sẽ huỷ bỏ thư và gửi thư báo cho người gửi (A).



Hình 3.8 Minh họa hệ thống thư điện tử

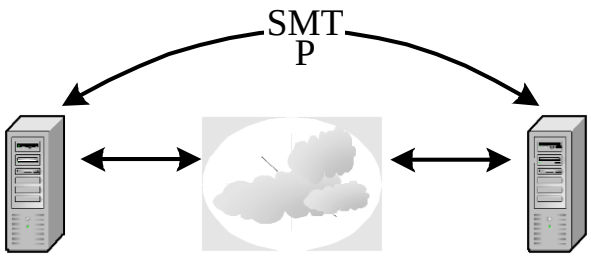
SMTP (Simple Mail Transfer Protocol) là giao thức gửi thư điện tử của tầng ứng dụng. SMTP sử dụng dịch vụ truyền dữ liệu tin cậy của TCP để truyền thư từ máy chủ thư điện tử của người gửi đến máy chủ thư điện tử của người nhận. Giống các giao thức khác ở tầng ứng dụng, SMTP có 2 phía: phía máy khách, trên máy chủ thư điện tử của người gửi và phía máy chủ trên máy chủ thư điện tử của người nhận. Tất cả các máy chủ thư điện tử đều chạy cả hai phía máy khách và máy chủ của SMTP. Máy chủ thư điện tử đóng vai trò máy khách khi gửi thư và đóng vai trò máy chủ khi nhận thư.

3.2.3.1 SMTP

SMTP là trái tim của dịch vụ gửi thư trên Internet và được đặc tả trong RFC821. SMTP truyền các bản tin (thư) từ máy chủ thư điện tử của người gửi đến máy chủ thư điện tử của người nhận. SMTP ra đời trước HTTP khả lâu (RFC đặc tả SMTP có từ năm 1982 và SMTP đã xuất hiện trước đó một thời gian dài). mặc dù có nhiều ưu điểm nên được tất cả máy chủ thư điện tử trên Internet sử dụng, SMTP vẫn là một kỹ thuật cũ nên chắc chắn có những đặc tính lạc hậu. Ví dụ SMTP đòi hỏi phần thân của tất cả các bản tin e-mail phải mã hoá theo bảng mã ASCII 7 bit. sự hạn chế này là do trong những năm đầu thập kỷ 80, với số đường truyền ít ỏi, không ai gửi thư cùng với những phần đính kèm lớn, hay gửi kèm các file hình ảnh, âm thanh có kích thước lớn. Nhưng trong kỷ nguyên đa phương tiện ngày nay, việc giới hạn mã ASCII 7 bit là một hạn chế lớn vì dữ liệu đa phương tiện nhị phân phải được chuyển sang mã ASCII trước khi được gửi đi qua SMTP và sau đó lại phải giải mã thành mã nhị phân sau khi thư đến đích.

Để minh họa hoạt động cơ bản của SMTP, hãy xét ví dụ sau giả sử A muốn gửi cho B một bản tin ASCII đơn giản:

- Đầu tiên, A sử dụng user agent của mình, đánh địa chỉ e-mail của B (B@ptit.edu.vn), soạn e-mail và yêu cầu user agent gửi thư đi.
- User agent của A gửi thư tới máy chủ thư điện tử của A. tại đây thư được đặt vào hàng thư đợi gửi .
- SMTP máy khách chạy trên máy chủ thư điện tử của A thấy thư trong hàng đợi. Nó tạo kết nối TCP tới SMTP máy chủ trên máy chủ thư điện tử của B.
- Sau giai đoạn khởi tạo 3 bước, SMTP máy khách gửi thư của A qua kết nối TCP.
- Tại máy chủ thư điện tử của B, SMTP máy chủ nhận thư và đặt thư vào hộp thư của B.
- Cuối cùng, khi thuận tiện B sẽ sử dụng user agent của mình để đọc thư.



Hình 3.9 Trao đổi thư giữa hai máy chủ Email

Kịch bản này được minh họa trên hình 3.9, một điểm quan trọng cần chú ý là SMTP không sử dụng máy chủ thư điện tử trung gian để gửi thư - ngay cả khi máy chủ thư điện tử gửi và nhận ở xa nhau . Ví dụ nếu mail máy chủ của A đặt ở Hongkong và máy chủ thư điện tử của B ở Hà Nội, thì giữa hai máy chủ thư điện tử ở Hong Kong và Hà Nội vẫn có đường kết nối TCP trực tiếp. Đặc biệt nếu máy chủ thư điện tử của B bị hỏng thì thư vẫn còn trong máy chủ thư điện tử của A và đợi cho lần gửi sau. Bản tin không được gửi qua máy chủ thư điện tử trung gian.

SMTP có nhiều đặc điểm tương tự như những quy tắc trong giao tiếp trực diện của con người. Đầu tiên, SMTP máy khách (chạy trên máy chủ thư điện tử gửi) thiết lập kết nối TCP với cổng 25 tại SMTP máy chủ (chạy trên máy chủ thư điện tử nhận). Trong trường hợp máy chủ không làm việc, máy khách sẽ cố gắng thử lại lần sau. Ngay khi kết nối được thiết lập, máy chủ và máy khách thực hiện một vài thủ tục bắt tay. Quá trình này tương tự như hai người tự giới thiệu về bản thân trước khi tiến hành nói chuyện. Trong thủ tục trao đổi, SMTP máy khách thông báo với SMTP máy chủ địa chỉ người gửi và địa chỉ người nhận. Ngay sau quá trình giới thiệu, máy khách sẽ gửi thư bằng dịch vụ truyền dữ liệu tin cậy của TCP. Sau đó, máy khách sẽ lặp lại các bước này khi vẫn còn bản tin khác để gửi từ máy chủ, còn nếu không, máy khách yêu cầu TCP đóng kết nối lại.

Ví dụ sau là đoạn hội thoại giữa máy khách (C) và máy chủ (S). Tên máy tính máy khách là yahoo.com và máy chủ là hamburger.edu. Dòng hội thoại mở đầu bằng chữ C: là đoạn hội thoại máy khách gửi qua socket TCP và dòng hội thoại bắt đầu với chữ S: là đoạn hội thoại máy chủ gửi đi thông qua socket TCP. Đoạn hội thoại bắt đầu ngay sau khi thiết lập được kết nối TCP:

S: 220 ptit.edu.vn
C: 250 Hello yahoo.com
S: 250 Hello crepes. fr, pleased to meet you
C: MAIL FROM: <A@crepes. fr>
S: 250 A@crepes. fr. . . Sender ok
C: RCPT TO: <B@ptit.edu.vn >
S: 250 B ptit.edu.vn . . .Recipient ok
C : DATA
S: 354 Enter mail, end with “. “ Oa a line by itself
C: Do you like ketchup? C: How about pickles ?
S: 250 Message accepted for delivery
C : QUIT

S: 221 ptit.edu.vn closing connection.

Trong ví dụ trên, máy khách gửi một bản tin ("Do you like ketchup? How abou pickles?") từ máy chủ thư điện tử yahoo.com tới máy chủ thư điện tử ptit.edu.vn. Client sử dụng 5 câu lệnh : HELO (viết tắt của HELLO), MAIL FROM, RCPT TO, DATA và QUIT. Ý nghĩa của những câu lệnh này có thể đoán được qua tên gọi của nó. Máy chủ gửi trả kết quả thực hiện mỗi lệnh, mỗi câu trả lời gồm một mã trạng thái và một lời giải thích tiếng Anh. Ở đây SMTP sử dụng kết nối liên tục: Nếu có nhiều thư để gửi từ cùng một máy chủ thư điện tử thì máy chủ thư điện tử gửi sẽ gửi tất cả các thư trên cùng một kết nối TCP. với mỗi bản tin, máy khách bắt đầu tiến trình gửi bằng lệnh HELO yahoo.com và chỉ gửi lệnh QUIT sau khi gửi tất cả thư.

Giao thức SMTP và HTTP đều được sử dụng để gửi file giữa các máy tính. HTTP chuyển file hoặc đối tượng tớ Web máy chủ tới Web máy khách (trình duyệt Web), SMTP chuyển file (là bản tin thư điện tử) giữa các máy chủ thư điện tử. Khi truyền file cả 2 giao thức HTTP và SMTP cùng sử dụng kết nối liên tục. Điểm khác biệt cơ bản giữa hai giao thức là HTTP là giao thức kiểu kéo (Pull protocol) - máy khách kéo thông tin từ máy chủ về. Phía nhận (máy khách) là phía thiết lập kết nối TCP. SMTP lại là giao thức theo kiểu đẩy (Push protocol) - máy khách đẩy thông tin lên máy chủ. Phía gửi (máy khách) là phía thiết lập kết nối TCP trước. Ngoài dữ liệu văn bản, bản tin còn có thể chứa các kiểu dữ liệu khác như âm thanh hình ảnh. HTTP đặt các đối tượng này trong các bản tin riêng rẽ để gửi. với SMTP tất cả các đối tượng này được đặt trong cùng một thư điện tử.

Khi A gửi thư cho B, A sẽ đặt thư vào phong bì, ghi rõ địa chỉ gửi và địa chỉ nhận, nhân viên bưu điện sẽ đóng dấu ngày tháng vào phong bì. Thư điện tử cũng giống như vậy, bên cạnh nội dung bức thư (phần thân) cũng cần có địa chỉ người gửi, địa chỉ người nhận. Những thông tin phụ trợ này sẽ được đặt trong các dòng tiêu đề. Các dòng tiêu đề và phần thân của thư được tách biệt với nhau bằng lặp ký tự CR-LF. RFC 822 đặc tả đầy đủ các dòng tiêu đề cũng như ý nghĩa của chúng. Giống HTTP, tiêu đề gồm tớ khoá, theo sau là dấu hai chấm và một giá trị nào đó. với SMTP có một số trường bắt buộc, một số trường không bắt buộc. Tiêu đề phải có trường **From:** và trường **To:** một số trường như **Subject:** có thể có hoặc không, những trường này khác những lệnh SMTP mà đã được đề cập đến (mặc dù chúng cũng có ‘‘from’’ và ‘‘To’’). Các lệnh. là một phần trong giai đoạn khởi tạo của SMTP trong khi các trường nằm ngay trong thư.

Một bức thư thường có tiêu đề như sau:

From: A@yahoo.com

To: B@ptit.edu.vn

Subject: Searching for the meaning of life

Sau phần tiêu đề bản tin là một dòng trống, tiếp đến là thân bản tin (dạng mã ASCII). Bản tin kết thúc bằng một dòng chỉ chứa một dấu chấm câu.

Phần tiêu đề bản tin được đặc tả trong RFC 822 phù hợp cho việc gửi văn bản nhưng lại không đầy đủ để gửi thư chứa nội dung đa phương tiện (multimedia) - là thư có đính kèm ảnh, audio, video hoặc các thư chứa các ký tự khác tiếng Anh. Để gửi dữ liệu không thuộc dạng văn bản ASCII, user agent gửi phải gửi thêm một số trường trong tiêu đề của thư. Những trường này được đặc tả trong RFC 2045 và RFC 2046, là phần mở rộng MIME (Multipurpose Intemet Man Extension) cho RFC 822.

Hai trường MIME hỗ trợ multimedia là **Content-Type**: và **Content-Transfer-encoding.** Trường Content-Type cho phép phía nhận thực hiện các thao tác thích hợp trên thư nhận được. Ví dụ, nếu chỉ ra thân bản tin chứa ảnh JPEG, user agent nhận có thể gửi thân bản tin tới chương trình giải nén JPG. Để gửi bản tin văn bản không mã hoá theo bảng mã ASCII (ví dụ văn bản tiếng Trung Quốc, Nhật bản), người ta phải mã hoá nó theo bảng mã ASCII mà không làm ảnh hưởng tới SMTP. Trường Content-Transfer- encoding: xác định phần thân bản tin đã được mã hoá theo bảng mã ASCII và phương pháp mã hoá được sử dụng. Vì vậy khi user agent nhận được một bản tin với hai tiêu đề trên, đầu tiên nó sử dụng giá trị của tiêu đề Content-Transfer-encoding: để chuyển đổi thân bản tin về dạng ban đầu (không theo định dạng ASCII) và sau đó sử dụng trường Content-Type để xác định thao tác thực hiện kế tiếp.

Xét ví dụ sau, giả sử A muốn gửi một ảnh JPEG cho B . Để thực hiện điều này, A sử dụng phần mềm Eudora, đánh địa chỉ mail của B , chủ đề của e-mail và chèn ảnh JPEG vào thân bản tin. Sau khi hoàn tất việc soạn thảo, A nhấn nút send. Sau đó, user agent của A tạo ra một bản tin MIME có nội dung sau:

From: A@yahoo.com

To: B@ptit.edu.vn

Subject: Picture of yummy crepe. MIME-Version: 1.0

Content-Transfer-encoding: base64

Content-Type: Image/jpeg

(base64 encoded data . . .

...base64 encoded data)

Với bản tin MIME trên, chúng ta thấy rằng user agent của A mã hoá ảnh JPEG sử dụng kỹ thuật mã hoá base64. Đây là một trong những kỹ thuật mã hoá chuẩn trong MIME [RFC 2045] để biến đổi sang định dạng mã ASCII 7 bit.

Khi B đọc thư, user agent của B xử lý bản tin MIME này. Thấy trường **Content- Transfer-endcoding**: base64, nó thực hiện giải mã thân bản tin đã được mã hoá bằng kỹ thuật base64. Trường Content-Type: image/jpeg giúp cho user agent của B xác định rằng thân của bản tin phải được giải nén theo chuẩn JPEG. Cuối cùng, bản tin chứa trường MIME-Version: xác định phiên bản MIME đang được sử dụng. lưu ý rằng, bản tin cũng phải tuân theo khuyến nghị RFC 822/SMTP. Theo đặc tả MIME trong khuyến nghị RFC 2046, trường Content-Type: có khuôn dạng sau:

Content-Type: type/subtype; parmeters

Phần tham số sau dấu chấm phẩy có thể không bắt buộc. Trong khuyến nghị RFC 2046, trường Content-Type được sử dụng để xác định kiểu dữ liệu trong phần thân của bản tin MIME, gồm hai giá trị: kiểu dữ liệu và kiểu con. Sau phần kiểu và kiểu con là phần tham số. Nói chung, kiểu cao nhất (top-level) được sử dụng để khai báo kiểu dữ liệu chung, kiểu con (subtype) xác định định dạng đặc biệt trong kiểu dữ liệu chung. Các tham số bổ nghĩa cho kiểu và không ảnh hưởng tới bản chất kiểu dữ liệu. tập hợp tham số phụ thuộc vào kiểu và kiểu con.

Được thiết kế để có thể mở rộng, số lượng các lặp type/subtype và những tham số đi kèm trong MIME ngày càng tăng. Để bảo đảm là tập hợp này phát triển có trình tự, được đặc tả rõ ràng, MIME cần thiết lập quá trình đăng ký với IAAN (Internet Assigned Numbers Authority) là cơ quan đăng ký trung tâm. Tiến trình đăng ký kiểu dữ liệu được đặc tả trong khuyến nghị RFC 2048. Hiện nay, mới có định nghĩa cho bẩy nhóm dữ liệu chính. với mỗi kiểu lại có một danh sách các kiểu con và danh sách này đang tăng lên hàng năm. Dưới đây là 5 nhóm dữ liệu chính:

- **Văn bản (Text)**: Kiểu văn bản được sử dụng để xác định thân bản tin chứa thông tin dạng văn bản một kiểu con thường gặp là plain (trơn). Văn bản trơn không có lệnh hay chỉ dẫn định dạng khuôn dạng và do đó không cần phần mềm đặc biệt nào để hiển thị. Nếu nhìn tiêu đề MIME của thư trong hộp thư bạn có thể sẽ thấy trên tiêu đề có trường text/plain; charset"-us-ascii" hay text/plain; charset="ISO-8859-1". Những tham số này xác định bộ mã mà bản tin sử dụng. Một kiểu con khác cũng rất thông dụng là text/html. Kiểu con html yêu cầu máy chủ thư điện tử thông dịch những thẻ HTML gắn trong bản tin. Điều này cho phép user agent nhận hiển thị bản tin dưới dạng một trang Web (với font, hyperlink, applet và v. v. v).
- **Ảnh (Image)**: Kiểu ảnh được dùng để xác định thân bản tin là ảnh. Hai kiểu con thông dụng là image/gif và image/jpeg. với kiểu con image/gif, muốn hiển thị ảnh, user agent phải giải nén ảnh GIF.
- **Âm thanh (Audio)**: Kiểu audio yêu cầu nội dung được gửi ra thiết bị audio (speaker hoặc telephone). Kiểu con thông dụng là basic (mã theo luật 8-bit cơ sở) và 32kadpcm (định dạng 32kps được đặc tả trong RFC 1911).
- **Video**: Kiểu video có kiểu con là mpeg và quicktime.
- **Kiểu ứng dụng (Application)**: Kiểu ứng dụng dành cho dữ liệu không thuộc bất kỳ kiểu nào khác. Nó thường được áp dụng cho loại dữ liệu phải qua một ứng dụng khác xử lý trước khi người nhận có thể sử dụng được. Ví dụ khì người gửi gắn một tài liệu MS Word vào thông điệp E-mail, user agent đặt giá trị application/msword vào trường type/subtype. Khi user agent thấy giá trị application/msword trong trường type/subtype, nó khởi động ứng dụng MS Winword và chuyển phần thân bản tin MIME cho ứng dụng Word. một kiểu con quan trọng khác là là octerstream. Kiểu con này thường được dùng khi thân bản tin chứa dữ liệu nhị phân tuỳ ý. Khi nhận kiểu con này, mail reader sẽ yêu cầu người nhận lựa chọn để lưu bản tin trên đĩa xử lý sau.

Có một kiểu MIME đặc biệt quan trọng là kiểu đa phần (multipart). Bản tin e- mail cũng như trang Web có thể chứa nhiều đối tượng (như văn bản, ảnh, applet). Web gửi mỗi đối tượng trong một bản tin trả lời độc lập nhưng thư điện tử đặt tất cả các đối tượng trong cùng một bản tin. Đặc biệt, khi bản tin đa phương tiện có nhiều đối tượng thì bản tin đó có kiểu là multipart/mixed. Khi nhận được một bản tin mà trường content-type có giá trị multipart/mixed, user agent nơi nhận biết bản tin nhận được chứa nhiều đối tượng. Khi nhận được thông điệp như vậy, user agent phải xác định rõ:

- Điểm đầu và điểm cuối của đối tượng.
- Cách mã hoá các đối tượng không theo bảng mã ASCII.
- Kiểu của mỗi đối tượng.

Công việc này được thực hiện nhờ ký tự phân cách giữa các đối tượng và trường Content-type, Content-Transfer-Endcoding đứng trước mỗi đối tượng trong bản tin. Xét ví dụ sau: Giả sử A muốn gửi bản tin bao gồm một đoạn văn bản ASCII, một ảnh JPEG và cuối cùng là một đoạn văn bản ASCII cho B. Sử dụng user agent của mình, A đánh một đoạn văn bản, chèn ảnh JPEG sau đó đành tiếp đoạn văn bản còn lại. kết quả là user agent của A tạo ra một bản tin như sau:

From: A@yahoo.com.

To: B@ptit.edu.vn

Subject: Picture of yummy crepe with commentary

MIME-Version: 1.0

Content-Type:mu/tipartlmixed; Boundary=StartOfNextPart

--StartOfNextPart

Dear B ,

Please find a picture of an absolutely scrumption crepe.

- StartOfNextPart

Content-Transfer-Encoding: base64

Content-Type: image/jpeg

base64 encoded data . . . base64 encoded data

--StartOfNextPart

Let me know if you would like the recipe.

Qua bản tin trên chúng ta thấy rằng trường Content-Type: trong tiêu đề xác định cách thức phân cách các phần khác nhau trong cùng một bản tin. Việc phân cách được bắt đầu bằng 2 dấu gạch ngang (--) và kết thúc bằng lặp ký tự CRLF.

Bản tin e-mail bao gồm nhiều phần, lõi của bản tin là phần thân chứa dữ liệu thực sự được chuyển tớ người gửi đến người nhận. với bản tin nhiều phần, thân bản tin gồm nhiều phần và trước mỗi phần có một hoặc vài trường xác định kiểu. Đứng trước thân bản tin là lặp CRLF và một số trường. Những trường như From:, To:, và Subject: được đặc tả trong RFC 822 và những trường như

Content-type: và Content- Transfer- endcoding: là tiêu đề MIME. Nhưng chính máy chủ thư điện tử nhận bản tin cũng chèn vào bản tin một số trường khác, ví dụ Received: ở đầu bản tin. Trường này xác định tên của SMTP máy chủ gửi bản tin ("from"), tên của SMTP máy chủ nhận bản tin ("by") và thời gian khi bản tin tới đích. Người đọc sẽ đọc được bản tin như sau:

Received: from yahoo.com by ptit.edu; 10 Dec 11 18:37:39 GMT

From: A@yahoo.com

To: B@ptit.edu.vn Subject: Picture of yummy crepe. MIME-Version:1.0

Content-Transfer-encoding: base64

Content-Type: image/jpeg

base64 encoded data . . . . .

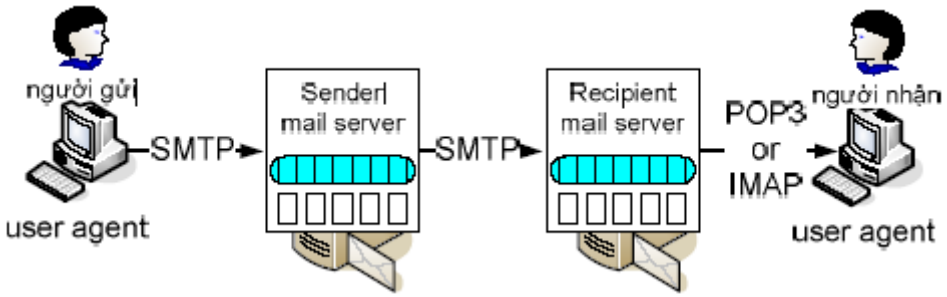
... base64 encoded data

Thực ra tất cả mọi người khi đùng e-mail đều nhìn thấy trường Received: đứng trước bản tin e-mail (trường này có thể nhìn thấy trực tiếp trên màn hình hoặc khi in thư). Có thể có những bản tin có nhiều trường Received: và trường Return-Path: phức tạp. Đó là vì bản tin này có thể được chuyển tiếp qua nhiều máy chủ thư điện tử trước khi đến tay người nhận. Ví dụ nếu B cấu hình máy chủ thư điện tử của mình (ptit.edu.vn) gửi chuyển tiếp tất cả các thư của B tới google.com khi đó tất cả các thư của B khi lấy từ google.com sẽ có những trường sau:

Received: from ptit.edu.vn by google.com; Dec 2011 18:03:01 GMT

Received: from crepes. fr by ptit.edu.vn ; 10 Dec 2011 18:03:07 GMT

Những trường này cho phép người nhận theo dõi vết đường đi của thư qua nhiều SMTP máy chủ cũng như thời gian thư tới mỗi máy chủ.



Hình 3.10 Giao thức email và các thực thể truyền thông

Mỗi khi SMTP gửi thư từ máy chủ thư điện tử của A tới máy chủ thư điện tử của B , thư được đặt trong mail bom của B . từ trước tới giờ, chúng ta chấp nhận giả thiết để đọc thư, B phải đăng nhập vào máy chủ thư điện tử và sử dụng một chương trình đọc thư (mail reader) nào đó cài ngay trên máy chủ thư điện tử. từ tận đầu những năm 90, mọi người vẫn thực hiện như vậy. Nhưng ngày nay mọi người thường đọc thư qua một user agent chạy trên máy tính cá

nhân của mình. Chạy user agent trên máy tính cá nhân, người sử dụng có được nhiều tính năng cao cấp, kể cả việc gửi và nhận những bản tin đa phương tiện.

Giả sử B (người nhận) chạy user agent của mình trên máy tính cá nhân. Có thể cài đặt máy chủ thư điện tử ngay trên may tính cá nhân của B . Tuy nhiên cách này có nhiều nhược điểm. Máy chủ thư điện tử quản lý nhiều mailbox, thực hiện cả chức năng máy khách và máy chủ của SMTP. Nếu cài máy chủ thư điện tử trên máy tính cá nhân của B thì PC đó lúc nào cũng phải bật và kết nối vào Intemet để có thể nhận thư mới (mà thư thì có thể đến bất cứ lúc nào). Điều này không thực tế với đa số người sử dụng Internet. Thông thường, người sử dụng chạy chương trình user agent trên máy tính cá nhân, truy cập vào hộp thư trên một máy chủ thư điện tử dùng chung (máy chủ thư điện tử này luôn luôn kết nối tới Internet và được chia xẻ giữa nhiều người dùng khác). Máy chủ thư điện tử thường được ISP của người dùng (và trường đại học hoặc công ty) quản lý.

Do user agent chạy trên máy tính cá nhân và máy chủ thư điện tử được quản lý Eởi các ISP nên cần có một giao thức cho phép user agent và máy chủ thư điện tử trao đổi với nhau. Đầu tiên chúng ta xét trường hợp thư được tạo ra tại PC của A được chuyển tới máy chủ thư điện tử của B như thế nào. Công việc này có thể được thực hiện một cách đơn giản bằng việc user agent của A trao đổi trực tiếp với máy chủ thư điện tử của B bằng giao thức SMTP. User agent của A sẽ khởi tạo một kết nối TCP tới máy chủ thư điện tử của B , gửi những lệnh khởi tạo SMTP, tải thư lên bằng lệnh DATA và sau đó đóng kết nối lại. Cách tiếp cận này hoàn toàn có thể thực hiện được nhưng ít khi được đùng vì nó không hỗ trợ trường hợp máy chủ thư điện tử phía nhận bị trục trặc. Trên thực tể, user agent gửi khởi tạo SMTP để tải thư của A tới chính maii máy chủ của A (chứ không phải là máy chủ thư điện tử của người nhận thư). Máy chủ thư điện tử của A sau đó sẽ thiết lập một phiên làm việc SMTP tới máy chủ thư điện tử của B để gửi tiếp thư từ máy chủ thư điện tử của B . Nếu máy chủ thư điện tử của B ngừng làm việc thì máy chủ thư điện tử của A sẽ giữ thư lại và sau đó cố gắng gửi lại. RFC SMTP có những lệnh để gửi tiếp thư qua nhiều SMTP máy chủ.

User agent chạy trên máy tính cá nhân của B lấy bản tin trong hộp thư trên máy chủ thư điện tử của B như thế nào ? Giải pháp là phải có một giao thức lấy thư cho phép chuyển thư từ máy chủ thư điện tử của B tới máy tính cục bộ. Hiện nay có 2 giao thức lấy thư thông dụng là POP3 (Post Once Protocol - Versíon 3) và IMAP (Internet Mail Access Protocol). User agent của B không thể sử dụng SMTP để lấy thư bởi vì lấy thư giống như việc kéo" trong khi SMTP là một giao thức "đẩy". Hình 3.10 minh họa về việc gửi và nhận thư. SMTP được dùng để chuyển thư giữa các máy chủ thư điện tử hay giữa user agent của người gửi và máy chủ thư điện tử của người gửi. POP3 hay IMAP được đùng để chuyển thư từ máy chủ thư điện tử tới user agent của người nhận.

3.2.3.2 POP3

POP3 được đặc tả trong RFC 1939 là giao thức lấy thư cực kỳ đơn giản và có rất ít chức năng. POP3 được khởi tạo khi user agent (máy khách) tạo kết nối TCP tới mail máy chủ (máy chủ) qua cổng 110. Sau khi thiết lập được kết nối TCP, POP3 gồm 3 giai đoạn: kiểm chứng, tiến hành xử lý và cập nhật. Trong giai đoạn kiểm chứng đầu tiên, user agent sử dụng tên và mật khẩu để xác nhận người sử dụng. Trong giai đoạn tiến hành xử lý thứ hai user agent tiến hành lấy thư. Nó có thể đánh dấu các thư để xoá hay hủy bỏ việc đánh đấu xoá. Giai đoạn ba - cập nhật, xảy ra sau khi máy khách ra lệnh quịt để kết thúc phiên làm việc POP3 . tại thời điểm đó mail máy chủ xoá tắt cả thư được đánh dấu.

Trong giai đoạn xử lý, user agent gửi lệnh và máy chủ trả lời kết quả thực hiện của mỗi lệnh đó. Mỗi lệnh có hai trạng thái kết quả: +OK thông báo lệnh vừa gửi được thực hiện đúng và ERR thông báo lệnh vừa gửi không thực hiện được. Giai đoạn kiểm chứng có 2 lệnh là user (tên truy nhập) và pass (mật khẩu). Giả sử máy chủ thư điện tử tên ptit.edu.vn, thực hiện lệnh telnet để kiểm thử như sau:

telnet ptit.edu.vn 110

+OK POP3 server ready

user B

+OK

pass xxxx

+OK user successfully logged on

Nếu đánh sai 1 lệnh thì máy chủ POP3 sẽ đáp lại bằng bản tin -ERR

Người sử dụng có thể cấu hình user agent ở một trong hai chế độ "Tải và xoá'' (download and delete) hay "tải và giữ" (download and keep). Chuỗi lệnh được user agent gửi phụ thuộc vào cấu hình này. Trong chế độ đầu, user agent sẽ phát ra chuỗi lệnh list, retr và dele. Giả sử người dùng có 2 bản tin trong hộp thư của mình. Trong đoạn hội thoại dưới đây C: (máy khách) là user agent và S: (máy chủ) là máy chủ thư điện tử. Khi đó giai đoạn xử lý công việc sẽ như sau:

C: list

S: 1 498

S: 2 912

S: .

C : retr 1

S : (blah blah . . .

S: …

S: … blah) S: .

C: dele 1

C: retr 2

S : (blah blah . . . .

S: …

S: ..blah)

C: dele 2

C : quit

S: +OK POP3 sever signing off

Đầu tiên, user agent yêu cầu mail máy chủ liệt kê kích thức của tất cả thư lưu trữ trong hộp thư. Sau đó, user agent bấy và xoá tìm thư trong hộp thư. Lưu ý rằng sau giai đoạn kiểm chứng người đùng chỉ còn 4 câu lệnh và list, retr, dele và quit. Cú pháp của các lệnh này được đặc tả trong RFC 1939. Sau khi xử lý lệnh quit, máy chủ POP3 vào giai đoạn cập nhật và xoá thư 1, 2 trong mailbox.

Trong chế độ này, khi B lấy thư từ những địa điểm khác nhau, thư của B sẽ nằm rải rác trên nhiều máy. Đặc biệt, nếu B đã lấy thư từ máy tính ở nhà thì sau đó sẽ không thể đọc lại thư đó trên máy tính ở cơ quan. Trong chế độ thứ hai ‘‘download and keep’’, user agent vẫn để lại thư trên mail máy chủ sau khi đã tải về. Khi đó B vẫn có thể đọc thư từ nhiều máy khác nhau.

Trong phiên làm việc POP3 giữa user agent và mail máy chủ, máy chủ POP3 sẽ ghi nhớ một vài thông tin trạng thái - ví dụ các thư đã bị đánh dấu xoá. Tuy nhiên máy chủ POP3 không chuyển thông tin trạng thái giữa các phiên làm việc khác nhau. Ví dụ không có thư nào được đánh đấu xoá ở đầu mỗi phiên làm việc. Điều này làm đơn giản công việc xây dựng một máy chủ POP3

3.2.3.3 IMAP

Sau khi tải thư về từ máy tính cá nhân, B có thể tạo những thư mục chứa thư và chuyển thư vào trong các thư mục đó. Sau đó B có thể xoá, chuyển thư giữa các thư mục hay tìm kiếm thư theo tên người gửi và chủ đề thư. Phương thức như vậy bất tiện với người sử dụng muốn đọc thư từ nhiều nơi vì họ thích duy trì phân cấp thư mục trên mail máy chủ để có thể truy cập được tớ bất kỳ máy tính nào. POP3 không đáp ứng được yêu cầu này.

IMAP (đặc tả trong RFC 2060) có thể giải quyết vấn đề này. Giống POP3, IMAP cũng là giao thức lấy thư nhưng có nhiều đặc tính và do đó phức tạp hơn POP3. IMAP được thiết kế cho phép người dùng thao tác trên những hộp thư ở xa một cách dễ dàng. IMAP cho phép B tạo những thư mục thư khác nhau trong hộp thư. B có thể đặt thư vào trong thư mục hay dịch chuyển thư từ thư mục này đến những thư mục khác. IMAP cũng có lệnh cho phép tìm kiếm trên thư mục theo tiêu chí xác định. IMAP phức tạp hơn POP3 nhiều vì máy chủ IMAP phải duy trì hệ thống thư mục cho mọi người dùng. Những thông tin trạng thái như thế phải được mail máy chủ lưu giữ cho tất cả các phiên làm việc. Nên nhớ rằng POP3 máy chủ không lưu giữ trạng thái của mỗi người dùng sau khi phiên làm việc kết thúc.

IMAP có một đặc tính quan trọng là có những lệnh cho phép user agent chỉ lấy một số thành phần trong thư. Ví dụ: user agent có thể lấy phần tiêu đề hoặc một phần trong thư có nhiều phần. Điều này rất có ích khi kết nối giữa user agent và mail máy chủ chậm, người dùng có thể không cần tải tất cả thư trong

hộp thư của mình. Đặc biệt có thể tránh tải những thư chứa nội dung âm thanh hay hình ảnh có kích thước lớn.

Phiên làm việc IMAP gồm 3 giai đoạn: giai đoạn thiết lập kết nối giữa máy khách (user agent) và IMAP máy chủ, giai đoạn máy chủ chấp nhận kết nối và giai đoạn tương tác máy khách lmáy chủ. tương tác máy khách/máy chủ của IMAP tương tự nhưng phong phú hơn nhiều tương tác trong POP3. Máy chủ luôn ở một trong bốn trạng thái. Trong trạng thái chưa kiểm chứng là trạng thái khởi đầu, khi đó người dùng phải đăng nhập hệ thống trước khi thực hiện các lệnh. Trạng thái đã kiểm chứng, người dùng phải chọn một thư mục trước khi gửi lệnh. Trong trạng thái lựa chọn, người dùng có thể sử dụng những lệnh có thể tác động tới bản tin (như lấy, xoá, chuyển thư). Cuối cùng là trạng thái thoát (logout) khi kết thúc phiên làm việc.

3.3 Một số ứng dụng quen thuộc

3.3.1 Trình duyệt web

Trình duyệt web là một phần mềm ứng dụng cho phép người sử dụng xem và tương tác với các văn bản, hình ảnh, đoạn phim, nhạc, trò chơi và các thông tin khác ở trên một trang web của một địa chỉ web trên mạng toàn cầu hoặc mạng nội bộ. Văn bản và hình ảnh trên một trang web có thể liên kết tới các trang web khác của cùng một địa chỉ web hoặc địa chỉ web khác. Trình duyệt web cho phép người sử dụng truy cập các thông tin trên các trang web một cách nhanh chóng và dễ dàng thông qua các liên kết đó. Trình duyệt web đọc định dạng HTML để hiển thị, do vậy một trang web có thể hiển thị khác nhau trên các trình duyệt khác nhau.

Trình duyệt web thường giao tiếp với máy chủ web bằng cách sử dụng giao thức HTTP để lấy nội dung các trang web. HTTP cho phép các trình duyệt web gửi thông tin đến các máy chủ web, cũng như lấy các trang web về. HTTP được sử dụng rộng rãi nhất là HTTP/1.1 được định nghĩa trong tài liệu RFC 2616. Các trang được xác định bằng cách thức của một URL (Uniform Resource Locator), được coi như một địa chỉ bắt đầu bằng cụm http:// để báo cho biết sử dụng giao thức HTTP.

Định dạng file của một trang web thường là HTML (ngôn ngữ đánh dấu siêu văn bản) và được xác định bởi giao thức HTTP sử dụng kiểu nội dung MIME. Phần lớn các trình duyệt hỗ trợ nhiều định dạng file khác bên cạnh HTML, như là các định dạng ảnh JPEG, PNG, GIF... và có thể mở rộng để hỗ trợ nhiều hơn nhờ sử dụng các plug-in. Sự kết hợp của kiểu nội dung HTTP và đặc tả giao thức URL cho phép các nhà thiết kế trang web có thể đưa ảnh, hoạt hình, video, âm thanh và đa phương tiện được streaming vào trang web, hoặc có thể truy cập chúng thông qua trang web.

World Wide Web là một thư viện khổng lồ với nhiều trang được lưu trữ trong các máy tính khác nhau trên khắp thế giới. Cùng với www, người sử dụng có thể làm được nhiều việc hơn là chỉ đọc thông tin như một tạp chí thông thường. Để truy cập vào WWW bạn cần một chương trình gọi là trình duyệt web (Web browsers). Hai trình duyệt web thông dụng nhất là Netscape Navigator và

Microsoft Internet Explorer. Nếu nhà cung cấp dịch vụ Internet (ISP Internet Service Provider) của bạn cung cấp cho bạn một phiên bản cũ bạn có thể tự download phiên bản mới nhất từ mạng Internet.

Để truy nhập một trong rất nhiều các trang thông tin điện tử này, người sử dụng phải khởi động trình duyệt và nhập vào địa chỉ của trang Web. Tất cả các tài nguyên trên Internet đều có URL (Uniform Resource Locator, một xâu ký tự có định dạng để xác định vùng tài nguyên) hoặc địa chỉ (address), ví dụ http://www.microsoft.com. Http là viết tắt của HyperText Transfer Protocol(Giao thức truyền siêu văn bản). Nó thông báo cho trình duyệt của bạn đây là một tài liệu Web và trình duyệt sẽ dùng giao thức truyền siêu văn bản để truy xuất thông tin.

Tiếp theo là cụm từ (www.microsoft.com) tên vùng của máy chủ mà người sử dụng cần truy nhập, đó chính là địa chỉ của máy chủ chứa thông tin trên mạng Internet. Tên vùng là xác định nên chúng ta có thể nhớ địa chỉ các máy tính trên mạng một cách dễ dàng. Thực tế địa chỉ là một loạt các chữ số và máy tính phải tìm trong một danh sách lớn các địa chỉ và tìm ra địa chỉ khớp với nó. Mọi từ theo sau tên vùng đều là đường dẫn đến thư mục và file mà trình duyệt cần truy nhập.

Trên trang thông tin đã được được tải về sẽ bao gồm các thành phần cơ bản của một trang Web như là văn bản, hình ảnh, và một vài từ với những màu sắc khác nhau. Những từ có màu khác này thường là những liên kết (hyper links) đến các trang khác, nếu nhấn chuột một trang mới tương ứng với liên kết sẽ được nạp. Đôi khi việc liên kết này cũng được bố trí trên các hình ảnh/biểu tượng, khi đó nếu ấn vào hình ảnh hoặc biểu tượng trên sẽ được nối đến một địa chỉ khác giống như là những cụm từ đổi màu.

Khi duyệt Web, người sử dụng có thể mở liên tiếp từ trang này sang trang khác, từ liên kết này đến liên kết khác, nếu muốn quay lại một trang đã qua thì nhấn nút Back (hầu hết mọi trình duyệt đều có chức năng này trên thanh công cụ của trình duyệt). Thông thường, người sử dụng cần phải hiểu và sử dụng thành thạo các chức năng sau:

- Nút Home sẽ đưa về trang mặc định, người sử dụng có thể tự đặt trang này theo yêu cầu.
- HyperLink (khi đưa chuột vào đây con trỏ chuột sẽ chuyển thành hình bàn tay): Đây là một liên kết, nhấn chuột vào sẽ dẫn tới một trang khác. Các liên kết này thường có màu khác với màu của phần văn bản khác. Một liên kết đã được ấn sẽ có màu khác với màu ban đầu của nó, điều này giúp bạn nhận biết những trang nào người dùng đã xem.
- Addresss: là nơi điền địa chỉ của trang Web muốn truy nhập.
- Navigation bar (thanh điều hướng) được thiết kế để giúp người dùng chọn nhanh một số trang Web mà chỉ cần nhấn nút. Trên thanh có nhiều phần khác nhau tương ứng với các địa chỉ.

3.3.2 Phần mềm đọc thư điện tử

Thư điện tử (Email) là phương pháp gửi một bức thư qua mạng Internet. Việc gửi thư điện tử có những ưu thế hơn hẳn phương pháp gửi thư thông thường: người sử dụng có thể gửi nó bất cứ lúc nào trong ngày mà không cần phải rời khỏi nhà hay phòng làm việc và nó sẽ được đưa tới hộp thư người nhận trong vài phút sau đó.

Để sử dụng email, người sử dụng phải truy nhập vào trang Web mail của nhà cung cấp dịch vụ thư điện tử và thao tác theo hướng dẫn của nhà cung cấp dịch vụ này. Đa số các máy tính của người sử dụng cài đặt phần cần phần mềm gọi là email client, chương trình email client cho PC phổ biến hiện nay là Outlook Express (Sẽ được trình bày chi tiết trong chương sau).

Địa chỉ email thường gồm tên, theo sau là @, và tiếp theo là tên miền của nhà cung cấp dịch vụ thư điện tử. Phần tên do người sử dụng tự qui định, phần tên miền phải tuân thủ theo tên miền của nhà cung cấp dịch vụ. Ví dụ @yahoo.com là tên miền của YAHOO....

3.3.3 Trình đa phương tiện

Trình đa phương tiện (Media player) là một thuật ngữ đặc thù để chỉ những phần mềm máy tính có chức năng thực thi các tập tin đa phương tiện. Hầu hết các trình đa phương tiện đều hỗ trợ một số các định dạng tập tin media, trong đó có cả các tập tin audio (âm thanh số) và video (hình ảnh số). Một số trình đa phương tiện lại chỉ tập trung vào các định dạng âm thanh (audio) hoặc video, và được gọi tương ứng với đó là trình audio (audio player) và trình video (video player). Những nhà sản xuất ra các trình ứng dụng này thường chú trọng vào việc cung cấp cho người dùng những trải nghiệm tốt nhất về các định dạng mà họ tập trung hướng đến.

Trình đa phương tiện quen thuộc với nhiều người dùng nhất là Windows Media Player được tích hợp sẵn trên Microsoft Windows. Hiện phiên bản mới nhất của ứng dụng này là Windows Media Player 11, được tích hợp cùng với Windows Vista, và cũng có thể tải về cài và sử dụng trên Windows XP SP2. Người dùng Mac OS X có thể dùng Quicktime Player để xem các định dạng Quicktime và iTunes để thực thi nhiều định dạng media khác nhau. Winamp là trình đa phương tiện chỉ chạy trên Windows xong hỗ trợ cả Apple iPod và nhiều thiết bị di động như Creative's Zen hay chơi audio và video. Với các bản phân phối của Linux, cũng có một số lượng đa dạng các trình đa phương tiện như VLC, MPlayer, xine, hay Totem.

3.3.4 Tiện ích Telnet, rlogin, ssh

TELNET (viết tắt của TErminaL NETwork) là một tiện ích mạng giao diện dòng lệnh được dùng để cung cấp những phiên giao dịch đăng nhập vào các máy trên mạng, tạo cảm giác như một thiết bị cuối được gắn vào một máy tính khác. TELNET vốn được cài đặt sẵn trong hầu hết các hệ điều hành, song với sự tiến triển gần đây SSH (Secure Shell) trở nên một giao thức có ưu thế hơn trong việc truy cập từ xa. SSH cũng được cài đặt sẵn cho hầu hết các loại máy vi tính.

CHƯƠNG 4: TẦNG VẬN TẢI

Nằm giữa tầng ứng dụng và tầng mạng, tầng vận tải là tầng trung tâm trong kiến trúc phân tầng với nhiệm vụ cung cấp dịch vụ truyền thông giữa các tiến trình ứng dụng chạy trên các máy tính khác nhau. Giao thức tầng vận tải cung cấp một kênh truyền logic (ảo) giữa các tiến trình ứng dụng chạy trên máy tính khác nhau. Gọi là logic vì không tồn tại một đường truyền vật lý thực sự giữa hai tiến trình. Các tiến trình ứng dụng sẽ sử dụng đường truyền ảo này để trao đổi bản tin mà không phải bận tâm về cơ sở hạ tầng của môi trường vật lý thực sự.

Ở phía gửi, thực thể vận tải chèn bản tin mà nó nhận được từ tiến trình ứng dụng vào các 4-PDU (là đơn vị dữ liệu của giao thức tầng vận tải - Protocol Data Unit). Công việc được thực hiện bằng cách chia bản tin thành nhiều đoạn nhỏ, bổ sung vào đầu mỗi đoạn tiêu đề của tầng vận tải để tạo ra gói dữ liệu của tầng vận tải (4-PDU). Sau đó tầng vận tải truyền gói dữ liệu (4-PDU) xuống tầng mạng, tại đây mỗi gói này được đặt trong gói dữ liệu của tầng mạng (3-PDU). Ở phía nhận, tầng vận tải nhận gói dữ liệu tớ tầng mạng, loại bỏ phần tiêu đề của gói dữ liệu 4-PDU, ghép chúng lại thành một bản tin hoàn chỉnh và chuyển cho tiến trình ứng dụng nhận. Trên mạng máy tính có thể có nhiều giao thức hoạt động ở tầng vận tải. Mỗi giao thức có thể cung cấp các dịch vụ với chất lượng khác nhau cho ứng dụng. Tất cả giao thức tầng giao vận đều cung cấp dịch vụ ghép kênh **(**multiplex**)** và phân kênh **(**demultiplex**),** điều này sẽ nói cụ thể trong các phần sau. Ngoài dịch vụ ghép kênh/phân kênh, tầng vận tải còn có thể cung cấp các dịch vụ khác cho tiến trình ứng dụng như truyền dữ liệu tin cậy.

4.1 Ghép kênh và phân kênh, các giao thức TCP và UDP

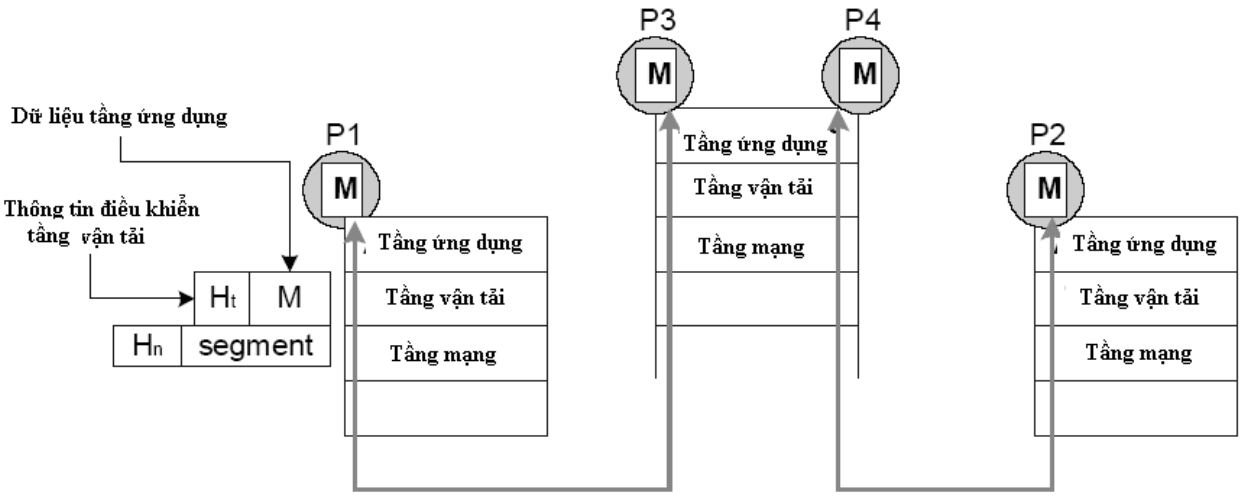
4.1.1 Ghép kênh và phân kênh

Ghép kênh/phân kênh không phải là một trong những chức năng chính của tầng vận tải, nhưng nó lại là công việc rất cần thiết. Để hiểu tại sao như vậy, ta thấy rằng IP truyền dữ liệu giữa hai thiết bị đầu cuối, mỗi thiết bị có một địa chỉ IP nhất định. IP không truyền dữ liệu giữa các tiến trình ứng dụng chạy trên các máy tính. Mở rộng việc gửi - từ máy tính đến máy tính - từ tiến trình đến tiến trình là công việc ghép kênh và phân kênh.

Tại máy tính nhận, tầng vận tải nhận đoạn dữ liệu từ tầng mạng ngay phía dưới và có trách nhiệm gửi dữ liệu trong đoạn dữ liệu này tới tiến trình ứng dụng thích hợp trên máy tính. Giả sử tại một thời điểm máy tính của người sử dụng đang tải trang Web, chạy một phiên FTP và hai phiên Telnet. Như vậy có tất cả 4 tiến trình đang chạy tại tầng ứng dụng: 2 tiến trình Telnet, 1 tiến trình FTP, và 1 tiến trình HTTP. Khi tầng vận tải trong máy tính của người sử dụng nhận được dữ liệu từ tầng mạng chuyển lên, nó phải gửi dữ liệu trong đó tới 1 trong 4 tiến trình trên.

Mỗi đoạn tin của tầng vận tải có trường xác định tiến trình nhận dữ liệu. Tầng vận tải bên nhận sẽ sử dụng trường này để xác định rõ tiến trình nhận và

gửi dữ liệu trong đoạn tin tới tiến trình đó. Công việc chuyển dữ liệu trong đoạn tin tới đúng tiến trình ứng dụng được gọi là phân kênh. Tại thiết bị gửi, tầng vận tải nhận dữ liệu từ nhiều tiến trình ứng dụng khác nhau, tạo đoạn tin chứa dữ liệu cùng với một số thông tin tiêu đề và cuối cùng chuyển đoạn tin xuống tầng mạng. Quá trình trên được gọi là ghép kênh.



Hình 4.1 Dịch vụ ghép kênh, phân kênh

UDP và TCP thực hiện việc ghép kênh và phân kênh nhờ hai trường đặc biệt ở đầu đoạn tin: trường định danh cổng tiến trình gửi (cổng nguồn - source port number) và trường định danh cổng tiến trình nhận (cổng đích - destination port number), hai trường này được minh họa trên hình 4.2. Chúng xác định một tiến trình ứng dụng duy nhất chạy trên máy tính. Tất nhiên UDP và TCP còn có nhiều trường khác mà chúng ta sẽ nghiên cứu sau.

Cổng nguồn	Cổng đích
Các thông tin điều khiển khác	
Dữ liệu của tầng ứng dụng	

Hình 4.2 Trường địa chỉ tiến trình gửi, tiến trình nhận trong đoạn tin

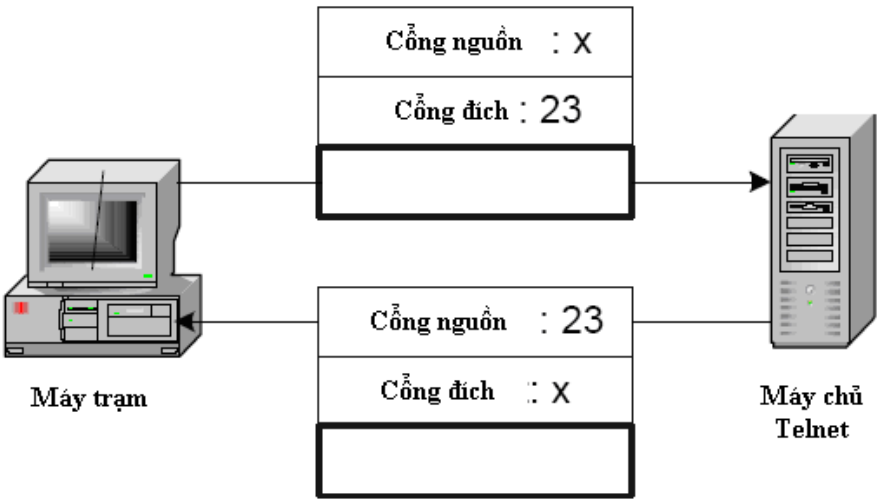
Số hiệu cổng là một con số 16 bit, nhận giá trị tớ 0 tới 65535. Các giá trị tớ 0 đến 1023 là các giá trị dùng cho các dịch vụ công cộng, tức là chỉ để cho các ứng dụng thông dụng như HTTP, FTP sử dụng. HTTP sử dụng cổng 80, FTP sử dụng cổng 20 và 21. Danh sách các cổng thông dụng có thể tham khảo

trong RFC 1700. Khi xây dựng một ứng dụng mới, phải xác định số hiệu cổng cho ứng dụng này.

Bảng 4.1: Một số ứng dụng và giao thức tại tầng vận tải

Ứng dụng	Giao thức tầng ứng	Tầng vận tải tương ứng
Thư điện tử	SMTP	TCP
Truy cập từ xa	Telnet	TCP
Web	HTTP	TCP
Truyền file	FTP	TCP
File máy chủ	NFS	thường là UDP
Đa phương tiện	Phụ thuộc hãng sản	thường là UDP
Điện thoại qua	Phụ thuộc hãng sản	thường là UDP
Quản lý mạng	SNMP	thường là UDP
Định tuyến	RIP	thường là UDP
Tên miền	DNS	thường là UDP

Mỗi ứng dụng chạy trên thiết bị đầu cuối có số hiệu cổng nhất định. Bởi vậy vấn đề đặt ra là tại sao mỗi đoạn tin ở tầng vận tải đều có trường số hiệu cổng nguồn và đích. Một thiết bị đầu cuối có thể chạy đồng thời hai tiến trình cùng kiểu, như vậy số hiệu cổng đích chưa đủ để phân biệt các tiến trình. Giả sử máy chủ web chạy tiến trình HTTP xử lý các bản tin yêu cầu; khi máy chủ web phục vụ nhiều yêu cầu cùng một lúc thì máy chủ sẽ chạy nhiều tiến trình trên cổng 80. Để gửi dữ liệu đến tiến trình nhận, phải xác định số hiệu cổng của phía gửi (cổng nguồn).

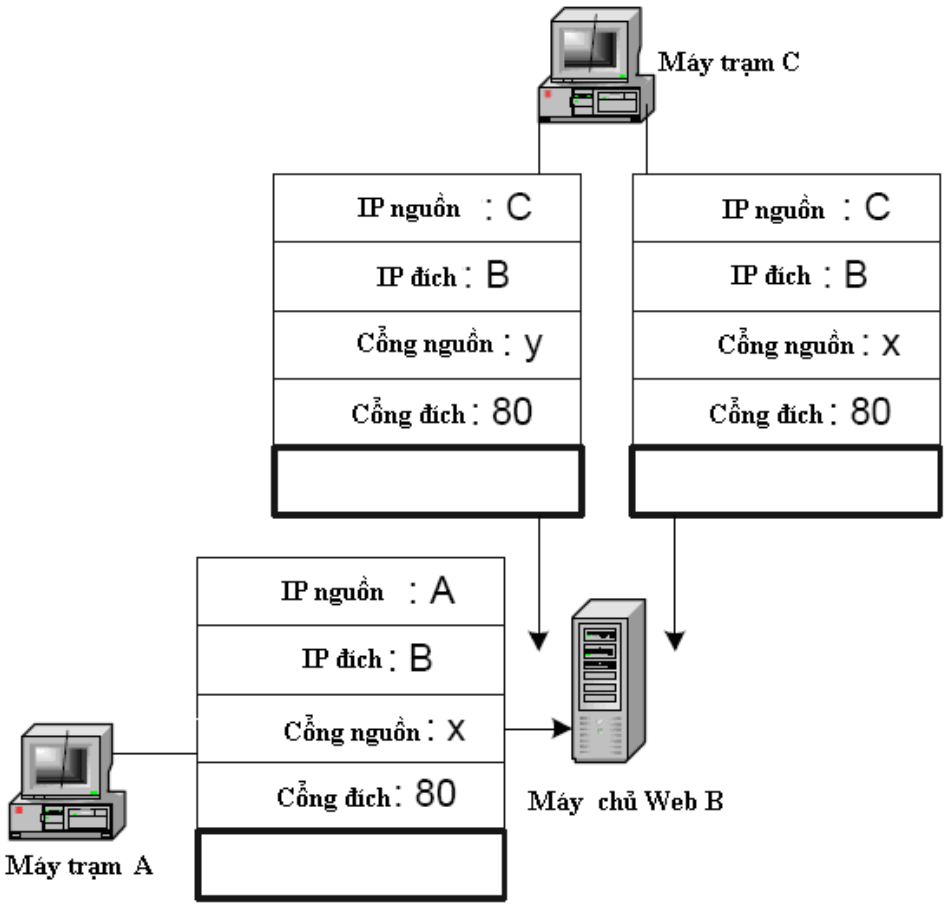


Hình 4.3 Sử dụng số hiệu cổng nguồn/đích trong trình ứng dụng khách/chủ

Thông thường, máy tính nào khởi đầu trước đóng vai trò máy khách, máy tính kia đóng vai trò máy chủ. Xét ví dụ một tiến trình người sử dụng Telnet máy chủ. Đoạn tin ở tầng vận tải khi rời máy khách chuyển tới máy chủ có số

hiệu cổng nguồn là một giá trị ngẫu nhiên (không được phép là giá trị của cổng đã có một tiến trình nào đó sử dụng) và số hiệu cổng đích (cổng của tiến trình nhận) bắt buộc phải là 23. Giả sử phía máy khách chọn số hiệu cổng là X thì mỗi đoạn tin được gửi từ ứng dụng Telnet có cổng nguồn là x, cổng đích là 23. Khi đoạn tin tới máy chủ, căn cứ vào số hiệu cổng là 23, máy chủ sẽ chuyển dữ liệu trong đoạn tin đó tới tiến trình Telnet.

Đoạn tin truyền từ máy chủ tới máy khách sẽ đảo ngược số hiệu cổng. Cổng nguồn bây giờ sẽ là cổng của ứng dụng có giá trị 23, còn cổng đích sẽ là X (là số hiệu cổng nguồn trong đoạn tin gửi từ máy khách tới máy chủ). Khi đoạn tin tới, máy khách cũng sẽ căn cứ vào số hiệu cổng nguồn và đích để gửi dữ liệu trong segment tới đúng tiến trình ứng dụng, hình 4.3 minh họa quá trình trên.



Hình 4.4 Các tiến trình cùng tạo kết nối đến một dịch vụ

Xét trường hợp hai máy khách khác nhau cùng thiết lập phiên làm việc tới một máy chủ và mỗi máy khách đều chọn cổng nguồn là X, điều này hoàn toàn có thể xảy ra với những máy chủ có nhiều người truy cập. Khi đó máy chủ phải sử dụng địa chỉ IP trong gói dữ liệu IP chứa đoạn tin. Trên hình 4.4, máy C có hai phiên làm việc và máy A có một phiên làm việc HTTP tới cùng máy chủ B. cả ba máy A, B, C đều có địa chỉ IP phân biệt lần lượt là A, B, C.

Máy C sử dụng hai cổng nguồn (X,Y) khác nhau cho hai kết nối HTTP tới B. A chọn số hiệu cổng nguồn độc lập với C nên nó có thể gán cổng nguồn x cho kết nối HTTP của mình. Tuy nhiên máy chủ B vẫn có thể thực hiện phân kênh hai đoạn tin có lặp cổng giống nhau do địa chỉ IP nguồn khác nhau. Tóm lại, bên nhận sử dụng cả ba giá trị (địa chỉ IP nguồn, số hiệu cổng nguồn, số hiệu cổng đích) để xác định tiến trình ứng dụng nhận.

4.1.2 Giao thức TCP

TCP là một giao thức tầng vận tải cung cấp dịch vụ truyền dữ liệu tin cậy, nó sử dụng nhiều nguyên lý để đảm bảo truyền tính tin cậy như: phát hiện lỗi, đánh số thứ tự các đoạn tin, cơ chế xác nhận. Giao thức TCP thuộc loại kết nối có hướng vì trước khi gửi dữ liệu của lớp ứng dụng phải có thủ tục "Bắt tay", nghĩa là chúng phải gửi một số đoạn tin đặc biệt để xác định các tham số đảm bảo cho quá trình truyền dữ liệu. TCP được đặc tả trong các khuyến nghị RFC 793, RFC 1122, RFC 1323, RFC 2018, RFC 2581. Giao thức TCP có các đặc điểm sau:

- Định hướng kết nối: Trước khi truyền dữ liệu phải thực hiện thủ tục thiết lập liên kết, sau khi truyền dữ liệu xong thì một trong hai bên hoặc cả hai bên gửi tín hiệu yêu cầu hủy bỏ liên kết.
- Đánh số tuần sự: Mỗi đoạn tin trước khi gửi đi phải được đánh số tuần tự, dựa vào cơ chế này mà bên nhận sẽ sắp xếp lại các đoạn tin chính xác và đồng thời phát hiện được những đoạn tin bị thất lạc.
- Đảm bảo tính tin cậy: Lỗi có thể xảy ra khi một đoạn tin nào đó bị thất lạc hoặc nội dung đoạn tin bị thay đổi. Trong cả hai trường hợp trên, giao thức TCP sẽ yêu cầu gửi lại.
- Điều khiển lưu lượng: năng lực xử lý của mỗi máy tính chỉ có hạn nhất định (thường phụ thuộc ba tài nguyên cơ bản trong mỗi máy tính: tốc độc CPU, dung lượng bộ nhớ, tốc độ đọc/ghi thiết bị lưu trữ, ngoài ra có thể xét thêm các yếu tố khác như: số lượng người dùng, băng thông mạng...). Nếu không có cơ chế điều khiển lưu lượng thì bên nhận có thể không kịp xử lý các đoạn tin gửi đến và dẫn đến tình trạng lỗi như: tràn bộ nhớ, treo hệ thống.

4.1.3 Giao thức UDP

Giao thức UDP (RFC 768) là giao thức thuộc tầng vận tải, ngoài chức năng ghép kênh/phân kênh, UDP có thêm cơ chế phát hiện lỗi đơn giản (chức năng phát hiện và sửa lỗi sẽ được chuyển lên tầng ứng dụng). Có thế nói, nếu sử dụng UDP thì gần như ứng dụng làm việc trực tiếp với tầng mạng IP. UDP lấy bản tin từ tiến trình ứng dụng, chèn thêm một số trường điều khiển, trong đó có hai trường địa chỉ cổng nguồn và đích cho chức năng Ghép kênh/phân kênh để tạo nên dữ liệu (gọi là datagram) và được chuyển xuống tầng mạng. Tầng mạng đặt datagram này trong gói tin và cố gắng gửi gói IP tới máy tính nhận. UDP không đòi hỏi bên gửi và bên nhận phải thiết lập liên kết trước khi trao đổi dữ liệu. Vì vậy UDP được xem là giao thức kết nối vô hướng hay không liên kết. Giao thức UDP dường như không có có nhiều ưu điểm như giao thức TCP: truyền dữ liệu

tin cậy, kiểm soát lưu lượng…, tuy nhiên trên thực tế giao thức UDP được sử dụng nhiều hơn vì những đặc điểm sau:

- **Không cần thiết lập liên kết**: Giao thức TCP sử dụng cơ chế "bắt tay" ba bước trước khi truyền dữ liệu nhưng giao thức UDP không cần cơ chế. Vì vậy UDP sẽ không phải mất thời gian để thiết lập đường truyền. Đây chính là nguyên nhân dịch vụ DNS chạy trên UDP chứ không phải là TCP, DNS sẽ chạy chậm nếu sử dụng TCP. HTTP sử dụng TCP vì các đối tượng Web cần được tải về chính xác, do đó yêu cầu một đường truyền tin cậy.
- **Không duy trì trạng thái kết nối**. TCP ghi nhớ trạng thái kết nối trên hệ thống đầu cuối. Trạng thái kết nối bao gồm vùng đệm của bên nhận và bên gửi, các tham số kiểm soát tắc nghẽn, số tuần tự phát và số biên nhận. Các thông số đó giúp TCP triển khai dịch vụ truyền tin tin cậy và cơ chế kiểm soát tắc nghẽn. UDP không phải lưu giữ những thông tin như vậy, do đó nếu phía máy chủ sử dụng UDP thì có khả năng phục vụ đồng thời nhiều máy khách hơn.
- **Thông tin điều khiển ít hơn**: Thông tin điều khiển của đoạn tin TCP tối thiểu là 20 bytes trong khi UDP chỉ có 8 bytes.
- **Không kiểm soát tốc độ gửi**. TCP có cơ chế kiểm soát tắc nghẽn, điều chỉnh tốc độ gửi khi xảy ra tắc nghẽn. Cơ chế điều chỉnh này có thể ảnh hưởng tới những ứng dụng thời gian thực, đó là những ứng dụng chấp nhận mất mát dữ liệu nhưng lại đòi hỏi phải có một tốc độ truyền tối thiểu. Tốc độ truyền dữ liệu của UDP chỉ bị giới hạn bởi tốc độ sinh dữ liệu của tầng ứng dụng, khả năng máy tính nguồn (CPU, tốc độ đồng hồ) và tốc độ truy nhập mạng. Bên nhận không nhất thiết phải nhận toàn bộ dữ liệu, khi nghẽn mạng, một phần dữ liệu có thể bị mất do tràn vùng đệm ở các thiết bị mạng, tốc độ nhận có thể bị giới hạn do tắc nghẽn ngay cả khi tốc độ gửi không bị hạn chế.

Các ứng dụng phổ biến như: Email, truy cập từ xa, Web và truyền tập tin chạy trên nền TCP vì chúng cần đến dịch vụ truyền dữ liệu tin cậy. Tuy nhiên có một số ứng dụng khác thích hợp với UDP hơn TCP như: Giao thức cập nhật bảng định tuyến RIP sử dụng UDP, bởi vì việc cập nhật được thực hiện định kỳ, cho nên dù cập nhật bị mất nhưng sẽ có cập nhật mới sau một khoảng thời gian ngắn. UDP được sử dụng để gửi dữ liệu quản trị mạng SNMP, trong trường hợp này UDP thích hợp hơn TCP vì các tiến trình quản trị mạng thường hoạt động khi mạng có sự cố không thể truyền dữ liệu chính xác hay cơ chế kiểm soát tắc nghẽn không làm việc. DNS sử dụng UDP, do đó có thể tránh được thời gian trễ của giai đoạn thiết lập kết nối.

Ngày nay UDP thường được các ứng dụng đa phương tiện như điện thoại Internet, hội thảo từ xa, các ứng dụng thời gian thực. Các ứng dụng này chấp nhận mất mát, lỗi trên một phần dữ liệu, vì thế truyền dữ liệu tin cậy không phải là tiêu chí quan trọng nhất đánh giá sự thành công của ứng dụng. Hơn nữa các ứng dụng thời gian thực như điện thoại Internet và hội thảo từ xa không thích ứng được với cơ chế kiểm soát tắc nghẽn của TCP, do đó các ứng dụng đa phương tiện và thời gian thực thường lựa chọn UDP ở tầng vận tải.

Mặc dù đã triển khai trong thực tế, song việc các ứng dụng đa phương tiện sử dụng UDP gây ra nhiều tranh cãi. UDP không kiểm soát được tắc nghẽn nên đó là nguyên nhân tiềm ẩn gây nghẽn mạng, khi đó chỉ rất ít thông tin được chuyển trên mạng. Nếu tất cả mọi người đều xem phim trực tuyến thì các gói tin sẽ bị tràn bộ đệm ở các thiết bị mạng và không có ai sử dụng được dịch vụ này. Thiếu cơ chế kiểm soát tắc nghẽn là một trong những nhược điểm của giao thức UDP.

4.1.3.1 Cấu trúc dữ liệu của giao thức UDP

Cấu trúc dữ liệu của giao thức UDP được mô tả trong RFC 768. dữ liệu của ứng dụng nằm trong trường dữ liệu của UDP datagram. Ví dụ đối với DNS, trường dữ liệu chứa bản tin yêu cầu hay bản tin trả lời.

Cổng nguồn	Cổng đích
Chiều dài	Checksum
Dữ liệu tầng ứng dụng	

Hình 4.5 Cấu trúc UDP datagram

Phần thông tin điều khiển UDP có bốn trường, độ lớn mỗi trường là hai byte. Số hiệu cổng cho phép thiết bị gửi chuyển dữ liệu tới đúng tiến trình chạy trên thiết bị nhận. Trường Checksum được bên nhận sử dụng để kiểm tra trong datagram có lỗi hay không. Trường độ dài (Length) cho biết độ dài (tính theo byte) của toàn bộ UDP datagram, kể cả phần thông tin điều khiển.

4.1.3.2 Cách tính UDP checksum

UDP checksum được sử dụng để phát hiện lỗi, nó được tính như sau: tính giá trị bù một của tổng các từ 16 bit trong datagram, giá trị nhận được được đặt vào trường checksum trong UDP datagram. Giả sử có ba từ 16 bit sau đây:

```
0110011001100110
0101010101010101
0000111100001111
```

Tổng hai từ đầu là:

```
0110011001100110
0101010101010101
1011101110111011
```

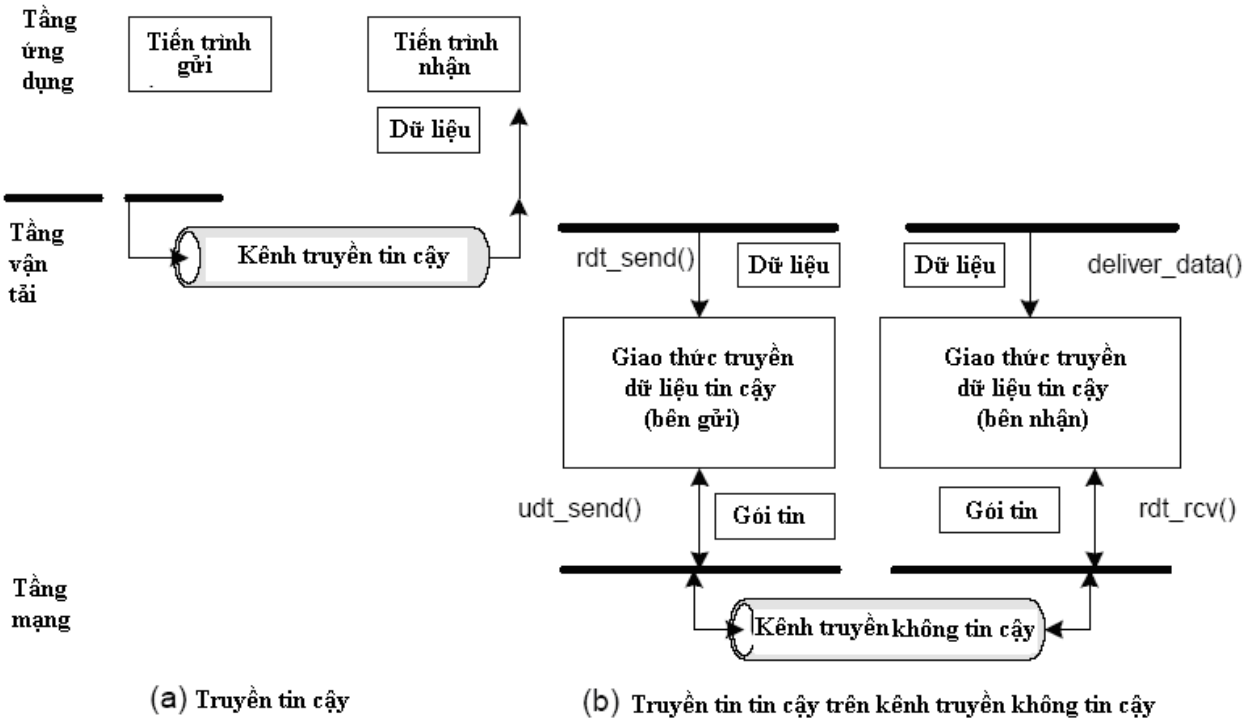
Cộng từ thứ ba vào, ta có:

```
1011101110111011
0000111100001111
1100101011001010
```

Cách lấy bù một là đảo 0 thành 1 và 1 thành 0. Vì vậy kết quả phép lấy bù một của 1100101011001010 là 0011010100110101 và đó chính là giá trị checksum. Tại phía nhận, tất cả bốn từ (kể cả checksum) được cộng lại. Nếu dữ liệu không có lỗi thì tổng nhận được phải chuỗi các bit 1, tức là 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1. Nếu có một bít nào đó bằng 0 thì chắc chắn dữ liệu nhận được đã bị lỗi. Mặc dù UDP có thể phát hiện lỗi nhưng nó không xử lý lỗi. Datagram lỗi có thể bị loại bỏ, nhưng cũng có thể được chuyển lên tầng ứng dụng kèm theo cảnh báo.

4.2 Các nguyên lý truyền tin cậy

Truyền dữ liệu tin cậy là một trong những chức năng chính của tầng vận tải, tuy nhiên chức năng này cũng xuất hiện ở tầng liên kết dữ liệu hay tầng ứng dụng. Hình 4.5a là sơ đồ cấu trúc của quá trình truyền dữ liệu tin cậy, tầng dưới cung cấp một dịch vụ truyền tin cậy cho các thực thể ở tầng trên.



Hình 4.5 Nguyên lý truyền dữ liệu tin cậy

Trên đường truyền tin cậy này, dữ liệu không bị lỗi (bit 0 biến thành bit 1 hoặc ngược lại), không bị mất và được nhận theo đúng thứ tự gửi. Đây chính là dịch vụ mà giao thức TCP cung cấp cho các ứng dụng Internet. Để thực hiện công việc này, người ta cần đến những giao thức truyền dữ liệu tin cậy. Nguyên nhân là tầng phía dưới của giao thức tin cậy là không tin cậy. Ví dụ TCP là giao thức truyền dữ liệu tin cậy nằm ở phía trên giao thức truyền không tin cậy (IP) giữa hai thiết bị đầu cuối trên mạng. Để đơn giản chúng ta coi tầng phía dưới là một đường truyền điểm nối điểm (point-to-point) không tin cậy (Hình 4.5b). Thực thể gửi sẽ nhận dữ liệu từ phía trên chuyển xuống qua hàm rdt-send(), phía nhận sử dụng hàm rdt_rcv() để lấy gói dữ liệu từ đường truyền. Để chuyển dữ liệu lên tầng trên, phía nhận sử dụng hàm deliver_data().

4.2.1 Xây dựng giao thức truyền dữ liệu tin cậy

Để tìm hiểu cơ chế truyền dữ liệu tin cậy, chúng ta sẽ mô tả trạng thái của phía nhận và phía gửi bằng kỹ thuật máy hữu hạn trạng thái (finite state machine - FSM).

4.2.1.1 Truyền dữ liệu tin cậy trên kênh tin cậy hoàn toàn

Giao thức đầu tiên, đơn giản nhất được đưa ra - rdt 1.0 sử dụng kênh truyền tin cậy ở phía dưới. FSM của bên gửi và bên nhận đều chỉ có một trạng thái (xem hình 4.6). Mặc dù mỗi FSM trong hình 4.6 chỉ có một trạng thái nhưng vẫn cần đến sự chuyển trạng thái để quay về chính trạng thái cũ. Sự kiện kích hoạt việc chuyển trạng thái được đặt phía trên đường kẻ nằm ngang, đó là nhãn sự kiện. Phía bên dưới đường kẻ nằm ngang là những hành động mà thực thể phải thực hiện ngay khi sự kiện đó xảy ra (thực hiện trước khi thực thể chuyển sang trạng thái mới). Với rdt1.0, việc gửi đơn giản chỉ là nhận dữ liệu từ tầng trên thông qua sự kiện rdt_send(data), tạo ra đoạn dữ liệu bằng hàm make_data (packet,data) và gửi đoạn dữ liệu lên kênh truyền. Trên thực tế, sự kiện rdt_send(data) là kết quả của một thủ tục, ví dụ khi ứng dụng phía trên sử dụng hàm rdt_send().

Hình 4.6 Giao thức cho kênh truyền tin cậy hoàn toàn

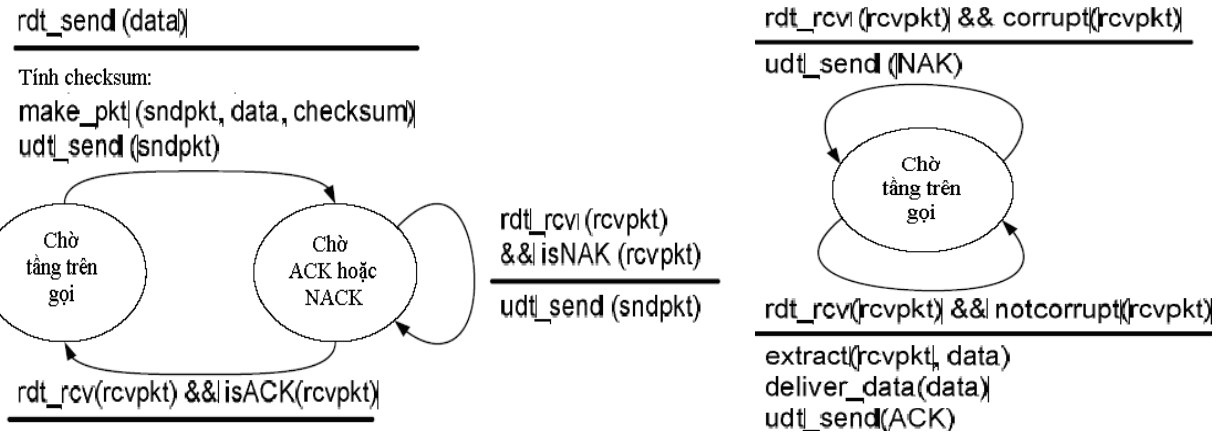
Ở bên nhận, rdt nhận gói dữ liệu (packet) từ kênh truyền bằng sự kiện rdt_rcv(packet), lấy dữ liệu ra khỏi gói dữ liệu bằng hàm extract (packet,data) và đưa dữ liệu lên tầng trên. Trên thực tế, sự kiện rdt_rcv(packet) là kết quả của một thủ tục, ví dụ khi ứng dụng phía trên sử dụng hàm rdt_rcv().

Trong giao thức đơn giản này, không có sự khác biệt giữa dữ liệu (data) với gói dữ liệu (packet). Như vậy, tất cả packet đều được truyền từ phía gửi cho phía nhận. với kênh truyền tin cậy, phía nhận không cần thiết phải phản hồi cho phía gửi vì nó chắc rằng không có chuyện gì xảy ra. Chú ý rằng, chúng ta đã giả thiết phía nhận có thể nhận dữ liệu với tốc độ phía gửi gửi. Vì vậy, phía nhận không cần yêu cầu phía gửi gửi chậm lại.

4.2.1.2 Truyền dữ liệu tin cây trên kênh truyền có lỗi bit

Một dạng kênh truyền thực tế hơn là gói dữ liệu trên kênh truyền có thể bị lỗi. Thường bit bị lỗi trên đường truyền vật lý của mạng. Giả sử tất cả các gói dữ liệu truyền đi đều đến được đích và theo đúng thứ tự gửi mặc dù các bit trong

gói dữ liệu có thể bị lỗi. Trong thực tế cuộc sống, khi hai người nói chuyện với nhau, nếu người nghe đã rõ thì xác nhận, ngược lại sẽ yêu cầu người nói nhắc lại. Cơ chế này đã được áp dụng cho rdt 2.0. Trong mạng máy tính, giao thức truyền tin cậy dựa trên cơ chế truyền lại như vậy được gọi là các giao thức ARQ (Automatic Repeat request).



Bên gửi Bên nhận

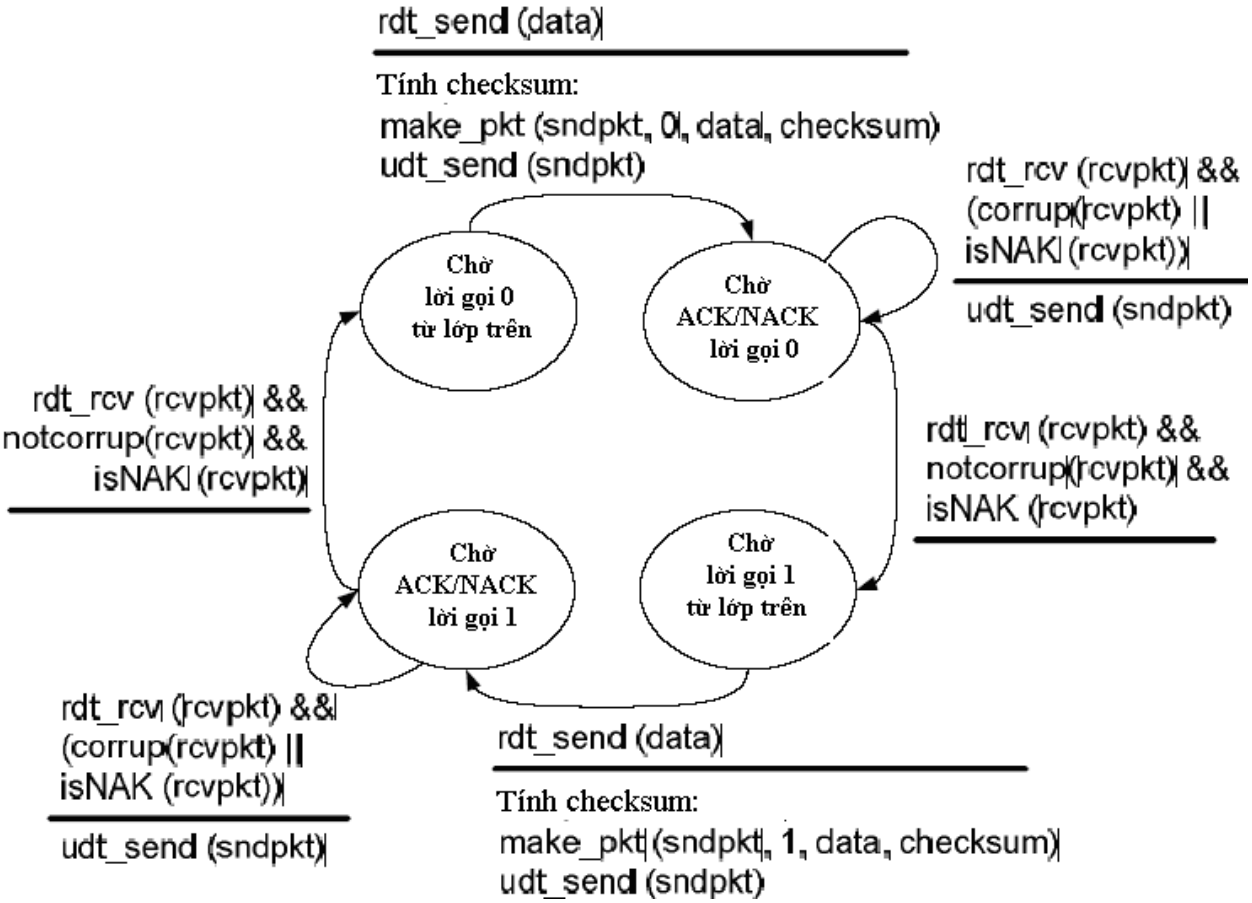
Hình 4.7 Giao thức cho kênh truyền có lỗi bit

Các giao thức ARQ cần phải có ba khả năng sau để xử lý trong trường hợp có lỗi bit:

- ***Phát hiện lỗi***: là cơ chế cho phép bên nhận phát hiện được khi nào trong gói dữ liệu có bit bị lỗi, thường sử dụng kỹ thuật CRC để thực hiện công việc này.
- ***Phản hồi từ phía nhận:*** Khi phía gửi và phía nhận nằm trên các thiết bị đầu cuối khác nhau - có thể cách nhau hàng nghìn km, cách duy nhất để phía gửi biết được kết quả gửi là phía nhận gửi thông tin phản hồi thông báo tình trạng nhận cho phía gửi. Báo nhận đúng ACK và báo nhận sai NAK trong ví dụ trên chính là các thông tin phản hồi. Giao thức rdt 2.0 yêu cầu phía nhận gửi phản hồi các bản tin ACK hay NAK cho phía gửi. Đoạn dữ liệu phản hồi chỉ cần sử dụng một bit, ví dụ giá trị 0 ứng với NAK và giá trị 1 ứng với ACK.
- ***Truyền lại:*** Đoạn dữ liệu bị lỗi sẽ được bên gửi phát lại.

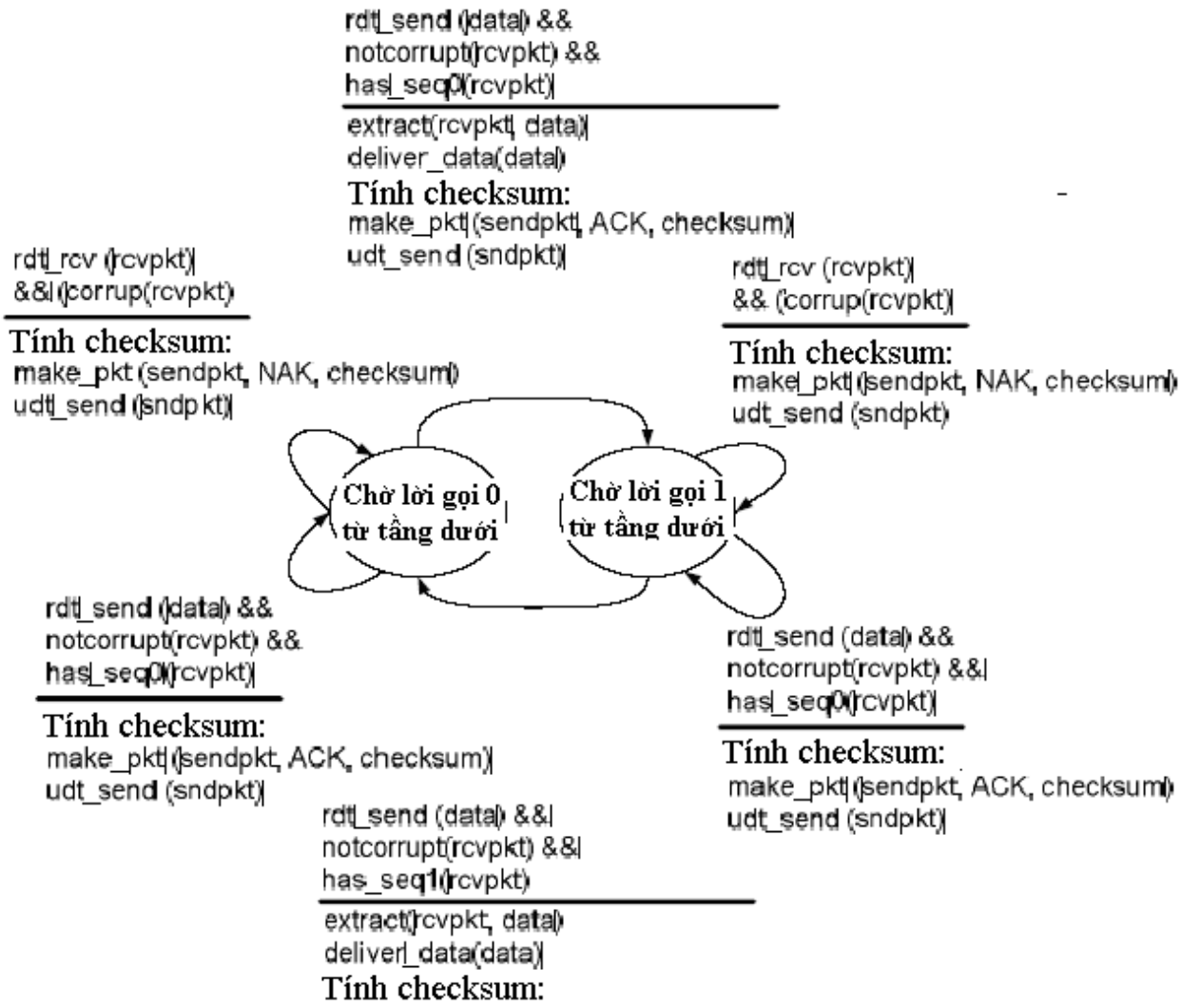
Trong giao thức rdt 2.0, phía gửi có hai trạng thái. Ở trạng thái thứ nhất, phía gửi đợi dữ liệu từ tầng trên. Trong trạng thái thứ hai, bên gửi đợi phản hồi ACK hoặc NAK từ bên nhận. Nếu nhận được ACK từ rdt_rcv(rcvpkt) && isACK(rcvpkt) tương ứng với sự kiện này, bên gửi biết được đoạn dữ liệu chuyển đến đích an toàn, vì vậy nó trở về trạng thái đợi dữ liệu từ tầng trên để chuyển tiếp. Nếu nhận được NAK, bên gửi gửi lại đoạn dữ liệu rồi quay lại trạng thái đợi phản hồi ACK hoặc NAK cho đoạn dữ liệu vừa gửi lại. Khi bên gửi ở trong trạng thái chờ phản hồi (ACK hoặc NAK), nó không thể nhận thêm dữ liệu từ tầng trên đưa xuống. Nó chỉ chấp nhận dữ liệu khi nhận được ACK và chuyển trạng thái. Bên gửi không gửi dữ liệu cho đến khi nó chắc chắn rằng bên nhận đã

nhận đúng đoạn dữ liệu đã gửi. Giao thức rdt 2.0 với hành vi như vậy thuộc kiểu dừng và chờ (stop and wait).



Hình 4.8 FSM của bên gửi trong rdt 2.1

FSM bên nhận trong giao thức rdt 2.0 chỉ có một trạng thái duy nhất. Khi nhận được đoạn dữ liệu, bên nhận gửi bản tin phản hồi ACK hoặc NAK, phụ thuộc vào đoạn dữ liệu đã nhận có lỗi hay không. Trong hình 4.7, rdt_rcv (rcvpkt) && corrups(rcvpkt) tương ứng với sự kiện đoạn dữ liệu nhận được bị lỗi.



Hình 4.9 FSM của phía nhận trong rdt 2.1

Giao thức rdt 2.0 vẫn còn nhược điểm: chúng ta chưa tính đến khả năng chính đoạn tin ACK hoặc NAK có lỗi. Chính đoạn tin phản hồi (ACK hoặc NAK) cũng cần phải có trường kiểm tra lỗi để bên gửi (lúc này lại là bên nhận) có khả năng phát hiện lỗi trong chính gói phản hồi. Vấn đề ở đây là khi nhận được một gói phản hồi bị lỗi, phía gửi không thể xác định nó là ACK nay NAK, do đó không xác định được đoạn dữ liệu nó gửi từ đích có bị lỗi hay không. Trong trường hợp này, có ba giải pháp xử lý ACK hoặc NAK bị lỗi:

- Giải pháp thứ nhất, dùng xác nhận khác tương tự như ACK, tuy nhiên có thể xảy ra lỗi liên hoàn, do đó giải pháp này khó có thể áp dụng.
- Giải pháp thứ hai là thêm vào trường kiểm tra một số bit để không những cho phép bên nhận phát hiện mà còn sửa được các bit lỗi. Đây hoàn toàn có thể là giải pháp trung gian cho những kênh truyền có lỗi nhưng không xử lý được trường hợp toàn bộ đoạn dữ liệu bị mất.
- Giải pháp thứ ba, phía gửi phát lại gói dữ liệu nếu phát hiện lỗi trong gói phản hồi (ACK hoặc NAK). Tuy nhiên, phương pháp này có thể dẫn đến sự trùng lặp dữ liệu. Bên nhận không biết được ACK/NAK mà nó gửi phản hồi

có bị lỗi trên đường truyền không. Vì thế nó không xác định được gói dữ liệu vừa nhận được là gói dữ liệu mới hay gói cũ.

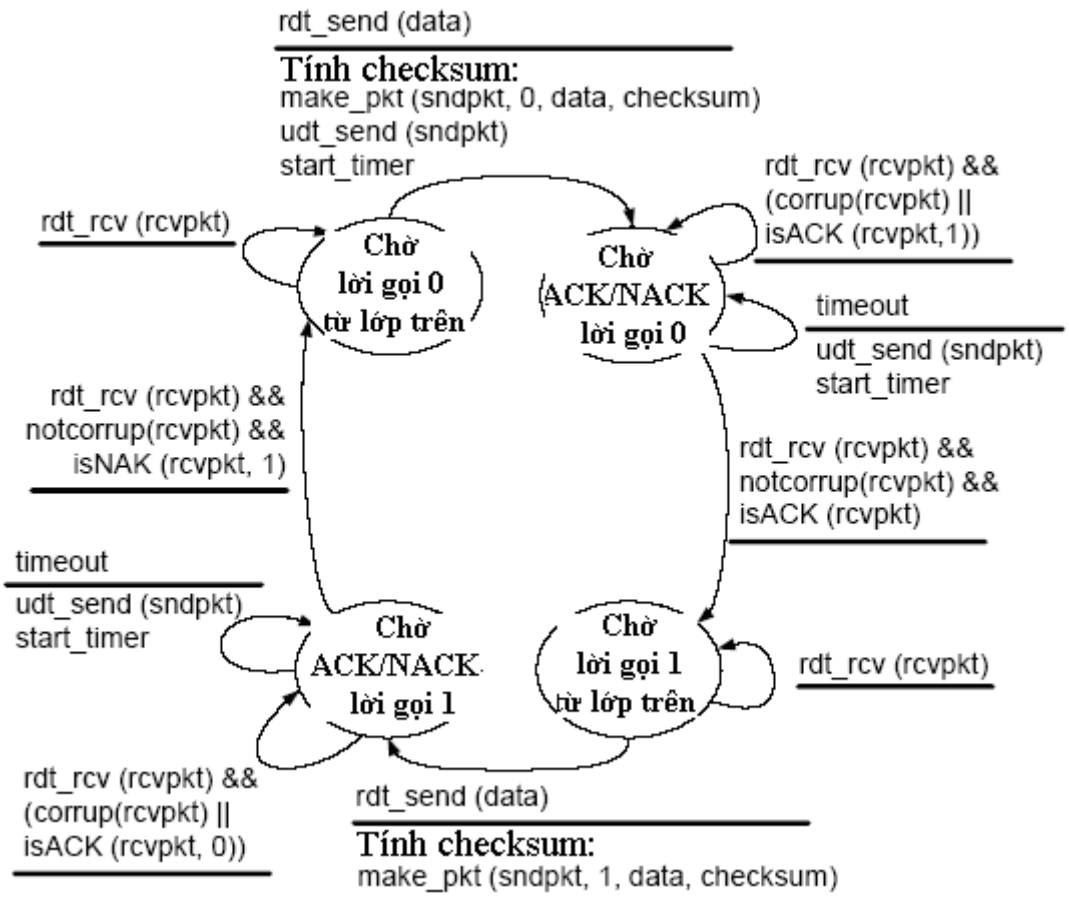
Giải pháp đơn giản nhất cho vấn đề này (sẽ được áp dụng cho nhiều giao thức, kể cả TCP) là thêm một trường số thứ tự cho đoạn dữ liệu, phía gửi đánh số cho các gói dữ liệu và đặt giá trị này vào trường số thứ tự (sequence number). Bên nhận chỉ cần kiểm tra số thứ tự để xác định đoạn dữ liệu nhận được là đoạn mới hay đoạn truyền lại. Với giao thức stop and wait đơn giản, chỉ cần một bit số thứ tự. Bên nhận có thể xác định bên gửi gửi lại đoạn dữ liệu đã gửi lần trước (số thứ tự của đoạn dữ liệu nhận được trùng với số thứ tự với đoạn dữ liệu nhận được lần trước) hay đoạn dữ liệu mới (có số thứ tự khác nhau, tăng lên theo module 2). Vì chúng ta vẫn giả định toàn bộ đoạn dữ liệu không bị mất trên kênh truyền, nên trong đoạn dữ liệu phản hồi (ACK/NAK) không cần chỉ ra số thứ tự của đoạn dữ liệu mà chúng biên nhận. Bên gửi biết rằng đoạn dữ liệu ACK/NAK (có thể bị lỗi hoặc không) là biên nhận cho đoạn dữ liệu gần nhất nó gửi.

Hình 4.8 và 4.9 là FSM của bên gửi và nhận trong giao thức rdt 2.1. Trong rdt 2.1, FSM của bên gửi và nhận đều có số trạng thái tăng gấp đôi. Đó là vì trạng thái giao thức phải biểu diễn gói dữ liệu được gửi (bởi bên gửi) và gói dữ liệu được đợi (tại bên nhận) có số thứ tự là 0 hay 1. Chú ý rằng các hành động trong trạng thái gói dữ liệu có số thứ tự 0 được gửi (phía gửi) hoặc được mong đợi (phía nhận) ngược với trạng thái gói dữ liệu có số thứ tự 1 được gửi hay được đợi.

Giao thức rdt 2.1 sử dụng cả biên nhận đúng (ACK) và biên nhận sai (NAK). NAK được gửi khi nhận được gói dữ liệu bị lỗi hay không đúng số thứ tự. Chúng ta có thể không cần sử dụng NAK: thay vì việc gửi NAK, chúng ta gửi ACK cho gói dữ liệu cuối cùng đã được nhận đúng. Nếu nhận hai ACK cho cùng một gói dữ liệu (hiện tượng trùng ACK - duplicate ACK) bên gửi xác định được bên nhận không nhận đúng gói dữ liệu sau gói dữ liệu đã biên nhận ACK hai lần. TCP sử dụng sự kiện ‘‘3 lần nhận được ACK trùng nhau’’ (‘‘tripie dupiicate ACKs’’) để kích hoạt việc gửi lại rdt 2.2 là giao thức truyền dữ liệu tin cậy trên kênh truyền có bít lỗi không sử dụng NAK.

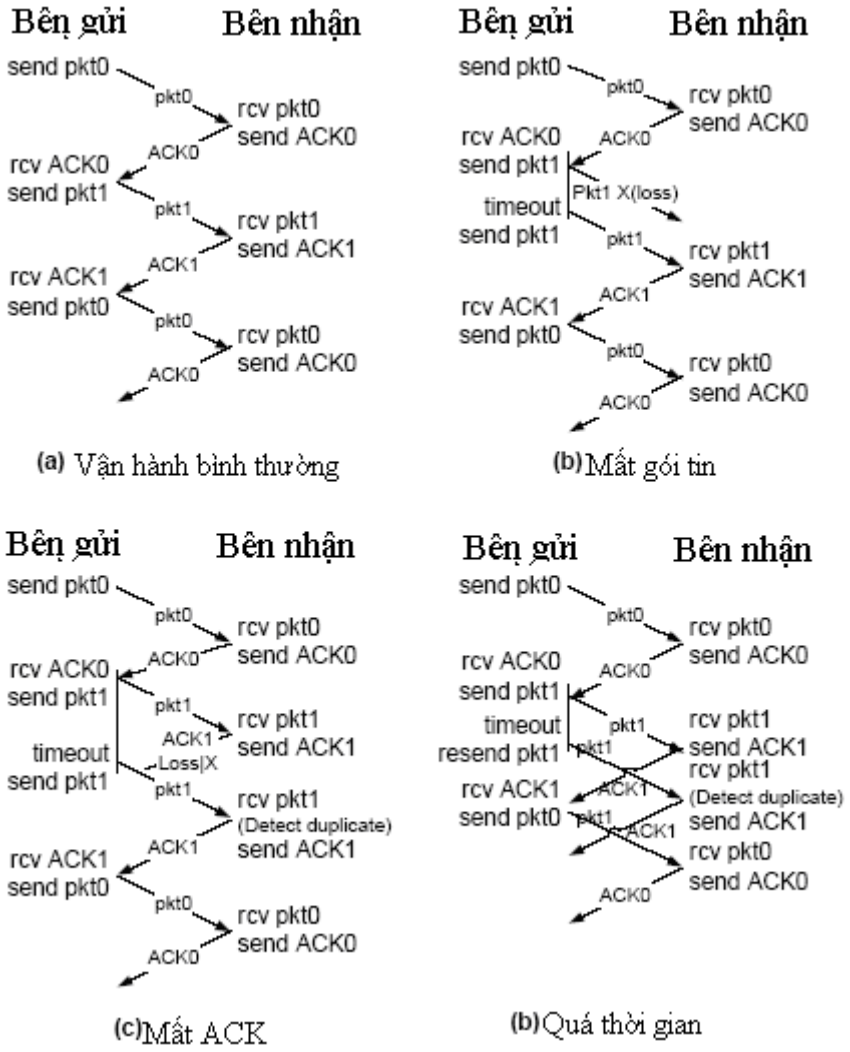
4.2.1.3 Truyền dữ liệu tin cây khi có lỗi

Dữ liệu trên kênh truyền không những bị lỗi mà còn có thể bị mất, đây là tình huống không phải không phổ biến trong mạng máy tính ngày nay, kể cả Internet. Lúc này giao thức cần phải giải quyết hai vấn đề: làm thế nào để phát hiện gói dữ liệu bị mất và làm gì khi mất gói dữ liệu. sử dụng cơ chế phát hiện lỗi nhờ checksum, số thứ tự, biên nhận ACK và truyền lại gói dữ liệu - đã được phát triển trong giao thức rdt 2.2 - cho phép chúng ta giải quyết được vấn đề thứ hai. Để giải quyết vấn đề thứ nhất, chúng ta cần đến một cơ chế mới.



Hình 4.10 FSM của bên gửi trong rdt 3.0

Có nhiều giải pháp xử lý việc mất mát dữ liệu. Ở đây chúng ta trình bày giải pháp lựa chọn bên gửi là nơi phát hiện và xử lý việc mất dữ liệu. Giả sử phía gửi gửi đi gói dữ liệu nhưng chính gói dữ liệu đó hoặc biên nhận ACK cho nó bị mất trên đường truyền. Trong cả hai trường hợp, bên gửi đều không nhận được biên nhận cho gói dữ liệu đã gửi. Giải pháp được đưa ra là sau khi gửi một khoảng thời gian nào đó mà không nhận được biên nhận ACK (có thể gói dữ liệu bị mất) thì bên gửi sẽ gửi lại. Nhưng phía gửi phải đợi trong bao lâu để chắc chắn rằng gói dữ liệu đã bị mất? Ít nhất phía gửi phải đợi trong khoảng thời gian để gói tin đi đến được phía nhận, phía nhận xử lý gói tin và thông tin biên nhận quay lại. Trong nhiều mạng, rất khó dự đoán và ước lược thời gian này. Lý tưởng là phải xử lý việc mất gói tin ngay khi có thể, đợi một khoảng thời gian dài đồng nghĩa với việc chậm trễ khi xử lý gói tin bị mất. Trên thực tế, phía gửi sẽ chọn một khoảng thời gian đợi nào đó, mặc dù không đảm bảo chắc chắn là gói tin bị mất. Nếu không nhận được ACK trong khoảng thời gian này, bên gửi sẽ gửi lại gói dữ liệu Chú ý rằng, nếu gói dữ liệu đến trễ, phía gửi sẽ gửi lại gói dữ liệu - ngay cả khi gói dữ liệu đó và cả ACK đều không bị mất. Điều này gây ra trùng lặp dữ liệu tại phía nhận. Tuy nhiên, giao thức rdt 2.2 đã có đủ khả năng (nhờ số thứ tự) để ngăn chặn sự trùng lặp dữ liệu.



Hình 4.11 Hoạt động của giao thức rdt 3.0

Đối với phía gửi, truyền lại là giải pháp "vạn năng". Phía gửi không biết được gói dữ liệu bị mất, gói biên nhận ACK bị mất hay chỉ đơn giản là chúng bị trễ. Trong tất cả các trường hợp, hành động của nó là giống nhau: truyền lại. Để thực hiện cơ chế truyền lại theo thời gian, một bộ định thời đếm ngược (countdown timer) được sử dụng để nhắc phía gửi thời gian đợi đã hết. Do vậy, phía gửi phải có khả năng (1) khởi tạo timer mỗi khi gửi gói dữ liệu (gói dữ liệu gửi lần đầu hay gói dữ liệu được truyền lại), (2) phản ứng với ngắt của timer (đưa ra những hành động thích hợp) và (3) dừng timer.

Sự trùng lặp các gói dữ liệu do phía gửi tạo ra, sự mất mát các gói dữ liệu (cả gói dữ liệu lẫn gói biên nhận) gây khó khăn cho phía gửi khi xử lý các gói biên nhận ACK. Nếu nhận được ACK, làm thế nào để phía gửi biết được ACK đó là biên nhận cho gói dữ liệu gửi đi gần đây nhất, hay là ACK biên nhận cho gói dữ liệu nào đó đã gửi từ trước nhưng đến trễ? Giải pháp là ta thêm vào gói ACK trường số thứ tự biên nhận (acknowledge number). Giá trị của trường này - do phía nhận tạo ra - là số thứ tự của chính gói dữ liệu cần được biên nhận. Bằng cách kiểm tra giá trị trường biên nhận, phía gửi có thể xác định được số thứ tự của gói dữ liệu được biên nhận. Thời gian dịch chuyển theo chiều từ trên xuống. Thời điểm nhận gói dữ liệu chậm hơn thời điểm gửi gói dữ liệu vì tính

đến thời gian gói dữ liệu lan toả trên đường truyền. Trong hình 4.11b-d, ngoặc vuông xác định thời điểm timer được thiết lập và thời điểm "timeout". Vì số thứ tự của gói dữ liệu thay đổi lần lượt giữa 0 và 1 nên đôi khi giao thức rdt 3.0 được gọi là giao thức một bit luân chuyển (alternate bit protocol).

4.3 Điều khiển lưu lượng

Điều khiển lưu lượng là hành động thay đổi tốc độ chuyển dữ liệu giữa bên gửi và bên nhận để tránh hiện tượng bên nhận không kịp xử lý. Nhiệm vụ của nó là đảm bảo rằng bên gửi không thể tiếp tục truyền dữ liệu nhanh hơn mức mà bên nhận có thể tiếp thu được. Điều khiển lưu lượng được thực hiện bằng cách bên nhận thông báo cho bên gửi biết về khả năng xử lý dữ liệu của nó.

Điều khiển tắc nghẽn thực hiện nhiệm vụ đảm bảo cho mạng có khả năng vận chuyển lưu lượng đưa vào, đây là vấn đề toàn cục trên mạng, liên quan đến hành vi của mọi nút mạng, quá trình chứa và chuyển tiếp trong mỗi nút mạng và các yếu tố khác có khuynh hướng làm giảm thông lượng của mạng. Điều khiển lưu lượng và điều khiển tắc nghẽn là hai khái niệm khác nhau nhưng liên quan chặt chẽ với nhau. Điều khiển lưu lượng là để tránh tắc nghẽn, còn điều khiển tắc nghẽn là để giải quyết vấn đề tắc nghẽn hoặc có dấu hiệu tắc nghẽn sắp xảy ra.

Việc triển khai giải pháp điều khiển lưu lượng và điều khiển tắc nghẽn có thể thực hiện trên các thuật toán riêng biệt, đôi khi cả hai thuật toán này cùng được cài đặt trong một giao thức, thể hiện ra như là một thuật toán duy nhất, thí dụ trong giao thức TCP. Điều khiển lưu lượng ở một vài tầng trong mạng: Tầng vận tải, tầng mạng và tầng liên kết dữ liệu. Điều khiển lưu lượng ở tầng vận tải (còn gọi là điều khiển lưu lượng đầu cuối - đầu cuối) nhằm tránh cho bộ đệm của bên nhận không khỏi bị tràn. Điều khiển lưu lượng trên từng chặng sẽ tránh cho từng đường truyền khỏi bị tắc nghẽn. Tuy nhiên, việc điều khiển lưu lượng trên từng chặng sẽ có ảnh hưởng đến các chặng khác, do đó nó cũng có tác dụng tránh tắc nghẽn cho các đường truyền có nhiều chặng. Trong mô hình tham chiếu OSI, điều khiển lưu lượng theo từng chặng được thực hiện ở tầng mạng và tầng liên kết dữ liệu.

Giải pháp điều khiển lưu lượng được sử dụng rộng rãi nhất là dùng cơ chế cửa sổ trượt, có thể áp dụng tại một hay nhiều tầng của mạng (thường ở tầng liên kết dữ liệu, tầng mạng nhưng phổ biến nhất là tầng vận tải). Cơ chế điều khiển lưu lượng bằng cửa sổ trượt cho phép bên gửi phát đi liên tiếp một số đơn vị dữ liệu nhất định rồi dừng lại và chờ thông báo về kết quả nhận trước khi tiếp tục phát các đơn vị dữ liệu tiếp theo. Bên nhận điều khiển lưu lượng bằng cách kìm lại hay gửi ngay biên nhận, đó là một đơn vị dữ liệu có chứa thông tin điều khiển để báo cho bên gửi biết về việc đã nhận một hay một số đơn vị dữ liệu. Tại mọi thời điểm, bên gửi phải ghi nhớ danh sách chứa số tuần tự các đơn vị dữ liệu đã gửi, các đơn vị dữ liệu này được coi là đang nằm trong cửa sổ gửi. Tương tự, bên nhận cũng duy trì một danh sách gọi là cửa sổ nhận, tương ứng với các đơn vị dữ liệu liệu mà nó đã nhận. Hai cửa sổ gửi và nhận không nhất thiết phải có độ lớn bằng nhau. Người ta đã đề xuất và sử dụng một số phương

thức quản lý cửa sổ khác nhau: biên nhận riêng rẽ cho mỗi đơn vị dữ liệu nhận được, biên nhận ở cuối cửa sổ, biên nhận ở đầu cửa sổ v.v.

4.4 Nâng cao hiệu năng bằng đường ống Pipeline

Mặc dù hoạt động đúng nhưng hiệu suất hoạt động của rdt 3.0 chưa cao, điểm yếu của vấn đề là do thao tác dừng và chờ. Sau khi phát một đoạn dữ liệu, bên phát dừng gửi dữ liệu để chờ nhận thông báo trả lời của bên nhận. Nếu kết quả đúng, bên phát sẽ gửi đoạn tin kế tiếp. Nếu kết quả sai, bên phát sẽ gửi lại đoạn dữ liệu đó. Để ước lược hiệu suất của giao thức stop and wait, hãy xét trường hợp lý tưởng với hai thiết bị đầu cuối, một ở bờ biển phía đông, một ở bờ biển phía tây nước Mỹ. Thời gian trễ giữa hai thiết bị (dù tín hiệu lan truyền với tốc độ ánh sáng, trong thực tế độ trễ trên các thiết bị mạng lớn hơn rất nhiều) là Pprop xấp xỉ 15 ms. Giả sử hai thiết bị được kết nối bằng đường truyền tốc độ C =1 Gbit/s. Kích thước của đoạn dữ liệu SP=1Kbyte/packet, thời gian cần thiết để truyền toàn bộ gói dữ liệu trên kênh truyền tốc độ 1 Gbps được tính bởi công thức:

$$T_{trans} = \frac{SP}{C} = \frac{8Kbit / packet}{1Mbit / sec} = 8 \text{ microseconds}$$

Với giao thức stop and wait, nếu phía gửi bắt đầu gửi gói dữ liệu tại thời điểm t = 0 thì tại thời điểm t = 8 microsecond, bit cuối cùng mới được bên gửi đẩy ra đường truyền. Tiếp theo phải mất 15 ms để cả gói dữ liệu đi từ phía gửi sang phía nhận như vậy bit cuối cùng của gói dữ liệu đến đích tại thời điểm t = 15.008ms. Để đơn giản, ta giả thiết gói ACK có cùng độ dài với gói dữ liệu và phía nhận gửi ngay gói ACK khi nhận được bit cuối cùng của gói dữ liệu. Khi vậy bit cuối cùng của gói ACK được truyền tới đích tại thời điểm t = 30.016 ms. Trong khoảng thời gian 30.016ms, phía gửi chỉ hoạt động (gửi hoặc nhận) trong 0.016 ms. Nếu định nghĩa hiệu suất của bên gửi (hay kênh truyền) là tỷ lệ thời gian phía gửi hoạt động (gửi dữ liệu trên kênh truyền), chúng ta có hiệu suất U_{sender} rất thấp:

$$U_{sender} = \frac{0.008}{30.016} = 0.00015$$

Điều đó có nghĩa là phía gửi chỉ hoạt động trong khoảng 0.15 phần nghìn thời gian. Theo cách tính khác, phía gửi gửi 1 Kbyte trong 30,016 milisecond tương đương với tốc độ truyền là 33 Kbyte/s thấp hơn nhiều so với tốc độ có thể là 1 Gigabit/s. Người quản trị mạng "bất hạnh" này phải trả một số tiền khổng lồ để thuê đường truyền 1 Gigabit/s nhưng cuối cùng chỉ nhận được một đường truyền có tốc độ 33 Kbyte/s. Đây là một ví dụ sống động minh họa việc phần

mềm có thể giới hạn các khả năng của phần cứng phía dưới. Trong trường hợp này Chúng ta đã bỏ qua thời gian xử lý của các giao thức tầng dưới ở cả phía gửi và phía nhận cũng như thời gian xử lý và thời gian trễ của gói tin tại các router trung gian. Nếu tính cả những yếu tố này, hiệu suất hoạt động thực sự sẽ còn thấp hơn nữa.

Giải pháp cho vấn đề hiệu suất sẽ là cho phép phía gửi gửi đồng thời nhiều gói dữ liệu mà không cần phải đợi ACK. Có thể hình dung các gói dữ liệu nối tiếp nhau trên đường truyền từ phía gửi đến phía nhận giống như Nước chảy trong một đường ống. Vì thế kỹ thuật gửi liên tiếp này được gọi là kỹ thuật đường ống **(pipeline)**. Kỹ thuật này làm tăng hiệu suất của giao thức lên nhiều lần, tuy nhiên nó đòi hỏi những yêu cầu sau:

- Khoảng số thứ tự phải tăng, bởi vì mỗi gói dữ liệu được truyền đi (không tính các gói dữ liệu truyền lại) phải có một số thứ tự duy nhất. Trên đường truyền có thể có đồng thời nhiều gói dữ liệu được gửi chưa được biên nhận.
- Phía gửi và phía nhận có thể phải có bộ đệm (buffer) cho nhiều gói dữ liệu. Ít nhất phía gửi có vùng đệm cho các gói dữ liệu đã được truyền đi nhưng chưa được biên nhận. Phía nhận cũng có thể cần vùng đệm cho cả các gói dữ liệu đã nhận đúng, như sẽ thảo luận dưới đây.

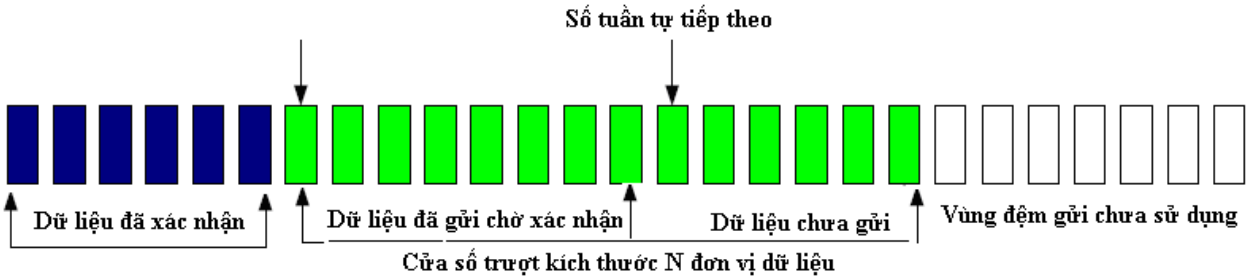
Yêu cầu về khoảng số thứ tự cần thiết cũng như về vùng đệm phụ thuộc vào cách giao thức xử lý việc mất dữ liệu, dữ liệu bị lỗi, bị trễ. Có hai cách tiếp cận chính được trình bày ở đây: Quay lại N (Go-Back-N) và lặp lại có Lựa chọn (Sesective Repeat).

4.4.1 Giao thức Go-back-N

Trong giao thức Go-Back-N, phía gửi cho phép truyền đi đồng thời nhiều gói dữ liệu mà không phải đợi biên nhận. Tuy nhiên tổng số gói dữ liệu không phải là vô hạn mà bị giới hạn bởi giá trị N - tổng số gói dữ liệu tối đa chưa được biên nhận trong đường ống. Hình 3.18 là khoảng số thứ tự trong giao thức Go-Back-N. Định nghĩa base là số thứ tự của gói dữ liệu đã được truyền đi lâu nhất chưa được biên nhận và next seqnum là số thứ tự nhỏ nhất chưa được sử dụng (là số thứ tự của đoạn tiếp theo sẽ gửi). Có bốn khoảng số thứ tự như sau: Khoảng [0,base-1] ứng với số thứ tự của các gói dữ liệu đã được truyền đi và đã được biên nhận. Khoảng [base, nextseqnum-1] ứng với các gói dữ liệu đã được gửi đi nhưng chưa được biên nhận. Khoảng [nextseqnum, base +N- 1] có thể được sử dụng làm số thứ tự cho các gói sẽ được gửi nếu như có dữ liệu từ tầng trên chuyển xuống. Khoảng từ [base+n] trở lên chưa được sử dụng cho đến khi các gói tin đợi biên nhận được biên nhận.

Trong hình 4.12, khoảng cho phép số thứ tự của những gói dữ liệu đã được gửi nhưng chưa được biên nhận có thể xem là một "cửa sổ" kích thước N nằm trong phạm vi số thứ tự. Khi giao thức vận hành, cửa sổ này có thể "trượt" trên toàn bộ khoảng số thứ tự. Vì vậy, N thường được xem là độ lớn cửa sổ (window size) và giao thức GBN là giao thức cửa sổ trượt (sliding- window). tại sao ngay từ đầu chúng ta phải giới hạn số lượng tối đa các gói dữ liệu được gửi mà chưa cần biên nhận bởi giá trị N. tại sao không để giá trị N này là vô hạn.

Chúng ta sẽ thấy trong phần 3.5, kiểm soát lưu lượng là một trong những lý do bắt buộc ta phải đặt giới hạn phía gửi.



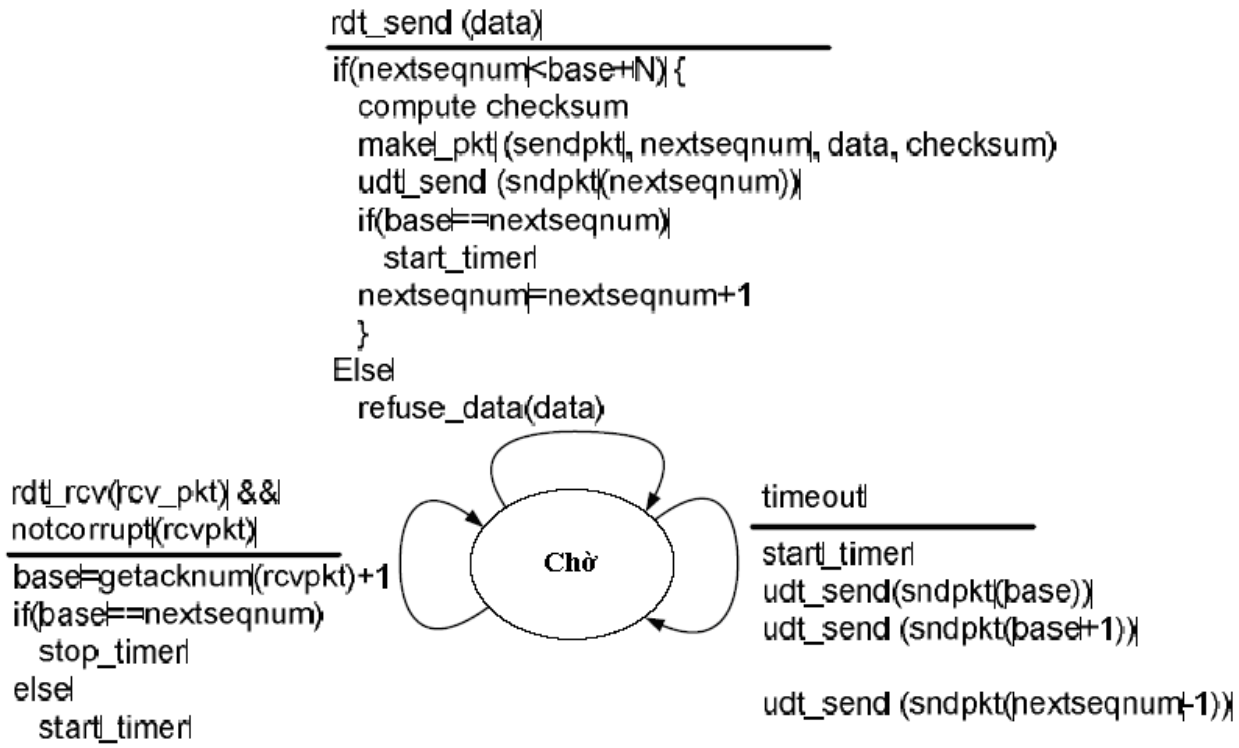
Hình 4.12 Khoảng số thứ tự của bên gửi trong giao thức Go-Back-N

Trong thực tế, số thứ tự của đoạn dữ liệu được đặt trong một trường có độ dài cố định trong phần thông tin điều khiển của đoạn dữ liệu. Nếu k là độ lớn trường số thứ tự (tính theo bit) của đoạn dữ liệu thì khoảng số thứ tự sẽ là $[0,2^k - 1]$. Vì khoảng số thứ tự bị giới hạn, nên tất cả các thao tác trên số thứ tự sẽ được thực hiện theo module 2^k (khoảng số thứ tự có thể xem là một vòng tròn với 2^k giá trị, sau giá trị 2^k-1 là giá trị 0). Giao thức rdt 3.0 chỉ sử dụng 1 bit làm số thứ tự nên khoảng số thứ tự là [0,1].

Gọi là FSM mở rộng (extended FSM) vì được thêm vào các biến (base và nextseqnum - giống như biến trong ngôn ngữ lập trình), các thao tác và hành động có điều kiện liên quan đến các biến này. Trong giao thức GBN, phía gửi phải đáp ứng ba sự kiện sau:

- Có dữ liêu từ trên chuyển xuống : khi rdt_send() được phía trên sử dụng để chuyển dữ liệu xuống, phía gửi phải kiểm tra xem cửa sổ đã đầy chưa (tức là đã có N gói dữ liệu gửi đi chưa được biên nhận không). Nếu cửa sổ chưa đầy, phía gửi tạo ra và sau đó gửi gói dữ liệu đồng thời cập nhật các biến. Nếu cửa sổ đầy, phía gửi không chấp nhận dữ liệu từ tầng trên và thông báo cửa sổ đã đầy. Khi đó, tầng trên sẽ phải gửi lại. Trên thực tế, phía gửi sẽ đưa dữ liệu vào vùng đệm (nhưng chưa gửi ngay) hoặc có cơ chế đồng bộ (sử dụng semaphore hay cờ) chỉ cho phép tầng ứng dụng sử dụng rdt_send() khi cửa sổ chưa đầy.
- Nhân được một ACK: trong giao thức GBN, giá trị biên nhận gói tin có số thứ tự n sẽ mang giá trị tích luỹ, nghĩa là toàn bộ gói dữ liệu có số thứ tự nhỏ hơn hoặc bằng n đều đã được phía nhận nhận đúng. Chúng ta sẽ quay lại vấn đề này khi xem xét phía nhận trong giao thức GBN.
- Hết thời gian đợi (timeout): tên giao thức - "Go-Back-N" bắt nguồn từ hành vi của phía gửi khi dữ liệu bị mất hay bị trễ. Giống như trong giao thức stop and wait, đồng hồ định thời được sử dụng để xử lý việc mất gói dữ liệu hay gói phản hồi. Khi hết thời gian đợi (timeout), phía gửi sẽ gửi lại tất cả các gói dữ liệu đã được gửi đi trước đó nhưng chưa được biên nhận. Trong hình 4.13, phía gửi chỉ sử dụng duy nhất một đồng hồ định thời, có thể xem

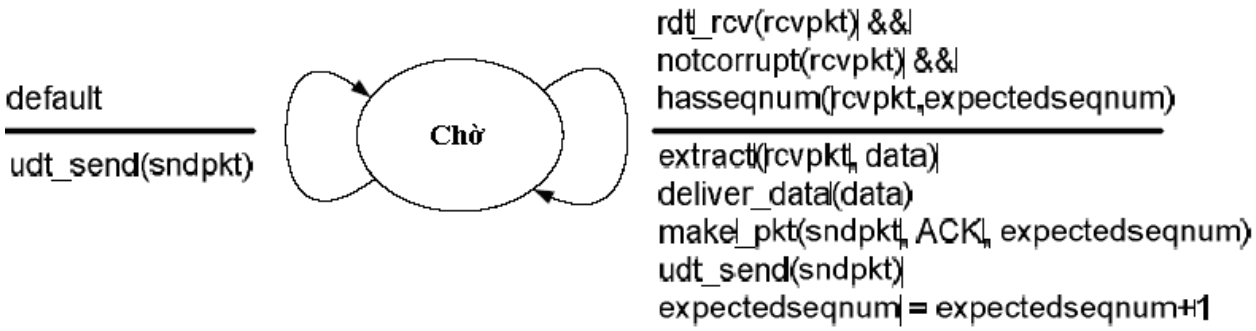
là thời gian của đoạn dữ liệu đã được truyền đi lâu nhất nhưng chưa được biên nhận. Nếu ACK nào đó được nhận nhưng vẫn còn gói dữ liệu gửi đi chưa được biên nhận thì đồng hồ định thời sẽ được khởi động lại. Nếu tất cả các đoạn dữ liệu đã gửi đều được biên nhận thì có thể ngừng đồng hồ định thời.



Hình 4.13 FSM mở rộng của bên gửi trong GBN

Các hành động của phía nhận trong giao thức GBN đơn giản. Nếu nhận được đúng đoạn dữ liệu và đoạn dữ liệu này đúng thứ tự thì phía nhận gửi ACK cho đoạn dữ liệu nhận được và chuyển dữ liệu trong đoạn dữ liệu này bên trên. Trong tất cả các trường hợp còn lại, phía nhận loại bỏ đoạn dữ liệu và gửi lại ACK cho đoạn dữ liệu đúng thứ tự cuối cùng nó nhận được. Chú ý rằng đoạn dữ liệu được chuyển lên tầng trên một lần duy nhất nên nếu đoạn dữ liệu thứ k được nhận và chuyển lên trên thì nghĩa là tất cả các đoạn dữ liệu có số thứ tự nhỏ hơn k cũng đã được chuyển lên. Sử dụng ACK tích luỹ là sự lựa chọn tốt nhất cho giao thức GBN. Trong giao thức GBN, bên nhận loại bỏ đoạn tin không theo thứ tự.

Dường như lãng phí khi loại bỏ đoạn tin đã nhận đúng nhưng không đúng thứ tự, nhưng có vài nguyên nhân cho hoạt động trên. Bên nhận phải chuyển dữ liệu lên tầng trên theo đúng thứ tự. Giả sử đoạn tin N đang được đợi nhận nhưng đoạn tin thứ (N+1) lại đến trước. Trong trường hợp ấy, để dữ liệu chuyển lên hợp lệ, bên nhận có thể lưu tạm đoạn tin (N+1) và chỉ chuyển đoạn tin này bên tầng trên sau khi đã nhận đúng đoạn tin thứ N. Tuy nhiên theo quy tắc truyền lại của bên gửi, nếu đoạn tin thứ N bị mất thì đoạn tin này và cả đoạn tin N+l sẽ được truyền lại.

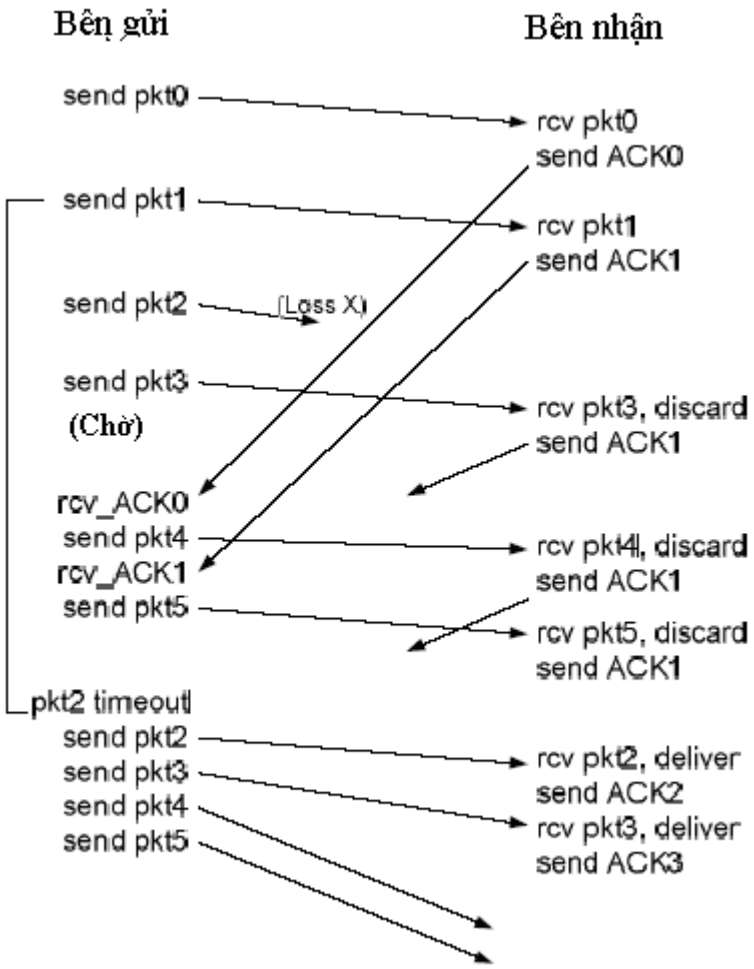


Hình 4.14 FSM mở rộng của bên nhận trong GBN

Như vậy, bên nhận có thể loại bỏ đoạn tin N+1. Ưu điểm của giải pháp này là bên nhận triển khai vùng đệm (buffer) đơn giản bởi không cần lưu lại các đoạn tin không đúng thứ tự. Nếu bên gửi phải ghi nhớ các cận của cửa sổ (base, base+n) và vị trí nextseqnum trong cửa sổ, thì bên nhận chỉ phải nhớ số thứ tự của đoạn tin hợp lệ tiếp theo. Giá trị này được giữ trong biến expectedseqnum (số thứ tự được mong đợi). Tất nhiên, nhược điểm của việc loại bỏ đoạn tin đã nhận đúng (nhưng không theo thứ tự) là khi truyền lại đoạn tin có thể bị mất hay lỗi, do đó phải truyền đi truyền lại nhiều lần.

Với độ lớn giới hạn, bên gửi sẽ chỉ được gửi các gói tin tớ 0 đến 3 nhưng sau đó phải đợi biên nhận cho các đoạn tin này trước khi tiếp tục gửi tiếp. Khi nhận được các ACK liên tiếp nhau (ví dụ ACK0 và ACK1), cửa sổ sẽ trượt về phía trước, bên gửi có thể truyền đoạn tin mới (lần lượt là pkt4 và pkt5). Ở phía bên nhận, đoạn tin số 2 bị mất, do đó đoạn tin 3,4,5 gửi đến không theo đúng thứ tự và bị loại bỏ. Với GBN, có một chú ý quan trọng là triển khai GBN tương tự FSM mở rộng. Hình thức triển khai bao gồm nhiều thủ tục khác nhau, mỗi thủ tục thực hiện một nhóm các hành động nào đó đáp lại các sự kiện khác nhau có thể xảy ra. với lập trình hướng sự kiện, các thủ tục sẽ được các thủ tục khác gọi hay là kết quả của việc gọi ngắt. Ở phía bên gửi, sự kiện có thể là:

- Thực thể tầng trên truyền dữ liệu xuống qua thủ tục rdt_send()
- Ngắt khi thời gian đợi hết
- Tầng dưới chuyển dữ liệu bên qua hàm rdt_rcv().



Hình 4.15 Giao thức Go-Back-N trong quá trình hoạt động

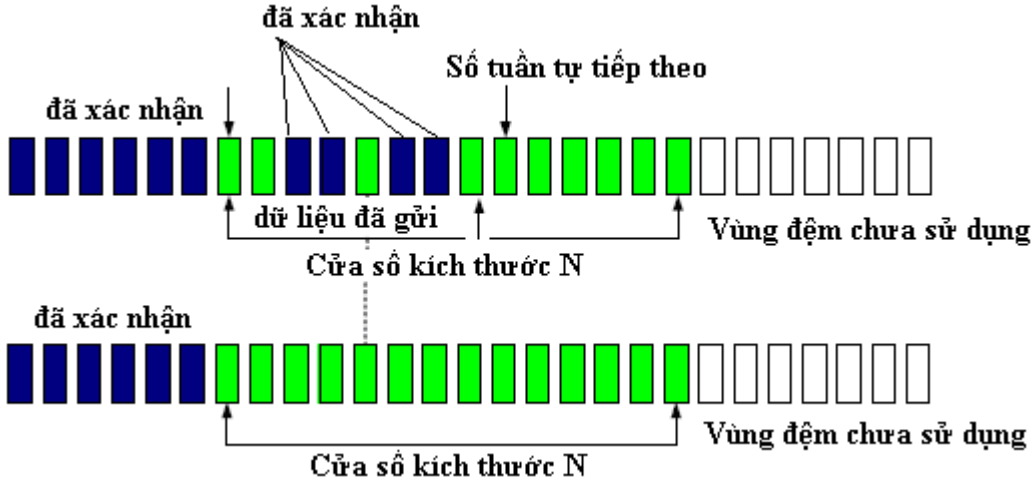
Trong thực tế, TCP là giao thức "tựa" GBN. Tuy nhiên có sự khác biệt giữa GBN và TCP. Nhiều phiên bản TCP lưu lại các segment không theo thứ tự nhận đúng. Trong phương án nâng cấp TCP, sử dụng biên nhận có lựa chọn [RFC 258] cho phép bên nhận có thể biên nhận tuỳ ý một đoạn tin không theo thứ tự (chứ không sử dựng giá trị biên nhận tích luỹ). Biên nhận có lựa chọn chính là lớp giao thức gửi liên tiếp thứ hai: Lặp lại có lựa chọn (selective repeat - SR). Có thể xem TCP là sự kết hợp của cả hai giao thức GBN và SR.

4.4.2 Giao thức lặp lại có lựa chọn

Giao thức GBN khắc phục được hiệu suất thấp của giao thức stop and wait. Tuy nhiên trong một vài tình huống, chính hiệu suất của giao thức GBN cũng rất thấp. Ví dụ khi kích thước cửa sổ và thời gian truyền một đoạn tin lớn, có thể có nhiều đoạn tin ở trên đường truyền, một đoạn tin bị lỗi có thể khiến GBN phải truyền lại nhiều đoạn tin.

Giao thức lặp lại có lựa chọn (SR - Selective Repeat) tránh việc truyền lại không cần thiết bằng cách bên gửi chỉ gửi lại các đoạn tin mà nó cho là có lỗi (hoặc mất). Để truyền lại từng đoạn tin khi cần thiết, bên nhận cần biên nhận cho từng đoạn tin nhận đúng. Vẫn sử dụng lại kích thước cửa sổ là N để giới hạn tổng số đoạn tin chưa được biên nhận trên đường truyền. Tuy nhiên khác

với GBN, bên gửi sẽ nhận được biên nhận ACK cho một số đoạn tin trong cửa sổ.

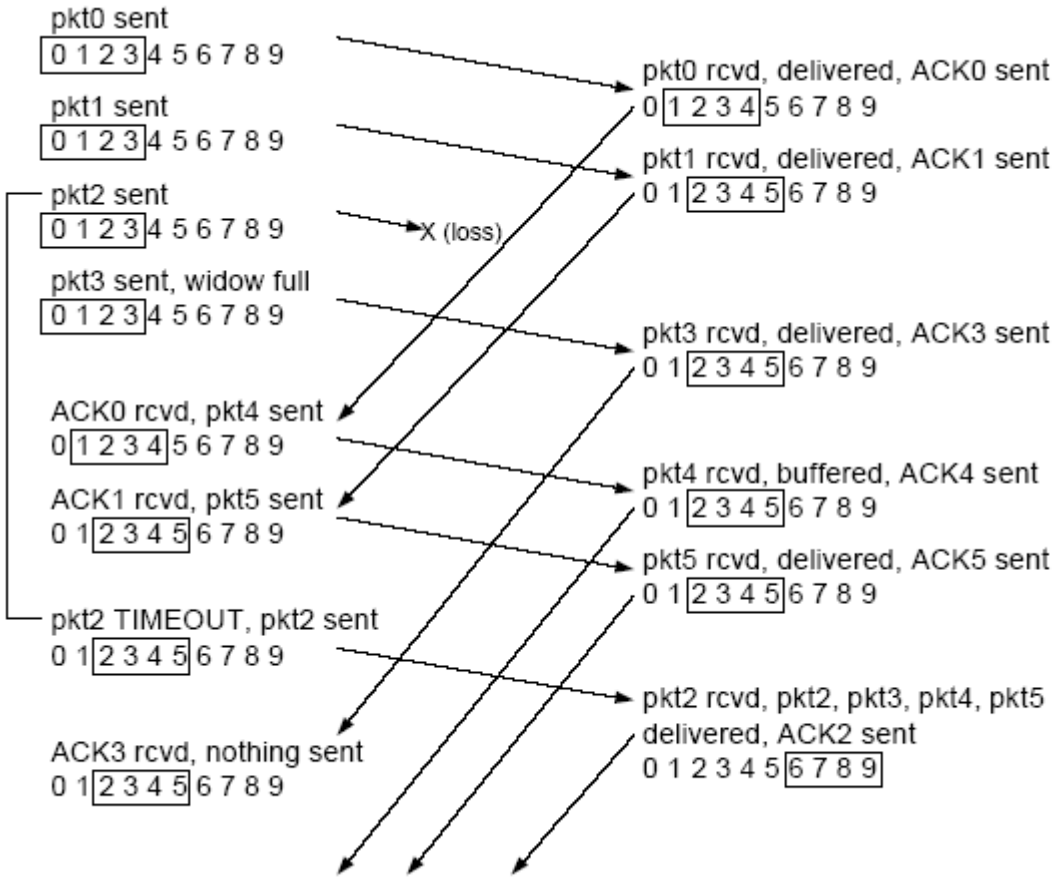


Hình 4.16 Khoảng số thứ tự của bên gửi và bên nhận

Bên nhận Selective Repeat sẽ biên nhận cho bất kỳ đoạn tin nhận đúng, cho dù không theo đúng thứ tự. Đoạn tin không đúng thứ tự vẫn được lưu giữ lại cho đến khi tất cả các đoạn tin còn thiếu (đoạn tin có số thứ tự nhỏ hơn) được chuyển đến, khi đó tất cả các đoạn tin sẽ được chuyển lên tầng trên theo đúng thứ tự.

Sự kiện và phản ứng của bên gửi

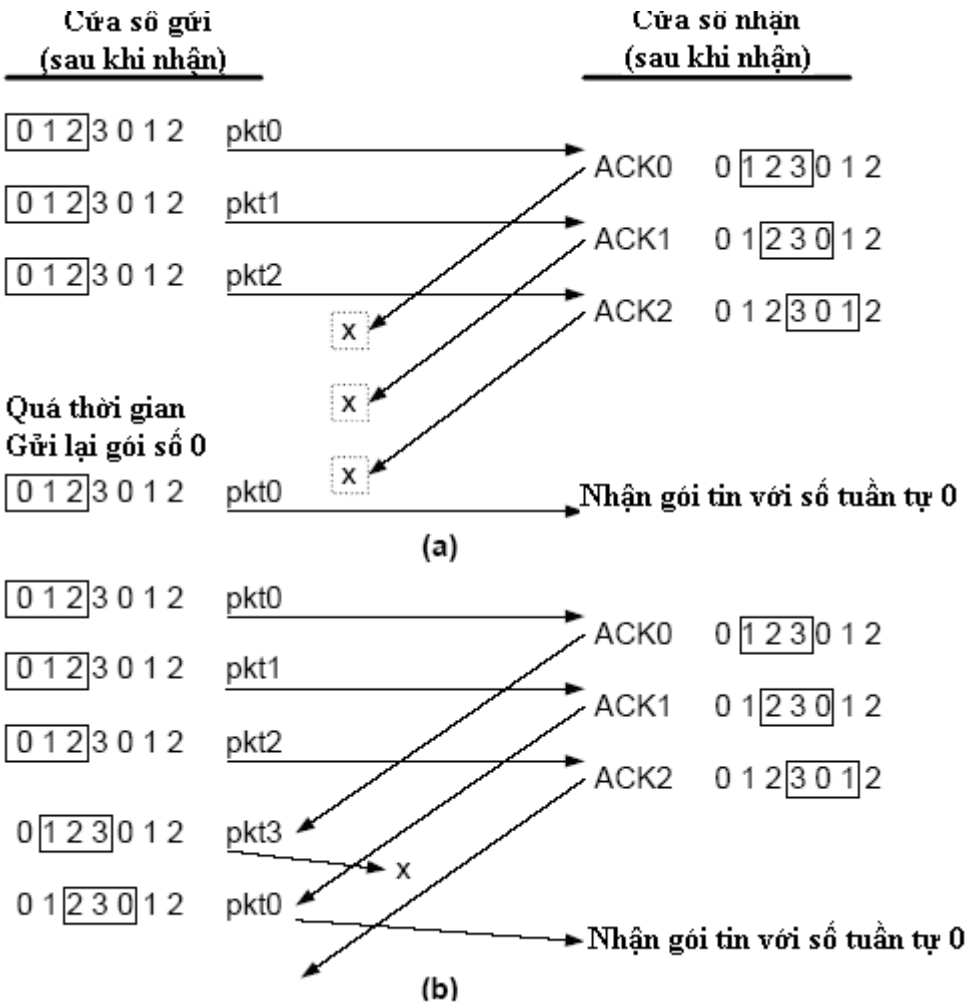
- Dữ liệu nhận được từ phía trên: Khi nhận được dữ liệu từ phía trên, bên gửi SR kiểm tra số thứ tự sẽ gửi. Nếu số thứ tự sẽ gửi nằm trong cửa sổ gửi, dữ liệu được đóng gói và gửi đi, ngược lại thì dữ liệu được lưu giữ trong bộ đệm hoặc gửi trả lên tầng trên để gửi sau, giống GBN.
- Hết thời gian đợi – Timeout: Timer lại được sử dụng để phát hiện mất gói tin. Tuy nhiên, mỗi gói tin gửi đi có một timer riêng, bởi vì chỉ có duy nhất một gói tin được gửi lại khi hết thời gian đợi. Có thể sử dụng đồng hồ hệ thống giữ vai trò đồng bộ cho các timer.
- Nhận được ACK: Nếu nhận được ACK, bên gửi đánh dấu gói tin đã được chuyển đúng. Nếu số thứ tự của gói tin vừa được biên nhận bằng send_base, cánh cửa sổ sẽ trượt tới gói tin có số thứ tự nhỏ nhất chưa được biên nhận. Nếu cửa sổ di chuyển và còn các gói tin chưa được truyền thì các gói tin đó sẽ được gửi.



Hình 4.17 SR trong quá trình hoạt động

Sự kiện và phản ứng của bên nhận

- Nhận đúng gói tin với số thứ tự trong khoảng [rcv_base,rcv_base+N-1. Trong trường hợp này, gói tin nhận được nằm trong cửa sổ nhận. Bên nhận gửi biên nhận cho gói tin này. Nếu gói tin đó chưa được nhận từ trước, nó sẽ được ghi lại trong bộ đệm. Nếu gói tin đó có số thứ tự bằng với cận dưới của cửa sổ nhận (rcv_base trong hình 4.16) thì nó cùng các gói tin có số thứ tự liên tiếp đã lưu giữ từ trước (bắt đầu tớ rcv_base) được chuyển lên tầng trên. Cửa sổ nhận sẽ trượt về phía trước một khoảng bằng với khoảng số gói tin đã chuyển lên tầng trên. với ví dụ trên hình 4.17 khi nhận được gói tin có số thứ tự rcv_base 2 thì gói tin này cùng với gói tin rcv_base+1 và gói tin rcv_base+2 được chuyển lên tầng trên.
- Nhận được gói tin với số thứ tự trong [rcv_base-N,rcv_base-1].Trong trường hợp này, gửi biên nhận lại cho gói tin (mặc dù đã biên nhận từ trước).
- Các trường hợp khác. Bỏ qua gói tin đó đi.



Hình 4.18 Truyền lại hay gói mới

Trong bước hai, bên nhận phải biên nhận lại (chứ không được bỏ qua) cho gói tin đến với số thứ tự nhỏ hơn giá trị biên của cửa sổ hiện thời. Điều này hết sức cần thiết. Ví dụ với không gian số thứ tự của bên gửi và bên nhận trong hình 4.16, nếu không nhận được ACK từ bên nhận xác nhận gói tin send_base đã được nhận, bên gửi sẽ gửi lại gói tin send_base, Phc dù rõ ràng rằng (với chúng ta, chứ không phải bên gửi) bên nhận đã nhận được gói tin đó. Nếu bên nhận không biên nhận gói tin này, cửa sổ bên gửi có thể sẽ không bao giờ trượt tới phía trước. Ví dụ này minh họa một đặc điểm quan trọng của giao thức SR (và nhiều giao thức tương tự khác). sự xác định của bên gửi và bên nhận về cái gì đã được nhận, cái gì chưa được nhận không phải luôn luôn giống nhau. với giao thức SR, điều này có nghĩa là cửa sổ bên gửi và bên nhận không bao giờ trùng khớp nhau.

Thiếu sự đồng bộ giữa cửa sổ bên gửi và bên nhận có thể gây hậu quả nghiêm trọng khi khoảng số thứ tự nhỏ. Ví dụ điều gì có thể xảy ra với khoảng số thứ tự là 4, các gói tin được đánh số là 0, 1, 2 và 3 , độ lớn cửa sổ là 3. Giả sử các gói tin từ 0, 1 , 2 được truyền đi và nhận chính xác tại phía bên nhận. Bên nhận gửi biên nhận cho 3 gói tin này. Khi đó, cửa sổ bên nhận tiến lên các gói tin thứ 4,5 và 6 với số thứ tự tương ứng là 3, 0, 1. Bây giờ xem xét hai trường

hợp. Trường hợp đầu tiên, hình 3.26(a), ACK của 3 gói tin đầu tiên bị mất, bên gửi truyền lại các gói tin đó. Khi đó bên nhận nhận được tiếp theo gói tin có số thứ tự 0 - lại chính là gói tin 0 đầu tiên được gửi ban đầu.

Trong trường hợp thứ hai, hình 4.18(b), ACK cho ba gói tin được chuyển đi thành công. Như vậy cửa sổ bên gửi sẽ trượt về phía trước và gửi đi các gói tin 4, 5, và 6 với số thứ tự tương ứng là 3, 0, 1. Nếu gói tin với số thứ tự 3 bị mất, lúc ấy gói tin có số thứ tự 0 đến, gói tin này chứa dữ liệu mới (không phải gói tin 0 truyền lại). Rõ ràng có một bức "màn chắn" giữa bên gửi và bên nhận vì bên nhận không thể "nhìn" thấy hành động từ bên gửi. Bên nhận chỉ quan sát được gói tin nào nó nhận được hay gửi đi. Hai trường hợp trong là tương tự nhau. Không có phương pháp nào phân biệt được gói 0 được truyền lại hay gói 5 được truyền lần đầu tiên. Rõ ràng nếu kích thước cửa sổ nhỏ hơn khoảng số thứ tự một đơn vị thì hệ thống không còn làm việc đúng đắn. Nhưng độ lớn cửa sổ nên là bao nhiêu? Người ta chứng minh được rằng độ lớn cửa sổ phải bé hơn hoặc bằng một nửa khoảng số thứ tự với giao thức SR.

Chúng ta giả thiết môi trường truyền không tin cậy ở dưới dẫn đến việc các gói tin có thể bị giữ lại trên đường truyền. Đây là việc ít khi xảy ra khi kênh truyền giữa phía gửi và phía nhận là một môi trường vật lý thực sự. Tuy nhiên khi kênh truyền này lại là một mạng máy tính thì việc một gói tin bị giữ lại trên kênh truyền hoàn toàn có thể xảy ra. Hệ quả của nó là xuất hiện một gói tin với số thứ tự hay số biên nhận là x trong khi cả cửa sổ nhận và gửi đều không chứa x. Trong trường hợp này, kênh truyền bị coi là một bộ đệm, có thể tuỳ ý phát mại gói tin ở bất cứ thời điểm nào. Vì số thứ tự có thể được sử dụng lại nên trong một số trường hợp sẽ xảy ra hiện tượng trùng lặp gói tin. Giải pháp trong thực tế là bảo đảm số thứ tự không được sử dụng lại cho đến khi bên gửi có thể tương đối chắc chắn về gói tin với số thứ tự x được gửi trước đây không còn tồn tại trong mạng. Điều này được thực hiện với giả thiết một gói tin không thể "Tồn tại" trong mạng trong một khoảng thời gian lớn hơn một khoảng thời gian cố định nào đấy. Thời gian "sống" lớn nhất của gói tin xấp xỉ là 3 phút với mạng TCP cao tốc [RFC 1323].

CHƯƠNG 5: LẬP TRÌNH SOCKET

Socket là một phương pháp để thiết lập kết nối truyền tin giữa một chương trình yêu cầu dịch vụ và một chương trình cung cấp dịch trên mạng và đôi khi trong máy tính. Theo định nghĩa các chức năng mỗi tầng trong mô hình OSI, tầng mạng đảm nhiệm liên kết giữa đầu cuối với đầu cuối thì tầng vận tải đảm nhiệm liên kết giữa hai tiến trình. Mỗi socket là một điểm cuối trong một kết nối và có thể được xem như kết hợp của tầng mạng và tầng vận tải, do đó nó phải bao gồm một cặp địa chỉ logic của tầng mạng và số hiệu cổng của tầng vận tải. Một socket trên máy yêu cầu dịch vụ có địa chỉ mạng được cấp sẵn để gọi một socket trên máy cung cấp dịch vụ. Một khi socket đã được thiết lập phù hợp, hai máy tính có thể trao đổi dịch vụ và dữ liệu.

Khái niệm lập trình socket do trường đại học Berkeley để xuất vào những năm 1980, xuất hiện lần đầu tiên trong hệ điều hành Unix là Berkeley Sockets Interface, một chương trình thiết bị được thiết kế để giúp máy tính nối mạng có thể trao đổi thông tin với nhau. Giữa những năm 1990, Microsoft đã tạo riêng socket của họ là Windows Sockets (còn gọi là WinSock), nhờ vậy các ứng dụng Windows có thể trao đổi thông tin với nhau qua mạng máy tính.

5.1 Khái niệm về socket

5.1.1 Mô hình client/server

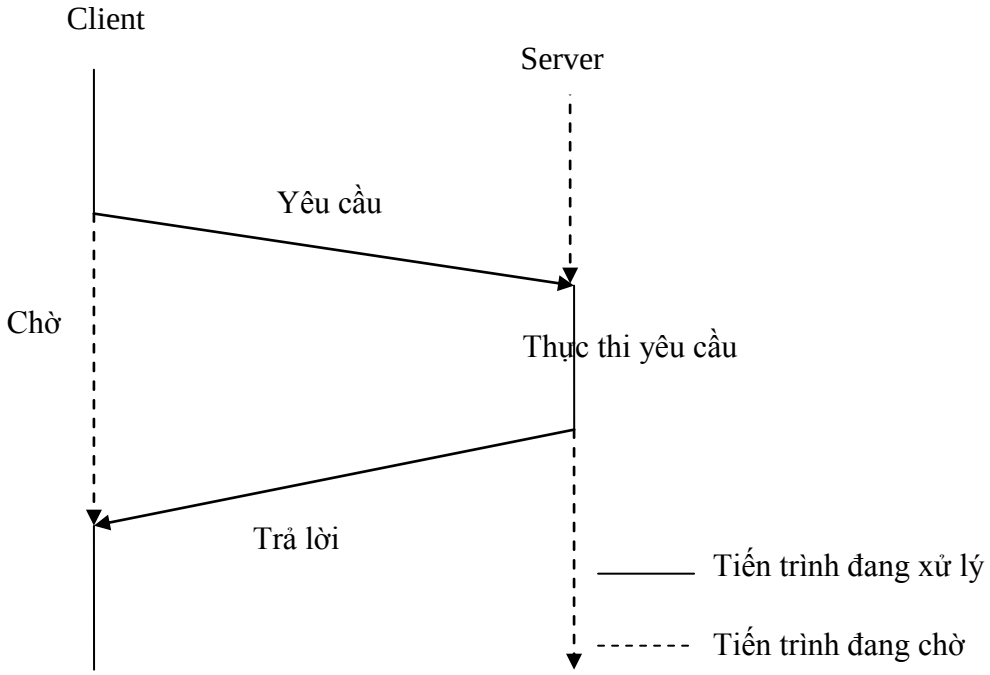
Mô hình client/server cung cấp một cách tiếp cận tổng quát để chia sẻ tài nguyên trong các hệ thống phân tán, trong đó tiến trình cung cấp dịch vụ gọi là server và ngược lại tiến trình sử dụng dịch vụ gọi là client. Tài nguyên sẽ được quản lý bởi một tập các tiến trình gọi là các tiến trình server và mọi tiến trình client muốn thực hiện truy xuất tới tài nguyên thông qua tiến trình server đó. Bản thân các tiến trình server cũng cần phải truy xuất tới các tài nguyên dùng chung được quản lý bởi một tiến trình khác, vì vậy một số tiến trình có thể đóng cả hai vai trò client và server. Các tiến trình client gửi yêu cầu tới tiến trình server, nếu yêu cầu hợp lệ thì tiến trình server sẽ thực hiện yêu cầu và trả về kết quả cho client. Mô hình truyền tin client/server hướng tới việc cung cấp dịch vụ. Quá trình trao đổi dữ liệu bao gồm:

- Tiến trình client gửi yêu cầu tới tiến trình server
- Tiến trình server kiểm tra và xử lý yêu cầu của tiến trình client
- Tiến trình server gửi kết quả xử lý yêu cầu cho client

Như vậy, mô hình client/server thực chất gồm hai bước truyền thông điệp và một bước xử lý, như vậy nảy sinh vấn đề đồng bộ client và server. Tiến trình client phải ở trạng thái chờ cho đến khi nhận được kết quả trả về từ tiến trình server. Mô hình client/server thường được cài đặt dựa trên các thao tác cơ bản là gửi và nhận. Quá trình giao tiếp giữa client và server có thể diễn ra theo một trong hai chế độ: đồng bộ hoặc không đồng bộ.

Chế độ đồng bộ: Trong chế độ đồng bộ, khi tiến trình client hoặc server phát ra lệnh gửi dữ liệu, việc thực thi của tiến trình sẽ bị tạm ngừng cho tới khi tiến trình nhận phát ra lệnh nhận dữ liệu. Tương tự đối với tiến trình nhận dữ liệu, nếu tiến trình nào đó phát ra lệnh nhận dữ liệu, mà tại thời điểm đó chưa có dữ liệu gửi tới thì việc thực thi của tiến trình cũng sẽ bị tạm ngừng cho tới khi có dữ liệu gửi tới.

Chế độ không đồng bộ: Trong chế độ này, khi tiến trình client hay server phát ra lệnh gửi dữ liệu thực sự, việc thực thi của tiến trình vẫn được tiến hành mà không quan tâm đến việc có tiến trình nào phát ra lệnh nhận dữ liệu đó hay không. Tương tự cho trường hợp nhận dữ liệu, khi tiến trình phát ra lệnh nhận dữ liệu, nó sẽ nhận dữ liệu hiện có, việc thực thi của tiến trình vẫn được tiến hành mà không quan tâm đến việc có tiến trình nào phát ra lệnh gửi dữ liệu tiếp theo hay không.



Hình 5.1 Tương tác giữa Client và Server

Khi phát triển các ứng dụng mạng, lập trình viên cần chú ý tiến trình ở trạng thái bị chờ, vì nó có thể dẫn đến tình huống một tiến trình nào đó sẽ rơi vào vòng lặp vô hạn.

5.1.2 Các kiến trúc Client/Server

5.1.2.1 Client/Server hai tầng

Trong ứng dụng client/server hai tầng, khối lượng công việc xử lý được dành cho phía client trong khi server chỉ đơn giản đóng vai trò như là chương

trình kiểm soát luồng vào ra giữa ứng dụng và dữ liệu. Có thể tìm thấy những ứng dụng loại này như dịch vụ truy nhập từ xa (telnet), truyền tập tin (ftp), thậm chí là cả dịch vụ Web (web tĩnh). Kiến trúc này có nhược điểm sau:

- Tăng lưu lượng thông tin lưu chuyển trên mạng: Trong quá trình xử lý, tiến trình client cần nhiều thông tin phụ trợ do đó phải gửi nhiều yêu cầu tới tiến trình server.
- Giảm hiệu năng xử lý:các máy trạm thường có cấu hình thấp hơn các máy chủ.
- Khó khăn trong việc bảo trì và nâng cấp hệ thống: muốn bảo trì và nâng cấp hệ thống phải thay đổi cả phía server lẫn client.

5.1.2.2 Client/Server ba tầng

Kiến trúc client/server ba tầng hạn chế các nhược điểm của kiến trúc client/server hai tầng bằng cách thêm một tầng mới tách biệt việc xử lý dữ liệu với việc cung cấp dịch vụ. Trong kiến trúc ba tầng, một ứng dụng được chia thành ba tầng tách biệt nhau về mặt. Tầng đầu tiên là tầng trình diễn thường chỉ đóng vai trò hiển thị các thông tin cho người sử dụng. Tầng thứ hai, thực thi các công việc xử lý nghiệp vụ. Tầng thứ ba chứa dữ liệu cần thiết cho ứng dụng, dữ liệu này có thể bao gồm bất kỳ nguồn thông tin nào (Oracle, SQL Server hoặc tài liệu XML…). Về cơ bản, tầng thứ ba thực hiện các lời gọi hàm/thủ tục để truy xuất dữ liệu cần thiết. Sự tách biệt giữa chức năng xử lý với giao diện đã tạo nên sự độc lập và linh hoạt cho việc phát triển các ứng dụng. Nhiều giao diện người dùng được xây dựng và triển khai mà không làm thay đổi logic ứng dụng.

5.1.2.3 Kiến trúc n- tầng

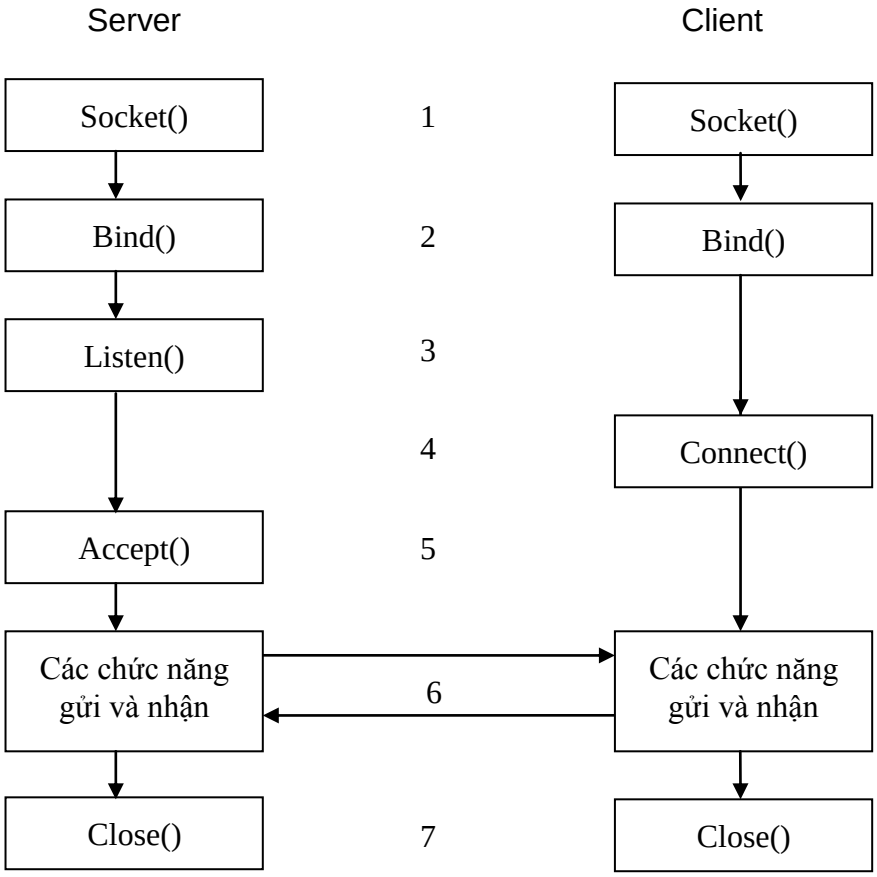
Kiến trúc ba tầng được ứng dụng tương đối phổ biến, tuy nhiên trong một số tình huống chúng ta có thể mở rộng thành kiến trúc n-tầng. Việc thêm mỗi tầng sẽ phức tạp hơn trong vấn đề xử lý, tăng giá thành sản phẩm phần mềm. Về cơ bản việc phân tầng phải đảm bảo các nguyên tắc sau:

- Tầng ứng dụng: quản lý tương tác của người dùng với ứng dụng.
- Tầng trình diễn: Xác định cách thức hiển thị giao diện người dùng và cách quản lý các yêu cầu của người dùng.
- Tầng nghiệp vụ: Mô hình hóa các quy tắc nghiệp vụ.
- Tầng dịch vụ hạ tầng: Cung cấp các chức năng cần thiết phục vụ cho tầng nghiệp vụ

5.1.3 Mô hình truyền tin socket

Hai giao thức TCP và UDP là các giao thức tầng vận tải để truyền dữ liệu, mỗi giao thức có những ưu và nhược điểm riêng. Giao thức TCP là giao thức có liên kết, nó có độ tin cậy truyền tin cao nhưng tốc độ truyền tin bị hạn chế do phải có giai đoạn thiết lập và giải phóng liên kết, khi đoạn tin có lỗi hay bị thất lạc thì giao thức TCP phải thực hiện truyền lại. Giao thức UDP thuộc loại không liên kết, có tốc độ truyền tin rất nhanh vì không cần phải thiết lập và giải phóng

liên kết. Khi lập trình cho giao thức TCP thì sử dụng các socket dạng luồng, còn đối với giao thức UDP sẽ sử dụng DatagramSocket và DatagramPacket.



Hình 5.2 Các bước cơ bản của tiến trình Client/Server

Truyền tin có liên kết nghĩa là cần phải có giai đoạn thiết lập và giải phóng liên kết trước và sau khi truyền tin. Dữ liệu được truyền trên mạng máy tính dưới dạng các gói tin, mỗi gói tin chứa thông tin điều khiển bao gồm địa chỉ + cổng nguồn và đích. Tuy nhiên do các gói tin có chiều dài hữu hạn nên bên gửi thường phải phân chia dữ liệu thành nhiều gói tin và khôi phục lại dữ liệu ban đầu từ các gói ở bên nhận. Trong quá trình truyền tin có thể có thể có một hay nhiều gói bị mất hay bị hỏng và cần phải truyền lại hoặc các gói tin đến không theo đúng trình tự. Để giải quyết những vấn đề này, bên gửi phải chia dữ liệu thành các gói và thêm thông tin điều khiển, bên nhận phải phân tích thông tin điều khiển và quản lý danh sách các gói tin nhận được, ... rất nhiều công việc cần phải thực hiện và việc phát triển phần mềm mạng trở nên phức tạp. Tuy nhiên, những công việc phức tạp đó đã được socket xử lý, chúng cho phép người lập trình xem một liên kết mạng như là một luồng mà có thể đọc dữ liệu ra hay ghi dữ liệu vào từ luồng này. Socket coi tất cả các thao tác gửi/nhận tương tự như thao tác đọc/ghi tập tin trên ổ đĩa máy tính, các thao tác mức thấp hoàn toàn trong suốt đối với người lập trình.

Một socket có thể thực hiện bảy thao tác cơ bản:

- Kết nối với một máy ở xa
- Gửi dữ liệu
- Nhận dữ liệu
- Ngắt liên kết
- Gán cổng
- Nghe dữ liệu đến
- Chấp nhận liên kết từ các máy ở xa trên cổng đã được gán.

Mỗi khi liên kết được thiết lập, một kênh truyền logic được hình thành giữa các hai tiến trình mạng và chúng có thể trao đổi thông tin với nhau theo chế độ song công. Khi việc truyền dữ liệu hoàn thành, một hoặc cả hai tiến trình sẽ gửi yêu cầu ngắt liên kết. Với sự phát triển của công nghệ thông tin, lập trình viên có nhiều lựa chọn khác để phát triển các ứng dụng mạng như: gọi thủ tục từ xa (Remote Procedure Call - RPC), Java RMI, CORBA, webservice…, nhưng socket vẫn tiếp tục đóng vai trò quan trọng trong việc duy trì các luồng truyền thông trên mạng máy tính. Các ứng dụng có liên quan đến mạng máy tính đều viết ở lớp bên trên socket.

5.2 Java sockets

Trong Java, thư viện Java.net.Socket là lớp được dùng trong việc tạo ra các socket phía client và java.net.ServerSocket được dùng cho phát triển ứng dụng phía server. Với những công cụ này, các lập trình viên có thể nhanh chóng tạo ra các socket mà không cần phải vất vả trong việc tạo các và quản lý các liên kết tại lớp vận tải. Lớp Socket của Java được sử dụng bởi cả client và server, có các phương thức tương ứng với bốn thao tác đầu tiên. Ba thao tác cuối chỉ cần cho server để chờ các client liên kết với chúng và được cài đặt bởi trong lớp ServerSocket.

5.2.1 Socket cho phía server

Lớp ServerSocket trong thư viện java.net cung cấp các phương thức để viết các ứng dụng phía server bằng ngôn ngữ Java. Nó cho phép tạo các đối tượng ServerSocket mới, các phương thức để lắng nghe các liên kết trên một cổng xác định, và các phương thức trả về một Socket khi thiết lập thành công một liên kết, và sẵn sàng cho việc gửi và nhận dữ liệu. Vòng đời của một server bao gồm :

1. Một ServerSocket mới được tạo ra trên một cổng xác định bằng cách sử dụng phương thức ServerSocket(số hiệu cổng), số hiệu cổng phải tuân thủ các qui định của IANA (Assigned Numbers Authority). Nếu số hiệu cổng bằng 0 thì hệ thống tự gán cổng, nếu cổng đã sử dụng thì hệ phương thức sẽ báo lỗi.
2. ServerSocket lắng nghe liên kết đến trên cổng đó bằng cách sử dụng phương thức accept(). Phương thức accept() ở trạng thái phong tỏa cho tới khi một client gửi một yêu cầu thiết lập liên kết, nếu chấp

nhận nó trả về một đối tượng Socket cho client để thiết lập một liên kết giữa client và server.

3. Tùy thuộc vào kiểu server, hoặc phương thức getInputStream(), getOutputStream() hoặc cả hai được gọi để nhận các luồng vào ra để truyền tin với client.
4. Server và client tương tác theo một giao thức thỏa thuận sẵn cho tới khi một trong hai phía hoặc cả hai phía gửi yêu cầu hủy bỏ liên kết.
5. Server, client hoặc cả hai gửi yêu cầu hủy bỏ liên kết
6. Server trở về bước hai và đợi yêu cầu thiêt lập liên kết tiếp theo.

Tạo socket cho server:

Ví dụ: tạo một server socket cho cổng 8080:

```
try
{
        ServerSocket httpd = new ServerSocket(8080);
}
catch(IOException e)
{
        System. err.println(e);
}
```

Chấp nhận và hủy liên kết:

Một đối tượng ServerSocket hoạt động trong một vòng lặp chấp nhận các liên kết. Mỗi lần lặp nó gọi phương thức accept(), phương thức này trả về một đối tượng Socket thể hiện liên kết giữa client và server phục vụ cho việc truyền tin giữa client và server. Khi hoàn thành một giao tác, server gọi phương thức close() của đối tượng socket. Nếu client ngắt liên kết trong khi server vẫn đang hoạt động, các luồng vào ra kết nối server với client sẽ đưa ra ngoại lệ InterruptedException trong lần lặp tiếp theo.

public Socket accept() throws IOException

Khi hoàn thành thiết lập liên kết và sẵn sàng chấp nhận các yêu cầu liên kết của phía client thì phải gọi phương thức accept() của lớp ServerSocket. Phương thức này ở trạng thái bị phong tỏa, nó dừng quá trình xử lý và đợi cho tới khi client được kết nối. Khi client thực sự kết nối, phương thức accept() trả về đối tượng Socket. Ta sử dụng các phương thức getInputStream() và getOutputStream() để truyền tin với client. Ví dụ:

```
try
{
ServerSocket theServer = new ServerSocket(8080);
while(true)
{
                Socket con = theServer.accept();
                PrintStream p = new PrintStream(con.getOutputStream());
                p.println("Ready connected");
                con.close();
}
```

```
}
catch(IOException e)
{
        System.err.println(e);
}
```

Khi kết thúc làm việc với một đối tượng server socket thì cần phải đóng lại đối tượng này, sử dụng phương thức public void close() throws IOException. Ví dụ cài đặt một server cung cấp thời gian hệ thống như sau:

```
import java.net.*;
import java.io.*;
import java.util.Date;
public class daytimeServer
{
        public final static int daytimePort =13;
        public static void main(String[]args)
        {
                ServerSocket  theServer;
                Socket con;
                PrintStream p;
                Try
                {
                        theServer = new ServerSocket(daytimePort);
                        try
                        {
                                p= new PrintStream(con.getOutputStream());
                                p.println(new Date());
                                con.close();
                        }
                        catch(IOException e)
                        {
                                theServer.close();
                                System. err.println(e);
                        }
                }
                catch(IOException e)
                {
                        System. err.println(e);
                }
        }
}
```

5.2.2 Socket cho phía Client

Hàm public Socket(string host,int port) throws UnknownHostException, IOException tạo một socket TCP, trong đó host là địa chỉ hoặc tên miền (trong trường hợp này phải sử dụng dịch vụ DNS) và port là số hiệu cổng cung cấp dịch vụ của server. Ví dụ:

```
Try
{
        Socket s = new Socket( “www.ptit.edu.vn”,80);
}
catch(UnknownHostException e)
```

```
{
        System.err.println(e);
}
catch(IOException e)
{
        System.err.println(e);
}
```

Một số phương thức thường dùng:

Đối tượng Socket có một số trường thông tin riêng mà ta có thể truy nhập tới chúng thông qua các phương thức trả về các thông tin này.

Phương thức	Ý nghĩa
getInetAddress()	Trả về địa chỉ đích
getPort()	Trả về số hiệu cổng đích
getLocalAddress()	Trả về địa chỉ nguồn
getLocalPort()	Trả về số hiệu cổng nguồn
getInputStream()	Trả về một luồng nhập để đọc dữ liệu từ một socket vào chương trình
getOutputStream()	Trả về một luồng xuất thô để ghi dữ liệu từ ứng dụng ra đầu cuối của một socket

Đóng Socket:

Các socket được đóng một cách tự động khi một trong hai luồng đóng lại, hoặc khi chương trình kết thúc, hoặc khi socket được thu hồi. Tuy nhiên, lập trình viên không nên để hệ thống tự đóng socket, đặc biệt là khi các chương trình chạy trong khoảng thời gian vô hạn. Mỗi khi một Socket đã bị đóng lại, ta vẫn có thể truy xuất tới các trường thông tin InetAddress, địa chỉ cục bộ, và số hiệu cổng cục bộ thông qua các phương thức getInetAddress(), getPort(), getLocalHost(), và getLocalPort(). Tuy nhiên khi ta gọi các phương thức getInputStream() hoặc getOutputStream() để đọc dữ liệu từ luồng đọc InputStream hoặc ghi dữ liệu OuputStream thì xuất hiện ngoại lệ IOException. Lập trình viên có thể sử dụng các phương thức sau để đóng kết nối:

public void close() throws IOException

public void shutdownInput() throws IOException

public void shutdownOutput() throws IOException

Phương thức close() đóng cả các luồng nhập và luồng xuất từ socket. Trong một số trường hợp ta chỉ muốn đóng một nửa kết nối, hoặc là luồng nhập hoặc là luồng xuất. Bắt đầu từ Java 1.3, các phương thưc shutdownInput() và shutdownOutput() cho phép ta thực hiện điều này.

5.3 Máy chủ đa xử lý

Mỗi ứng dụng cung cấp dịch vụ trên máy chủ thường phải phục vụ đồng thời yêu cầu của nhiều client, nếu xử lý tuần tự thì thời gian chờ của client sẽ kéo dài và đôi khi vượt quá khoảng thời gian chờ cho phép và dẫn đến tình trạng lỗi. Để giải quyết vấn đề này, các ứng dụng trên server thường được xây dựng theo kiểu đa tiến trình (multiprocessing), mỗi tiến trình lại được xây dựng dựa trên nguyên lý xử dụng đa luồng (multithread). Về bản chất, thread là đơn vị nhỏ nhất của tiến trình được định thời bởi hệ điều hành và được bao hàm trong các tiến trình thực thi của máy tính. Mỗi thread có một ngăn xếp lời gọi cho các phương thức, đối số và biến cục bộ của thread đó.

Với cơ chế đa luồng, các ứng dụng có thể thực thi đồng thời nhiều dòng lệnh cùng lúc, nó có thể làm nhiều công việc đồng. Có thể hiểu một cách đơn giản: hệ điều hành với cơ chế đa nhiệm cho phép nhiều ứng dụng chạy cùng lúc thì ứng dụng với cơ chế đa luồng cho phép thực hiện nhiều công việc cung lúc. Tuy nhiên, lập trình viên cần lưu ý việc tạo và quản lí các tiến trình (process) tốn nhiều tài nguyên hệ thống hơn rất nhiều so với việc tạo ra một luồng (thread). Do đó giải pháp thường được áp dụng là tạo ra nhiều luồng xử lý song song. Trong ngôn ngữ lập trình Java, thư viện lớp java.lang.Thread cung cấp các lớp để quản lý luồng.

Mặc dù các thread chia sẻ các tài nguyên của tiến trình những vẫn có thể được thực thi một cách độc lập. Multi-thread hữu dụng khi dùng để lập trình đa nhiệm, khi một chương trình hoặc tiến trình có sự cố, toàn bộ hệ thống sẽ vẫn an toàn. Đặc biệt hữu dụng trong trường hợp một tiến trình duy nhất mà sinh ra nhiều thread con của một hệ thống đa nhiệm. Các ứng dụng Multi-thread chạy nhanh hơn trên các máy tính hỗ trợ xử lý đa luồng. Tuy nhiên, việc sử dụng multithreading không hề đơn giản. Khó khăn trong lớn nhất là việc quản lí đồng thời nhiều luông và phải lưu ý vấn đề tương tranh hệ thống, một trong những nguyên nhân cơ bản dẫn đến tình trạng khóa cứng (deadlock).

5.4 Lập trình socket với ngôn ngữ C

Các hàm phục vụ cho lập trình socket với ngôn ngữ C nằm trong thư viện socket.h. Giống như lập trình socket trong Java, lập trình viên cần phải xây dựng ứng dụng phía server và phía client. Các bước tạo lập một socket phía server gồm:

- Tạo một socket bằng cách gọi hàm socket().
- Nhúng socket đến địa chỉ của máy chủ sử dụng hàm bind(), bao gồm địa chỉ máy chủ và số hiệu cổng mà dịch vụ sẽ cung cấp.
- Nghe các yêu cầu kết nối đến từ client bằng cách sử dụng hàm listen().
- Tiếp nhận các kết nối sử dụng hàm accept().
- Trao đổi thông tin với client bằng các hàm read() và write().
- Đóng kết nối bằng hàm close().

Các bước thiết lập một socket phía client gồm:

- Tạo một socket bằng hàm socket().

- Gửi yêu cầu kết nối đến server bằng hàm connect().
- Trao đổi thông tin với server bằng các hàm read() và write().
- Đóng kết nối bằng hàm close().

CHƯƠNG 6: GIAO THỨC TCP

6.1 Cấu trúc segment

Đoạn tin của giao thức TCP bao gồm các trường thông tin điều khiển và dữ liệu của lớp ứng dụng. Khi cần gửi một tập tin có kích thước lớn - ví dụ tập tin hình ảnh, nó phải chia tập tin thành các đoạn có kích thước nhỏ hơn hoặc bằng kích thước tối đa của đoạn dữ liệu (MSS – Maximum Segment Size).

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Source Port						Destination Port
Sequence Number						
Acknowledgment Number						
HLEN	reserved	N S	C W R	E C N	Control Bits	Window
Checksum						Urgent Pointer
Options and padding :::						
Data :::						

Hình 6.1 Cấu trúc đoạn dữ liệu của giao thức TCP

Hình 4.19 minh họa cấu trúc của đoạn tin giao thức TCP, phần thông tin điều khiển bao gồm:

- Source Port, Destination Port: Trường số hiệu cổng nguồn, số hiệu cổng đích để thực hiện dịch vụ ghép kênh/phân kênh dữ liệu cho các ứng dụng lớp trên.
- Sequence Number, Acknowledgment Number: Trường số thứ tự (sequence number) 32 bit và trường số biên nhận (acknowledge number) 32 bit được bên gửi và bên nhận sử dụng trong việc cung cấp dịch vụ truyền dữ liệu tin cậy.
- Trường độ dài tiêu đề (length field) 4 bit xác định độ dài của phần thông tin điều khiển (đơn vị là 4 bytes), độ dài thay đổi phụ thuộc trường option (Nếu trường option rỗng, thì chiều dài là 5 x 4 bytes =20 bytes).
- ECN (Explicit Congestion Notification): Sử dụng trong điều khiển tắc nghẽn.
- 6 bit điều khiển. Bit ACK được sử dụng để chỉ ra rằng giá trị đặt trong trường biên nhận là đúng. Các bit RST, SYN và FIN được sử dụng trong việc thiết lập hay đóng kết nối. Khi bit PSH được bật, thì đây là dấu hiệu để yêu cầu bên nhận phải chuyển dữ liệu lên tầng trên ngay lập tức. Cuối cùng, bit URG được dùng để báo hiệu dữ liệu trong đoạn tin được thực thể tầng trên bên gửi tạo ra là "khẩn cấp". Vị trí byte cuối cùng của dữ liệu khẩn cấp được xác định bởi con trỏ dữ liệu khẩn 16 bit (ptr to urgent data). TCP phải

báo cho tầng trên biết có dữ liệu khẩn và đặt con trỏ vào cuối dữ liệu khẩn (Trong thực tế, PSH, URG và con trỏ dữ liệu khẩn không được sử dụng).

- Window: Trường độ lớn cửa sổ 16 bit được sử dụng để kiểm soát lưu lượng, đó là số lượng byte dữ liệu tối đa mà bên nhận có thể chấp nhận được.
- Checksum: Giá trị kiểm tra lỗi, được tính bằng phần bù của tổng chuỗi 16 bit.
- Urgent Pointer: Vị trí byte cuối cùng của dữ liệu khẩn cấp.
- Trường option có thể thay đổi tuỳ ý. Trường này được sử dụng để bên gửi và bên nhận có thể thương lượng về giá trị MSS hoặc giá trị gia tăng của cửa sổ trong mạng cao tốc, lựa chọn nhãn thời gian[1].

6.2 Truyền dữ liệu tin cậy

Giao thức IP chuyển các gói tin không đảm bảo tính chính xác, thứ tự cũng như tính toàn vẹn dữ liệu. Các gói tin IP có thể bị tràn tại bộ đệm của thiết bị định tuyến và đo đó không bao giờ đến được đích, dữ liệu có thể đến không đúng thứ tự hay các bit trong gói tin có thể bị thay đổi. Các đoạn dữ liệu của tầng vận tải được đặt trong gói tin IP để truyền qua mạng chúng hoàn toàn có thể bị mất hoặc bị thay đổi giá trị. Giao thức TCP tạo ra một đường truyền dữ liệu tin cậy trên nền giao thức IP không tin cậy. Dịch vụ truyền dữ liệu tin cậy của TCP đảm bảo dòng dữ liệu tới tiến trình nhận không có lỗi, liên tục, không trùng gặp và đúng thứ tự, nghĩa là dòng byte nhận được giống hệt dòng byte gửi đi.

Giao thức TCP cung cấp dịch vụ truyền dữ liệu tin cậy bằng cách sử dụng cơ chế phối hợp số tuần tự, xác nhận số tuần tự ACK và đồng hồ xác định thời gian quá hạn mỗi đoạn tin cần phải phản hồi. Giao thức TCP biên nhận cho dữ liệu đã được nhận chính xác bằng cách gửi lại số tuần tự nhận của đoạn tin kế tiếp đang chờ nhận (số tuần tự của đoạn tin đã nhận chính xác cộng thêm 1). TCP cũng thực hiện việc gửi liên tục (cơ chế đường ống), cho phép bên gửi có thể gửi nhiều đoạn tin mà chưa cần nhận biên nhận ngay. Cơ chế này cho phép nâng cao đáng kể hiệu suất của đường truyền. số lượng tối đa các đoạn tin được gửi chưa cần biên nhận ngay phụ thuộc vào cơ chế kiểm soát lưu lượng và kiểm soát tắc nghẽn của giao thức TCP.

Để đơn giản, giả thiết kết nối TCP giữa hai máy A và B chuyển dữ liệu từ máy A tới máy B. Tại phía gửi (máy A), thực thể TCP lấy dữ liệu của tầng ứng dụng, đóng gói trong các đoạn dữ liệu và chuyển cho tầng mạng. Ngay sau khi chuyển đoạn tin cho tầng mạng, TCP khởi động đồng hồ thời gian cho đoạn tin đó. Thời gian đợi kết thúc mà vẫn chưa nhận được biên nhận cho đoạn tin đã gửi sẽ sinh ra một ngắt thời gian, khi đó máy A phải xử lý bằng cách truyền lại đoạn tin đã tạo nên ngắt thời gian.

Nhận được một đoạn tin chứa giá trị trường biên nhận ACK hợp lệ, thực thể TCP phía gửi phải quyết định đó là ACK lần đầu tiên nhận được (tức là biên nhận cho một đoạn tin đã gửi nhưng chưa được biên nhận) hay chỉ là ACK trùng

[1] Xem thêm RFC 854 và RFC 1323.

lặp (biên nhận lại một gói tin đã từng được biên nhận). Trong trường hợp là ACK đầu tiên thì bên gửi sẽ biết rằng tất cả các đoạn tin có số thứ tự không vượt giá trị biên nhận vừa nhận được đã được nhận đúng tại phía bên nhận. Khi đó, bên gửi có thể cập nhật biến trạng thái TCP kiểm soát số thứ tự của đoạn tin cuối cùng mà nó cho rằng đã được nhận chính xác và theo đúng thứ tự tại phía bên nhận. Các hành động của bên nhận được tóm tắt trong bảng sau:

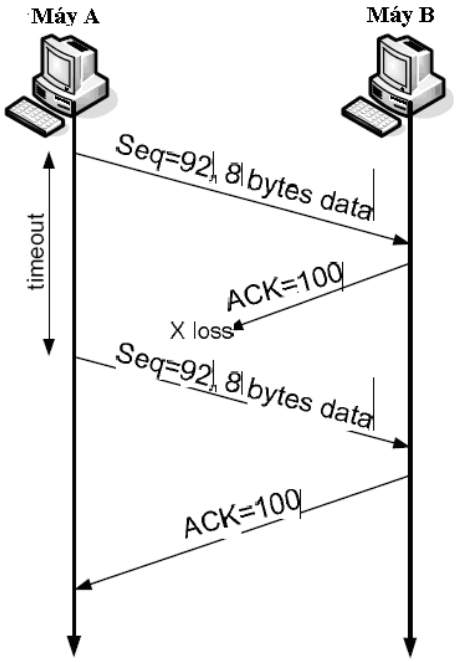
Sự kiện	Hành động bên nhận
Đoạn tin đến có số thứ tự là số thứ tự mong muốn. Tất cả dữ liệu đến số thứ tự mong muốn đã được biên nhận. Không có khoảng trống trong dữ liệu nhận được	Đợi một thời gian nhất định (500ms), nếu không có đoạn tin mới thì gửi ACK với số tuần tự biên nhận là số thứ tự của đoạn tin đến trước khi chờ +1.
Đoạn tin đến có số thứ tự là số thứ tự mong muốn. Đoạn tin đến trước đang đợi gửi biên nhận. Không có khoảng trống trong dữ liệu nhận được	Ngay lập tức gửi ACK.
Đoạn tin không đúng thứ tự đến, có số thứ tự cao hơn số thứ tự mong muốn nhận. Phát hiện có khoảng trống dữ liệu.	Ngay lập tức gửi ACK trùng lặp và chỉ ra số thứ tự của đoạn tin mong muốn nhận tiếp theo.
Segment đến lấp đầy một phần hoặc toàn bộ trống trong dữ liệu nhận được	Ngay lập tức gửi đi ACK biên nhận cho đoạn dữ khoảng liệu đúng thứ tự liên tục lớn nhất nhận được

. Khi bên nhận TCP nhận đoạn tin có số thứ tự lớn hơn số thứ tự đúng thứ tự đang được mong đợi, nó phát hiện có đoạn trống trong dòng dữ liệu - nghĩa là thiếu đoạn tin. Giao thức TCP không sử dụng biên nhận phủ định nên nó biên nhận lại đoạn tin đúng thứ tự cuối cùng mà nó nhận được (tạo ra ACK trùng lặp). Nếu bên gửi TCP nhận được 3 ACK trùng lặp cho cùng một đoạn tin, nó cho rằng đoạn tin ngay sau đoạn tin được biên nhận ba lần bị lỗi. Trong trường hợp này, TCP thực hiện cơ chế truyền lại nhanh ([RFC 258]), gửi lại đoạn tin đó trước khi đồng hồ thời gian của của đoạn tin lỗi thực hiện ngắt.

Bên gửi của TCP

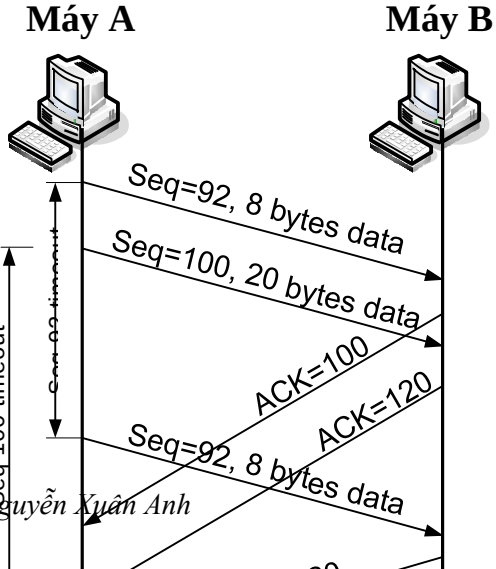
```
/* assume sender is not contrained b y TCP flow or congestion control, that data from
above is less than MSS in size, and that data transfer is in one direction only*/
sendbase = initial_sequence number nextseqnum = initial_sequence number
loop (forever) {
    switch (event)
      event: data received from application above
         create TCP segment with sequence number nextseqnum start timer for
         segment nextseqnum
         pass segment to IP
         nextseqnum = nextseqnum + length(data)
         break;  /* end of event data received from above */
      event: timer timeout for segment with sequence number y retransmit segment
         with sequence number y
         compute new timeout interval for segment y break: /* end of timeout event */
      event: ACK received with ACK field value of y
```

```
            if (y > sendbase) { /* cumulative ACK of all data up to y */ cancel all timers
               for segments with sequence numbers < y sendbase = y
               }
            else {/* a duplicate ACK for already ACKed segment */ Increment number of
               duplicate ACKs received for y
               If (number of duplicate ACKs received for y == 3) {
                     /* TCP fast retransmit */
                     Resend segment with sequence number y
                     Restart timer for segment y
                     }
               } break; /* end of ACK received event */
} /* end of loop forever */
```



Hình 6.2 Truyền lại vì mất ACK

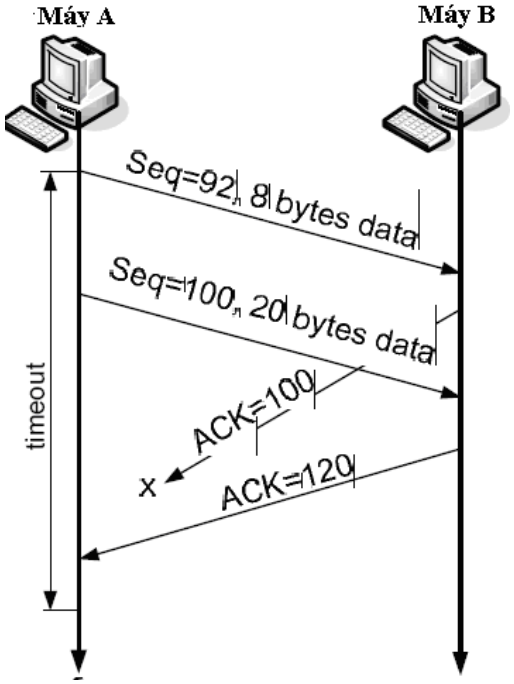
Giả sử máy A đang chờ một đoạn tin ACK từ máy B với giá trị biên nhận 100. Đoạn tin gửi từ máy A đã đến máy B nhưng ACK gửi từ máy B đến máy A bị mất. Trong trường hợp này, khi hết thời gian đợi, máy A truyền lại một đoạn tin có số tuần tự là 99 cho B. Tất nhiên khi nhận được đoạn tin truyền lại, máy B sẽ phát hiện sự trùng lặp nhờ trường số tuần tự, vì vậy thực thể TCP trên máy B sẽ loại bỏ đoạn tin truyền lại này.



Hình 6.3 Không cần truyền lại đoạn tin vì ACK đến trước thi hết thời gian đợi

Giả sử máy A gửi hai đoạn tin liên tiếp, đoạn tin đầu tiên có số thứ tự là 92, segment thứ hai có số thứ tự là 100. Giả sử cả hai đoạn tin này đều đến máy B nguyên vẹn và máy B gửi biên nhận ACK riêng rẽ cho từng đoạn tin, ACK cho đoạn tin đầu tiên có số biên nhận là 100 và cho segment thứ hai là 120. Giả thiết cả hai ACK đều không đến được máy A trước khi hết thời gian đợi của đoạn tin đầu tiên. Khi hết thời gian đợi, máy A gửi lại đoạn tin đầu tiên có số thứ tự 92. Máy A chỉ gửi lại đoạn tin thứ hai nếu hết thời gian đợi trước khi ACK có số biên nhận 120 hoặc lớn hơn đến. Vì vậy, như minh họa trên hình 4.21, nếu ACK thứ hai không mất và đến trước thời hạn của đoạn tin thứ hai thì máy A sẽ không phải gửi lại đoạn tin thứ hai. Trường hợp nếu ACK của đoạn tin đầu tiên bị mất, nhưng trước khi hết thời gian đợi của đoạn tin đầu tiên, máy A nhận được ACK có số biên nhận 120 - do đó máy A hiểu rằng mảy B đã nhận được tất cả các đoạn tin đến tận đoạn tin thứ 119, vì vậy máy A không phải gửi lại đoạn tin nào trong hai đoạn tin trên.

Mặc dù TCP là giao thức kiểu Go- Back-N nhưng chúng không giống hoàn toàn. Điểm khác biệt ở chỗ Go-Back-N truyền tất cả các đoạn tin có số tuần tự nhỏ hơn hoặc bằng số tuần tự của đoạn tin bị lỗi trong khi đó TCP chỉ truyền lại bản tin bị lỗi đó. Giả thiết cả hai giao thức cùng chuyển các đoạn tin 1…N, tất cả đoạn tin này đều được nhận đúng thứ tự và không có lỗi. Giả sử ACK của đoạn tin số M<N bị mất nhưng ACK của N- 1 segment còn lại đến bên nhận trước khi hết thời gian đợi của từng đoạn tin.



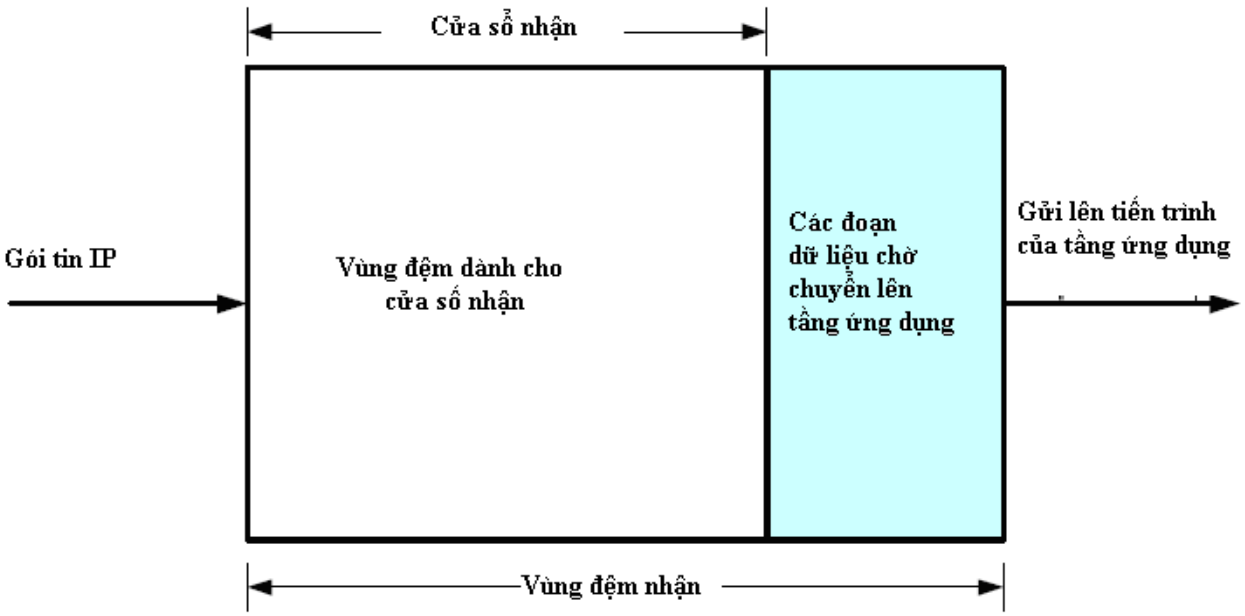
Hình 6.4 ACK tích lũy tránh việc truyền lại đoạn tin đứng trước

Trong trường hợp này, Go-Back-N sẽ truyền lại tất cả các đoạn tin M…N trong khi đó TCP sẽ truyền lại nhiều nhất là một đoạn tin có số tuần tự là thứ M, thậm chí TCP sẽ không truyền lại đoạn tin thứ M nếu ACK cho đoạn tin thứ M+1 đến trước khi hết thời gian đợi của đoạn tin thứ M. RFC 2018 đã đề xuất mở rộng cơ chế biên nhận của TCP giống kiểu giao thức lặp có lựa chọn, bên nhận phải cung cấp cho bên gửi những thông tin tường minh về các đoạn tin nào đã được nhận đúng và các đoạn tin nào bị lỗi hoặc chưa nhận được.

6.3 Điều khiển luồng

Khi kết nối TCP nhận được các đoạn dữ liệu, nó sẽ đặt chúng vào một vùng nhớ tạm thời gọi là đệm nhận. Một tiến trình tương ứng của tầng ứng dụng sẽ đọc dữ liệu từ bộ đệm này, sau khi đọc xong thì phải giải phóng bộ đệm để dành cho các đoạn tin kế tiếp. Vì một lý do nào đó, tiến trình đọc chưa đọc kịp dữ liệu trong bộ đệm, khi đó sẽ xảy ra tình trạng tràn bộ đệm. Để giải quyết vấn đề này, TCP cung cấp dịch vụ kiểm soát lưu lượng (flow control), thực chất đó là quá trình làm tương thích về tốc độ gửi/nhận.

Để kiểm soát lưu lượng, TCP bên gửi sử dụng biến receive window. Đây là giá trị mà bên nhận báo cho bên gửi biết độ lớn vùng đệm còn rỗi của nó. Trong kết nối hai hướng, ở mỗi phía kết nối có giá trị receive window phân biệt. Giá trị receive window động, có nghĩa là nó sẽ thay đổi trong thời gian kết nối. Giả sử máy A gửi một tập tin đếni máy B qua kết nối TCP. Máy B sẽ khởi tạo bộ đệm cho kết nối này với độ lớn RcvBuffer. Tiến trình ứng dụng trên B đọc dữ liệu tớ bộ đệm.



Hình 6.5 Cửa số và bộ đệm nhận

Giả thiết:

LastByteread = số thứ tự của byte cuối cùng trong dòng dữ liệu mà tiến trình ứng dựng trong máy B đọc từ buffer

LastByteRcvd = số byte cuối cùng trong dòng dữ liệu đến từ mạng và được để trong receive buffer của máy B

Vì TCP không được phép tràn bộ đệm nên chúng ta phải có:

LastByteRcvd - LastByteread < RcvBuffer

Receive window là giá trị RcvWindow, là độ lớn vùng đệm rỗi:

RcvWindow = RcvBuffer - [LastByteRcvd - LastByteread]

Vì độ lớn vùng đệm rỗi thay đổi theo thời gian nên giá trị RcvWindow động. kết nối sử dụng biến RcvWindow để cung cấp dịch vụ kiểm soát lưu lượng như thế nào? máy B báo cho máy A độ lớn vùng rỗi mà nó có trong bộ đệm là bao nhiêu bằng cách đặt giá trị RcvWindow hiện thời vào trong trường window của tất cả các segment gửi từ A. Ban đầu máy B thiết lập RcvWindow RcvBuffer. Rõ ràng để đạt được điều này thì máy B phải kiểm soát vài biến kết nối.

Máy A cũng có hai biến LastByteSent và LastByteAcked. Độ lệch giữa hai biến này, LastByteSent - LastByteAcked là số lượng dữ liệu chưa được biên nhận mà A gửi qua kết nối. Bằng cách khống chế số lượng dữ liệu chưa được biên nhận nhỏ hơn giá trị RcvWindow, A đảm bảo không làm tràn bộ đệm tại B. Do vậy trong suốt thời gian kết nối, A phải đảm bảo:

LastByteSent - LastByteAcked <= RcvWindow

Một vấn đề kỹ thuật nhỏ nẩy sinh ở đây. Giả sử bộ đệm ở máy B đầy, có nghĩa là RcvWindow = 0. Sau khi thông báo tới máy A là RcvWindow = 0, máy B không có gì để gởi tới máy A. Khi tiến trình ứng dụng ở máy B lấy dữ liệu lên

làm cho bộ đệm rỗng thì TCP không gửi đoạn tin mới cùng với giá trị RcvWindow mới tới máy A - TCP chỉ gửi đoạn tin tới A khi có dữ liệu hoặc ACK để gửi. Bởi vậy máy A sẽ không bao giờ được thông báo đã có thêm khoảng trống trong bộ đệm ở B. Máy A bị "khoá" và không truyền thêm dữ liệu. Để giải quyết vấn đề này, đặc tả TCP yêu cầu máy A tiếp tục gửi đoạn tin với một byte dữ liệu khi receive window của máy B bằng 0. Những đoạn tin này sẽ được B biên nhận. Khi bộ đệm bắt đầu có vùng rỗng thì trong gói biên nhận sẽ có cả giá trị khác không của RcvWindow.

Khác với TCP, giao thức UDP không có cơ chế kiểm soát lưu lượng. UDP đặt các datagram vào trong một hàng đợi có độ lớn hữu hạn ứng với socket nào. Tiến trình đọc lần lượt từng datagram trong hàng đợi. Nếu tốc độ đọc của tiến trình không đủ nhanh thì hàng đợi sẽ tràn và các datagram đến sau sẽ bị mất.

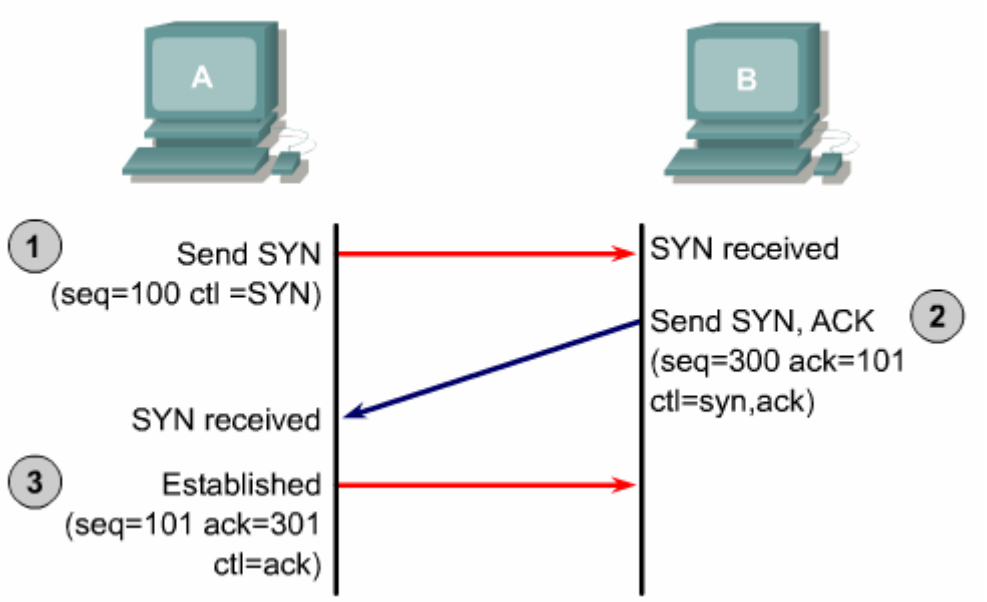
6.4 Quản lý kết nối

Giao thức TCP thuộc loại kết nối có hướng, do đó một kết nối của nó phải được thực hiện qua ba giai đoạn: thiết lập liên kết, truyền dữ liệu và hủy bỏ liên kết. Để thực hiện thiết lập liên kết và hủy bỏ liên kết, các tiến trình TCP trên các thiết bị đầu cuối của người sử dụng phải trao đổi các đoạn tin đặc biệt (những đoạn tin này sử dụng các cờ trong trường Control Bits). Giả sử tiến trình chạy trên máy máy khách muốn khởi tạo một kết nối tới một tiến trình trên máy chủ. Đầu tiên tiến trình ứng dụng trên máy khách yêu cầu thực thể TCP của nó thiết lập một kết nối tới một tiến trình trên máy chủ, sau đó thực thể TCP máy khách khởi tạo kết nối TCP tới thực thể TCP trên máy chủ qua ba bước sau:

Bước 1: Máy khách tạo một đoạn tin đặc biệt (không chứa dữ liệu của tầng ứng dụng nhưng chứa các thông tin điều khiển): cờ SYN trong Control Bits được đặt giá trị bằng 1, số tuần tự ban đầu (client_isn)… và chuyển xuống lớp mạng để gửi tới máy chủ.

Bước 2: Khi nhận được đoạn tin SYN, tiến trình TCP của máy chủ sẽ cấp phát bộ đệm nhận và các biến TCP phục vụ kết nối đồng thời gửi đi một đoạn đặc biệt thông báo chấp nhận kết nối từ máy khách. Đoạn tin này cũng không chứa dữ liệu của tầng ứng dụng mà chỉ thiết lập bit SYN và bit ACK (do đó gọi là đoạn tin SYNACK) trong trường Control bits, thiết lập số tuần tự của mình (server_isn), số tuần tự xác nhận bằng client_isn+1, thiết lập giá trị trường cửa số và các tham số khác.

Bước 3: Nhận được đoạn tin SYNACK, tiến trình TCP của máy khách cũng khởi tạo bộ đệm và các biến phục vụ kết nối đồng thời gửi đoạn tin thứ ba biên nhận với các tham số: Bit ACK được đặt bằng 1, số tuần tự là client_isn+1, số xác nhận là server_isn + 1.

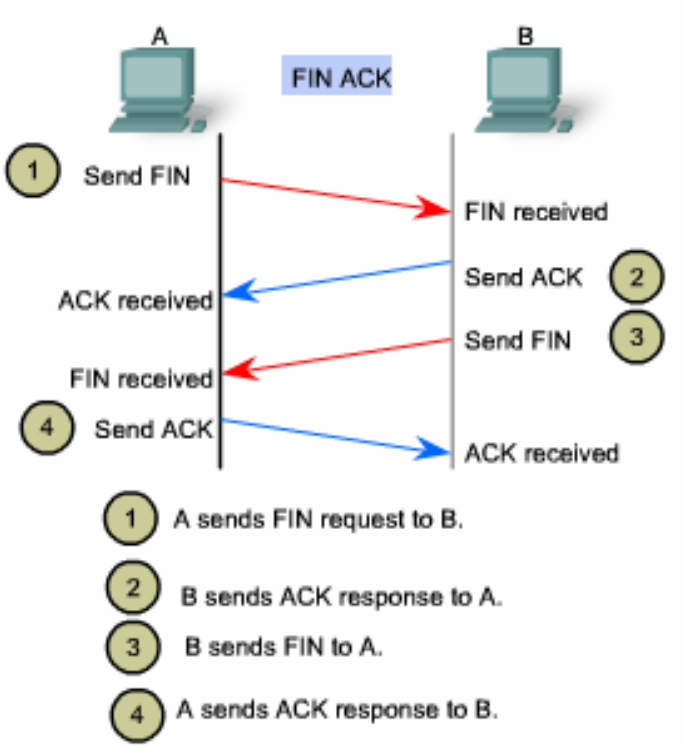


Hình 6.6 Ba bước bắt tay trong giai đoạn thiết lập kết nối của TCP

Sau khi đã thực hiện 3 bước trên, một kênh truyền logic đã được thiết lập, máy khách và máy chủ có thể trao đổi đoạn dữ liệu của lớp ứng dụng. Như vậy, để thiết lập được kết nối hai máy phải trao đổi 3 đoạn tị, vì vậy thủ tục kết nối được xem là quá trình bắt tay ba bước.

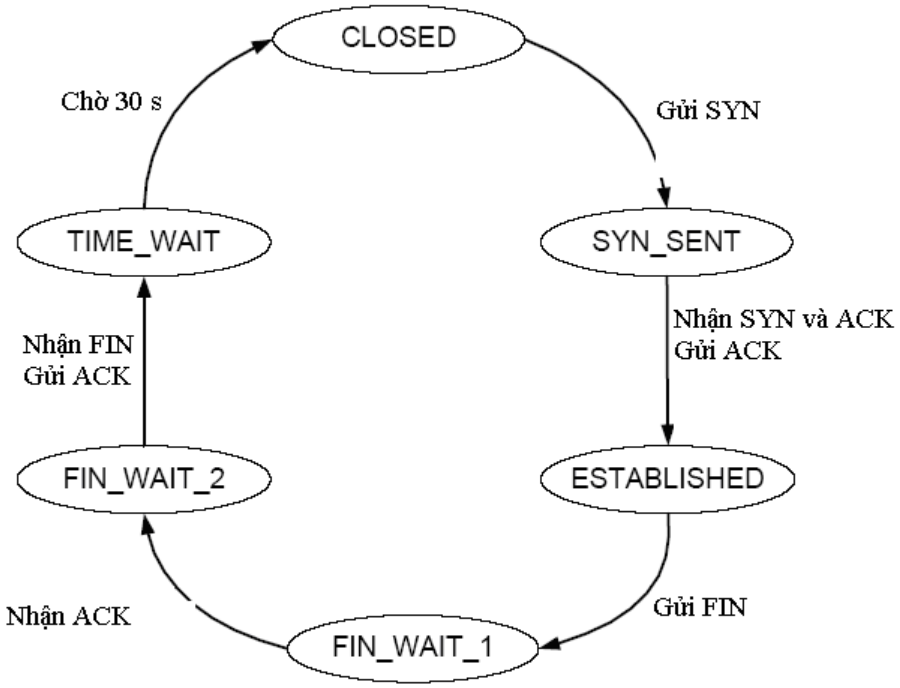
Sau khi truyền xong dữ liệu thì phải giải phóng tài nguyên mạng, do đó máy khách và máy chủ phải chuyển sang giai đoạn thứ ba là hủy bỏ kết nối, việc hủy bỏ kết nối không nhất thiết phải bắt đầu từ máy khách. Khi kết nối kết thúc thì các tài nguyên (bộ đệm và các biến TCP) trong máy được giải phóng. Ví dụ máy khách quyết định đóng kết nối. Tiến trình ứng dụng máy khách sẽ đưa ra lệnh đóng. Khi đó tiến trình TCP của máy khách gửi một đoạn tin đặc biệt đến tiến trình máy chủ: cờ FIN trong segment này được đặt giá trị 1. Nhận được đoạn tin FIN, máy chủ lần lượt gửi hai đoạn tin cho máy khách: một đoạn tin ACK biên nhận đoạn tin FIN của máy khách và một đoạn tin kết thúc FIN. Cuối cùng máy khách gửi đoạn tin ACK biên nhận đoạn tin FIN từ máy chủ. Sau bốn bước trên, tất cả nguyên của hai máy phục vụ cho kết nối đều được giải phóng.

Trong suốt thời gian kết nối TCP, giao thức TCP chạy trên mỗi máy chuyển qua các trạng thái TCP (TCP state, sử dụng lệnh netstat –a để kiểm chứng các trạng thái này). Hình 4.27 minh họa quá trình thay đổi trạng thái TCP xảy ra bên phía máy khách. TCP máy khách bắt đầu ở trạng thái đóng (CLOSED). Ứng dụng bên phía máy khách khởi tạo một kết nối TCP. Điều này đòi hỏi TCP máy khách gửi đoạn tin SYN tới máy chủ. Sau khi gửi đoạn tin SYN, TCP máy khách chuyển sang trạng thái SYN_SENT. Trong trạng thái SYN_SENT, TCP máy khách đợi đoạn tin SYNACK. Khi nhận được đoạn tin này, máy khách chuyển sang trạng thái ESTABLISHED. Ở trạng thái ESTABLISHED, máy khách có thể gửi và nhận những đoạn tin chứa dữ liệu của tầng ứng dụng.

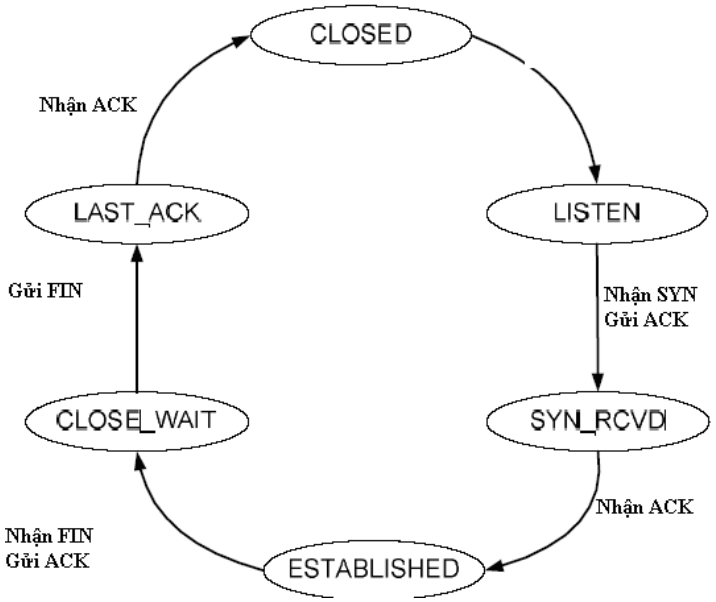


Hình 6.7 Kết thúc kết nối TCP

Giả sử ứng dụng máy khách quyết định đóng kết nối (máy chủ tương tự). Khi đó TCP máy khách gửi đoạn tin FIN và chuyển sang trạng thái FIN-WAIT_1. Trong trạng thái này, TCP máy khách đợi đoạn tin biên nhận từ phía máy chủ. Sau khi nhận được đoạn tin này, TCP máy khách chuyển sang trạng thái FIN_WAIT_2. Trong trạng thái FIN_WAIT_2, TCP máy khách đợi FIN đoạn tin từ máy chủ. Sau khi nhận đoạn tin này, TCP máy khách gửi đoạn tin ACK biên nhận tới máy chủ và chuyển sang trạng thái TIME_WAIT. Trong trạng thái TIME_WAIT, TCP máy khách có thể gửi lại biên nhận ACK trong trường hợp ACK trước bị mất. Thời gian đợi ở trạng thái TIME_WAIT phụ thuộc vào phần mềm triển khai TCP, nhưng thường nhận các giá trị 30 giây, một phút, hai phút. Sau khi hết thời gian đợi, kết nối chính thức được đóng và tất cả tài nguyên phía máy khách (bao gồm cả số hiệu cổng) được giải phóng.



Hình 6.8 Quá trình biến đổi trạng thái của Client



Hình 6.9 Quá trình biến đổi trạng thái của Server

6.5 Điều khiển tắc nghẽn

Một chức năng quan trọng khác của giao thức TCP là cơ chế kiểm soát tắc nghẽn. Cơ chế này của TCP chỉ dựa vào các thiết bị đầu cuối chứ không dựa vào cơ chế kiểm soát tắc nghẽn của tầng mạng vì tầng IP không cung cấp cho TCP các thông tin minh bạch khi có tắc nghẽn. Kết nối TCP kiểm soát tốc độ truyền của nó bằng cách giới hạn số lượng các đoạn tin đã gửi nhưng chưa được biên nhận. Gọi W là số lượng cho phép các đoạn tin chưa cần biên nhận, thường coi như kích thước cửa sổ của TCP. Trường hợp lý tưởng, kết nối TCP cho phép

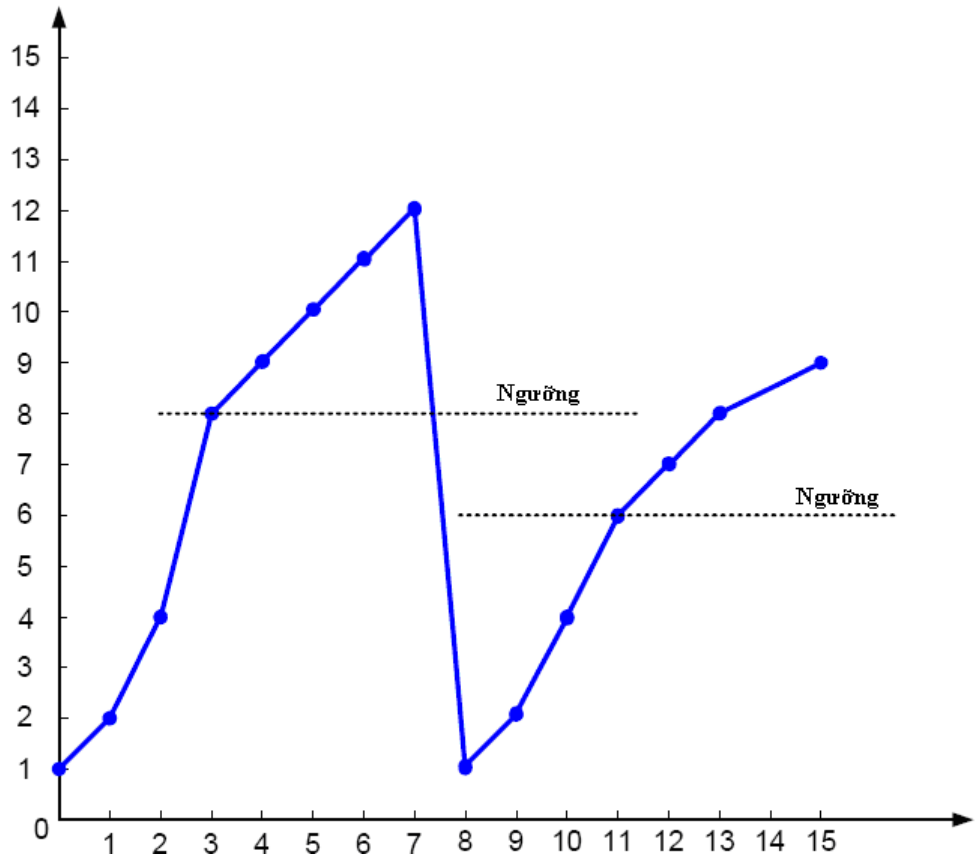
truyền với tốc độ tối đa có thể (càng nhiều đoạn tin chưa được biên nhận) chừng nào mà chưa xảy ra hiện tượng mất đoạn tin do bị tắc nghẽn. Nói chung kết nối TCP bắt đầu với giá trị W tương đối nhỏ và sau đó thăm dò kênh truyền còn dỗi hay không bằng cách tăng dần giá trị W. Kết nối TCP tiếp tục được tăng W cho đến khi xảy ra mất dữ liệu (sự kiện hết thời gian đợi hay nhận được các biên nhận trùng lặp). Khi đó TCP sẽ giảm W tới một giá trị an toàn và sau đó lại bắt đầu thăm dò kênh truyền rỗi bằng cách tăng dần giá trị W.

Mỗi kết nối của TCP có bộ đệm gửi, bộ đệm nhận và một vài biến (LastByteRead, RcvWin). Cơ chế kiểm soát tắc nghẽn của TCP bổ sung thêm hai biến nữa: cửa sổ tắc nghẽn (congestion window) và ngưỡng (threshold). Cửa sổ tắc nghẽn, ký hiệu là CongWin biểu thị số lượng dữ liệu tối đa mà người gửi có thể gửi qua kết nối. Như vậy khối lượng dữ liệu được gửi không được vượt quá CongWin và RcvWin, tức là:

LastByteSend - LastByteAcked <= min {CongWin,RcvWin}

Ngưỡng ký hiệu là threshold sẽ ảnh hưởng tới quá trình tăng của CongWin. Giả thiết bên gửi cần gửi một lượng dữ liệu lớn và bộ đệm bên nhận đủ lớn để có thể chứa được lượng dữ liệu chuyển đến, khi đó số lượng dữ liệu gửi chưa cần được biên nhận chỉ bị giới hạn bởi CongWin. Khi kết nối TCP đã được thiết lập giữa hai hệ thống đầu cuối, tiến trình ứng dụng gửi chuyển dữ liệu tới bộ đệm gửi của TCP. TCP chia dữ liệu thành các khối với kích thước MSS, đặt các khối dữ liệu trong đoạn tin, và chuyển chúng tới tầng mạng. Cửa sổ tắc nghẽn của TCP điều tiết số lượng đoạn tin được gửi. Ban đầu, CongWin nhận giá trị 1 MSS, TCP gửi đoạn tin đầu tiên và được biên nhận. Nếu đoạn tin này được biên nhận trước khi quá thời gian, phía gửi tăng CongWin và gửi đi hai đoạn tin. Nếu những đoạn tin này được biên nhận trong thời gian đợi của chúng, CongWin lại được tăng thêm 1 MMS cho mỗi đoạn tin được biên nhận. Khi đó CongWin mới là 4 MMS và phía gửi gửi đi bốn đoạn tin. Thủ tục này được thực hiện liên tục cho tới khi CongWin vượt ngưỡng hoặc không nhận được biên nhận trong thời gian chờ biên nhận.

Trong giai đoạn này, cửa sổ tắc nghẽn CongWin tăng theo hàm số mũ. Ban đầu nó nhận giá trị 1 MSS, sau đó tăng lên 2 MMS, 4 MMS, 8 MMS . . . Đây là giai đoạn khởi đầu chậm vì giá trị cửa sổ khởi đầu với giá trị nhỏ, nó sẽ tăng khá nhanh theo qui luật hạt thóc trên bàn cờ châu Âu. Giai đoạn khởi đầu chậm kết thúc khi CongWin vượt ngưỡng, lúc đó giá trị CongWin sẽ tăng tuyến tính chứ không còn tăng theo hàm số mũ. Tức là nếu CongWin = W, sau khi nhận được biên nhận cho W đoạn tin, giá trị CongWin sẽ tăng thêm 1, CongWin = W+1, giai đoạn này còn gọi là tránh tắc nghẽn. Giai đoạn tránh tắc nghẽn tiếp tục khi vẫn nhận được biên nhận trong thời gian đợi. Tuy nhiên giá tri cửa sổ cũng như tốc độ gửi dữ liệu không thể tăng mãi. Đến lúc nào đó sẽ xảy ra sự cố mất gói dữ liệu trên mạng. Điều này dẫn đến hiện tượng quá thời gian ở bên gửi. Lúc này giá trị ngưỡng (threshold) nhận giá trị bằng một nửa CongWin, CongWin được đặt bằng 1 MMS. Bên gửi sẽ tiếp tục tăng nhanh giá trị CongWin theo hàm số mũ cho đến khi nó vượt ngưỡng.



Hình 6.10 Cửa sổ kiểm soát tắc nghẽn

Tóm lại:

- Khi cửa sổ tắc nghẽn chưa vượt ngưỡng, cửa sổ sẽ tăng theo hàm mũ. Khi cửa sổ tắc nghẽn vượt ngưỡi, cửa sổ sẽ tăng tuyên tính.
- Khi hết thời gian đợi, giá trị ngưỡng bằng một nửa giá trị cửa sổ tắc nghẽn hiện thời và cửa sổ tắc nghẽn nhận giá trị 1.

Nếu bỏ qua giai đoạn khởi đầu chậm, TCP tăng độ lớn cửa sổ theo cấp số cộng khi mạng chưa bị tắc nghẽn và giảm độ lớn cửa sổ theo cấp số nhân (chia 2) ngay khi mạng bị tắc nghẽn. Vì vậy, TCP được coi là thuật toán AIMD (additive-increase, multiplicative-decrease). Độ lớn cửa sổ tắc nghẽn của TCP được minh họa trong hình 6.10, ngưỡng ban đầu bằng 8 MSS. Cửa sổ tắc nghẽn tăng nhanh theo luỹ thừa 2 trong giai đoạn khởi đầu chậm và đạt ngưỡng tại t=3, giá trị cửa sổ tắc nghẽn tăng tuyến tính đến khi xuất hiện mất dữ liệu. Giả sử khi dữ liệu bị mất, cửa sổ tắc nghẽn có giá trị 12 MMS. Ngưỡng mới được đặt bằng 6 MSS và cửa sổ tắc nghẽn bằng 1 MMS, quá trình lại được tiếp tục. Thuật toán này do V.Jacobson đề xuất, hiện nay có nhiều biến thể của thuật toán này.

Thuật toán Tahoe, Reno và Vegas:

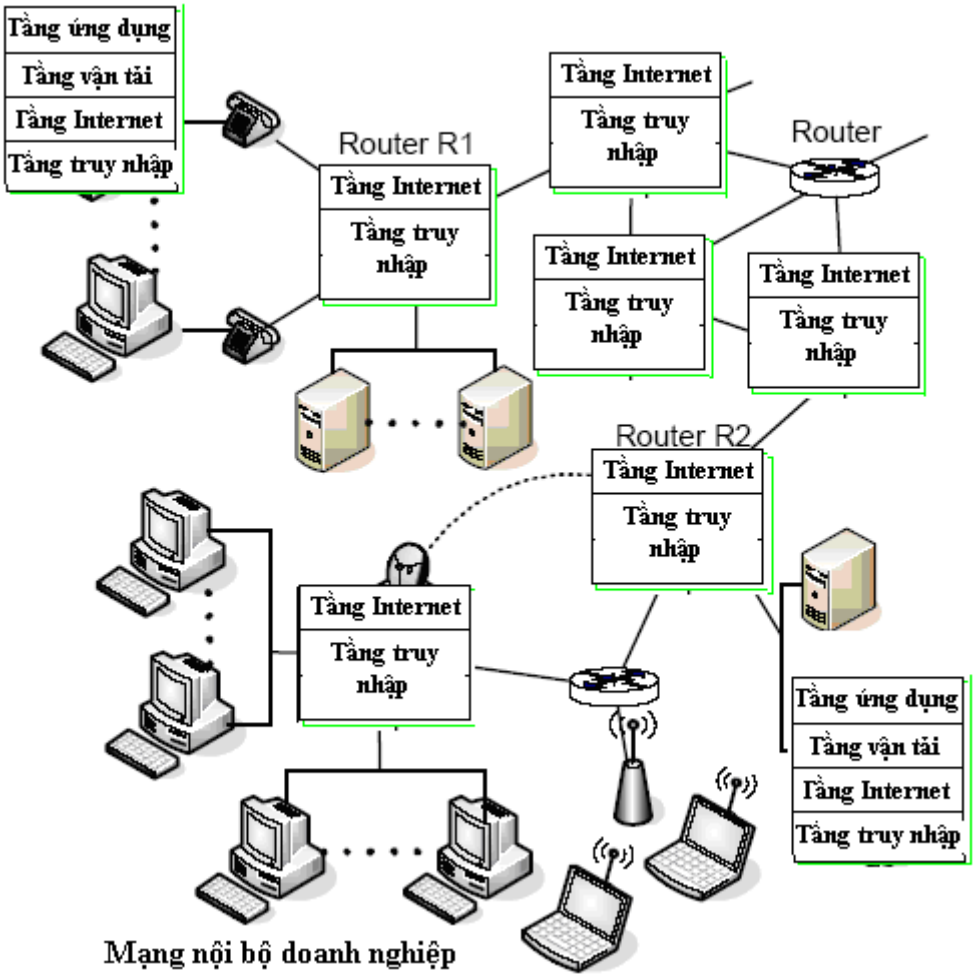
Thuật toán kiểm soát tắc nghẽn trên đây được gọi là Tahoe, tuy nhiên nó có nhược điểm khi một đoạn tin bị mất, bên phát có thể phải đợi trong một khoảng thời gian dài để gửi lại. Vì vậy một biến thể của thuật toán Tahoe gọi là Reno đã được triển khai trong phần lớn các hệ điều hành. Giống Tahoe, Reno đặt độ lớn

cửa sổ tắc nghẽn bằng 1 khi quá thời gian của bộ định thời. Tuy nhiên Reno có cơ chế truyền lại nhanh, bên phát sẽ gửi lại đoạn tin đã nhận được biên nhận ba lần ngay cả khi chưa hết thời gian đợi của đoạn tin này. Reno cũng sử dụng cơ chế khôi phục nhanh, hiện nay phần lớn thực thể TCP sử dụng thuật toán Reno, tuy nhiên cũng đã xuất hiện một số thuật toán cải tiến hiệu suất của Reno như thuật toán Vegas.

CHƯƠNG 7: TẦNG MẠNG VÀ GIAO THỨC IP

7.1 Mô hình dịch vụ tầng mạng

Tầng vận tải cung cấp dịch vụ truyền thông giữa hai tiến trình đang chạy trên hai máy tính khác nhau. Để có thể cung cấp được dịch vụ này, tầng vận tải phải sử dụng dịch vụ cung cấp đường truyền giữa hai máy tính của tầng mạng. Nói cụ thể hơn, tầng mạng chuyển đoạn tin của tầng vận tải từ máy tính này đến máy tính khác. Tại bên gửi, tất cả các đoạn dữ liệu của tầng vận tải được chuyển xuống tầng mạng. Nhiệm vụ của tầng mạng là chuyển những đoạn dữ liệu này đến máy tính đích và từ đó sẽ chuyển lên tầng vận tải của máy nhận. Không giống tầng vận tải, tầng mạng gồm nhiều máy tính và các thiết bị mạng trung gian, vì vậy giao thức tầng mạng là một trong những giao thức phức tạp nhất.



Hình 7.1 Mạng máy tính và các thiết bị mạng

Hình 7.1 minh họa một mạng đơn giản với hai máy tính H1 và H2 và một số thiết bị định tuyến trên đường truyền giữa H1 và H2. Nếu H1 gửi gói tin đến H2 thì tầng mạng trên H1 sẽ truyền gói tin này đến thiết bị định tuyến gần nhất: R1. Tại máy tính nhận H2, tầng mạng sẽ nhận gói tin từ thiết bị định tuyến gần nó nhất (trong trường hợp này là R2) và chuyển lên cho tầng vận tải tại. Vai trò

chính của thiết bị định tuyến là chuyển gói tin từ một cổng giao diện sang cổng giao diện khác. Vai trò của tầng mạng đơn giản chi là chuyền gói tin từ máy tính gửi đến máy tính nhận. Vì thế, tầng mạng có ba chức năng quan trọng sau đây:

Xác định đường đi: Tầng mạng phải xác định các tuyến đường mà gói tin được truyền từ nguồn đến đích. Thuật toán xác định tuyến đường như vậy gọi là thuật toán định tuyến. Thuật toán định tuyến sẽ quyết định đường đi của các gói tin từ máy tính nhận đến máy tính gửi (trong ví dụ là máy tính H1 và máy tính H2). Trọng tâm của chương này là các thuật toán định tuyến. Độ phức tạp của thuật toán định tuyến tăng tỉ lệ với số lượng thiết bị định tuyến trên đường truyền, diều này dẫn đến vấn đề định tuyến phân cấp.

Chuyển mạch: Khi gói tin đến đầu vào của một thiết bị định tuyến, nó phải quyết định gửi gói tin đến đầu ra thích hợp nào. Ví dụ, gói tin từ máy H1 đến router R1 sẽ phải được chuyển đến router kế tiếp trên đường tới H2.

Thiết lập đường truyền: Tại tầng vận tải, hai thực thể truyền thông phải có một giai đoạn "Bắt tay" trước khi trao đổi dữ liệu thực sự. Điều này cho phép bên gửi và bên nhận thiết lập các thông tin trạng thái cần thiết. Một số kiến trúc mạng khác (ví dụ ATM) đòi hỏi các thiết bị định tuyến trên tuyến đường từ nguồn đến đích phải "Bắt tay" nhau trước khi bắt đầu truyền dữ liệu thực sự, quá trình này được gọi là thiết lập đường truyền. Tầng mạng trong mô hình TCP/IP không đòi hỏi công việc này.

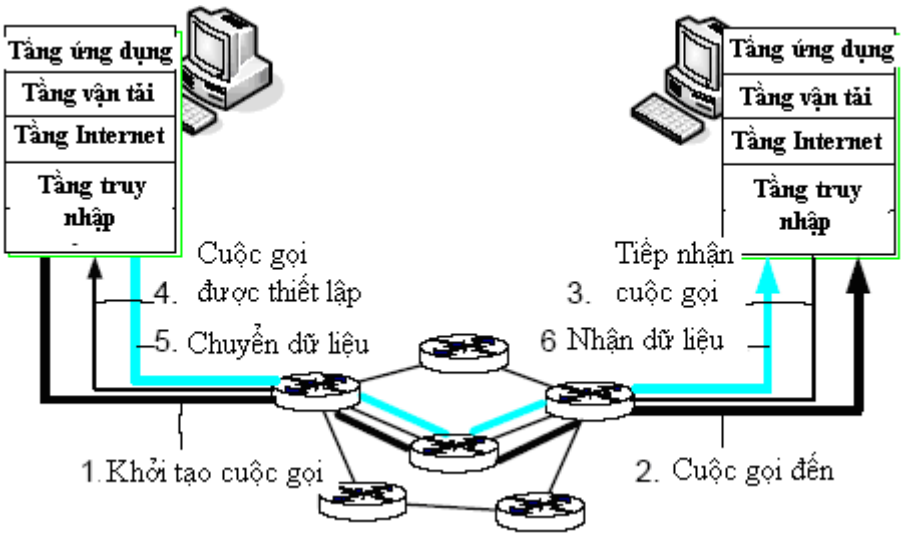
7.1.1 Nguyên lý chuyển mạch tầng mạng

Tầng mạng sử dụng mạch ảo (Virtual Circuit - VC) để truyền các gói tin, về khía cạnh nào đó mạch ảo tương tự mạng điện thoại truyền thống. Có ba giai đoạn trong chuyển mạch ảo:

- Thiết lập mạch ảo: trong cả giai đoạn thiết lập, nơi gửi thông báo địa chỉ nhận với tầng mạng, yêu cầu tầng mạng thiết lập mạch ảo. Tầng mạng xác định tuyến đường giữa bên gửi và bên nhận, tức là chuỗi các cung đường và các thiết bị chuyển mạch mà tất cả các gói dữ liệu sẽ đi qua. Điều này yêu cầu việc cập nhật bảng định tuyến và dự trữ tài nguyên trong mỗi thiết bị chuyển mạch.
- Truyền dữ liệu: Sau khi thiết lập được mạch ảo, dữ liệu có thể được chuyển trong mạch ảo đó.
- Đóng mạch ảo: Giai đoạn này bắt đầu khi phía gửi (hoặc phía nhận) báo cho tầng mạng yêu cầu đóng mạch ảo. Tầng mạng sẽ thông báo cho thiết bị đầu cuối bên kia cũng như các thiết bị chuyển mạch trên mạch ảo để cập nhật lại các bảng định tuyến, giải phóng tài nguyên.

Mạch ảo ở tầng mạng khác với việc thiết lập kết nối ở tầng vận tải. Thiết lập kết nối ở tầng vận tải chỉ liên quan đến các thiết bị đầu cuối của người sử dụng. Hai thiết bị đồng ý thiết lập kết nối và thoả thuận các thông số của kết nối (ví dụ số thứ tự khởi tạo, độ lớn cửa sổ kiểm soát lưu lượng). Hai thiết bị đầu cuối này sẽ nhận biết được về sự kết nối ở tầng vận tải, nhưng các thiết bị chuyển mạch ở giữa thì không biết các thông tin đó. Trái lại trong tầng mạng của mạng chuyển mạch ảo, tất cả các thiết bị chuyển mạch giữa hai thiết bị đầu cuối đều tham gia

vào quá trình thiết lập mạch ảo, và do đó đều nhận biết được tất cả các mạch ảo đi qua.

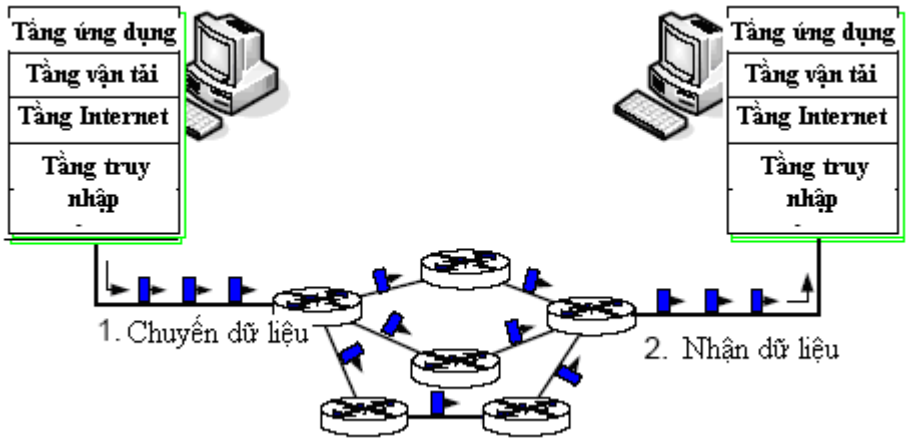


Hình 7.2 Mô hình dịch vụ chuyển mạch ảo

Bản tin trao đổi giữa các thiết bị đầu cuối yêu cầu khởi tạo hay kết thúc mạch ảo, bản tin trao đổi giữa các thiết bị chuyển mạch yêu cầu thiết lập mạch ảo (cập nhật bảng chuyển mạch) được gọi là bản tin báo hiệu. Giao thức được sử dụng để trao đổi những bản tin này và giao thức báo hiệu. Quá trình thiết lập mạch ảo được minh họa trong hình 5.2. ATM, Frame Relay và X.25 là ba kiến trúc mạng sử dụng chuyển mạch ảo.

Trong mạng chuyển mạch gói, khi thiết bị đầu cuối muốn gửi gói tin, nó đặt vào gói tin địa chỉ thiết bị nhận và sau đó chuyển gói tin vào mạng. Như minh họa trong hình 5.3, không có giai đoạn thiết lập mạch ảo. Những thiết bị trung chuyển trong mạng chuyển mạch gói (được gọi là bộ định tuyến - router) không duy trì bất kỳ trạng thái nào về mạch ảo. Thiết bị trung chuyển sẽ định tuyến gói tin đến đích bằng cách xác định địa chỉ đích, tìm kiếm trên bảng định tuyến và chuyển tiếp gói tin theo hướng đến đích (giống việc chuyển thư bình thường trong hệ thống bưu điện). Vì bảng định tuyến có thể được cập nhật liên tục, nên các gói tin được gửi từ thiết bị đầu cuối này đến thiết bị đầu cuối khác có thể đi theo nhiều tuyến đường khác nhau và đến đích không theo thứ tự. Mạng Internet công cộng ngày nay sử dụng dịch vụ chuyển mạch gói.

Để chuyển đoạn tin của tầng vận tải, tầng mạng thường đưa ra dịch vụ chuyển mạch ảo hoặc dịch vụ chuyển mạch gói nhưng không bao giờ cung cấp cả hai dịch vụ này. Ví dụ, dịch vụ của mạng ATM là mạch ảo trong khi mạng Internet cung cấp dịch vụ chuyển mạch gói. Dịch vụ chuyển mạch ảo được xếp vào lớp dịch vụ hướng nối vì phải thiết lập và kết thúc kết nối cũng như việc duy trì thông tin trạng thái của kết nối tại tất cả thiết bị chuyển mạch. Dịch vụ chuyển mạch gói được xếp vào lớp dịch vụ không hướng nối.



Hình 7.3 Mô hình chuyển mạch gói

Bảng 7.1 tóm tắt những nét chính của mô hình dịch vụ Internet và kiến trúc mạng ATM. Kiến trúc hiện nay của Internet chỉ cung cấp duy nhất dịch vụ chuyển mạch gói, một dịch vụ truyền số liệu theo kiểu hết sức cố gắng.

Bảng 7.1Những đặc điểm chính của mô hình dịch vụ Internet và mạng ATM

Mạng	Loại dịch vụ	Đảm bảo băng thông	Đảm bảo không mất	Thứ tự	Độ trễ	Kiểm soát Tắc nghẽn
Internet	Cố gắng tối đa	Không	Không	Không	Không	Không
ATM	CBR	Có	Có	Có	Có	Không tắc nghẽn
ATM	VBR	Có	Có	Có	Có	Không tắc nghẽn
ATM	ABR	tốc độ nhỏ nhất	Không	Có	Có	Không tắc nghẽn
ATM	UBR	Không	Không	Có	Có	Không

Tầng mạng của mô hình TCP/IP không đảm bảo thời gian gửi các gói tin giống nhau, các gói tin không được đảm bảo đến đích theo đúng thứ tự và thậm chí không đảm bảo gói tin đến được đích. Kiến trúc ATM cung cấp nhiều kiểu dịch vụ khác nhau (ATM có nhiều mô hình dịch vụ). Trong phạm vi cùng một mạng những kết nối khác nhau có thể được cung cấp những lớp dịch vụ khác nhau.

Dịch vụ truyền với tốc độ cố định (CBR- Constant Bit Rate): là mô hình dịch vụ ATM đầu tiên được chuẩn hoá có thể thấy được vai trò các công ty điện thoại đằng sau ATM. Dịch vụ mạng CBR là sự lựa chọn lý tưởng cho việc truyền dữ liệu đa phương tiện (ví dụ điện thoại số) theo thời gian thực với tốc độ truyền cố định. Mục tiêu của dịch vụ CBR là làm cho kết nối mạng trông giống như một đường kết nối thực sự giữa bên gửi và bên nhận. Trong dịch vụ CBR các tế bào được truyền qua mạng với một độ trễ nào. Biến thiên của độ trễ, tỷ lệ các tế bào bị mất hay đến trễ được đảm bảo không vượt quá một giá trị ngưỡng.

Tốc độ truyền tối đa của mỗi kết nối được xác định trước và bên gửi có thể gửi dữ liệu với tốc độ này.

Dịch vụ truyền với tốc độ không xác định (UBR - Unspecified Bit Rate): Không giống dịch vụ CBR (đảm bảo tốc độ, độ trễ mất mát dữ liệu), UBR không đảm bảo những điều này ngoại trừ việc gửi các cell theo đúng thứ tự. Như vậy dịch vụ UBR giống mô hình dịch vụ Cố gắng tối đa của Internet. dịch vụ UBR không cung cấp thông tin phản hồi cho bên gửi về việc các tế bào có đến được đích hay không. Với mạng UBR, tính tin cậy của truyền dữ liệu được triển khai trong các giao thức ở tầng cao hơn, dịch vụ UBR phù hợp với những ứng dụng truyền dữ liệu không cần tốc độ truyền cố định.

Dịch vụ truyền với tốc độ có sẵn (ABR - Available Bit Rate): tương tự như UBR nhưng bổ xung thêm hai tính năng:

- Tốc độ truyền tế bào nhỏ nhất (MRC) được đảm bảo cho kết nối ABR. Tuy nhiên khi tài nguyên của mạng rỗi, bên gửi có thể gửi với tốc độ cao hơn MCR.
- Có phản hồi về tắc nghẽn từ tầng mạng, mạng ATM có thể cung cấp thông tin phản hồi cho bên gửi (là bit thông báo tắc nghẽn hay tốc độ gửi thấp) để bên gửi điều chỉnh tốc độ gửi.

ABR không đảm bảo một băng thông tối thiểu nhưng cố gắng truyền dữ liệu nhanh nhất có thể. Như vậy, ABR phù hợp với các ứng dụng truyền dữ liệu yêu cầu độ trễ nhỏ (ví dụ duyệt Web).

Dịch vụ truyền với tốc độ biến đổi (variable bit rate - VBR): Trong dịch vụ VBR thời gian thực, tỷ lệ mất gói dữ liệu, độ trễ có thể chấp nhận được thỏa thuận trước giống dịch vụ CBR. Tuy nhiên, tốc độ gửi thực sự được phép thay đổi theo các tham số do người dùng đưa vào. Điều này cho phép sử dụng tài nguyên có hiệu quả hơn, nhưng xét theo các tiêu chí về mất mát dữ liệu, độ trễ thì VBR tương tự CBR.

7.1.2 Lịch sử chuyển mạch gói và chuyển mạch ảo

Lịch sử phát triển của mô hình dịch vụ mạng Internet và ATM phản ánh nguồn gốc của chúng. Vận hành dựa trên mạch ảo và cung cấp dịch vụ tốc độ cố định CBR, mạng ATM dường như thừa kế mô hình mạng điện thoại công cộng. Mặc dù sau này mạng ATM đã chú trọng các dịch vụ truyền số liệu, tuy nhiên với phương pháp sử dụng mạch ảo và phải hỗ trợ truyền dữ liệu theo thời gian thực đồng thời vẫn đảm bảo hiệu năng cho tất cả các dịch vụ, tầng mạng của ATM phức tạp.

Ngược lại mạng Internet (theo kiến trúc TCP/IP) xuất phát từ nhu cầu kết nối các máy tính (được xem là thiết bị thông minh) với nhau. Với thiết bị đầu cuối phức tạp, kiến trúc Internet lựa chọn mô hình dịch vụ mạng đơn giản nhất có thể và đặt các chức năng phụ trợ (ví dụ như truyền dữ liệu tin cậy), cũng như các ứng dụng mạng ở tầng cao hơn trên các thiết bị đầu cuối. Mô hình dịch vụ mạng của Internet không đảm bảo bất kỳ một dịch vụ nào do vậy có thể dễ dàng kết nối các mạng sử dụng những công nghệ kết nối rất khác nhau (ví dụ vệ tinh, Ethernet, cáp quang, sóng vô tuyến). Các công nghệ này có tốc độ và tỷ lệ mất

mát dữ liệu khác nhau. Những ứng dụng trên mạng Internet như thư điện tử, Web, và thậm chí cả dịch vụ của tầng mạng như DNS được triển khai trên các máy tính (và thiết bị đầu cuối). Các dịch vụ mới có thể nhanh chóng được sử dụng rộng rãi thông qua các giao thức tầng ứng dụng.

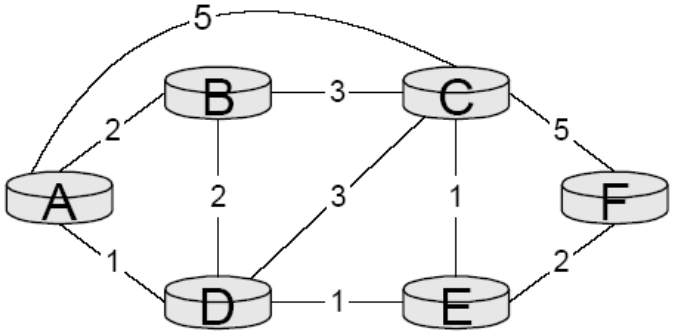
7.2 Nguyên tắc định tuyến

Để truyền gói dữ liệu từ máy tính gửi đến máy tính nhận, tầng mạng phải quyết định đường đi hoặc các thiết bị định tuyến mà gói dữ liệu phải đi qua. Dù mạng chuyển mạch gói (các gói tin khác nhau có thể đi theo các tuyến đường khác nhau) hay mạng mạch ảo (tất cả các gói tin được truyền trên cùng một tuyến đường định trước) thì tầng mạng đều phải xác định đường đi cho gói tin. Đây là công việc của các giao thức định tuyến ở tầng mạng.

Để chuyển một gói tin từ nguồn đến đích, trong mỗi thiết bị định tuyến phải tồn tại bảng định tuyến. Có hai cách xây dựng bảng định tuyến, định tuyến tĩnh và định tuyến động. Định tuyến tĩnh chỉ có thể áp dụng cho tuyến đường rất ít thay đổi, trong khi đó định tuyến động phù hợp với các tuyến đường khi lưu lượng mạng hay kiến trúc liên kết mạng bị thay đổi. Trong định tuyến động, một tiến tình định kỳ hoặc theo sự kiện liên tục gửi bản tin về tình trạng của các tuyến mạng.

Trọng tâm của giao thức định tuyến là thuật toán xác định đường đi cho gói tin hay còn gọi là thuật toán tìm đường, thuật toán định tuyến. Mục tiêu của thuật toán định tuyến hết sức đơn giản: với một tập hợp thiết bị định tuyến cùng với liên kết giữa các thiết bị định tuyến, thuật toán định tuyến phải xác định đường đi tốt nhất từ thiết bị nguồn đến thiết bị đích. Đường đi tốt nhất có thể là đường đi có giá nhỏ nhất, tuy nhiên trong thực tế nhiều vấn đề khác đã làm phức tạp hóa các thuật toán này, ví dụ các vấn đề liên quan đến chính sách, giá cả… Ví dụ: thiết bị định tuyến X thuộc tổ chức Y không được chuyển tiếp các gói tin được tạo ra từ mạng của tổ chức Z.

Người ta thường sử dụng lý thuyết đồ thị để áp dụng cho các thuật toán định tuyến. Hình 7.4 biểu diễn và mô hình hóa sơ đồ mạng dưới dạng đồ thị. Ở đây nút của đồ thị thể hiện thiết bị định tuyến, cung trong lý thuyết đồ thị nối các nút thể hiện đường truyền vật lý giữa các thiết bị định tuyến. Cung được đặc trưng bởi đại lượng giá là chi phí của việc gửi gói tin qua thiết bị định tuyến. Giá có thể phản ánh mức tắc nghẽn trên đường truyền (thời gian trễ trung bình) hoặc khoảng cách vật lý giữa hai thiết bị định tuyến. Để đơn giản chúng ta coi mỗi cung trên đồ thị có một giá và không quan tâm đến việc xác định giá đó bằng cách nào.



Hình 7.4 Mô hình mạng dưới dạng đồ thị

Với mô hình đồ thị, vấn đề tìm kiếm tuyến đường từ nguồn đến đích có chi phí thấp nhất yêu cầu xác định chuỗi các cung sao cho:

- Cung đầu tiên trong tuyến đường xuất phát từ nguồn.
- Đích của cung cuối cùng trong tuyến đường là đích.
- Với mọi i, cung thứ i và i-1 cùng kết nối vào một nút.

Với đường đi có giá nhỏ nhất, tổng chi phí của tất cả các cung trên trên tuyến đường là nhỏ nhất. Chú ý nếu tất cả các cung có giá như nhau thì đường đi có giá nhỏ nhất cũng là đường đi ngắn nhất giữa nguồn và đích. Ví dụ như trong hình 5.4, đường đi có giá nhỏ nhất giữa nút A (nguồn) và nút C (đích) là đường ADEC.

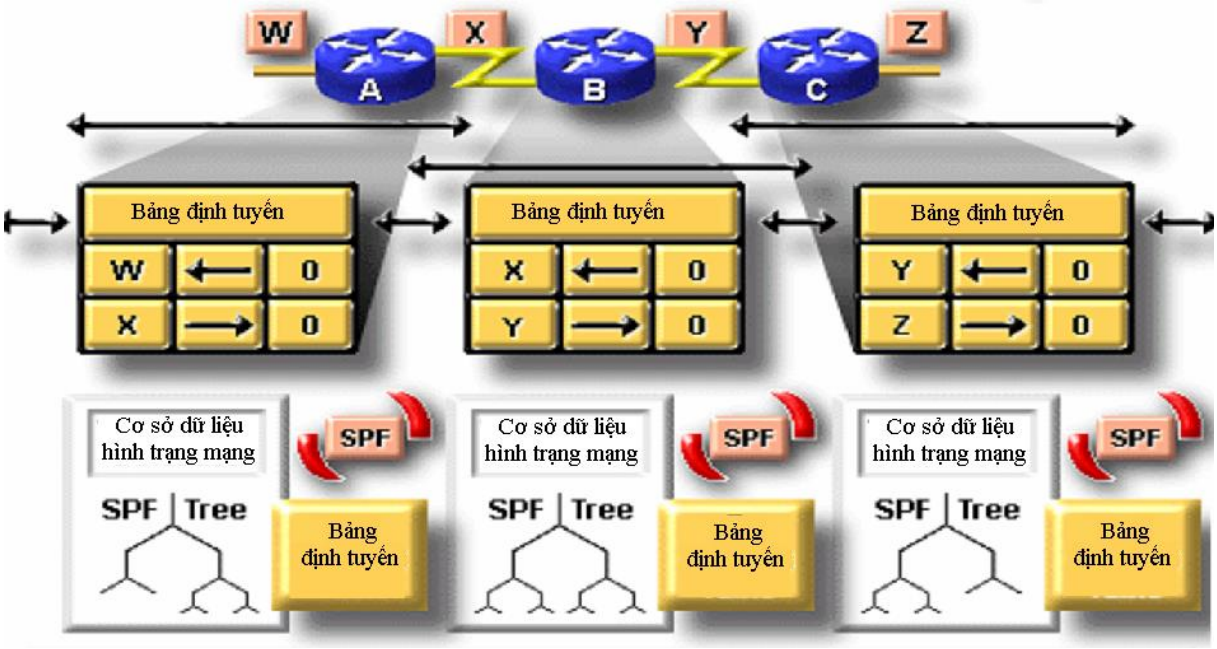
Để tìm đường đi có giá thấp nhất từ A đến F, phần lớn mọi người sẽ lần theo các thiết bị định tuyến từ A đến F qua nhiều con đường (có tất cả 12 tuyến đường khác nhau nối A và F) sau đó so sánh giá của mỗi tuyến đường. Quá trình như vậy chỉ có thể thực hiện được khi có đầy đủ thông tin về tất cả các nút và các cạnh trên đồ thị hay thuật toán đó gọi là thuật toán tìm đường tập trung. Thuật toán định tuyến toàn cục xác định đường đi với giá thấp nhất giữa nguồn và đích bằng cách sử dụng tất cả thông tin về tổng thể mạng. Đầu vào của thuật toán là tất cả các nút, cung và giá của các cung. Rõ ràng thiết bị định tuyến phải bằng một cách nào đó thu được các thông tin này trước khi bước vào giai đoạn tính toán thực sự. Thuật toán có thể được chạy tại một nơi (thuật toán định tuyến tập trung) hoặc chạy tại nhiều nơi. Tuy nhiên điểm phân biệt chính yếu là thuật toán định tuyến toàn cục phải có trước đầy đủ thông tin về đồ thị mạng. Trong thực tế, thuật toán như vậy được gọi là thuật toán trạng thái đường truyền (link state) vì thuật toán phải biết được giá của mỗi liên kết trên mạng.

Trong thuật toán phân tán, xác định đường đi có giá thấp nhất được thực hiện dần dần theo cách thức phân tán. Không nút nào có đầy đủ thông tin về giá của tất cả các liên kết trên mạng. Ban đầu mỗi nút chỉ biết về giá của các cung có nối trực tiếp với nó. Sau đó, thông qua các bước tính toán và trao đổi thông tin với các nút hàng xóm (hai nút được gọi là hàng xóm nếu giữa chúng có một đường kết nối vật lý trực tiếp, trong thuật ngữ đồ thị gọi là hai đỉnh kề nhau), nút dần dần xác định được đường đi có giá nhỏ nhất đến một tập hợp đích nào đó. Thuật

toán này là thuật toán vector khoảng cách (distance vector) vì mỗi nút không biết được đường đi cụ thể đến đích mà chỉ biết đến nút hàng xóm trên đường đến đích và tổng giá của đường đi đến đích.

7.2.1 Thuật toán định tuyến theo trạng thái đường truyền

Trong thuật toán trạng thái đường truyền, cấu trúc mạng và giá của tất cả các liên kết đều phải được xác định trước. Điều này được thực hiện bằng cách mỗi nút sẽ gửi thông báo quảng bá về định danh của mình và giá các cung liên kết trực tiếp đến nó tới tất cả các thiết bị khác trên mạng. Việc quảng bá rộng rãi trạng thái đường truyền có thể được thực hiện ngay khi nút không biết về đầy đủ các nút khác trên mạng. Ban đầu nút chỉ biết được thông tin về các hàng xóm của mình cũng như giá các cung đến các hàng xóm. Nhưng sau đó nó sẽ xác định được hình trạng của phần còn lại của mạng khi nhận những thông báo quảng bá từ các nút khác. kết quả của việc quảng bá trạng thái liên kết là tất cả các nút có thể đầy đủ thông tin về tổng thể mạng. Sau đó mỗi nút đều có thể chạy thuật toán trạng thái đường truyền và xác định đường đi có giá thấp nhất tới mọi nút.



Hình 7.5 Xây dựng bảng định tuyến dựa trên pháp trạng thái đường truyền

Thuật toán trạng thái đường truyền thường được sử dụng là thuật toán Dijkstra, đó là thuật toán xác định đường đi có giá thấp nhất từ một nút nguồn đến tất cả các nút khác trên mạng. Thuật toán Dijkstra có nhiều bước và sau k bước sẽ xác định được đường đi có giá thấp nhấp tới ít nút đích. Chúng ta định nghĩa một số ký hiệu sau:

c(i, j) : giá liên kết từ nút i đến nút j. Nếu nút i và nút j không có đường kết nối trực tiếp thì $c(i, j)=\infty$. Để đơn giản, chúng ta coi $c(i, j)= c(j, i)$.

D(v): giá hiện tại thấp nhất của tuyến đường đi từ nút nguồn đến nút v.

P(v) : nút phía trước nút v (hàng xóm của v) trên tuyến đường hiện có giá thấp nhất từ nguồn tới nút v.

N: tập hợp của các nút đã xác định được đường đi ngắn nhất tới.

Thuật toán Dijkstra gồm có bước khởi tạo cho vòng lặp, số các bước bằng tổng số nút trên mạng. Khi kết thúc, thuật toán sẽ xác định được đường đi ngắn nhất từ nút nguồn đến tất cả các nút khác trên mạng. Thuật toán Link state(LS):

```
Khởi tạo: N={A}
For (tất cả các nút v)
    if v kề A thì D(V)=c(A,v) else D(v)= ∞ Repeat
        Tìm w không ở trong N có D(w) nhỏ nhất
        Bổ sung w vào N
        cập nhật D(v) cho tất cả v kề với w và không
        nằm trong N: D(v)=min (D(v), D(w)+c(w,v))
        /* giá mới đến v khác cũ giá cũ đến v hoặc
        biết được  giá đường đi ngắn nhất
        đến w cộng với giá tớ w đến v */
    Until tất cả các nút nằm trong N
```

Bảng 7. 2 bảng trạng thái các bước cho mạng minh họa trên hình 5.4

Bước	N	D(b), p(B)	D(c), p(C)	D(D), p(D)	D(E), p(E)	D(F), p(F)
0	A	2, A	5, A	1, A	∞	∞
1	AD	2, A	4, D		2,D	∞
2	ADE	2, A	3, E			4, E
3	ADEB		3, E			4, E

Bước	N	D(b), p(B)	D(c), p(C)	D(D), p(D)	D(E), p(E)	D(F), p(F)
4	ADEBC					4, E
5	ADEBCF					

Xét đồ thị mạng trong hình 7.4 và tính đường đi có giá thấp nhất từ A đến tất cả các nút khác bảng 4.2 cho thấy các kết quả tính của thuật toán, mỗi dòng trong bảng ứng với trạng thái của thuật toán sau khi kết thúc một bước. Sau đây chúng ta sẽ phân tích một số bước đầu tiên:

- **Trong bước khởi tạo**, giá hiện tại thấp nhất của đường đi từ A đến các nút hàng xóm B, C và D tương ứng là 2, 5, và 1. Chúng ta có một chú ý nhỏ ở đây, giá đến C được đặt là 5 (ngay sau đây chúng ta sẽ thấy đây không phải là đường đi tốt nhất) vì đây là giá của đường nói trực tiếp từ A đến C. Giá

đến E và F được đặt là vô cùng vì giữa A và E, F không có đường kết nối trực tiếp.

- **Trong bước đầu tiên** chúng ta tìm kiếm trên những nút chưa được đưa vào tập N và xác định nút có giá đến thấp nhất. Đó là nút D với giá là 1 và do đó D được bổ sung vào N. Dòng 12 của thuật toán LS được thực hiện để cập nhật D(v) cho tất cả các nút v, kết quả nhận được được trình bày trong dòng thứ 2 (bước 1) trong bảng 4.2. Giá của đường đi đến B không đổi. Giá đường đi đến C (nhận giá trị 5 trong bước khởi tạo trước) qua D có giá trị nhỏ hơn là 4. Đây là đường ít hơn được chọn và nút phía trước của C trên đường đi ngắn nhất từ A sẽ là D. tương tự vậy, giá đường đi đến E (qua D) được tính là 2 và bảng được cập nhật tương ứng.
- **Trong bước thứ hai**, đường đi đến nút B và E đều có giá thấp nhất và chúng ta bổ sung E vào tập N (Bây giờ N chứa A, D và E). Giá đến các nút chưa nằm trong N (gồm B, C và F) được cập nhật trong dòng 12 của thuật toán LS, kết quả là dòng 3 của bằng 4.2..

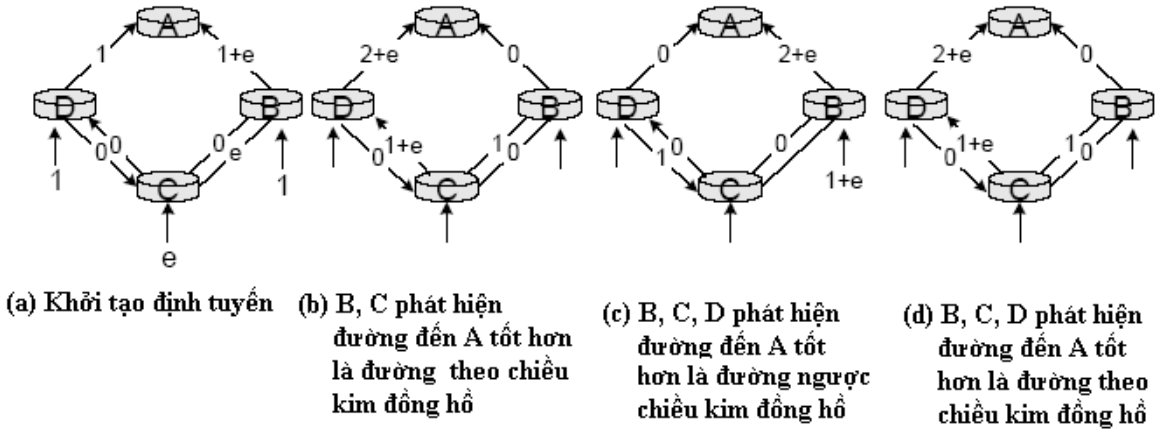
Khi thuật toán LS kết thúc, với mỗi nút chúng ta xác định được nút ngay trước nó trên tuyến đường có giá thấp nhất xuất phát từ nguồn. với mỗi nút phía trước chúng ta lại có nút phía trước nữa" Cuối cùng chúng ta xác định được toàn bộ đường đi từ nguồn đến tất cả các nút đích.

Độ phức tạp tính toán của thuật toán này bằng bao nhiêu? với n nút (không kể nút nguồn), để tìm đường đi có giá thấp nhất từ nguồn đến tất cả các đích, khối lượng tính toán là bao nhiêu trong trường hợp xấu nhất? Trong vòng lặp đầu tiên chúng ta cần kiểm tra qua tất cả n nút để xác định nút w có giá nhỏ nhất không nằm trong N; trong vòng lặp thứ hai, chúng ta cần kiểm tra n - 1 nút để xác định giá thấp nhất trong vòng lặp thứ ba là n - 2 nút … tổng số các nút mà chúng ta cần phải kiểm tra qua tất cả các bước là n(n+l)/2 và theo đó chúng ta có thể nói rằng thuật toán Link state có độ phức tạp là O(n2). (Thuật toán này có thể được cải tiến bằng cách sư dụng cầu trúc dữ liệu HEAP, độ phức tạp chỉ còn theo hàm logarit của n).

Trước khi kết thúc về thuật toán LS, chúng ta hãy xét ví dụ minh họa một cấu hình mạng giống như trên hình 5.5. Giá của mỗi liên kết (cung) bằng Tải hiện tại trên nó (như thế giá sẽ là độ trễ mà gói tin phải chịu). Trong trường hợp này giá không có tính chất đối xứng, nghĩa là c(A, B) chỉ bằng c(B, A) nếu tải trên cả hai hướng AB là như nhau. Giả sử hai nút B và D gửi một đơn vị dữ liệu, nút C gửi khối lượng dữ liệu là e tới A. Định tuyến ban đầu được minh họa trên hình 5.5(a), giá của mỗi cung ứng với tải trên cung đó.

Trong bước tiếp theo của thuật toán LS, nút C (với giá liên kết cơ bản đã được xác định trong hình 7.6a) nhận thấy đường đi đến A theo chiều kim đồng hồ có giá là 1, trong khi theo chiều ngược lại có giá là 1+e. Do đó, đường đi đến A có giá thấp nhất của C bây giờ là theo chiều kim đồng hồ. Tương tự, B nhận thấy đường đi đến A có giá thấp nhất mới cứng theo chiều kim đồng hồ, kết quả được trình bày trong hình 7.6b. Trong bước tiếp theo, nút B, C và D nhận thấy đường đi đến A ngược chiều kim đồng hồ có giá là 0 và tất cả các nút định lại

tuyến đường theo ngược chiều kim đồng hồ. Trong bước tiếp theo B, C và D lại thay đổi việc định tuyến theo chiều kim đồng hồ.



Hình 7.6 Xung đột định tuyến

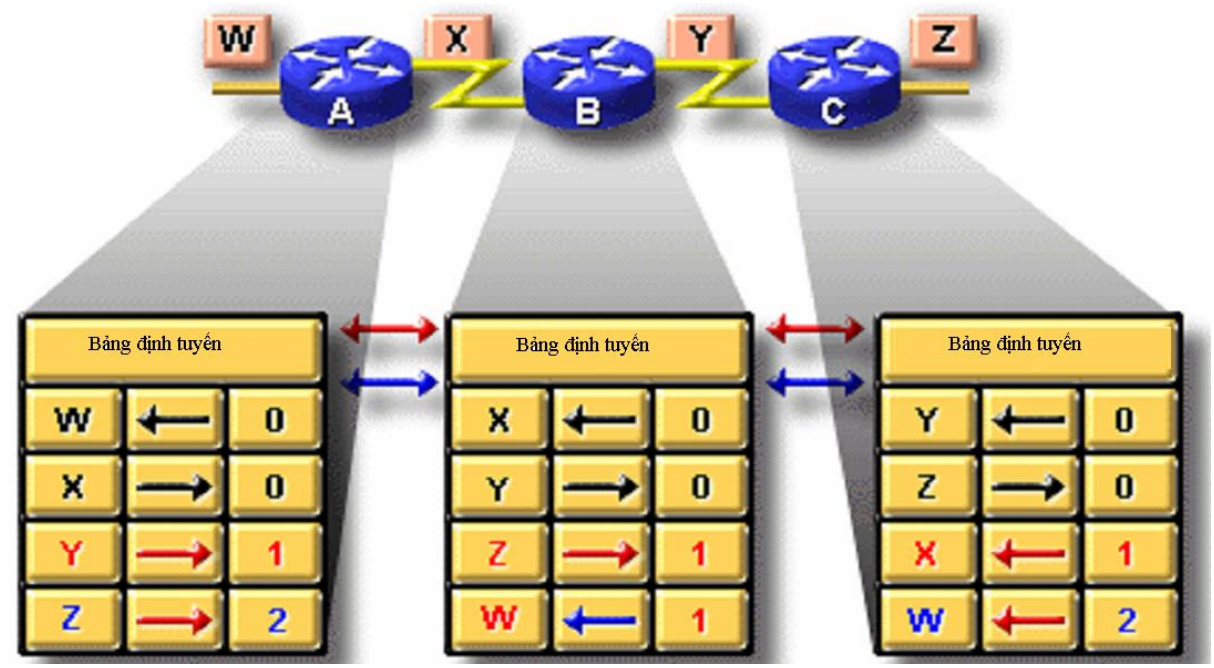
Lảm thế nào có thể ngăn ngừa sự dao động như trên (điều này luôn xuất hiện với những thuật toán chọn độ tắc nghẽn hoặc thời gian trễ làm giá cho đường truyền). Một giải pháp được đưa ra là định giá cho đường truyền không phụ thuộc vào tải trên đường đi, một giải pháp khó có khả năng chấp nhận vì mục tiêu của định tuyến là tránh những đường truyền hay tắc nghẽn (có độ trễ cao). Một giải pháp khác là làm thế nào để tất cả các thiết bị định tuyến không chạy thuật toán LS tại cùng một thời điểm. Giải pháp này dường như hợp lý hơn vì chúng ta hy vọng rằng thậm chí nếu các thiết bị định tuyến có chạy thuật toán LS với cùng chu kỳ, thì thuật toán sẽ đưa ra những kết quả khác nhau trên mỗi nút.

7.2.2 Thuật toán vector khoảng cách

Thuật toán vector khoảng cách (DV) là thuật toán lặp, không đồng bộ và phân tán. Thuật toán được xem là phân tán vì mỗi nút nhận thông tin từ những nút hàng xóm có đường kết nối trực tiếp đến nó, thực hiện các bước tính toán và gửi kết quả tính toán tới tất cả các nút hàng xóm. Tính lặp được thể hiện ở chỗ quá trình này được thực hiện liên tục cho đến khi không còn thông tin được trao đổi giữa các lặp hàng xóm. Thuật toán không đòi hỏi các nút phải dừng hoạt động trong khi trao đổi với những nút khác, đây chính là đặc điểm không đồng bộ.

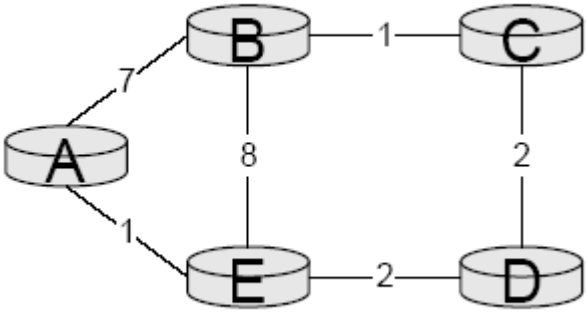
Cấu trúc dữ liệu chính trong thuật toán DV là bảng khoảng cách được đặt ở mỗi nút. Trong bảng khoảng cách, mỗi hàng ứng với một nút đích trên mạng và mỗi cột ứng với một nút hàng xóm có đường kết nối trực tiếp đến. Giả sử nút X muốn định tuyến đến đích Y qua nút hàng xóm Z. Trong bảng khoảng cách của nút X, DX(Y, Z) là tổng của giá đường liên kết trực tiếp giữa X và Z- c(x, Z) với giá đường đi bé nhất tớ Z đến Y. Đó là:

$$D^X(Y, Z) - c(X, Z) + \min_W\{D^Z(Y,w)\}$$



Hình 7.7 Xây dựng bảng định tuyến dựa trên vector khoảng cách

Biểu thức $\min_W$ trong đẳng thức trên được lấy trên tất cả các hàng xóm của Z (kể cả X, như chúng ta sẽ thấy dưới đây). Đẳng thức giúp chúng ta hình dung ý tưởng của thuật toán DV - mỗi nút phải biết được giá nhỏ nhất của đường đi tới tất cả các hàng xóm của nó đến bất kỳ đích nào. Và khi giá nhỏ nhất đến đích nào đó thay đổi, nút phải báo cho tất cả các hàng xóm biết.



	$D^E()$	Chi phí khi đi qua A	B	D
	A	1	14	5
Đích	B	7	8	5
	C	6	9	4
	D	4	11	2

Hình 7.8 Tính toán chi phí bằng khoảng cách

Xét kiến trúc mạng và bằng khoảng cách của nút E trên hình 7.8, đây là bảng khoảng cách của nút E sau khi thuật toán DV hội tụ. Hãy nhìn vào dòng đầu tiên với đích đến là A. Rõ ràng giá đường kết nối trực tiếp đến A từ E phải là 1, do đó $D^E(A, A) = 1$. $D^E(A, D)$ là giá đường đi từ E đến A qua D bằng tổng giá của đường đi từ E đến D (2) cộng với giá đường đi nhỏ nhất từ D đến E. Giá thấp nhất của đường đi từ D đến A là 3 (đường đi qua E), do vậy giá thấp

nhất từ E đến A qua D là 5. Tương tự trường trong bảng khoảng cách với đường đi qua B: $D^E(A, B) = 14$. Các vòng tròn trong bảng khoảng cách là giá nhỏ nhất của đường đi đến các đích. Cột ứng với vòng tròn xác định nút tiếp theo trên đường đi đến đích có giá thấp nhất. từ đó có thể dễ dàng xây dựng bảng định tuyến cho mỗi nút.

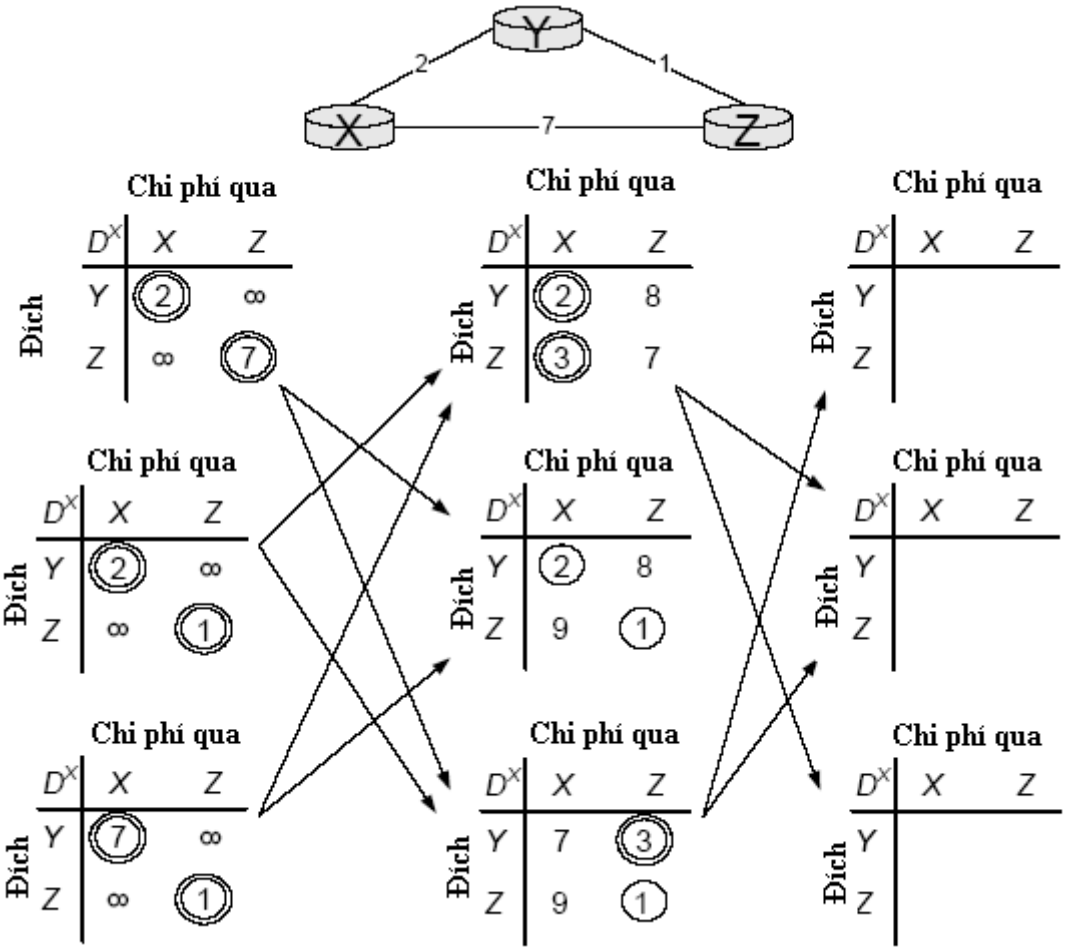
Thuật toán distance vector(DV)

tại mỗi nút X:

```
Khởi tạo:
for (tất cả các nút kể với v)
  D^x(*, v) = ∞/* dấu * là để chỉ cho tất cả các hàng*/
  D^x(v, v) = c(x, v)
  for (tất cả các đích Y)
    gửi minW D(Y, w) đến mỗi hàng xóm /* w chạy trong tập các hàng xóm của X*/
    loop
        Đợi cho đến  khi (thấy giá liên kết đến  hàng xóm V thay đổi          hoặc nhận
        được sự cập nhật của hàng xóm V)
        if (c(x, V) thay đổi một lượng d)/* thay đổi giá của tất cả các đường đi đến đích
        qua v bằng, d có thể dương hoặc âm*/
        for (tất cả các đích y: DX(y, V) = DX(y, V)  + d else if (nhận được cập nhật từ V
        đến Y)
        /* đường đi ngắn nhất từ V đến nút nào đó Y thay đổi*/
        /* V đã gửi giá trị mới của minW DW(Y, w)*/
        /* gọi giá trị mới nhận được là newval*/
        for (đích y): D^x y, V) = c(x, V)  + newval
        if (chúng ta có minW DX(y, w) mới cho đích Y nào đó) gửi giá trị minW D^x (y, w)
        mới đến tất cả các hàng xóm.
Forever
```

Mỗi nút sẽ chỉ biết thông tin về giá đường liên kết tới nút hàng xóm cũng như thông tin nó nhận được từ những hàng xóm này. Thuật toán DV còn được gọi là thuật toán Bellman - Ford. Nó được áp dụng trong nhiều giao thức định tuyến trong thực tế như: Internet BGP, ISO IDRP... Điểm cơ bản là việc cập nhật bảng khoảng cách khi nhận được sự thay đổi về giá của liên kết đến hàng xóm hoặc nhận được thông tin cập nhật tới hàng xóm của nó và gửi cập nhật đến tất cả các hàng xóm nếu đường đi có giá nhỏ nhất đến đích nào đó bị thay đổi.

Hình 7.9 minh họa hoạt động của thuật toán DV cho mạng đơn giản gồm có 3 nút. Hoạt động của thuật toán được thực hiện một cách đồng bộ: tất cả các nút đồng thời nhận được thông điệp từ hàng xóm của chúng, tính toán bảng khoảng cách mới và báo cho các hàng xóm về sự thay đổi giá đường đi ngắn nhất. Các ô có khoanh tròn một vòng trong hình vẽ ứng vời giá nhỏ nhất hiện tại đến đích nào đó nằm trong hàng tương ứng. Khoanh hai vòng tròn biểu diễn giá nhỏ nhất mới được xác định, khi đó các thông tin cập nhật sẽ được gửi đến các nút hàng xóm, ứng với mũi tên giữa các cột. Cột ngoài cùng bên trái là các bảng khoảng cách của nút X, Y và Z sau bước khởi tạo. Xét trường hợp nút X tính lại bảng khoảng cách sau khi nhận được thông tin cập nhật từ nút Y và Z.



Hình 7.9 Hoạt động của thuật toán vector khoảng cách

Khi nhận được cập nhật từ Y và Z, X thực hiện dòng 21 của thuật toán DV:

$D^X(Y,X)=c(X,Z)+\min_W D^Z(Y,w)$

$=7+1$

$=8$

$D^X(Z,Y) =c(X,Y)+\min_W D^Y(Z,w)$

$=2+1$

$=3$

X biết giá trị $\min_W D^Z(Y, w)$ và $\min_W D^Y(Z, w)$ vì nút Z và Y gửi những giá trị này đến X (và X cũng nhận được theo dòng 10 của thuật toán DV). Việc tính lại bảng khoảng cách của Y và Z ở cột giữa trong hình 7.7 được thực hiện tương tự. Giá trị $D^X(Z, Y) = 3$ có nghĩa là giá nhỏ nhất tớ X đến Z giảm từ 7 xuống 3. Do đó, X gửi cập nhật đến Y và Z để thông báo cho chúng giá thấp nhất mới đến Z. Chú ý X không cần cập nhật cho Y, Z về giá của nó đến Y vì giá trị này không bị thay đổi. Khi Y tính lại bảng khoảng cách không phát hiện

ra thay đổi, đo đó Y không gửi cập nhật đến X và Z.

Quá trình nhận cập nhật từ hàng xóm, tính lại bảng khoảng cách và cập nhật các thay đổi đến hàng xóm được thực hiện cho đến khi không còn bản tin nào được trao đổi. Trong trường hợp này vì không có thông tin cập nhật được gửi nên các nút không phải tính lại bảng khoảng cách và thuật toán ở trạng thái không hoạt động: tất cả các nút ở trạng thái đợi trong dòng 9 của thuật toán DV. Thuật toán DV sẽ ở trong trạng thái không hoạt động cho đến khi giá của một nên kết nào đó thay đổi.

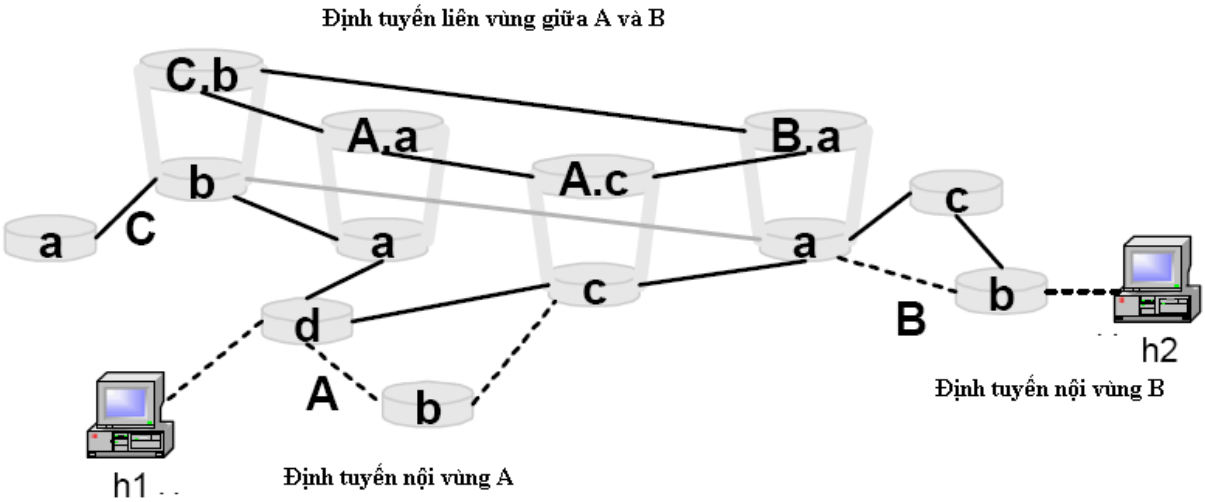
7.3 Định tuyến phân cấp

Mạng bao gồm tập hợp các thiết bị định tuyến được kết nối với nhau, sẽ rất đơn giản nếu chúng sử dụng cùng một thuật toán định tuyến để xác định đường đi trên toàn bộ hệ thống mạng. Khi số lượng các thiết bị định tuyến lớn, khối lượng thông tin phải tính toán, lưu trữ và trao đổi giữa các bảng chứa thông tin định tuyến trên mỗi thiết bị định tuyến trở nên rất lớn. Mạng Internet ngày nay bao gồm hàng triệu thiết bị định tuyến liên kết với nhau và hàng tỉ máy tính. Lưu trữ thông tin về tất cả các máy tính cũng như các thiết bị định tuyến đòi hỏi một lượng bộ nhớ khổng lồ. Các thông tin trao đổi cập nhật giữa các thiết bị định tuyến có thể sẽ tốn toàn bộ băng thông của đường truyền. Thuật toán DV trên hàng triệu thiết bị định tuyến chắc chắn sẽ không bao giờ hội tụ. Do đó cần phải giảm độ phức tạp trong việc xác định đường đi trên một mạng lớn như Internet. Đối với vấn đề quản trị cũng nãy sinh nhiều vấn đề khó có thể dung hòa được, ví dụ không thể ẩn cấu trúc mạng bên trong của tổ chức với bên ngoài.

Vấn đề trên có thể giải quyết bằng cách nhóm các thiết bị định tuyến thành các vùng tự quản (AS - Autonomous System). Các thiết bị định tuyến trong một AS sử dụng cùng một thuật toán định tuyến (ví dụ như thuật toán LS hay DV). Thuật toán định tuyến chạy trong mỗi vùng AS được gọi là giao thức định tuyến nội vùng. Nhu cầu tất yếu cần phải kết nối các AS với nhau, do đó một số thiết bị định tuyến trong AS phải có thêm nhiệm vụ định tuyến gói tin ra bên ngoài. Các thiết bị định tuyến định tuyến gói tin ra phía ngoài như vậy được gọi là thiết bị định tuyến cầu nối (gateway router). Để định tuyến gói tin đi giữa các AS (có thể phải đi qua nhiều AS trên toàn bộ tuyến đường các thiết bị định tuyến liên vùng phải biết cách xác định đường đi giữa các AS, thuật toán định tuyến được sử dụng tại các thiết bị định tuyến đó gọi là giao thức định tuyến liên vùng.

Trên hình 7.10 có 3 có ba vùng tự trị: A, B và C. Vùng A có 4 thiết bị định tuyến: A.a, A.b, A.c, và A.d, sử dụng cùng một thuật toán intra-AS của vùng A, cả bốn thiết bị định tuyến này đều có đầy đủ các thông tin về nhau cũng như các liên kết trong vùng A. Tương tự, vùng B có 3 thiết bị định tuyến và vùng C có 2 thiết bị định tuyến. Các thuật toán định tuyến trong các miền A, B, C không nhất thiết phải giống nhau. Các thiết bị định tuyến cầu nối là A.a, A.c, B.a, và C.b phải chạy các thuật toán định tuyến nội vùng để trao đổi với các thiết bị định tuyến trong miền chúng còn phải sử dụng thuật toán liên vùng để định tuyến giữa các vùng tự trị. Về mặt hình trạng, chúng sử dụng giao thức liên vùng sử dụng kết nối vật lý thực sự (giữa A.c và B.a), có thể là đường ảo (ví dụ giữa A.c

và A.a), A.c sử dụng cả giao thức định tuyến nội vùng để định tuyến với A.b, A.d và liên vùng để định tuyến với B.a.



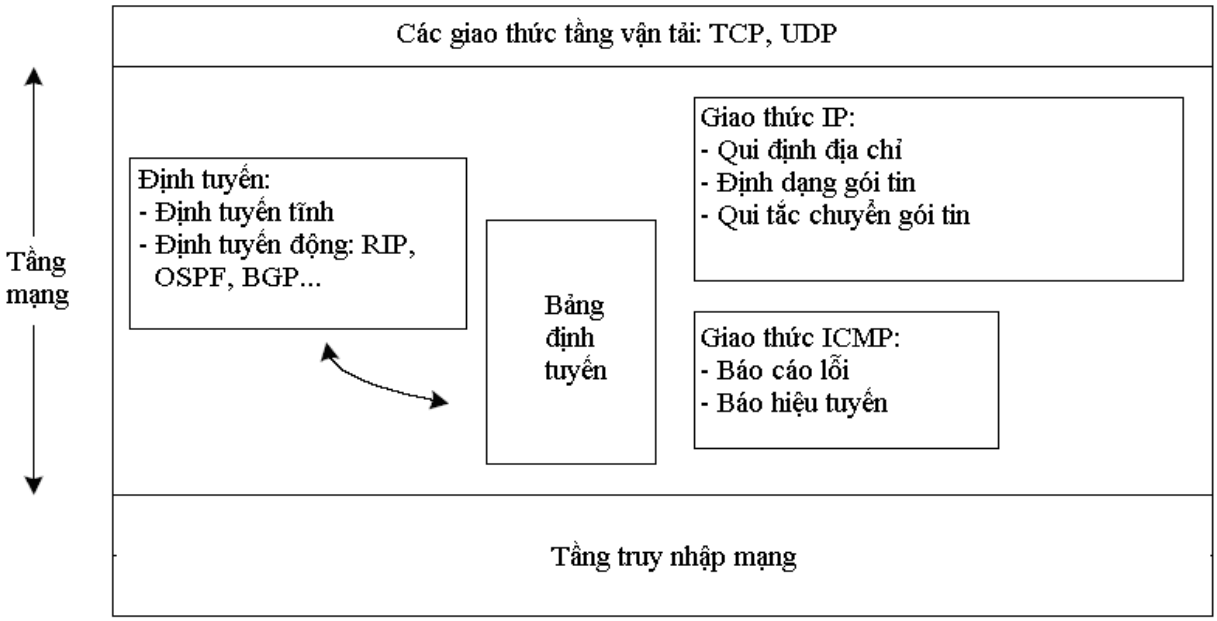
Hình 7.10 Định tuyến liên vùng

Giả sử máy tính h1 (nối với thiết bị định tuyến A.d) cần gửi gói tin tới máy tính h2 trong vùng tự trị B. Bảng định tuyến tại A.d cho biết, thiết bị định tuyến A.e chịu trách nhiệm gửi gói tin ra bên ngoài vùng tự trị. Gói tin từ A.d tới thiết bị định tuyến A.c sử dụng giao thức định tuyến nội vùng của A. Một điểm quan trọng cần chú ý là thiết bị định tuyến A.d không cần biết cấu trúc nội tại trong miền B và C và cũng như hình trạng giữa ba miền A, B và C. Thiết bị định tuyến A.c nhận gói tin, xác định đích của gói tin đó nằm ngoài miền A, bảng định tuyến liên vùng sẽ xác định rằng để gửi tới miền B thì phải chuyển tới B.a. Khi gói tin tới B.a, giao thức liên vùng xác định rằng gói tin này tới máy tính nào đó trong miền B và chuyển cho giao thức nội vùng của B. Cuối cùng thiết bị định tuyến B.a chuyển gói tin đó tới máy tính đích h2 sử dụng giao thức nội vùng của B.

7.4 Giao thức IP

Tầng mạng của Internet sử dụng dịch vụ chuyển mạch gói. Khi nhận được một đoạn tin từ tầng vận tải, tầng mạng đặt đoạn tin trong gói dữ liệu IP với các trường địa chỉ gửi, địa chỉ nhận và gửi gói tin này tới thiết bị định tuyến đầu tiên trên đường đi tới địa chỉ đích. Có một sự liên hệ tương đối về cách hoạt động của tầng mạng với hệ thống Bưu điện: một người viết một lá thư, đặt lá thư vào phong bì ghi địa chỉ người nhận và thả phong bì thư vào hộp thư. Tầng mạng của Internet và cũng như bưu điện đều không có bất kỳ một sự liên hệ trước nào với bên nhận trước khi chuyển thư.

Tầng mạng của Internet và bưu điện sẽ đều chỉ cung cấp một dịch vụ kiểu "Cố gắng hết sức", nghĩa là không có bất kỳ đảm bảo nào về gói tin sẽ đến đích, đến trong một khoảng thời gian xác định trước hay đúng thứ tự.



Hình 7.11 Tầng mạng trong Internet

Trong hình 7.11, tầng mạng trong kiểu mạng chuyển mạch gói giống như mạng Internet có 3 thành phần chính:

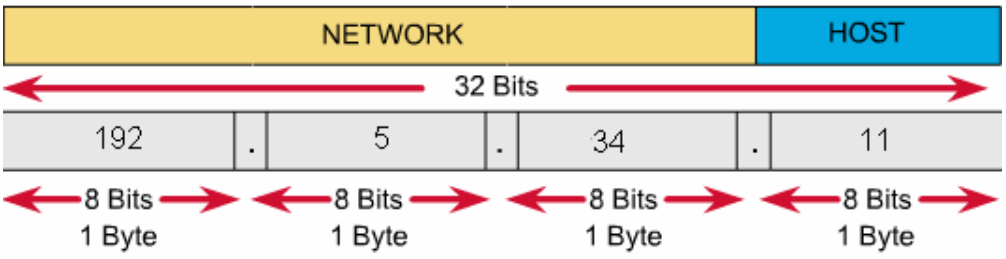
- Giao thức mạng: xác định địa chỉ tầng mạng; ý nghĩa của các trường trong và gói dữ liệu - PDU của tầng mạng, các hành động của các thiết bị định tuyến và thiết bị đầu cuối khi nhận được các gói tin. Giao thức mạng trong Internet gọi là giao thức liên mạng hay ngắn gọn gọi là giao thức IP. Hiện nay có hai phiên bản giao thức IP được sử dụng: IPV4 và IPV6.
- Bộ phận xác định đường đi: xác định tuyến đường của gói tin trên đường đi tới đích.
- Chức năng quản lý: Giao thức bản tin điều khiển Internet - ICMP.

7.4.1 Địa chỉ IPv4

Một máy tính thường có một đường kết nối duy nhất vào hệ thống mạng, khi thực thể IP trong máy máy tính muốn gửi gói tin, nó sẽ chuyên qua kết nối này. Nằm giữa máy tính và đường kết nối vật lý là một giao diện ghép nối. Thiết bị định tuyến khác với máy tính, chức năng của thiết bị định tuyến là chuyển một gói tin từ mạng này sang mạng khác (từ đường truyền này sang đường truyền khác, từ cổng này sang cổng khác). Thiết bị định tuyến có thể có nối đến nhiều kênh truyền và bộ phận nằm giữa thiết bị định tuyến và một kênh truyền cũng được gọi là giao diện. Như vậy thiết bị định tuyến có nhiều giao diện, mỗi giao diện ứng với một đường truyền vật lý. Vì tất cả các máy tính và thiết bị định tuyến đều phải có khả năng gửi và nhận gói tin IP nên mỗi giao diện phải có một địa chỉ, do đó địa chỉ IP ứng với một giao diện chứ không phải với máy tính hay thiết bị định tuyến.

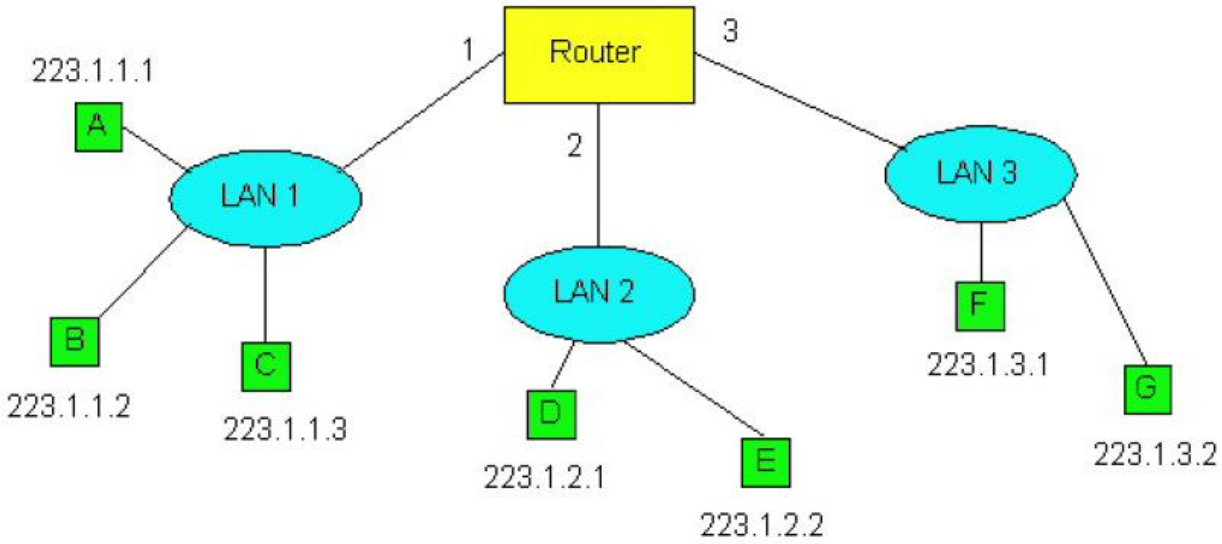
Địa chỉ IP có độ dài 32 bit (4 byte) và như vậy không gian địa chỉ có 2^{32} địa chỉ IP. Địa chỉ IP được viết theo ký pháp dấu chấm thập phân. Mỗi byte của địa chỉ được viết dưới dạng thập phân và phân cách với các byte khác bằng ký

tự chấm (.). Xét địa chỉ IP 193.32.216.9, giá trị 193 là số thập phân ứng với nhóm 8 bit đầu của địa chỉ, 32 là số thập phân ứng với nhóm 8 bit thứ hai của địa chỉ, bởi vậy địa chỉ 193.32.216.9 trong ký pháp nhị phân sẽ là 110000001 00100000 11011000 00001001.



Hình 7.12 Cấu trúc địa chỉ IP

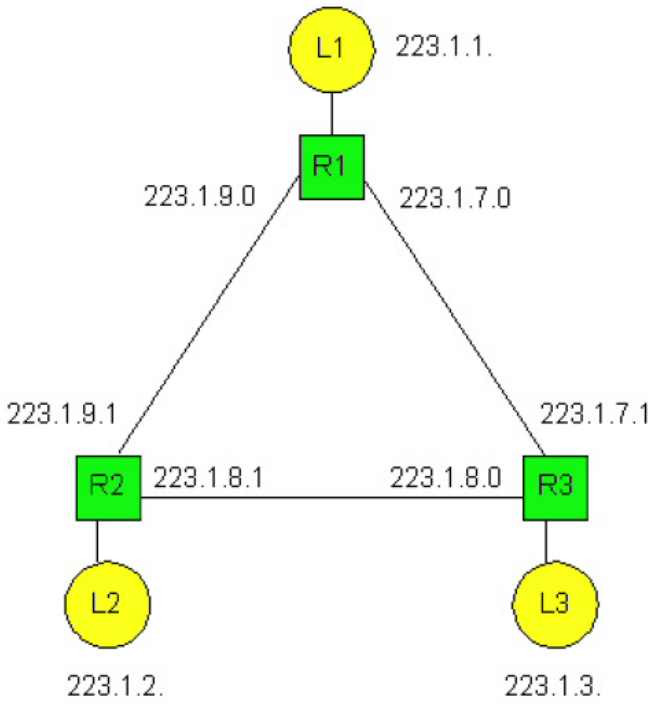
Mỗi giao diện ghép nối của máy tính hay thiết bị định tuyến trên mạng toàn cầu Internet phải có một địa chỉ IP xác định duy nhất. Những địa chỉ đó không thể chọn một cách tùy ý mà phụ thuộc vào mạng mà nó kết nối vào. Trong ngữ cảnh này, thuật ngữ "mạng" không có ý là một cấu trúc tổng thể gồm các máy tính, router và các liên kết giữa chúng. Hiện tại thuật ngữ này được sử dụng với ý nghĩa cụ thể hơn, có quan hệ chặt chẽ với địa chỉ IP.



Hình 7.13 Thiết bị định tuyến và các giao diện

Hình 7.13 minh họa một thiết bị định tuyến có 3 giao diện được sử dụng để kết nối 7 máy tính. Quan sát địa chỉ IP của mỗi giao diện ứng với mỗi máy tính và thiết bị định tuyến. Giao diện của 3 máy tính ở phần trên bên trái và thiết bị định tuyến nối với chúng đều có địa chỉ IP là 223.1.1.xxx.: nghĩa là 24 bit đầu của địa chỉ IP giống nhau. Chúng cũng được kết nối với nhau bằng một đường kết nối vật lý duy nhất (trong trường hợp này là môi trường quảng bá sử dụng cáp Ethernet) mà không cần qua bất kỳ thiết bị định tuyến trung gian nào. Các giao diện của những máy tính này và giao diện phía trên bên trái của router tạo nên mạng IP (IP network) hay đơn giản là mạng. 24 bit địa chỉ đầu giống

nhau là phần mạng trong cấu trúc địa chỉ IP, 8 bit còn lại là phần máy tính (host) của địa chỉ IP. Chính mạng này cũng có một địa chỉ là 22.3.1.1.0/24 trong đó kí hiệu /24 là mặt nạ mạng (network mask) với ý nghĩa 24 bit đầu tiên của địa chỉ 32 bit xác định địa chỉ mạng. Những bit này cũng được xem là tiền tố mạng (network prefix). mạng 22.3.1.1.0/24 gồm giao diện của 3 máy tính (223.1.1.1; 223.1.1.2; 223.1.1.3) và một giao diện của thiết bị định tuyến (223.1.1.4). Bất kỳ máy tính nào nối với mạng 223.2.2.0/24 đều phải có địa chỉ dưới dạng 223.1.1.xxx.



Hình 7.14 Kết nối liên mạng

Định nghĩa IP về mạng không chỉ với phân đoạn mạng nối nhiều máy tính với một thiết bị định tuyến. Trên hình 7.14 ba thiết bị định tuyến đôi một nối với nhau qua các đường liên kết điểm tới điểm (point-to-point). Mỗi thiết bị định tuyến có ba giao diện, hai giao diện kết nối tới hai thiết bị định tuyến kia và một giao diện dành cho kết nối quảng bá với các máy tính. Có bao nhiêu mạng? Ba mạng 223.1.1.0/24, 223.1.2.0/24 và 223.1.3.0/24, mạng 223.1.9.0/24 cho hai giao diện nối router R1 và R2, mạng 223.1.8.0/24 cho hai giao diện nối router R2 và R3 và mạng 223.1.7.0/24 ứng với hai giao điện nối router R3 và R1.

Với một hệ thống liên mạng gồm nhiều thiết bị định tuyến và máy tính, chúng ta có thể sử dụng một công thức để xác định các mạng trong hệ thống. Chúng ta loại bỏ tất cả giao diện của các máy tính và thiết bị định tuyến. Khi đó sẽ tạo ra các mạng cô lập, mỗi mạng cô lập đó được gọi là một mạng. Áp dụng cách thức này trong ví dụ trên hình 7.14, có 6 mạng IP tách biệt. Mạng Internet gồm hàng triệu hệ thống mạng như vậy, địa chỉ mạng có vai trò then chốt trong việc định tuyến trên Intemet.

Kiến trúc địa chỉ Internet đầu tiên đưa ra 5 lớp địa chỉ minh họa trên hình 7.15. Lớp địa chỉ thứ năm, bắt đầu bằng 11110 được dự trữ cho việc sử dụng sau này. Với lớp địa chỉ A, 8 bit đầu là địa chỉ mạng và 24 bit cuối là địa chỉ của máy tính trong mạng đó. Do đó có 127 mạng lớp A (bit đầu tiên trong 8 bit địa chỉ mạng luôn nhận giá trị 0), mỗi mạng lớp A lại có thể có 2^{24} địa chỉ. Lớp B có 2^{14} mạng, mỗi mạng lại có 2^{16} địa chỉ. Lớp địa chỉ C dùng 24 bit làm địa chỉ mạng và chỉ có 8 bit làm địa chỉ máy. Lớp D dự trữ làm địa chỉ nhóm và lớp E dùng để nghiên cứu.

Lớp địa chỉ	Phạm vi
Class A	1-126 (00000001-01111110) *
Class B	128-191 (10000000-10111111)
Class C	192-223 (11000000-11011111)
Class D	224-239 (11100000-11101111)
Class E	240-255 (11110000-11111111)

Hình 7.15 Các lớp địa chỉ IP

Năm lớp địa chỉ trình bày trên hiện tại không còn được áp dụng trong kiến trúc địa chỉ IP nữa. Điều kiện phần mạng của địa chỉ lớp có độ dài là một, hai hoặc ba byte không hợp lý khi số lượng các tổ chức với mạng cỡ nhỏ hay trung bình ngày càng tăng. một mạng lớp C (/24) chỉ có thể có 2^8 - 2 = 254 máy tính (2 trong số 2^8 địa chỉ được dự trữ cho một số mục đích khác) - một số lượng quá nhỏ với nhiều tổ chức. Tuy nhiên một mạng loại B (/16) có thể có tới 65.634 máy tính lại là quá lớn. một tổ chức với 2000 máy tính phải sử dụng địa chỉ lớp B (/16). với kiểu gán địa chỉ như vậy thì không gian địa chỉ lớp B sẽ nhanh chóng bị cạn kiệt và không gian địa chỉ không được sử dụng hiệu quả. Ví dụ, tổ chức đó sử dụng địa chỉ lớp B cho 2000 máy tính và có khoảng 63000 địa chỉ còn lại bị lãng phí trong khi có thể phân phối cho các tổ chức khác.

Năm 1993, IETF chuẩn hóa Định tuyến liên vùng không phân lớp (Classless interdomain routing - CLDR) [RFC1519] . Với địa chỉ mạng không phân lớp, phần mạng của địa chỉ IP có thể có độ dài tùy ý không nhất thiết phải là 8, 16 hay 24 bit. Khuôn dạng của một địa chỉ không phân lớp là a.b.c.d/x, trong đó x là số lượng bit dùng để làm địa chỉ mạng. Trong ví dụ trước, tổ chức cần không gian địa chỉ cho 2000 máy tính chỉ cần một không gian 2.048 địa chỉ dưới dạng a.b.c.d/21, cho phép khoảng 63000 địa chỉ còn lại được phân phối cho các tổ chức khác. Trong trương hợp này, 21 bit đầu tiên xác định địa chỉ mạng của tổ chức và tất cả địa chỉ lớp của máy tính trong tổ chức đều có địa chỉ mạng giống nhau, 11 bit còn lại xác định cụ thể máy tính nào trong tổ chức. Trên thực tế, có thể tiếp tục chia 11 bit đó để tạo ra các mạng con bên trong mạng a.b.c.d/21.

7.4.1.1 Vấn đề địa chỉ và định tuyến

Mỗi gói tin IP có có trường địa chỉ gửi và trường địa chỉ nhận. Máy tính gửi sẽ điền vào trường địa chỉ gửi 32 bit địa chỉ IP của mình và điền vào trường địa chỉ nhận 32 bit địa chỉ IP của máy tính nhận (giống như các trường FROM và TO trên phong bì thư). Trường dữ liệu của gói tin thường là các đoạn dữ liệu của tầng vận tải. Sau khi máy tính tạo ra gói tin, tầng mạng làm thế nào để gửi gói tin từ máy tính nguồn tới máy tính đích? Câu trả lời phụ thuộc vào việc máy tính nguồn và máy tính đích có nằm trong cùng một mạng hay không. Giả sử máy tính A muốn gửi gói tin IP tới máy tính B nằm trong cùng một mạng 223.1.1.0/24 với A. Điều này được thực hiện như sau: đầu tiên thực thể IP trong máy A dò trong bảng định tuyến cục bộ của nó (xem hình 7.15) và tìm thấy hàng 223.1.1.0/24 trùng với các bit đầu (địa chỉ mạng) trong địa chỉ IP của máy B. bảng định tuyến chỉ ra rằng số lượng các thiết bị trung gian để đến mạng 223.1.1.0 là 1, nghĩa là B nằm trong cùng một mạng với A. Do đó máy A có thể gửi trực tiếp đến B mà không cần qua các thiết bị định tuyến trung gian. Sau đó máy A chuyển gói dữ liệu IP cho tầng liên kết dữ liệu để chuyển gói dữ liệu đó đến B.

destination network	next router	number of hops to destination
223.1.1.	-	1
223.1.2.	223.1.1.4	2
223.1.3.	223.1.1.4	2

Hình 7.16 Bảng định tuyến trên máy tính

Trường hợp máy A muốn gửi gói dữ liệu tới một máy nằm trong mạng khác, chẳng hạn là E. Máy A dò trên bảng định tuyến và tìm thấy 223.1.1.0/24 có địa chỉ mạng trùng với phần mạng trong địa chỉ IP của E. Vì số lượng các thiết bị trung gian là 2, nên máy A biết máy đích nằm trên mạng khác và do đó sẽ phải chuyển qua các thiết bị định tuyến trung gian. Ngoài ra bảng định tuyến tại A cũng cho biết để gửi từ E, đầu tiên máy A phải gửi gói dữ liệu tới địa chỉ IP 223.1.1.4 là địa chỉ IP của giao diện thiết bị định tuyến kết nối vào cùng một mạng với A. Thực thể IP trong máy A chuyển gói dữ liệu xuống tầng liên kết dữ liệu và yêu cầu chuyển tới địa chỉ IP 223.1.1.4. Mặc dù gói dữ liệu IP được gửi từ giao diện của thiết bị định tuyến (qua tầng liên kết dữ liệu), địa chỉ đích trong gói dữ liệu vẫn là địa chỉ đích cuối cùng (địa chỉ của E) chứ không phải là địa chỉ thiết bị định tuyến trung gian.

Khi gói dữ liệu tới thiết bị định tuyến thì công việc của nó là chuyển gói dữ liệu hướng tới đích cuối cùng. Như vậy gói dữ liệu phải được chuyển đến giao diện của thiết bị định tuyến có địa chỉ IP là 223.1.2.9. Bởi vì số lượng các thiết bị trung gian tới đích là 1, nên router biết rằng đích (E) nằm trên cùng một

mạng với giao diện ứng với địa chỉ 223.1.2.9, do đó router chuyển gói dữ liệu tới giao diện (cổng) này, và sau đó gói dữ liệu được chuyển trực tiếp tới E.

destination network	next router	number of hops to destination	interface
223.1.1.	-	1	1
223.1.2.	-	1	2
223.1.3.	-	1	3

Hình 7.17 Bảng định tuyến trên thiết bị định tuyến

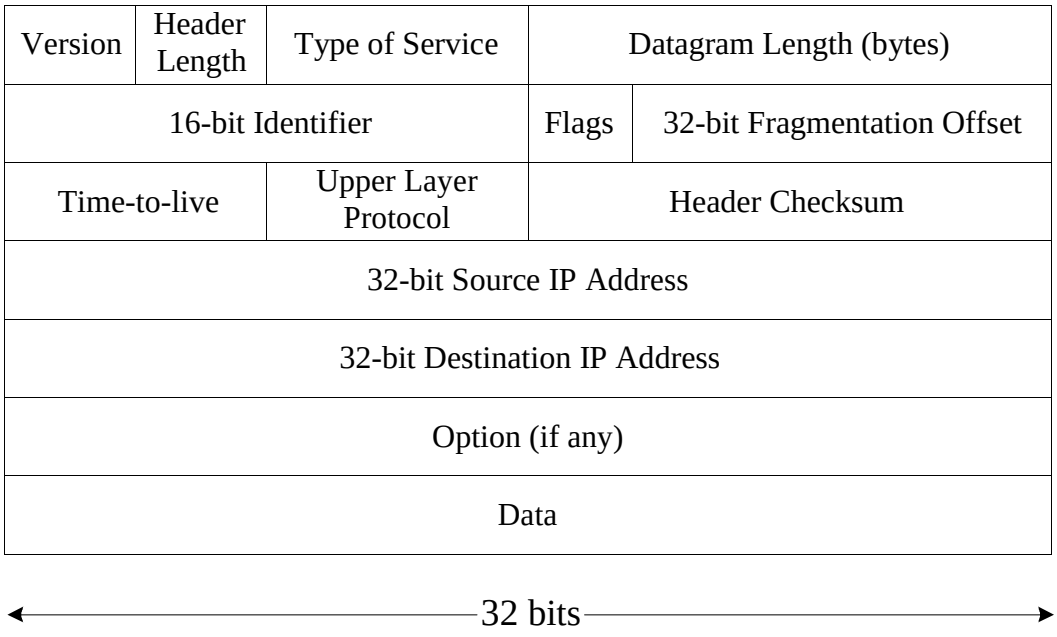
Trong hình 7.17, các hàng trong cột "next router" là rỗng vì các mạng (223.1.1.0/24, 223.1.2.0/24, và 223.1.3.0/24) được kết nối trực tiếp với thiết bị định tuyến. Trong trường hợp này, chúng không cần phải đi qua các thiết bị định tuyến trung gian để đến đích. Tuy nhiên nếu máy A và máy E cách nhau 2 thiết bị định tuyến thì trong trong bảng định tuyến của thiết bị định tuyến đầu tiên trên tuyến đường từ A tới E, dòng tương ứng với đích E sẽ chỉ ra phải qua 2 chặng nữa mới tới được đích và phải chỉ ra địa chỉ IP của thiết bị định tuyến thứ hai trên tuyến đường. Thiết bị định tuyến đầu tiên sẽ chuyển gói dữ liệu tới thiết bị định tuyến thứ hai nhờ vào giao thức của tầng liên kết dữ liệu giữa chúng. Sau đó thiết bị định tuyến thứ hai sẽ chuyển gói dữ liệu tới máy đích nhờ giao thức tầng liên kết dữ liệu giữa thiết bị định tuyến thứ hai và máy đích. Định tuyến cho gói dữ liệu trên mạng Internet cũng tương tự như hệ thống phân phát thư của Bưu điện, mỗi lá thư phải đi qua nhiều bưu cục mới đến được người nhận. Bảng định tuyến trong các thiết bị định tuyến đóng một vai trò then chốt trong việc định tuyến gói tin qua mạng Internet.

7.4.1.2 Khuôn dạng gói dữ liệu IP

Hình 7.18 minh họa khuôn dạng gói dữ liệu IP, các trường khoá trong gói dữ liệu IPv4 là như sau:

- Phiên bản (version): Trường 4 bit này xác định phiên bản giao thức IP của gói dữ liệu. Qua trường phiên bản, router mới xác định được ý nghĩa các trường còn lại trong gói dữ liệu IP. Các phiên bản IP khác nhau sử dụng các khuôn dạng dữ liệu khác nhau.
- Độ dài tiêu đề (Header length): Gói dữ liệu IPv4 có thể có nhiều trường mang tính lựa chọn (đặt trong tiêu đề gói dữ liệu IPv4). 4 bit này được dùng để xác định vị trí bắt đầu của dữ liệu thực sự trong gói dữ liệu IP. Tuy nhiên phần lớn gói dữ liệu IP không chứa các trường Lựa chọn nên tiêu đề của gói dữ liệu thường cố định là 20 byte
- Kiểu dịch vụ (Type of service - TOS): Trường kiểu dịch vụ (TOS) giúp phân biệt các kiểu khác nhau của gói dữ liệu IP, để tớ đó có thể xử lý theo những cách khác nhau. Ví dụ khi mạng quá tải, cần phân biệt được gói dữ liệu chứa

thông tin kiểm soát mạng (ICMP) với gói dữ liệu thực sự (bản tin HTML) hay giữa datagram chứa dữ liệu thời gian thực (ứng dụng điện thoại qua Internet) với datagram không chứa dữ liệu thời gian thực (ứng dụng FTP). gần đây Cisco (công ty chiếm thị phần router lớn nhất) đã sử dụng 3 bit đầu tiên của trường TOS để định nghĩa các mức dịch vụ khác nhau mà router có thể cung cấp. Các mức dịch vụ cụ thể được người quản trị router thiết lập theo những tiêu chí của tổ chức.



Hình 7.18 Khuôn dạng gói dữ liệu IP

- Độ dài gói dữ liệu (datagram length): đây là tổng độ dài tính theo byte của gói dữ liệu IP (cả phần tiêu đề lẫn phần dữ liệu). Độ dài trường này là 16 bit nên về lý thuyết kích thước tối đa của gói dữ liệu IP là 65.535 byte. Tuy nhiên, hiếm khi kích thước gói dữ liệu vượt quá 1500 bytes và thường giới hạn là 576 bytes.
- Định danh, cờ và vị trí phân đoạn (Identifer, Flags, Fragmention Offset): 3 trường này được sử dụng khi phân mảnh gói IP (fragmentation), phiên bản mới của IP (IPv6) không cho phép phân mảnh gói dữ liệu tại các thiết bị định tuyến.
- Thời gian tồn tại (Time-To-live-TTL): Trường thời gian tồn tại (TTL) được sử dụng để bảo đảm gói dữ liệu không được lưu chuyển mãi mãi (định tuyến lặp theo các đường vòng) trong mạng. Mỗi lần gói tin đi qua một thiết bị định tuyến sẽ bị giảm đi một (- 1). Nếu trường TTL bằng 0 thì thiết bị định tuyến sẽ loại bỏ gói tin.
- Giao thức (Protocol): Trường này chỉ được sử dụng khi gói dữ liệu IP đến được máy tính đích. Giá trị của trường này xác định giao thức tầng vận tải ở máy tính đích sẽ nhận được phần dữ liệu trong gói dữ liệu IP. Ví dụ giá trị 6 có ý nghĩa phần dữ liệu cần chuyển tới thực thể TCP, giá trị 17 có ý nghĩa phần dữ liệu phải chuyển đến thực thể UDP, RFC 1700 liệt kê các giá trị này.

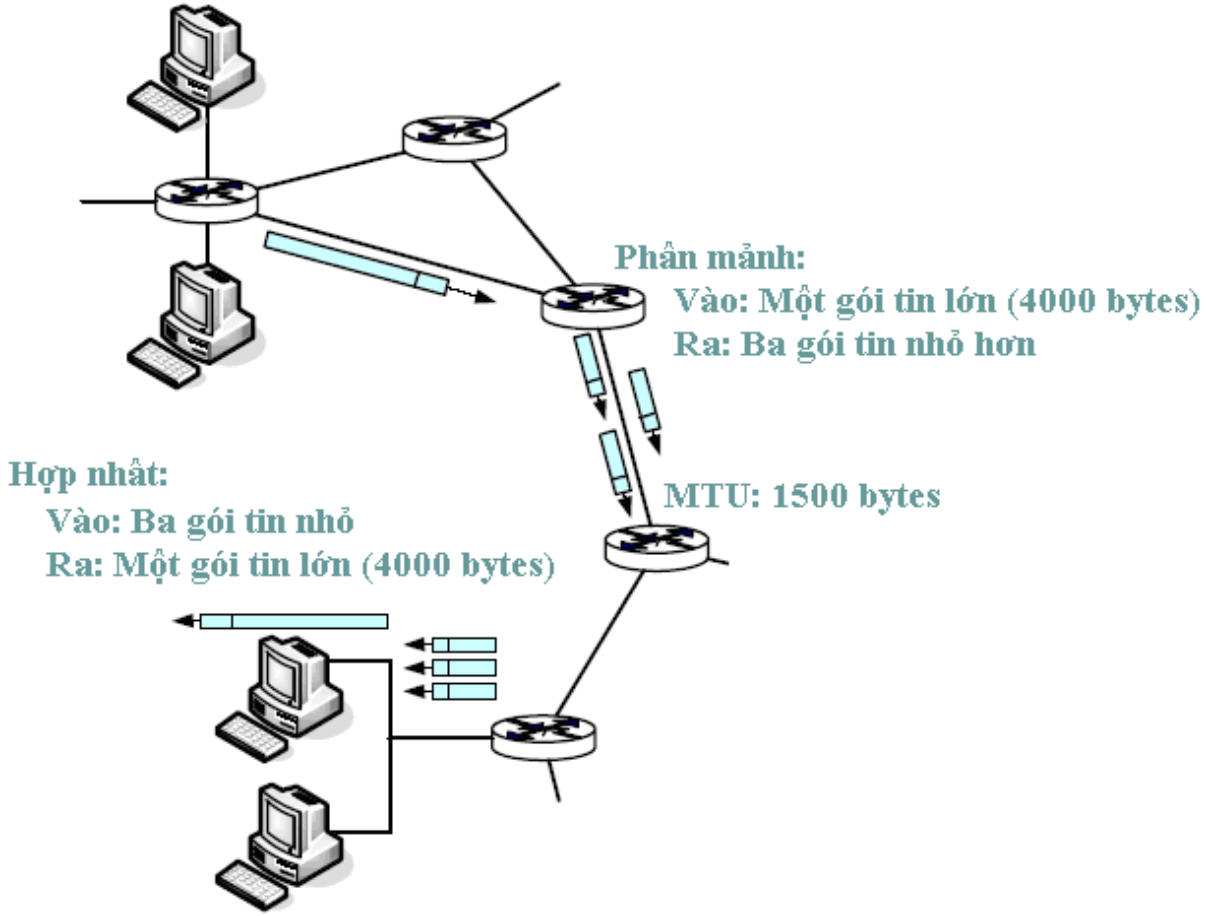
Vai trò của trường giao thức trong gói dữ liệu IP tương tự vai trò trường số hiệu cổng trong đoạn tin của tầng vận tải. Trường giao thức được xem là điểm nối giữa tầng mạng và tầng vận tải cũng như trường số hiệu cổng là điểm nối giữa tầng vận tải với ứng dụng cụ thể.

- Checksum của phần thông tin điều khiển (Header checksum): Trường checksum giúp thiết bị định tuyến phát hiện lỗi trong phần thông tin điều khiển của gói dữ liệu IP được gửi đến. Giá trị checksum được tính bằng cách xem phần thông tin điều khiển là một chuỗi các từ 2 bytes, cộng các từ này lại và sau đó lấy bù một. Thiết bị định tuyến tính lại Internet checksum cho mỗi gói dữ liệu IP nhận được và có thể phát hiện ra lỗi nếu như giá trị checksum tính lại xuất hiện bit 0. Thiết bị định tuyến thường loại bỏ những gói dữ liệu bị lỗi. Chú ý rằng router phải tính lại checksum, trường TTL và có thể một số trường khác. RFC 1071 trình bày phương thức tính checksum nhanh.
- Đia chỉ IP nguồn và đích: Những trường này là 32 bit địa chỉ IP của máy tính gửi và máy tính nhận. Tầm quan trọng của địa chỉ đích là rõ ràng. Địa chỉ IP cùng với số hiệu cổng tạo gọi là socket.
- Tùy chọn (Option): Các trường này cho phép mở rộng tiêu đề IP. Phần lựa chọn trong tiêu đề hiếm khi được sử dụng, sự tồn tại của phần lựa chọn trong tiêu đề làm phức tạp việc xử lý các gói tin vì tiêu đề của gói dữ liệu có phần lựa chọn không có độ dài cố định, do đó không xác định được vị trí bắt đầu của dữ liệu thực sự. Như vậy thời gian xử lý gói dữ liệu IP tại mỗi thiết bị định tuyến có thể khác nhau. Đây là nhược điểm của các mạng hiệu suất cao. Vì thế, IPv6 sẽ loại bỏ các trường Lựa chọn.
- Dữ liệu (data): Cuối cùng là trường quan trọng nhất - trường dữ liệu. Thông thường trường dữ liệu của gói IP là gói dữ liệu của tầng vận tải (TCP hay UDP) để chuyển đến nơi nhận. Tuy nhiên, trường dữ liệu có thể là các kiểu dữ liệu khác, ví dụ bản tin ICMP.

7.4.1.3 Phân mảnh và hợp nhất gói tin IP

Không phải tất cả các giao thức của tầng liên kết dữ liệu đều có khả năng truyền các gói tin có cùng độ lớn, một vài giao thức có thể gửi những gói tin lớn trong khi một vài giao thức chỉ có thể gửi những gói tin nhỏ. Ví dụ: mạng Ethernet có độ lớn không quá 1500 bytes, trong khi gói tin trên những liên kết ở mạng diện rộng có độ lớn không vượt quá 576 bytes, số lượng dữ liệu tối đa của một gói tin trên một đường truyền vật lý được định nghĩa là MTU (maximum transfer unit). Gói dữ liệu IP được đặt trong gói dữ liệu của tầng liên kết dữ liệu giữa hai thiết bị định tuyến kế tiếp nhau trên đường truyền. Do vậy giá trị MTU của giao thức của tầng liên kết dữ liệu giới hạn độ dài của gói tin IP. Giới hạn kích thước của gói tin IP không phải là vấn đề lớn, vấn đề ở đây là các kết nối giữa các thiết bị định tuyến dọc theo tuyến đường từ nơi gửi đến nơi nhận có thể sử dụng các giao thức liên kết dữ liệu khác nhau với những giá trị MTU khác nhau.

Một thiết bị định tuyến kết nối đến nhiều đường vật lý khác nhau, mỗi kết nối có một giao thức liên kết dữ liệu khác nhau với các giá trị MTU khác nhau. Giả sử khi nhận được một gói dữ liệu đến từ một liên kết nào đó, thiết bị định tuyến căn cứ vào địa chỉ đích kiểm tra trong bảng định tuyến để xác định cần gửi gói tin đi ra theo kết nối nào. Tuy nhiên đường kết nối ra ngoài này có giá trị MTU nhỏ hơn độ dài gói dữ liệu IP, khi đó thiết bị định tuyến phải phân mảnh (fragmentation) dữ liệu trong gói dữ liệu IP thành nhiều gói dữ liệu IP nhỏ hơn và sau đó gửi những gói dữ liệu nhỏ hơn này trên đường kết nối. Mỗi gói dữ liệu IP nhỏ này được coi là một mảnh (fragment).



Hình 7.19 Phân mảnh và hợp nhất gói tin

Các mảnh tách rời này cần được ráp lại trước khi chuyển lên tầng vận tải tại máy tính nhận. Rõ ràng là cả TCP và UDP đều mong muốn nhận được một đoạn tin đầy đủ, không bị phân mảnh từ tầng mạng. Việc hợp nhất các gói dữ liệu tại các thiết bị định tuyến sẽ làm giao thức phức tạp lên nhiều và làm giảm hiệu suất của thiết bị định tuyến. Giữ vững nguyên tắc thiết kế tầng mạng đơn giản nhất có thể, IPv4 quyết định việc hợp nhất các mảnh dữ liệu được thực hiện tại các thiết bị đầu cuối chứ không phải là tại các thiết bị định tuyến.

Khi máy tính đích nhận được một loạt các gói dữ liệu tớ cùng một nguồn, nó cần phải xác định liệu đấy là những gói dữ liệu độc lập hay là các mảnh của một gói dữ liệu lớn ban đầu. Trong trường hợp thứ hai, nó phải tiếp tục xác định

liệu đã nhận được đầy đủ các mảnh chưa và làm sao để ráp các mảnh này tại theo trật tự ban đầu để tạo ra gói dữ liệu nguyên thủy. Máy tính đích sẽ sử dụng các trường identifcation, flag và fragmentation để thực hiện công việc hợp nhất này. Khi tạo ra một gói dữ liệu IP, ngoài địa chỉ gửi và địa chỉ nhận, máy tính gửi sẽ đặt vào trường identifcation một số định danh. Giá trị của số định danh sẽ tăng dần. Khi một thiết bị định tuyến cần chia chia nhỏ một gói dữ liệu, thì tất cả các gói dữ liệu con được tạo ra đều có địa chỉ nguồn, địa chỉ đích và giá trị trường định danh giống hệt gói dữ liệu ban đầu. Khi đích nhận được một loạt các gói dữ liệu từ cùng một nơi gửi đến, nó có thể kiểm tra giá trị định danh để xác định liệu những gói dữ liệu đó có là các mảnh của một gói dữ liệu lớn hơn hay không. Vì dịch vụ IP không tin cậy nên một số mảnh có thể không đến được đích. Để máy tính nhận có thể chắc chắn là đã nhận được mảnh cuối cùng của gói dữ liệu ban đầu, thì trường cờ của mảnh cuối cùng phải có giá trị 0, trong khi trường cờ của các mảnh khác có giá trị là 1. Tương tự để máy nhận xác định được liệu có mất mảnh nào không (và để ghép các mảnh theo đúng thứ tự), trường Offset được sử dụng để xác định vị trí của mảnh trong gói dữ liệu IP ban đầu. Xét ví dụ trên hình 7.19, một gói dữ liệu có độ lớn 4000 bytes đến thiết bị định tuyến và phải gửi qua gói dữ liệu ban đầu phải được tách ra thành ba mảnh phân biệt (mỗi mảnh trở thành một gói dữ liệu IP độc lập). Giả sử gói dữ liệu ban đầu có trường định danh nhận giá trị 777. Giá trị các trường trong ba mảnh này được chỉ ra trong bảng sau:

Fragment	Bytes	ID	Offset	Flag
1	1480 byte trong trường dữ liệu	identification=777	offset=0 (dữ liệu Bắt đầu từ byte thứ 0)	flag=1 (còn mảnh nữa)
2	1480 byte trong trường dữ liệu	identification=777	offset=1480 (dữ liệu bắt đầu từ byte thứ 1480)	flag=1 (còn mảnh nữa)
3	1020 byte(3980-1480-1480) trong trường dữ liệu	identification=777	offset=2960 (dữ liệu bắt đầu từ byte thứ 2960)	flag=0 (đây là mảnh cuối cùng)

Dữ liệu của gói IP chỉ được chuyển đến tầng vận tải tại máy tính nhận khi tầng IP tái tạo hoàn chỉnh gói dữ liệu IP ban đầu. Nếu một số mảnh diệu bị mất không đến được đích, thì toàn bộ gói dữ liệu sẽ bị loại bỏ và không được chuyển lên tầng vận tải. Nếu sử dụng giao thức TCP ở tầng vận tải, thì thực thể TCP sẽ khắc phục mất mát do bên phát sẽ gửi lại đoạn tin. Phân đoạn và hợp nhất đặt thêm các nhiệm vụ cho các thiết bị định tuyến, vì vậy người ta cố gắng giảm thiểu việc phân mảnh dữ liệu. Điều này thường được thực hiện bằng cách giới hạn độ lớn gói dữ liệu của tầng vận tải (TCP hay UDP segment) bởi một giá trị tương đối nhỏ. Khi đó việc phân mảnh trở nên không cần thiết. Phần lớn các giao thức liên kết dữ liệu hỗ trợ IP có MTU tối thiểu là 536 byte, có thể loại bỏ hoàn toàn việc phân mảnh nếu đặt giá trị MSS là 536 byte với 20 byte tiêu đề của gói TCP và 20 byte tiêu đề của gói IP. Đây chính là lý do hầu hết các gói TCP khi truyền khối lượng lớn dữ liệu (chẳng hạn FTP) có độ dài từ 512 đến

536 byte. Nếu đoạn tin TCP được bọc trong gói IP và cả hai gói TCP và IP đều không có trường tùy chọn thì gói dữ liệu IP sẽ có 40 byte thông tin điều khiển, phần còn lại là dữ liệu của tầng ứng dụng.

7.4.2 Địa chỉ IP V6

Nhận thấy sự thiếu hụt địa chỉ IP trong khi mạng Interntet ngày càng mở rộng, đầu những năm 1990, IETF (Internet Engineering Task Force) bắt đầu lực phát triển giao thức mạng thay thế IPv4. Những người thiết kế IPv6 cũng chọn lọc các tính năng, cải tiến nhiều đặc điểm khác của IPv4 dựa trên cơ sở những kinh nghiệm thực tế của IPv4.

7.4.2.1 Định dạng gói tin IP V6

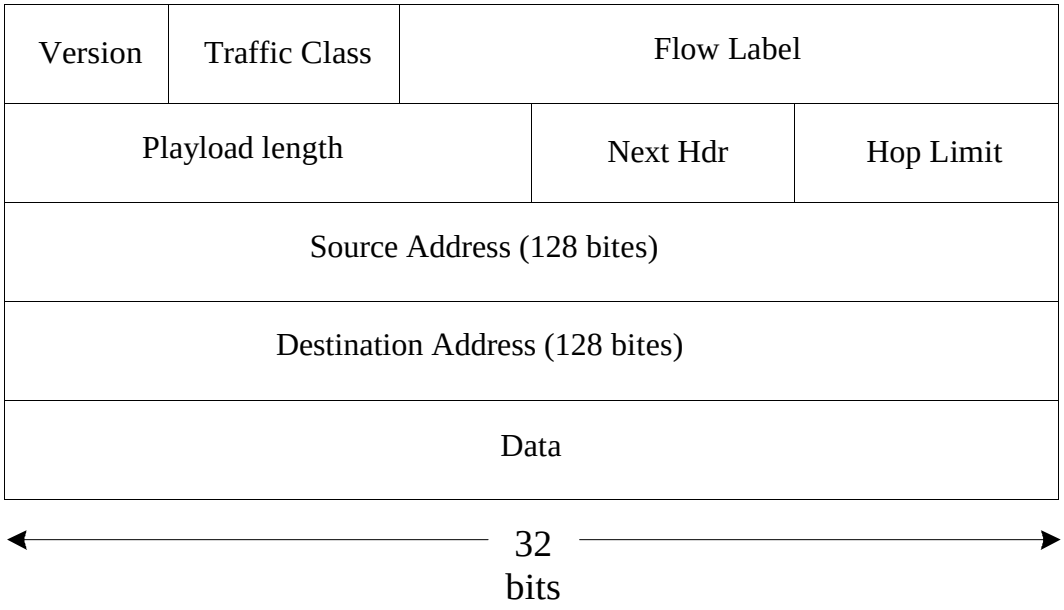
Khuôn dạng gói dữ liệu IPv6 được minh họa trên hình hình 7.20. Điểm thay đổi quan trọng nhất của IPv6 chính là khuôn dạng gói tin.

- Mở rộng khả năng đánh địa chỉ: IP v6 tăng kích thước địa chỉ IP từ 32 bit lên 128 bit. Với độ dài 128 bit có thể tạo được 2^{128} địa chỉ sẽ đảm bảo khả năng không bị thiếu địa chỉ IP. Bên cạnh địa chỉ duy nhất (unicast) và địa chỉ nhóm (multicast), IPv6 còn có một dạng địa chỉ mới gọi là "anycast", cho phép một gói tin với địa chỉ đích thuộc kiểu "anycast" có thể được chuyển từ một nhóm các máy tính (đặc điểm này sẽ được sử dụng ví dụ khi gửi thông điệp HTTP GET tới nhiều trang phụ (mirror site) chứa cùng một tài liệu nào đấy).
- Phần thông tin điều khiển có độ dài cố định 40 byte: một số trường IP v4 mang tính chất tùy chọn, do đó độ dài của nó có thể thay đổi. Độ dài phần thông tin điều khiển cố định cho phép xử lý các gói dữ liệu IP v6 nhanh hơn.
- Gán nhãn luồng (flow label) và độ ưu tiên (priority): IPv6 không có định nghĩa cho luồng một cách rõ ràng. Các khuyến nghị RFC 1752 và RFC 2460 cho phép gắn nhãn cho các gói tin thuộc về cùng một luồng. Các gói tin này đòi hỏi được xử lý một cách đặc biệt, như các dịch vụ thời gian thực với chất lượng tốt hơn. Ví dụ, các dữ liệu đa phương tiện có thể xem như một luồng liên tục. Dữ liệu các ứng dụng truyền thống, như truyền tập tin, thư điện tử không được xem như một luồng. Có thể dữ liệu của những người có độ ưu tiên cao (ví dụ người trả phí cao hơn) cũng có thể coi như một luồng. Rõ ràng ở đây những người thiết kế IPv6 đã dự đoán được nhu cầu phân biệt giữa các luồng dữ liệu ngay cả khi chưa định nghĩa chính xác được luồng là gì. Thông tin điều khiển của IP v6 cũng có trường Traffic Class 8 bit. Trường này giống trường TOS (Tyte of Service) trong IPv4 có thể được sử dụng cho những gói tin có quyền ưu tiên trong một luồng, hoặc cho những ứng dụng có độ ưu tiên cao (ví dụ gói tin ICMP).

So sánh khuôn dạng gói dữ liệu IP v4 và IP v6 ta thấy gói IP V6 có cấu trúc đơn giản hơn. Sau đây là một số trường trong gói dữ liệu IPV6:

- Phiên bản (version): Trường 4-bit này xác định phiên bản IP của gói dữ liệu. Rõ ràng gói IPv6 có giá trị "6" trong trường này. Chú ý không phải đặt giá trị "4" trong trường này thì gói dữ liệu là IPv4.

- Traffc class: Trường 8-bit này giống trường TOS trong IPv4.
- Nhãn luồng (Flow label): Trường 20 bit này xác định một luồng chứa gói dữ liệu.
- Độ lớn dữ liệu (Payload length): Độ lớn (tính theo byte) của phần dữ liệu không tính tiêu đề.
- Next header: Trường này xác định giao thức ở tầng phía trên sẽ nhận dữ liệu (ví dụ tới TCP hoặc UDP). Trường này giống trường Protocol của IPv4.
- Hop limit: Giá trị của trường này sự giảm đi 1 khi đi qua mỗi router. Nếu giá trị này bằng 0, gói dữ liệu bị loại bỏ.
- Địa chỉ nguồn và đích (source ang destination addresss): Khuôn dạng 128-bit địa chỉ IPv6 được đặc tả trong RFC 2373.
- Dữ liệu (data): Khi gói tin IPv6 tới đích, các tiêu đề sẽ bị loại bỏ và phần dữ liệu này sẽ được chuyển đến thực thể ở tầng phía trên.



Hình 7.20 Cấu trúc gói tin IP V6

Có một số trường trong IPv4 không xuất hiện trong IPv6 nữa:

- Phân mảnh, Hợp nhất gói tin: IPv6 không cho phép phân mảnh và hợp nhất gói tin tại các router trung gian. Nếu một gói dữ liệu IPv6 quá lớn để có thể gửi đi trên một đường liên kết ra của router, router sẽ loại bỏ gói tin này và gửi một thông báo lỗi ICMP "Packet Too Big" tới bên gửi. Sau đó bên gửi gửi lại dữ liệu, sử dụng các gói dữ liệu có kích thước nhỏ hơn. Việc phân mảnh và hợp nhất các gói tin IP chiếm nhiều thời gian xử lý của các router. Thực hiện những công việc này tại các thiết bị đầu cuối sẽ làm tăng tốc độ truyền trên mạng.
- Checksum. Do tầng vận tải (ví dụ, TCP và UDP) và các giao thức liên kết dữ liệu (ví dụ Ethenet) đã thực hiện kiểm tra lỗi, chức năng này không cần thiết trong tầng mạng nên đã được bỏ đi. Giá trị trường TTL trong tiêu đề của

IPv4 giảm đi một khi đi qua mỗi thiết bị định tuyến, nên giá trị trường checksum trong tiêu đề IPv4 cần phải được tính tại tại các thiết bị định tuyến. Như vậy, giống như phân mảnh và hợp nhất, việc này khiến thời gian xử lý gói tin IPv4 lâu hơn.

7.4.2.2 ICMP cho IPV6

Giao thức ICMP được sử dụng để thông báo lỗi và cung cấp một số các thông tin hạn chế tới thiết bị đầu cuối (ví dụ lệnh ping). Một phiên bản mới của ICMP được đặc tả cho IPv6 trong khuyến nghị RFC 2463. Bên cạnh các kiểu và mã cũ, ICMPv6 cũng đưa thêm vào nhiều kiểu và mã mới. Ví dụ kiểu mã lỗi "Packet Too Big" hay "Unrecognized IP v6 option".

7.4.3 Chuyển từ IPv4 sang IPv6

Một vấn đề đặt ra là công tác chuyển đổi địa chỉ IP từ phiên bản 4 sang phiên bản 6, tất nhiên các máy tính và thiết bị có khả năng tương thích ngược (làm việc với IP v4) nhưng các hệ thống cũ thì không thể tương thích với IP v6. Một số giải pháp đã được đề xuất như sau:

- Lựa chọn một thời điểm nào đó, tắt tất cả máy để nâng cấp lên IPv6. Ngay cả thời điểm đó khi Internet còn rất nhỏ và vẫn còn được quản trị bởi một nhóm nhỏ các chuyên gia, người ta cũng không thể chọn được một thời điểm như vậy. Giải pháp này ngày nay sẽ đòi hỏi sự tham gia của hàng trăm triệu máy tính và hàng triệu người quản trị mạng, rõ ràng đây là một điều không tưởng. RFC 1933 đưa ra hai giải pháp (có thể sử dụng đồng thời hay dùng riêng rẽ) để dần dần tích hợp các thiết bị sử dụng IPv6 vào thế giới IPv4 (dĩ nhiên mục tiêu dài hạn vẫn là chuyển tất cả các thiết bị sử dụng IPv4 sang IPv6).
- Có thể giải pháp đơn giản nhất để đưa vào các thiết bị hỗ trợ IPv6 là dual-stack. Các thiết bị triển khai cả IPv4 và IPv6. Thiết bị IPv6/IPv4 như vậy được đặc tả trong RFC 1933 có khả năng nhận và gửi cả hai gói dữ liệu IPv4 và IPv6. Khi trao đổi với một nút IPv4, nút IPv6/IPv4 sử dụng gói dữ liệu IPv4 và khi trao đổi với nút IPv6, sẽ sử dụng gói IPv6. Nút IPv4/IPV6 cần phải có cả hai địa chỉ IPv6 và IPv4. Chúng cần phải có khả năng xác định được một nút có khả năng IPv6 hay chỉ hỗ trợ IPv4. Có thể giải quyết vấn đề này nhờ hệ thống tên miền DNS.

7.5 Định tuyến trên Internet

Mạng Internet toàn cầu ngày nay là sự kết nối của nhiều mạng bao gồm các ISP khu vực, quốc gia và quốc tế. Các giao thức định tuyến được triển khai dựa trên hai loại giao thức đã đề cập trên. Giao thức định tuyến nội miền được sử dụng để cấu hình và duy trì bảng định tuyến trong tất cả các thiết bị định tuyến trong một miền. Hiện nay có 3 giao thức định tuyến nội miền được sử dụng rộng rãi: RIP (Routing lnformation Protocol), OSPF (Open Shortest Path First) và EIGRP (Cisco propriety Enhanced Interior Gateway Routing Protocol). Giao thức định tuyến liên miền thường được sử dụng là giao thức BGP.

7.5.1 Giao thức RIP

RIP là một trong những giao thức định tuyến nội miền đầu tiên, nó có một số đặc điểm sau:

- Định tuyến nội miền: Cho phép trao đổi thông tin giữa các thiết bị định tuyến trong một miền.
- Đo khoảng cách bằng chặng đường đi: Giá đường đi giữa hai thiết bị đầu cuối được xác định bằng số lượng các thiết bị định tuyến trung gian trên đường đi đó. Độ dài tối đa của một tuyến đường là 15, nghĩa là đường kính tối đa của một miền là 15 thiết bị định tuyến.
- Truyền thông không tin cậy: RIP sử dụng UDP để chuyển bản tin về các tuyến đường.
- Gửi tin dạng quảng bá và nhóm: RIP v1 sử dụng cách truyền quảng bá khi truyền giữa hai thiết bị định tuyến, RIP v2 cho phép truyền theo chế độ nhóm.
- Thuật toán vector khoảng cách: RIP sử dựng thuật toán vector khoảng cách. Các thiết bị định tuyến hàng xóm trao đổi bảng định tuyến cho nhau 90s một lần (có thể thay đổi chu kỳ này) trong các bản tin RIP (RIP response message, RIP advertisement), mỗi bản tin chứa tối đa 25 địa chỉ đích.
- Các máy tính có thể thụ động nhận thông tin từ các thiết bị định tuyến: RIP cho phép các thiết bị đầu cuối (chủ yếu là máy tính) lắng nghe và cập nhật bảng định tuyến. Điều này đặc biệt hữu dụng với các mạng có nhiều thiết bị định tuyến. Khi đó máy tính trong mạng có thể dễ dàng xác định được thiết bị định tuyến cần chuyển tới.

0	8	16	24	31

Command(1-5)	Version(2)	Must be zero
Family of Net 1		Route tag for Net 1
IP address of Net 1		
Subnet mask for Net 1		
Next hop for Net 1		
Distance to Net 1		
Family of Net 2		Route tag for Net 2
IP address of Net 2		
Subnet mask for Net 2		
Next hop for Net 2		
Distance to Net 2		
…		

Hình 7.21 Khuôn dạng bản tin RIP

Thiết bị định tuyến gửi một bản tin RIP liệt kê các mạng mà nó có thể kết nối với. Khi nhận được một quảng cáo như vậy, thực thể RIP trên thiết bị định tuyến sử dụng những thông tin này để cập nhật bảng định tuyến của nó. Mỗi một trường trong bản tin quảng cáo là một cặp địa chỉ mạng đích **n**, khoảng cách **r**, trong đó khoảng cách **r** là số lượng các thiết bị định tuyến trung gian từ thiết bị định tuyến gửi bản tin tới đích có địa chỉ mạng là **n**. Khi nhận được một bản tin, giả sử thiết bị định tuyến không có đường đi tới đích được quảng cáo trong bản tin hoặc có đường đi đến đích nhưng giá cao hơn, thiết bị định tuyến sẽ cập nhật bảng định tuyến để sử dụng tuyến đường mới (điểm đầu tiên trên tuyến đường này chính là thiết bị định tuyến gửi quảng cáo).

Ưu điểm chính của RIP là tính đơn giản, không đòi hỏi cấu hình nhiều. Người quản trị chỉ cần bật máy lên, cho phép router trao đổi thông tin với nhau, sau một thời gian ngắn, thiết bị định tuyến sẽ tự xây dựng được bảng định tuyến của mình. Doanh nghiệp có thể lựa chọn một thiết bị định tuyến trong miền làm thiết bị định tuyến ngầm định, thường là thiết bị định tuyến nối với ISP. Sau đó RIP sẽ thực hiện việc quảng cáo cho thiết bị định tuyến ngầm định này, các gói tin gửi ra phía ngoài sẽ được gửi qua thiết bị định tuyến ngầm định tới ISP. Hình 5.9 minh họa khuôn dạng bản tin cập nhật RIP. Mỗi trường trong bản tin ứng với một địa chỉ đích, mặt nạ mạng của địa chỉ đích (đo đó có thể sử dụng địa chỉ không phân lớp CIDR), khoảng cách tới đích và nút kế tiếp trên đường tới đích.

7.5.2 Giao thức OSPF

Giao thức RIP có một số nhược điểm của thuật toán vector khoảng cách. Độ dài của bản tin có thể lớn do phải liệt kê toàn bộ danh sách các địa chỉ đích và khoảng cách tới đó. Khi nhận được bản tin, thiết bị định tuyến nhận phải lấy ra tìm trường, so sánh trong bảng định tuyến của nó, như vậy thời gian xử lý bản tin trong mỗi thiết bị định tuyến lớn, gây ra một độ trễ nhất định. Do vậy giao thức RIP chỉ phù hợp với các mạng có kích cỡ nhỏ. Khi một tổ chức mạng tương đối lớn, người ta cần phải đưa ra giao thức phù hợp hơn. IETF đã đưa ra giao thức OSPF với các đặc điểm sau:

- Định tuyến nội miền: Cho phép trao đổi thông tin giữa các thiết bị định tuyến trong một miền.
- Hỗ trợ CIDR: hỗ trợ việc phân mạng, tạo lập mạng con.
- Trao đổi các thông tin đã được kiểm chứng. Hai thiết bị định tuyến trao đổi bản tin OSPF với nhau có thể tiến hành thủ tục kiểm tra để xác định mình nhận được bản tin từ đúng đối tượng, điều này ngăn ngừa các cuộc tấn công bằng phương pháp giả mạo.
- Sử dụng thuật toán tìm đường dựa trên trạng thái kênh truyền.
- Hỗ trợ phân cấp trong miền. Ưu điểm chính của OSPF là cho phép tiếp tục phân một miền thành nhiều miền con.

7.5.3 Giao thức BGP

Giao thức BGP (Border Gateway Protocol) được mô tả trong các khuyến nghị RFC 1771, 1772, 1773, được xem là một chuẩn hiển nhiên trong định tuyến

giữa các vùng trên mạng Internet. Nhiệm vụ của nó là định tuyến giữa các vùng được quản trị độc lập với nhau. BGP có những đặc điểm sau:

- Định tuyến liên vùng: BGP cho phép cung cấp các thông tin định tuyến giữa các vùng, mỗi tuyến đường được xem là một chuỗi các AS liên tiếp nhau.
- Hỗ trợ việc thiết lập chính sách: Người quản trị có thể áp dụng những chính sách nào đó, ví dụ hạn chế việc quảng cáo ra bên ngoài.
- Truyền thông tin cậy: Hai thực thể BGP sử dụng kết nối TCP để trao đổi bản tin.

7.6 Các giao thức khác

7.6.1 Giao thức ICMP

ICMP được các thiết bị đầu cuối, thiết bị định tuyến sử dụng để trao đổi các thông tin tầng mạng với nhau (chủ yếu cho việc báo lỗi), giao thức này được đặc tả trong RFC 792. Ví dụ khi chạy một phiên Telnet, FTP, hoặc HTTP, người dùng có thể gặp thông báo như "Destination network unreachable" (Không đến được mạng đích). Nếu thiết bị định tuyến không tìm được đường dẫn đến máy tính đích, nó sẽ tạo ra và gửi thông báo trên tới máy tính của người sử dụng để thông báo lỗi.

ICMP thường được coi là một phần của IP, nhưng về mặt kiến trúc lại nằm trên IP, bởi vì thông báo ICMP được đặt trong gói IP, giống như dữ liệu của giao thức TCP hoặc UDP nằm trong trường dữ liệu của gói tin IP. Tương tự như giao tiếp với tầng vận tải, khi nhận được một gói tin IP với trường protocol xác định giao thức ICMP, tầng mạng của máy tính nhận sẽ chuyển phần dữ liệu (và thông điệp ICMP) lên thực thể ICMP, giống như đã làm với TCP hay UDP.

Thông báo ICMP có trường kiểu (type) và trường mã (code) và chứa 8 byte đầu tiên của gói dữ liệu IP gây ra lỗi (nguyên nhân để tạo ra thông báo ICMP). Do đó bên gửi có thể xác định được gói tin nào gây ra lỗi. Bảng sau liệt kê một số mã của bản tin ICMP:

Loại ICMP	Mã	Nội dung
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestin control)
8	0	echo request
9	0	route advertisement
10	0	route discovery
11	0	TTL expired
12	0	IP header bad

Câu lệnh ***traceroute*** cho phép người sử dụng xác định tất cả các thiết bị định tuyến trên một tuyến đường giữa bất kỳ hai thiết bị đầu cuối nào. Chương trình ***traceroute*** cũng sử dụng các thông báo của ICMP để xác định định tên và địa chỉ của các router giữa nguồn và đích. Chương trình ***traceroute*** trong máy tính nguồn sẽ gửi đi một loạt các gói dữ liệu IP tới máy tính đích. Gói IP đầu tiên có trường TTL nhận giá trị 1, gói thứ hai là 2, gói thứ ba là 3, máy tính nguồn đặt thời gian cho mỗi gói IP gửi đi. Khi gói IP thứ n đến thiết bị định tuyến thứ n, nó thấy trường TTL của gói dữ liệu nhận giá trị 0, nên theo nguyên tắc của giao thức IP, thiết bị định tuyến sẽ loại bỏ gói dữ liệu và gửi thông điệp cảnh báo ICMP (kiểu 11 mã 0). Trong thông điệp cảnh báo này có tên và đại chỉ IP của thiết bị định tuyến. Khi nhận được thông báo ICMP, máy tính nguồn xác định được thời gian khứ hồi đến thiết bị định tuyến thứ n cũng như tên, và địa chỉ IP của thiết bị định tuyến đó. Ngoài ra, ICMP còn được sử dụng trong các dịch vụ định hướng lại cổng mặc định của máy tính.

7.6.2 Cấp phát địa chỉ IP

Mỗi thiết bị tham gia vào mạng phải được gán địa chỉ và địa chỉ đó phải duy nhất trên mạng, có hai cách cách gán địa chỉ IP:

- Gán thủ công: Người quản trị hệ thống thiết lập cấu hình
- Gán tự động: dịch vụ cấp phát địa chỉ động, sử dụng các giao thức như RARP, BOOTP hoặc DHCP (Dynamic Host Configuration Protocol). DHCP là phiên bản mở rộng của giao thức BOOTP với DHCP, khi máy chủ DHCP trong mạng nhận yêu cầu DHCP từ một máy khách, nó sẽ cấp phát một địa chỉ IP cho máy khách yêu cầu. DHCP được sử dụng rộng rãi trong mạng cục bộ hay truy cập Internet đến ISP.

7.6.2.1 Giao thức RARP

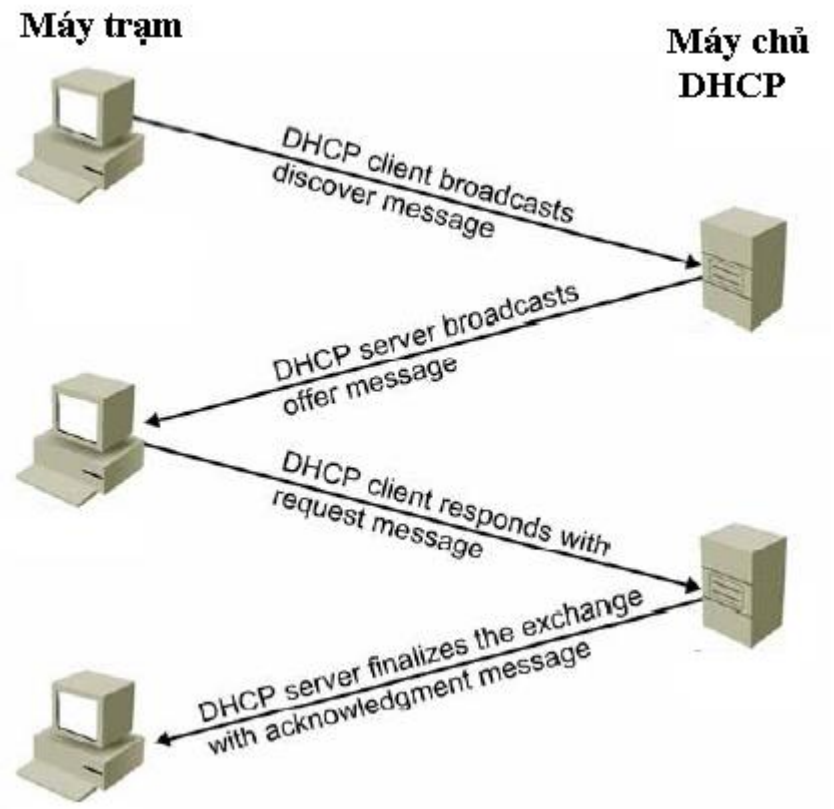
Giao thức phân giải địa chỉ ngược (RARP- Reverse Address Resolution Protocol) ra đời năm 1984 là giao thức dùng để tìm địa chỉ logic (IP) khi biết địa chỉ vật lý (MAC). Mỗi máy trạm chỉ cần bộ xử lý và bộ nhớ mà không cần phải có đĩa cứng, không gian lưu trữ được dùng chung trên máy chủ. Do không có tập tin cấu hình nên tiến trình khởi động của các máy tính này thường phải lấy thông tin từ máy chủ. Tuy nhiên, trước khi có thể nối kết đến được máy chủ, các máy trạm cần phải biết được địa chỉ IP của nó. Trên máy chủ phải có một bảng ánh xạ địa chỉ vật lý và địa chỉ IP của các máy trạm, bảng ánh xạ này do người quản trị mạng thiết lập thủ công. Khi nhận được yêu cầu RARP, máy chủ tìm trong bảng địa chỉ và trả về địa chỉ IP tương ứng cho máy trạm đã gởi yêu cầu. Giao thức RARP hoạt động trên lớp liên kết dữ liệu và do đó phụ thuộc vào phần cứng card mạng, đây là một trong những hạn chế của giao thức này. Ngoài ra, giao thức RARP không cung cấp một số thông tin cần thiết khác cho máy trạm như cổng mặc định, địa chỉ IP của máy chủ DNS...

7.6.2.2 Giao thức BOOTP

Giao thức BOOTP phục vụ cho việc trong việc cấp phát địa chỉ IP động, ra đời vào năm 1985 nhằm khắc phục một số hạn chế của của giao thức RARP. Giao thức BOOTP hoạt động ở lớp ứng dụng và dùng giao thức UDP tại lớp vận tải, do đó không phụ thuộc vào phần cứng card mạng. BOOTP cho phép một máy trạm nhận được các thông số cấu hình cần thiết bao gồm địa chỉ IP, mặt nạ mạng con, cổng ra mặc định và địa chỉ máy chủ. Tuy nhiên, người quản trị mạng vẫn phải ánh xạ địa chỉ MAC và địa chỉ IP bằng phương pháp thủ công.

7.6.2.3 Giao thức DHCP

Giao thức DHCP dùng để gán các địa chỉ IP động cho máy trạm khi tham gia vào mạng. Máy chủ DHCP quản lý địa chỉ IP theo phương pháp động, mỗi địa chỉ được cấp phát trong một khoảng thời gian nhất định (trong hệ điều hành windows thời gian thuê địa chỉ là 7 ngày), điều đó giúp cho việc sử dụng các địa chỉ hiệu quả hơn. DHCP có cơ chế tự động gán địa chỉ IP thích hợp cho từng mạng con. Để được cấp phát địa chỉ IP, máy trạm và máy chủ phải thực hiện bốn bước:



Hình 7.22 Các bước cấp phát địa chỉ IP bằng DHCP

- Máy trạm quảng bá bản tin DHCP Discover
- Máy chủ trả lời bằng bản tin DHCP Offer
- Máy trạm yêu cầu cấp phát địa chỉ IP bằng bản tin DHCP Request
- Máy chủ xác nhận yêu cầu được chấp thuận bằng bản tin DHCP ACK

Bước 1: Khi mới khởi động, máy trạm chưa có địa chỉ IP, nó tạo ra một bản tin DHCP Discover và gửi quảng bá lên mạng cho đến khi nhận được phản hồi từ máy chủ.

Bước 2: Tất cả các máy trong mạng nội bộ đều nhận được bản tin DHCP Discover , chỉ có máy chủ DHCP mới xử lý bản tin này. Nếu máy chủ có cấu hình hợp lệ cho máy trạm, nó chuẩn bị bản tin DHCP Offer chứa địa chỉ MAC của máy trạm, địa chỉ IP dự định cấp, mặt nạ mạng, địa chỉ IP của máy chủ, cổng mặc định, địa chỉ IP của máy chủ DNS, thời gian cho thuê …. Máy chủ DHCP gửi quảng bá bản tin này lên mạng.

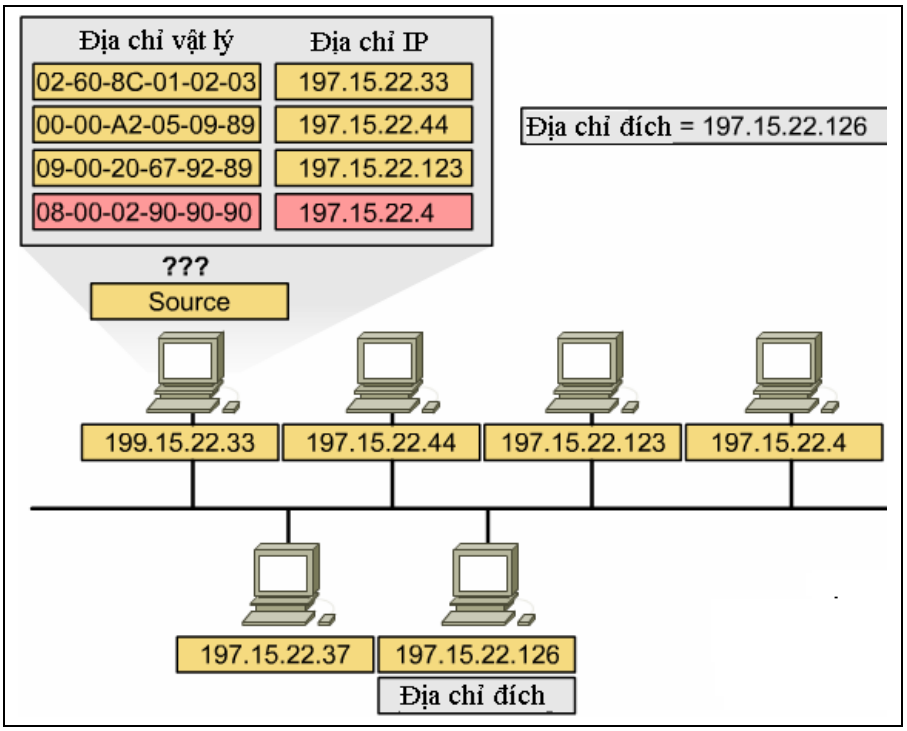
Bước 3: Máy trạm nhận được bản tin DHCP Offer và chấp nhận một trong các địa chỉ IP, nó chuẩn bị bản tin DHCP Request và gửi quảng bá lên mạng để khẳng định đã chấp nhận địa chỉ IP được cấp phát từ máy chủ DHCP.

Bước 4: Nhận được bản tin bản tin DHCP Request, máy chủ DHCP sẽ xác nhận đồng ý cấp phát bằng cách tạo bản tin DHCP ACK và gửi quảng bá lên mạng.

7.7 Chuyển đổi địa chỉ

7.7.1.1 Giao thức ARP

Để có thể trao đổi thông tin trên mạng, mỗi máy tính cần phải có cả địa chỉ vật lý và địa chỉ logic. Sẽ xảy ra tình huống: khi đã biết địa chỉ logic cần phải tìm địa chỉ vật lý tương ứng với địa chỉ logic tại thời điểm đó.



Hình 7.23 Xác định địa chỉ vật lý bằng giao thức ARP

Trên các mạng TCP/IP, giao thức ARP được dùng để tìm một địa chỉ MAC tương ứng với một địa chỉ IP. Khi muốn tìm một địa chỉ MAC tương ứng với địa chỉ IP, máy tính gửi quảng bá bản tin ARP chứa địa chỉ IP của máy tính

cần tìm. Nhận được bản tin này, mỗi trạm sẽ so sánh địa chỉ IP của mình, nếu trùng nhau nó sẽ gửi địa chỉ MAC của nó trở lại cho trạm có yêu cầu.

7.7.1.2 Chuyển đổi địa chỉ - NAT

NAT (Network Address Translation) là cơ chế dùng để chuyển đổi địa chỉ IP trong mạng nội bộ với mạng bên ngoài. Cơ chế này được dùng chủ yếu cho hai mục đích:

- Giải quyết vấn đề thiếu hụt địa chỉ IP
- Che giấu địa chỉ IP trong mạng nội bộ nhằm tăng cường khả năng bảo mật mạng.

Có 2 kiểu chuyển đổi địa chỉ IP cơ bản NAT (Network Address Translation) và PAT (Port Address Translation). Trong khi cơ chế NAT (Network Address Translation) chỉ cho phép ánh xạ ở mức địa chỉ IP thì PAT cho phép ánh xạ thêm tới mức cổng, do đó tại một thời điểm số lượng liên kết ra mạng bên ngoài sẽ nhiều hơn. Đối với từng cơ chế lại cho phép cấu hình tĩnh hay động, tùy từng tình huống, người quản trị mạng có thể lựa chọn phương pháp NAT hay PAT.

7.8 Chia mạng

Các nhà cung cấp dịch vụ Internet được cấp phát dải địa chỉ tương đối lớn và sẽ chia sẻ một phần trong dải địa chỉ này cho tổ chức hoặc thuê bao. Ví dụ ISP được cấp phát dải địa chỉ 200.23.16.0/20. Đến lượt mình ISP chia không gian địa chỉ này thành 8 không gian địa chỉ nhỏ hơn bằng nhau và gán 8 không gian này cho các tổ chức thuê. Địa chỉ IP được Tổ chức ICANN (The Internet Corporation for Assigned Names ang Numbers) quản lý theo các nguyên tắc chỉ đạo ghi trong RFC 2050. Vai trò của tổ chức phi lợi nhuận ICANN không chỉ là phân phối địa chỉ IP mà còn quản trị các DNS máy chủ gốc. Nó cũng còn có nhiệm vụ đặt tên miền cũng như giải quyết các tranh chấp về tên miền.

Hiện nay việc phân phối địa chỉ IP được cơ quan đăng kí Internet cấp vùng quản lý. Giữa năm 2000, có 3 cơ quan đăng kí như vậy: American Registry for Internet Number (ARIN) cho châu Mỹ và một số phần của châu Phi. Reseaux IP Europeans (RIPE) cho châu Âu và một số nước chung quanh và Asia Pacific Network lnformation Center (APNIC). Các máy tính có khả năng di động (mobile) có thể thay đổi mạng theo cách động (khi di chuyển) hoặc tĩnh (theo thời gian). Khi định tuyến, phải xác định mạng trước và sau đó mới tới máy tính trong mạng, nên địa chỉ IP của máy tính di động sẽ phải thay đổi khi máy tính thay đổi mạng. Việc phân chia mạng con mang lại những ích lợi sau:

- Dễ quản lý hơn (vì số trạm ít hơn).
- Hạn chế miền quảng bá, tăng hiệu quả truyền thông trong mạng. Ví dụ: một mạng LAN có 10 máy, nếu dùng mạng lớp C sẽ có miền quảng bá tới 254 host, nếu dùng subnet mask thì miền quảng bá sẽ giảm xuống, hiệu quả mạng sẽ tăng lên. Tăng cường mức độ bảo mật mức thấp cho LAN: mỗi mạng có một danh sách truy cập, theo danh sách này mạng có thể cho phép hay từ chối truy cập vào nó. Có thể bán hoặc cho thuê các đại chỉ không được dùng đến: các công ty được sử hữu các mạng lớn lớp A, B có thể không dùng hết số địa chỉ của họ, họ có thể bán hoặc cho thuê.

CHƯƠNG 8: TẦNG LIÊN KẾT

8.1 Mô hình dịch vụ tầng liên kết dữ liệu

Giao thức tầng liên kết dữ liệu được sử dụng để truyền các khung dữ liệu trên một môi trường vật lý. Giao thức tầng liên kết dữ liệu định nghĩa khuôn dạng đơn vị dữ liệu trao đổi giữa các nút ở mỗi đầu của đường truyền, cũng như những công việc các nút thực hiện khi nhận và gửi những đơn vị dữ liệu này. Chức năng của giao thức tầng liên kết dữ liệu khi gửi và nhận khung dữ liệu bao gồm: phát hiện lỗi truyền lại, điều khiển lưu lượng và truy cập ngẫu nhiên. Giao thức tầng liên kết dữ liệu rất đa dạng : Ethernet, token ring, FDDI và PPP, đôi khi ATM và frame relay có thể cũng được coi là giao thức thuộc tầng liên kết dữ liệu.

Nếu nhiệm vụ của tầng mạng là chuyển gói dữ liệu của tầng vận tải từ máy gửi từ máy nhận thì giao thức của tầng liên kết dữ liệu có nhiệm vụ chuyển gói dữ liệu tầng mạng giữa hai nút kế tiếp trên đường truyền. Một đặc điểm quan trọng của tầng liên kết dữ liệu là gói dữ liệu tầng mạng có thể được xử lý bởi nhiều giao thức khác nhau trên đường truyền. Ví dụ, gói dữ liệu này có thể được chuyển bởi giao thức Ethernet trong mạng nội bộ, sau đó là các giao thức PPP, HDLC, frame relay… trên các đường truyền giữa các thiết bị định tuyến. Mặc dù dịch vụ cơ bản của bất kì tầng liên kết dữ liệu nào là chuyển gói dữ liệu của tầng mạng giữa hai nút kế tiếp, song cụ thể dịch vụ này được thực hiện như thế nào lại phụ thuộc vào giao thức tầng liên kết dữ liệu sử dụng trên đường truyền đó. Nói chung giao thức tầng liên kết dữ liệu có thể cung cấp những dịch vụ sau:

- Đóng gói dữ liệu (frame) và truy cập đường truyền (link access): Phần lớn các giao thức tầng liên kết dữ liệu đặt gói dữ liệu tầng mạng vào trong gói dữ liệu tầng liên kết dữ liệu (frame) trước khi truyền lên trên đường truyền. Frame gồm trường dữ liệu là gói dữ liệu của tầng mạng cùng với một số trường tiêu đề khác. Chú ý frame có thể có cả trường tiêu đề đầu và cuối (header và trailer). Giao thức tầng liên kết dữ liệu xác định khuôn dạng của frame cũng như giao thức truy cập kênh truyền (cách thức truyền). với đường truyền kiểu point-to-point có một phía gửi và một phía nhận thì giao thức truy cập đường truyền là đơn giản (gần như không tồn tại) - bên gửi có thể gửi frame bất cứ lúc nào đường truyền rỗi. Trường hợp phức tạp hơn xẩy" ra khi nhiều nút cùng chia sẻ đường truyền duy nhất: đây là vấn đề đa truy cấp. Chúng ta sẽ thấy các giao thức liên kết dữ liệu khác nhau sẽ có những khuôn dạng dữ liệu khác nhau (chú ý gói dữ liệu của tầng liên kết dữ liệu được gọi là frame). Chúng ta sẽ thấy rằng tiêu đề frame thường chứa trường địa chỉ vật lý của nút, (khác địa chỉ mạng của nút).
- Dịch vụ truyền tin cậy: Nếu cung cấp dịch vụ truyền tin cậy, giao thức tầng liên kết dữ liệu bảo đảm chuyển chinh xác gói dữ liệu tầng mạng trên một đường truyền. Trong chương 3 chúng ta thấy rằng giao thức tầng vận tải (TCP) cũng cung cấp dịch vụ truyền tin cậy. tương tự như trong tầng vận tải, truyền tin cậy ở tầng liên kết dữ liệu được thực hiện qua cơ chế báo nhận và truyền lại. dịch vụ truyền tin cậy ở

- Tầng liên kết dữ liệu thường được sử dụng trên đường truyền có tỉ lệ lỗi cao (ví dụ trên đường truyền không dây). mục đích là sửa lỗi ngay trên đường truyền bị lỗi chứ không phải truyền lại dữ liệu tớ thiết bị gửi từ thiết bị nhận bởi giao thức tầng vận tải hoặc tầng ứng dụng. Tuy nhiên tầng liên kết dữ liệu không cần cung cấp dịch vụ truyền tin cậy cho các đường truyền ít lỗi (ví dụ cáp quang). Vì vậy phần lớn các giao thức tầng liên kết dữ liệu phổ biến không cung cấp dịch vụ tự tin cậy.
- Kiểm soát lưu lượng: Khả năng lưu trữ tạm thời (buffer) các frame tại các nút trên mỗi phía của đường truyền không phải là vô hạn. Đây sẽ là một vấn đề khi tốc độ tới của các frame nhanh hơn tốc độ mà nút nhận có thể xử lý được. Nếu không kiểm soát lưu lượng, bộ đệm phía nhận có thể bị tràn và frame sẽ bị mất. Giống như tầng vận tải, tầng liên kết dữ liệu cung cấp cơ chế kiểm soát lưu lượng để ngăn chặn phía gửi gửi quá khả năng nhận của phía nhận.
- Phát hiện lỗi: Nút nhận có thể nhận được bit 0 trong khi phía gửi gửi bit 1 hay ngược lại. Nguyên nhân bit bị lỗi có thể do tín hiệu bị suy hao hay nhiễu điện từ. Rõ ràng không cần chuyển đi gói dữ liệu có lỗi, nhiều giao thức tầng liên kết dữ liệu cung cấp cơ chế phát hiện lỗi. Điều này được thực hiện thông qua phía gửi thiết lập một số bit phát hiện lỗi trong frame và phía nhận thực hiện việc kiểm tra lỗi. Phát hiện lỗi và một dịch vụ rất phổ biến trong nhiều giao thức tầng liên kết dữ liệu. Trong chương 3 và chương 4 chúng ta thấy rằng tầng vận tải và tầng mạng trên Intemet cũng có khả năng phát hiện lỗi. Tuy nhiên phát hiện lỗi trong tầng liên kết dữ liệu thường phức tạp hơn rất nhiều và được triển khai bằng phần cứng.
- Sửa lỗi: sửa lỗi cũng tương tự phát hiện lỗi. Tuy nhiên không chỉ phát hiện được lỗi trong frame nhận được mà phía nhận còn có khả năng xác định chính xác nơi lỗi xuất hiện trong frame (và do đó có thể sửa được những lỗi này.
- Bán song công và song công (Half duplex, full duplex). Trong chế độ truyền song công, hai phía của đường truyền có thể đồng thời truyền dữ liệu. Trong chế độ truyền bán song công, tại một thời điểm thiết bị không thể cùng truyền và nhận.

Nhiều dịch vụ của tầng liên kết dữ liệu tương đương với nhiều dịch vụ của tầng vận tải, ví dụ cả hai tầng đều có dịch vụ truyền tin cậy, mặc dù cơ chế thực hiện dịch vụ truyền tin cậy ở cả hai tầng là như nhau, nhưng hai dịch vụ truyền tin cậy này không giống nhau. Giao thức tầng vận tải cung cấp dịch vụ truyền tin cậy giữa hai tiến trình trên cơ sở đầu cuối (end-to-end). Giao thức tầng liên kết dữ liệu cung cấp dịch vụ truyền tin cậy giữa hai nút có một kết nối vật lý trực tiếp. tương tự với dịch vụ kiểm soát lưu lượng và phát hiện lỗi.

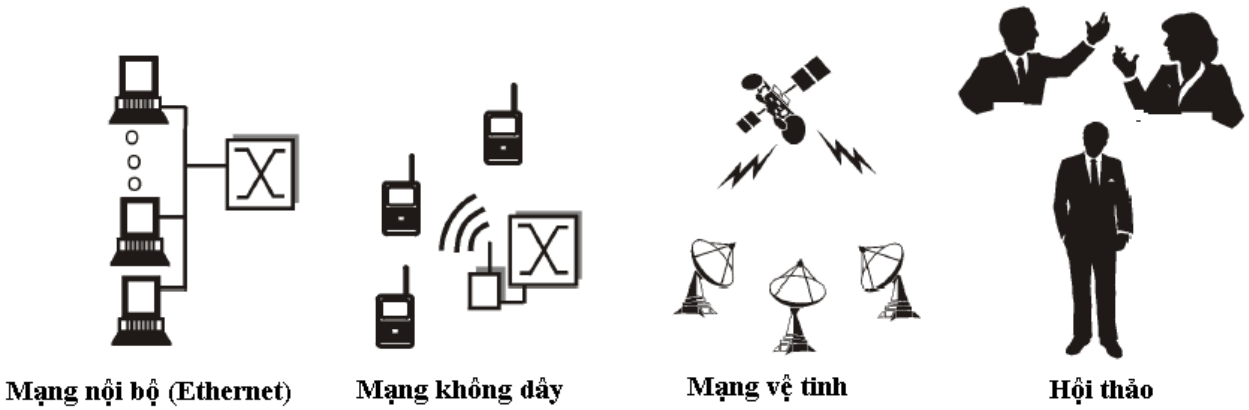
8.2 Giao thức đa truy nhập

Có hai loại đường truyền: truyền điểm-điểm và truyền quảng bá. Đường truyền điểm-điểm chỉ có một bên gửi và một bên nhận duy nhất ở hai đầu của đường truyền. Nhiều giao thức tầng liên kết dữ liệu đã được thiết kế cho đường truyền điểm-điểm như HDLC (High Data Link Control) và PPP (point-to-point

Protocol). Kiểu truyền thứ hai, kiểu quảng bá cho phép có nhiều nút gửi và nút nhận cùng kết nối đến một kênh truyền duy nhất. Thuật ngữ quảng được sử dụng ở đây vì khi bất kỳ một nút nào đó truyền đi một khung dữ liệu, kênh truyền sẽ quảng bá khung dữ liệu đó và tất cả các nút khác đều nhận được một bản sao của khung dữ liệu. Ethernet là công nghệ quảng bá được triển khai rộng rãi nhất. Trong phần này, chúng ta sẽ nghiên cứu một trong những vấn đề quan trọng nhất của tầng liên kết dữ liệu: làm thế nào để phối hợp việc truy cập vào kênh truyền chung của nhiều nút - vấn đề đa truy cập. Kênh truyền quảng bá thường được sử dụng trên mạng cục bộ - là mạng giới hạn trong một khu vực địa lý.

Phát thanh, truyền hình là những dịch vụ quảng bá, nhưng đó chỉ là quảng bá một chiều, trong khi các nút trên kênh truyền quảng bá trong mạng máy tính yêu cầu vừa có thể và nhận dữ liệu. Một ví dụ tương tự trong xã hội loài người là tại một cuộc hội thảo, mọi người tập trung trong một hội trường (không khí cung cấp môi trường quảng bá). Ví dụ thứ hai là một lớp học - nơi giáo viên và sinh viên cùng nhau chia sẻ môi trường quảng bá duy nhất. Vấn đề trung tâm trong cả hai trường hợp này là việc quyết định ai sẽ là người được nói (nghĩa là được chiếm được kênh truyền). Chúng ta đã có những quy tắc giao tiếp theo phép lịch sự để chia sẻ kênh truyền chung:

- Mỗi người đều có cơ hội nói
- Im lặng cho đến khi chưa được quyền nói
- Không được phép nói liên tục
- Giơ tay yêu cầu nếu muốn nói
- Đừng ngắt lời khi có người đang nói
- Đừng ngủ khi có người đang nói

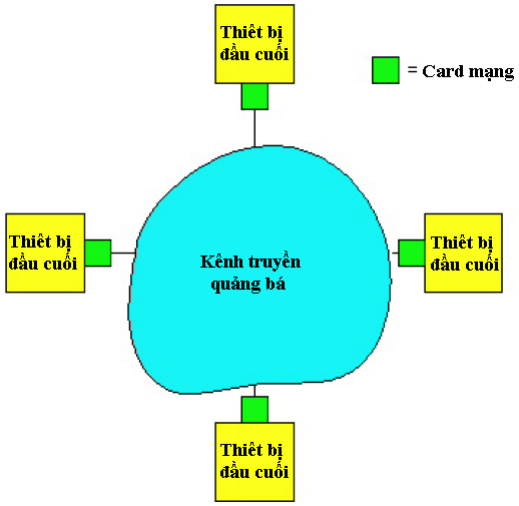


Hình 8.1 Chia sẻ kênh truyền dùng chung

Tương tự như vậy, mạng máy tính cũng có những giao thức gọi là giao thức đa truy cập (multiple access protocol) cho phép các nút điều chỉnh việc truyền thông của chúng trên kênh truyền quảng bá dùng chung. Như minh họa trên hình 8.1, giao thức đa truy cập rất cần thiết trong nhiều kiểu môi trường mạng: mạng không dây, mạng có dây và cả mạng vệ tinh. Hình 6.2 là một ví dụ về truyền quảng bá chia sẻ kênh truyền dùng chung, tất cả các nút đều có khả

năng truyền khung dữ liệu nên có thể xảy ra tình trạng tại một thời điểm nhiều nút cùng truyền có yêu cầu chiếm đường truyền. Khi đó, tất cả các nút cùng lúc nhận được nhiều khung dữ liệu, hậu quả là xảy ra xung đột đường truyền. Thông thường khi xung đột xảy ra, không nút nào có thể nhận chính xác dữ liệu vì tín hiệu trong các khung đan xen vào nhau, do đó tất cả các khung dữ liệu đang truyền trong thời điểm xung đột đều bị lỗi, tức là kênh truyền dùng chung không được sử dụng trong khoảng thời gian xảy ra xung đột. Rõ ràng khi nhiều nút thường xuyên muốn truyền dữ liệu, xác suất xảy ra xung đột sẽ lớn và phần lớn băng thông của kênh truyền bị lãng phí.

Để hiệu suất của kênh truyền quảng bá đạt giá trị tối đa khi nhiều nút muốn gửi dữ liệu cần phải có cơ chế phối hợp giữa những nút có nhu cầu truyền, cơ chế phối hợp này là trách nhiệm của giao thức đa truy cập. Đã có nhiều công trình nghiên cứu về vấn đề này, tuy nhiên người ta có thể phân loại các giao thức đa truy cập vào ba loại: giao thức phân chia kênh truyền, giao thức truy cập ngẫu nhiên và giao thức truy cập lần lượt.



Hình 8.2 Mạng quảng bá

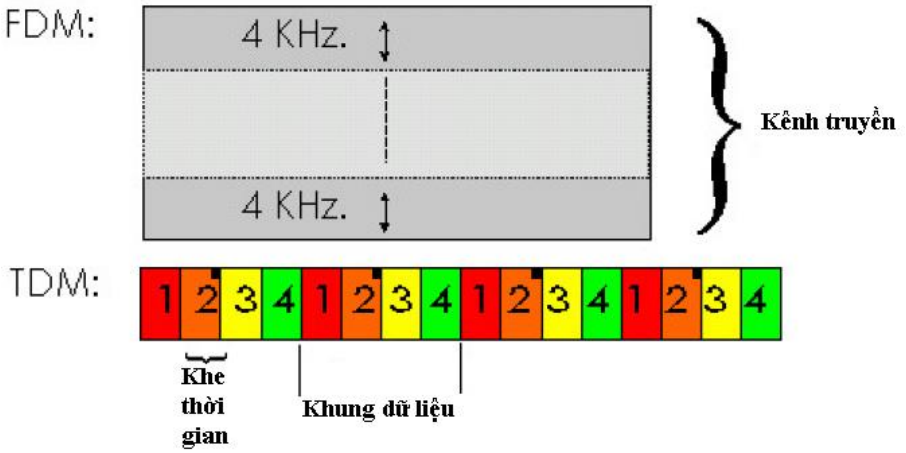
Giao thức đa truy cập trên kênh truyền quảng bá với tốc độ R b/s lý tưởng sẽ có những đặc điểm sau:

- Khi chỉ có một nút có dữ liệu gửi đi, nút đó gửi với thông lượng R b/s.
- Khi M nút có dữ liệu gửi đi, mỗi nút gửi với thông lượng R/M b/s. Yêu cầu này không có nghĩa rằng mỗi nút trong M nút luôn luôn truyền với tốc độ tức thời R/M mà đây chỉ là tốc độ trung bình xác định trong một khoảng thời gian.
- Giao thức được triển khai phân tán, nghĩa là không có một nút đóng vai trò điều phối, nếu không toàn bộ hệ thống sẽ ngừng hoạt động nếu nút điều phối bị hỏng.
- Giao thức phải đơn giản sao cho việc cài đặt không tốn kém.

8.2.1 Giao thức phân chia kênh truyền

Phân kênh theo thời gian (TDM) và theo tần số (FDM) là hai kỹ thuật có thể được sử dụng để phân chia băng thông của kênh truyền giữa tất cả các nút dùng chung kênh truyền đó. Giả sử kênh truyền hỗ trợ N nút và tốc độ truyền của kênh là R b/s. TMD chia thời gian thành các khoảng thời gian (độc lập với đơn vị dữ liệu khung ở tầng liên kết dữ liệu) và sau đó lại chia mỗi khoảng thời gian thành N khe thời gian. Mỗi khe thời gian được gán cho một trong số N nút, bất kỳ nút nào có nhu cầu gửi dữ liệu, nó truyền các bit dữ liệu của nó vào trong khe thời gian được gán cho nó trong mỗi khoảng thời gian, thường khoảng thời gian được chọn sao cho một khung dữ liệu có thể truyền trọn vẹn trong một khe thời gian. Hình 8.3 minh họa một ví dụ đơn giản về kỹ thuật TDM cho 4 nút. Trong một cuộc họp, điều này tương tự như quy định mỗi người chỉ được phát biểu trong khoảng thời gian quy định trước và sau đó phải dừng lại để người khác, như vậy mọi thành viên đều có cơ hội để phát biểu.

Kỹ thuật TDM đã loại trừ được xung đột và hoàn toàn công bằng: mỗi nút có được tốc độ truyền riêng trung bình R/N b/s. Nếu TDM dùng chung các kênh truyền quảng bá theo thời gian thì phương pháp FDM chia kênh truyền R b/s thành các tần số khác nhau (mỗi tần số có băng thông R/N) và gán một tần số đó cho mỗi nút. Vì thế, FDM tạo ra N kênh truyền nhỏ R/N b/s từ kênh truyền lớn R b/s. Tuy nhiên cả hai đều có chung nhược điểm: mỗi nút bị giới hạn bởi tốc độ trung bình R/N b/s và phải đợi đến thời gian truyền của mình ngay cả khi nó là nút duy nhất có nhu cầu gửi.



Hình 8.3 Ví dụ về phân chia theo FDM và TDM

Giao thức phân chia kênh truyền thứ ba là chia mã (CDMA - Code division multiple access). Nếu TDM và FDM gán khoảng thời gian và tần số cho các nút thì CDMA gán cho mỗi nút một mã khác nhau. Sau đó nút sử dụng mã duy nhất này để mã hoá dữ liệu gửi đi. Chúng ta sẽ thấy CDMA cho phép nhiều nút gửi đồng thời và các nút nhận tương ứng nhận đúng dữ liệu gửi cho mình, miễn là nó biết được mã của nút gửi. CDMA đã từng được sử dụng trọng hệ thống quốc phòng nhờ đặc tính chống nhiễu và ngày nay đang được áp dụng phổ biến cho mục đích dân sự, đặc biệt trong đa truy cập kênh truyền không dây.

8.2.2 Giao thức đa truy cập ngẫu nhiên

Trong giao thức truy cập ngẫu nhiên, nút truyền luôn luôn truyền dữ liệu với tốc độ cao nhất của kênh truyền R b/s. Khi có xung đột, nút liên quan đến xung đột truyền lại khung dữ liệu cho đến khi khung dữ liệu đó đến đích an toàn, việc truyền lại khung dữ liệu sẽ được thực hiện sau một khoảng thời gian ngẫu nhiên nào đó. Mỗi nút liên quan đến xung đột chọn thời gian đợi ngẫu nhiên một cách độc lập, vì thế sau mỗi xung đột xác suất hai nút cùng truyền lại cùng một lúc không lớn. Nhiều giao thức truy cập ngẫu nhiên đã được nghiên cứu, trong tài liệu này sẽ chỉ giới thiệu hai loại đại diện: giao thức ALOHA và giao thức đa truy cập cảm nhận sóng mang (CSMA/CD).

8.2.2.1 Slotted ALOHA

Nguyên tắc hoạt động của slotted ALOHA như sau:

- Tất cả khung dữ liệu có L bit.
- Thời gian được chia thành các khoảng L/R s, là khoảng thời gian đủ để truyền một khung dữ liệu.
- Nút bắt đầu truyền khung dữ liệu tại đầu mỗi khoảng thời gian.
- Tất cả các nút được đồng bộ hoá sao cho mỗi nút đều xác định được khi nào là đầu của khoảng thời gian.
- Nếu có nhiều khung dữ liệu xung đột trong khoảng thời gian nào đó thì tất cả các nút đều phát hiện sự kiện xung đột ngay trong khoảng thời gian đó.

Gọi p là xác suất (0<=p<=1). Hoạt động của slotted ALOHA trong mỗi nút như sau:

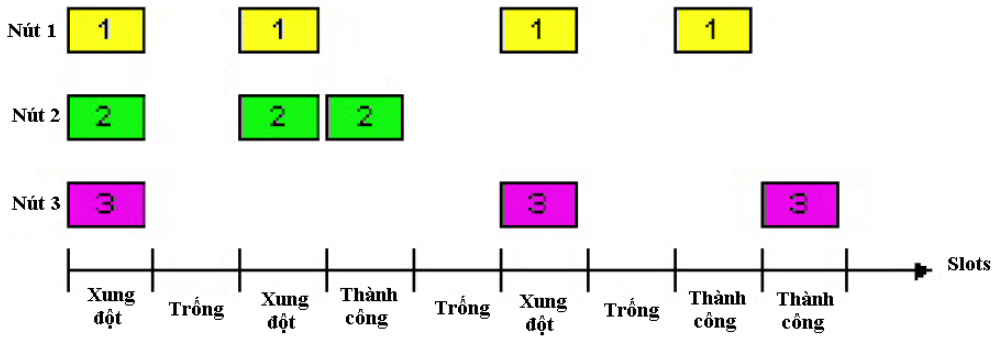
- Khi nút có khung dữ liệu mới để gửi đi, nó sẽ đợi đến thời điểm đầu của khoảng thời gian kế tiếp và gửi toàn bộ frame trong khoảng thời gian đó.
- Nếu không xảy ra xung đột, nút truyền thành công khung dữ liệu và do đó không cần thiết phải truyền lại (nút có thể chuẩn bị khung dữ liệu mới để truyền, nếu có).
- Nếu có xung đột, nút phát hiện xung đột ngay trong khoảng thời gian và sẽ truyền lại khung dữ liệu trong khoảng thời gian tiếp theo với xác suất p cho đến khi truyền thành công.

Truyền lại với xác suất p giống như việc tung đồng xu: kết quả mặt ngửa ứng với việc truyền lại xảy ra với xác suất p, kết quả mặt sấp ứng với việc bỏ qua khoảng thời gian này và tung lại đồng xu trong khoảng thời gian kế tiếp xảy ra với xác suất (1- p). Không giống phân chia kênh truyền, slotted ALOHA cho phép nút có nhu cầu gửi được phép liên tục gửi khung dữ liệu với tốc độ cao nhất của kênh truyền. Slotted ALOHA là một thuật toán phân tán vì mỗi nút khi phát hiện ra xung đột sẽ quyết định khi nào truyền lại một cách độc lập, tuy nhiên slotted ALOHA đòi hỏi phải có cơ chế đồng bộ trên tất cả các nút.

Khi không có nút nào phát dữ liệu, kênh truyền sẽ ở trạng thái trống. Nếu nhiều nút cùng đồng thời phát dữ liệu sẽ dẫn đến tình trạng xung đột và cũng không có khung dữ liệu nào được chuyển đi. Khoảng thời gian mà chỉ có duy nhất một nút gửi dữ liệu gọi là khoảng thời gian thành công, hiệu suất của giao

thức tỷ lệ các khoảng thời gian truyền thành công đó. Như vậy, nếu không có cơ chế điều khiển truy cập và nút truyền lại ngay sau mỗi lần xung đột, hiệu suất sẽ bằng 0.

Giả thiết trên mạng có N nút, mỗi nút truyền khung dữ liệu trong mỗi khoảng thời gian với xác suất p (mỗi nút luôn có một khung dữ liệu để gửi đi và khung dữ liệu này được gửi đi với xác suất p cho dù đây là khung mới hay khung dữ liệu gửi lại). Xác suất thành công của một khoảng thời gian nào đó là xác suất chỉ có một nút duy nhất truyền và N-1 nút còn lại không truyền trong khoảng thời gian đó. Xác suất một nút nào đó truyền là p, xác suất mà các nút còn lại không truyền là $(1-p)^{N-1}$. Do vậy xác suất để một nút náo đó truyền trong khi các nút khác không truyền là $p\ (1-p)^{N-1}$. Vì có N nút, nên xác suất để có khoảng thời gian thành công bằng $Np(1-p)^{N-1}$, do đó khi có N nút tích cực, hiệu suất của slotted ALOHA là $Np(1-P)^{N-1}$.



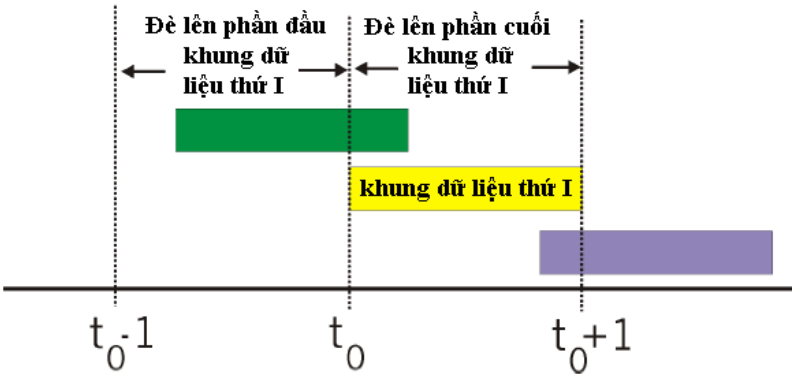
Hình 8.4 Nguyên lý hoạt động Slotted Aloha

Để đạt được hiệu suất lớn nhất, chúng ta phải xác định p* sao cho biểu thức này đạt giá trị lớn nhất. Và để đạt được hiệu suất lớn nhất khi có nhiều nút tích cực, chúng ta phải tính giới hạn của $NP^*(1-P^*)^{N-1}$ khi N tiến tới vô cùng. Áp dụng toán học, chúng ta sẽ xác định được hiệu suất lớn nhất của giao thức là 1/e = 0.37. Nghĩa là, khi nhiều nút cùng ở trạng thái tích cực thì trong điều kiện tốt nhất chỉ 37% thời gian đường truyền được sử dụng có ích. Vì vậy, tốc độ truyền hiệu quả của kênh truyền không phải là R b/s mà chỉ là 0,37R b/s. Phân tích tương tự chỉ ra rằng 37% thời gian đường truyền không được sử dụng và 26% thời gian xảy ra xung đột trên đường truyền. Như vậy một khung dữ liệu nào đó có thể được truyền với tốc độ tối đa R nhưng về tổng thể thông lượng truyền thành công của toàn bộ kênh truyền không vượt quá 0.37R.

8.2.2.2 ALOHA thuần túy

Giao thức slotted ALOHA) đòi hỏi tất cả các nút đồng bộ việc truyền tại đầu mỗi khoảng thời gian. Giao thức ALOHA thuần túy không chia khoảng thời gian, khi có dữ liệu từ lớp mạng chuyển xuống nó chuẩn bị các khung dữ liệu và ngay lập tức chuyển khung dữ liệu đầu tiên lên kênh truyền quảng bá. Nếu xảy ra xung đột thì ngay sau khi truyền xong khung dữ liệu này, nút đó ngay lập tức truyền lại khung dữ liệu đó với xác suất p. Nếu không có xung đột xảy ra, nút

mạng đợi trong một khoảng thời gian bằng với thời gian truyền khung dữ liệu, sau thời gian này, nút mạng truyền khung dữ liệu kế tiếp với xác suất là p, hoặc đợi một khoảng thời gian truyền khung dữ liệu với xác suất (1-p).



Hình 8.5 Nhiễu truyền dẫn trong ALOHA thuần túy

Giả thiết như trong trường hợp slotted ALOHA thời gian truyền khung dữ liệu là một đơn vị thời gian. Tại bất kì thời điểm nào, xác suất để nút truyền khung dữ liệu là p. Giả sử khung dữ liệu này bắt đầu truyền tại thời điểm t_0 như minh họa trong hình 8.5, để khung dữ liệu này được truyền thành công thì truyền trong khoảng thời gian $[t_0 - 1, t_0]$ không nút nào chiếm đường truyền. Xác suất để tất cả các nút khác không được bắt đầu truyền trong khoảng thời gian này là $(1-p)^{N-1}$. Tương tự, xác xuất để không nút nào được bắt đầu truyền trong khi nút đang xét đang truyền cũng là $(1-p)^{N-1}$. Vì vậy xác suất nút nào đó truyền thành công là $p(1-p)^{2(N-1)}$. Bằng cách lấy giới hạn như trong trường hợp slotted ALOHA, chúng ta thấy rằng hiệu suất lớn nhất của giao thức ALOHA là l/(2e) - bằng một nửa của slotted ALOHA.

8.2.2.3 Đa truy cập cảm nhận sóng mang

Trong cả hai giao thức ALOHA thuần túy và slotted ALOHA, quyết định truyền của một nút mạng được đưa ra độc lập với các nút khác. Cụ thể hơn, một nút không cần phải kiểm tra xem đường truyền bận hay rỗi. Giao thức ALOHA giống như hành vi của người bất lịch sự cứ thích là nói mà không cần để ý có ai đó đang nói hay không. Xã hội loài người có những quy tắc ứng xử cho phép xử sự một cách lịch sự và làm giảm "xung đột" với những người khác. Đặc biệt, có hai quy tắc quan trọng cho một cuộc đối thoại của người lịch sự:

- Nghe trước khi nói: nếu có ai đang nói, hãy đợi đến khi họ nói xong. Trong mạng máy tính, điều này được gọi là cảm nhận sóng mang (carrier sense) – một nút phải nghe kênh truyền trước khi truyền. Nếu kênh truyền bận thì nút sẽ chờ một khoảng thời gian ngẫu nhiên sau đó lại nghe kênh truyền. Sau khoảng thời gian chờ đó, nếu kênh truyền rỗi thì nút mạng mới bắt đầu truyền khung dữ liệu, nếu không lại đợi một khoảng thời gian ngẫu nhiên khác và quá trình lặp lại tương tự.
- Nghe trong khi nói: Nếu có ai đó đang nói mà thấy người khác nói thì tạm ngừng nói ngay lập tức. Trong mạng máy tính điều này được gọi là phát hiện xung đột, nút đang truyền tiếp tục lắng nghe kênh truyền trong khi đang

truyền. Nếu phát hiện có nút khác truyền xen vào, nút sẽ dừng truyền và sử dụng giao thức nào đó để quyết định khi nào sẽ thử truyền tiếp.

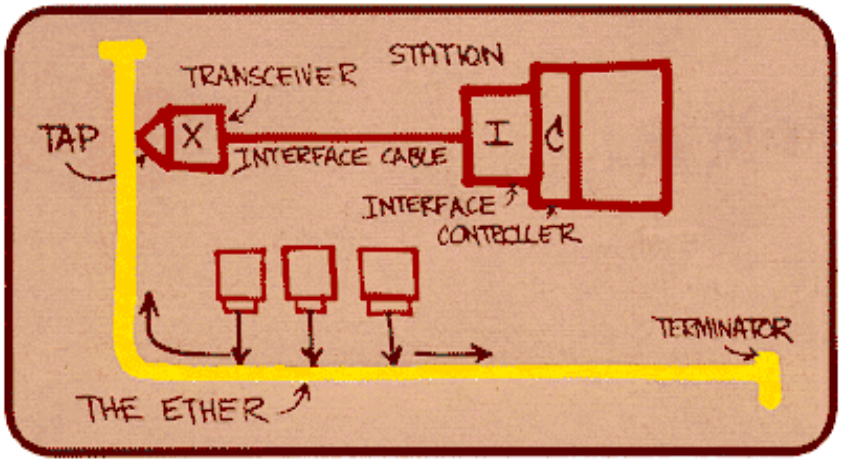
Hai quy tắc trên là ý tưởng chủ đạo của giao thức CSMA (Carrier Sense Multuple Access) và CSMA/CD (CSMA with Collision Detection). Có nhiều biến thể của CDMA và CDMA/CD đã được đưa ra với việc thực hiện các chiến lược chờ đợi khác nhau, công nghệ Ethernet sử dụng giao thức này. Ngay cả khi tất cả các nút thực hiện cảm nhận sóng mang thì xung đột vẫn có khả năng xuất hiện, nguyên nhân là do độ trễ của tín hiệu khi lan truyền trên mạng.

8.3 Các công nghệ kết nối

8.3.1 Công nghệ Ethernet

Hiện nay Ethernet gần như thống trị thị trường mạng cục bộ. Mới chỉ đầu những năm 1980 đến đầu những năm 1990, Ethernet còn phải đối đầu với nhiều thách thức từ những công nghệ LAN khác như FDDI, token-ring, ATM. Ra đời vào giữa những năm 70, Ethernet liên tục phát triển hoàn thiện và vượt xa các công nghệ LAN khác và trong tương lai gần ít có khả năng công nghệ khác thay thế được Ethernet.

Có rất nhiều lý do dẫn đến sự thành công của Ethernet. Thứ nhất, Ethernet là mạng cục bộ tốc độ cao được triển khai rộng rãi đầu tiên. Được triển khai tương đối sớm nên các nhà quản trị mạng lập tức trở nên quen thuộc với Ethernet nên ngại chuyển sang những công nghệ LAN mới. Thứ hai, so với các công nghệ khác như: token-ring, FDDI, ATM…, công nghệ Ethernet tương đối dễ lắp đặt và giá thành rẻ. Thứ ba, lý do chính đáng nhất để sử dụng các công nghệ LAN khác (FDDI hay ATM) là do công nghệ mới có tốc độ cao hơn, tuy nhiên Ethernet liên tục nâng cấp về tốc độ và đồng thời đảm bảo tính tương thích giữa các tốc độ khác nhau.



Hình 8.6 Bản phác họa mạng Ethernet của Bob Metcalf

Kiến trúc Ethernet được Bob Metcalf đưa ra vào khoảng giữa những năm 70. Hình 6.6 là sơ đồ nguyên gốc về ý tưởng mạng Ethernet, theo đó: các máy tính được kết nối với nhau bằng một kênh truyền chung tương tự như hệ thống ống dẫn nước sinh hoạt trong một khu tập thể. Mỗi máy tính kết nối vào mạng

phải có bộ phận tiếp giáp gọi là giao diện mạng (NIC – Network Interface Controler).

Ethernet có thể chạy trên hình trạng vật lý dạng bus hoặc dạng sao, môi trường truyền dẫn có thể là cáp đồng hoặc cáp quang, tốc độ 10 Mbit/s, 100 Mbit/s hay 1 Gbit/s, thậm chí đang thử nghiệm triển khai tốc độ 10 Gbit/s.

8.3.1.1 Cấu trúc khung dữ liệu Ethernet

Các công nghệ Ethernet khác nhau có mặt trên thị trường hiện nay đều có chung cấu trúc khung dữ liệu, cho dù đó là công nghệ Ethernet sử dụng cáp đồng trục hay cáp quang, chạy với tốc độ 10 Mbps, 100 Mbps hay 1 Gbps.

Ethernet					
8	6	6	2	46 to 1500	4
Preamble	Destination Address	Source Address	Type	Data	Frame Check Sequence

Hình 8.7 Khuôn dạng khung dữ liệu Ethernet

Hình 8.7 thể hiện cấu trúc khung dữ liệu Ethernet:

- Preamble (mào đầu, 8 byte): Khung dữ liệu Ethernet bắt đầu bằng trường preamble dài 8 byte, trong đó 7 byte đầu tiên có giá trị là 10101010, byte thứ tám có giá trị 10101011 . Bảy byte đầu tiên của phần mở đầu làm nhiệm vụ đánh thức Card mạng bên nhận và đồng bộ hoá đồng hồ bên gửi với đồng hồ bên nhận. Tại sao các đồng hồ lại không đồng bộ hoá? Chú ý rằng Card mạng A truyền frame với tốc độ 10 MBPS, 100 MBPS hay 1 Gbps phụ thuộc vào kiểu Ethernet. Tuy nhiên, bởi vì không có gì là tuyệt đối hoàn toàn nên Card mạng A chưa chắc đã truyền khung dữ liệu với tốc độ xác định mà với tốc độ nào đó. Card mạng nhận có thể chốt đồng hồ của Card mạng A bằng cách chốt tất cả các bit trong 7 byte đầu tiên. Hai bit cuối cùng trong byte thứ 8 (hai bit 1 liên tiếp nhau) báo cho Card mạng B biết rằng dữ liệu chính thức chuẩn bị đến. Khi máy tính B thấy hai bit 1 liên tiếp nhau, nó biết rằng 6 byte tiếp theo là địa chỉ đích. Card mạng có thể phát hiện khung dữ liệu đã được truyền xong khi không thấy dòng điện.
- Destination Address (Địa chỉ đích, 6 byte): Trường này chứa địa chỉ vật lý của Card mạng nhận, BB-BB- BB-BB-BB-BB. Khi Card mạng B nhận bất kỳ khung dữ liệu nào, nó sẽ kiểm tra địa chỉ đích của khung đó. Nếu địa chỉ đích là BB-BB-BB- BB-BB-BB (địa chỉ của chính nó), hoặc địa chỉ quảng bá LAN (FF- FF-FF-FF-FF-FF) thì Card mạng mới chuyển gói tin trong trường dữ liệu của khung lên tầng mạng.
- Source Address (Địa chỉ nguồn, 6 byte): Trường này chứa địa chỉ vật lý của Card mạng gửi khung dữ liệu, trong ví dụ này là AA-AA-AA-AA-AA-AA.
- Type (Trường kiểu, 2 byte): Trường này cho phép Ethernet hỗ trợ nhiều giao thức tầng mạng khác nhau. Cần chú ý rằng máy tính có thể sử dụng

nhiều giao thức tầng mạng không chỉ là IP. Trên thực tế, máy tính nào đó có thể hỗ trợ nhiều giao thức tầng mạng và sử dụng các giao thức khác nhau cho những ứng dụng khác nhau. Vì thế khi nhận được một khung Ethernet, Card mạng B cần xác định giao thức tầng mạng nào sẽ nhận nội dung của trường dữ liệu. Những giao thức tầng mạng như IP, Novell IPX hoặc APPLETALK đều có một mã định danh (là một số) đã được chuẩn hóa. hơn nữa, giao thức ARP cũng có một định danh. Trường kiểu tương tự với trường protocol trong gói dữ liệu IP hay trường số hiệu cổng trong tầng vận tải; mục đích của tất cả các trường này là kết hợp giao thức ở tầng dưới với giao thức ở tầng trên nó.

- Data (Trường dữ liệu, từ 46 đến 1500 byte): trường này chứa gói dữ liệu IP, MTU (Maximum Transfer Unit) của Ethernet là 1500 byte. Điều này có nghĩa là nếu gói dữ liệu IP vượt quá 1500 byte thì máy tính phải chia nhỏ gói dữ liệu ra. Kích thước tối thiểu của trường này là 46 byte. Điều này có nghĩa là nếu gói dữ liệu nhỏ hơn 46 byte, trường dữ liệu phải được chèn thêm một số dữ liệu giả cho đủ 46 byte. Khi bên gửi chèn thêm dữ liệu vào thì tầng mạng ở bên nhận cũng nhận được cả gói dữ liệu IP dẫn dữ liệu được chèn thêm vào, khi đó nó phải sử dụng trường độ dài trong gói dữ liệu IP để loại bỏ phần thêm vào.
- CRC (Mã kiểm tra dư thừa vòng - Cyclic Redundancy Check, 4 byte): mục đích của trường CRC là cho phép Card mạng phát hiện lỗi trong khung dữ liệu nhận được. Nguyên nhân lỗi bit là do hiện tượng suy hao năng lượng điện từ của tín hiệu hay tỏa nhiệt trong Card mạng Ethernet hay cáp mạng. Việc phát hiện lỗi được thực hiện như sau. Khi tạo ra khung dữ liệu Ethernet, máy tính A tính giá trị trường CRC dựa trên trường dữ liệu thực sự Công việc kiểm tra tại B xem dữ liệu thực sự và CRC có mâu thuẫn không được gọi là kiểm tra CRC. Nếu nếu giá trị trong CRC không phù hợp với phần dữ liệu thì máy tính B xác định trong khung dữ liệu nhận được đã có lỗi xuất hiện.

8.3.1.2 Dịch vụ truyền số liệu không liên kết

Tất cả công nghệ Ethernet cung cấp cho tầng mạng dịch vụ không liên kết. Nghĩa là khi Card mạng A muốn gửi gói dữ liệu đến Card mạng B, Card mạng A sẽ đặt gói dữ liệu trong khung dữ liệu và gửi lên mạng mà không phải bắt tay trước với Card mạng B. dịch vụ không kết nối ở tầng này tương tự với dịch vụ IP ở tầng Internter và dịch vụ UDP ở tầng vận tải.

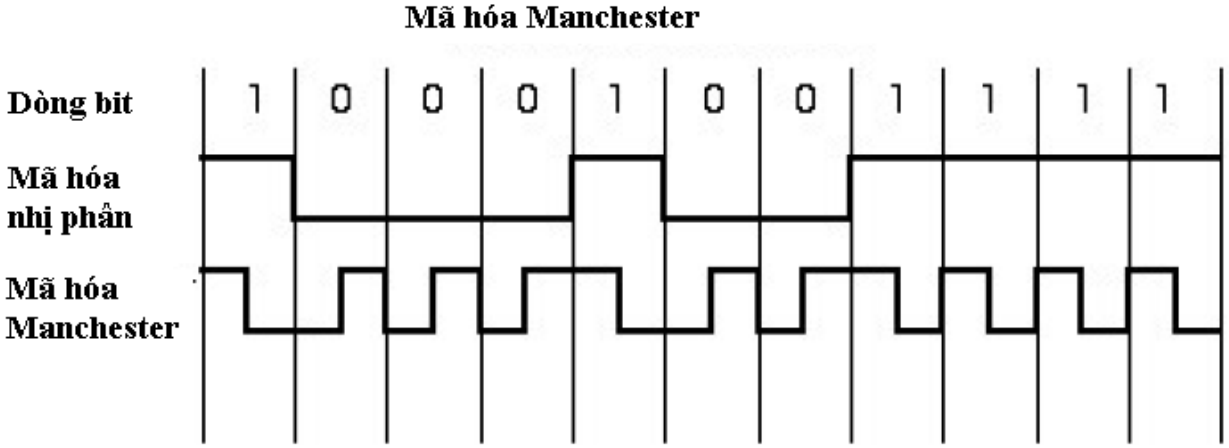
Công nghệ Ethernet cung cấp dịch vụ không tin cậy cho tầng mạng. Cụ thể, khi nhận được khung dữ liệu từ Card mạng A, Card mạng B sẽ không gửi phản hồi cho A. Card mạng A không thể xác định liệu khung dữ liệu nó truyền đi có được nhận đúng hay không. Nếu phát hiện lỗi khi kiểm tra CRC, Card mạng B sẽ loại bỏ khung dữ liệu. Chính điều này giúp Ethernet đơn giản và rẻ. Nhưng dòng dữ liệu chuyển tới tầng mạng có thể bị gián đoạn.

Nếu có sự gián đoạn do một số khung dữ liệu Ethernet bị loại bỏ, giao thức tầng úng dụng tại máy B có phát hiện được sự gián đoạn đó không? Điều này phụ thuộc việc ứng dụng sử dụng UDP hay TCP. Nếu ứng dụng dùng UDP

thì giao thức tầng ứng dụng trong máy B sẽ không phát hiện được gián đoạn trong dữ liệu. Mặt khác, nếu ứng dụng dùng TCP thì thực thể TCP trong máy B sẽ không gửi biên nhận cho những dữ liệu đã bị loại bỏ, do vậy thực thể TCP trong máy A phải gửi lại. Khi TCP gửi lại dữ liệu, thì cuối cùng dữ liệu cũng sẽ đi qua các Card mạng Ethernet, tức là khung dữ liệu đã được truyền lại nhưng Ethernet coi đó là một gói dữ liệu mới.

8.3.1.3 Dải tần cơ sở và mã hoá Manchester

Ethernet sử dụng băng tần cơ sở (baseband) nghĩa là Card mạng gửi tín hiệu số trực tiếp vào kênh truyền dùng chung. Card mạng không chuyển tín hiệu sang đại tần số khác như trong một số công nghệ khác. Ethernet sử dụng mã hoá Manchester (Hình 8.8).



Hình 8.8 Mã hóa Manchester

Trong phương pháp mã hoá Manchester, mỗi bít ứng với một quá trình chuyển trạng thái xung truyền: bit 1 chuyển từ trên xuống dưới, bit 0 chuyển từ dưới lên trên. Lý do sử dụng mã hoá Manchester là đồng hồ của card mạng gửi và nhận không đồng bộ hoàn toàn với nhau. Khi xuất hiện sự chuyển ngay trong phần giữa mỗi bit, máy tính nhận có thể đồng bộ đồng hồ của nó với đồng hồ của máy tính gửi. Sau khi đồng hồ của card mạng nhận được đồng bộ hoá, phía nhận có thể thu được tín hiệu của mỗi bit và xác định nó là 0 hay 1 . Mã hoá Manchester được sử dụng nhiều trong tầng vật lý chứ không phải trong tầng liên kết dữ liệu.

8.3.1.4 CSMA/CD

Các nút trên mạng cục bộ Ethernet được kết nối qua một kênh truyền quảng bá dùng chung, vì vậy khi card mạng gửi đi một khung dữ liệu, tất cả các card mạng trên LAN đều nhận được khung đó. Ethernet dùng thuật toán đa truy cập CSMA/CD, nó sử dụng các cơ chế sau:

- Card mạng có thể bắt đầu truyền tại bất kì thời điểm nào, nghĩa là không chia khoảng thời gian.
- Card mạng không bao giờ truyền khung dữ liệu khi nó nhận thấy Card mạng khác đang truyền.

- Card mạng đang truyền sẽ ngừng ngay lập tức nếu phát hiện ra Card mạng khác cũng đang truyền.
- Trước khi cố gắng thử truyền lại, Card mạng đợi một khoảng thời gian ngẫu nhiên theo thuật toán backoff.

Những cơ chế này giúp hiệu suất của CSMA/CD được cải thiện đáng kể so với slotted ALOHA khi vận hành trong môi trường mạng LAN. Trong thực tế, nếu thời gian trễ để tín hiệu lan truyền giữa hai nút là rất nhỏ thì hiệu suất của CSMA/CD có thể đạt tới 100%. Cơ chế thứ hai và thứ ba kể trên yêu cầu Card mạng Ethernet có khả năng cảm nhận được khi nào thì có một Card mạng khác đang truyền và phát hiện xung đột trong khi truyền. Card mạng Ethernet thực hiện hai nhiệm vụ này bằng việc đo mức điện áp trước và trong khi truyền. Card mạng dùng giao thức CSMA/CD không cần kết hợp với Card mạng khác trên Ethernet.

Trên một Card mạng, giao thức CSMA/CD làm việc như sau:

- Card mạng nhận gói tin từ tầng mạng, tạo ra khung dữ liệu Ethernet và đặt vào trong bộ đệm.
- Nếu Card mạng thấy kênh truyền rỗi (không có tín hiệu trên kênh truyền – cường độ dòng điện bằng 0) thì Card mạng bắt đầu truyền. Nếu Card mạng thấy kênh truyền bận, nó sẽ đợi cho đến khi phát hiện thấy kênh truyền rỗi.
- Trong khi truyền, Card mạng kiểm tra xem có năng lượng tín hiệu đến từ Card mạng khác hay không. Nếu không phát hiện được năng lượng sau khi đã truyền xong khung dữ liệu thì khung dữ liệu đóó được xem là đã truyền thành công.
- Nếu Card mạng phát hiện năng lượng tín hiệu từ Card mạng khác trong khi đang truyền thì lập tức nó dừng lại không truyền và gửi đi tín hiệu báo nhiễu 32 bit (tín hiệu JAM). Mục đích của tín hiệu báo nhiễu là bảo đảm tất cả các Card mạng đang truyền khác đều phát hiện ra xung đột.
- Sau khi dừng phát và gửi tín hiệu báo nhiễu, Card mạng sẽ thực hiện thuật toán exponential backoff. Khi truyền frame nào đó, nếu thấy frame đó bị xung đột n lần liên tiếp, Card mạng chọn một giá trị ngẫu nhiên K trong khoảng $(0,1,2...2^m-1)$ với $m = \min(n,10)$. Sau đó Card mạng sẽ đợi K*512 trước khi quay lại bước 2.

Xét ví dụ sau: giả sử adapter A bắt đầu truyền đi một khung dữ liệu và ngay trước khi tín hiệu từ A tới được Card mạng B, Card mạng B bắt đầu truyền. Do vậy B chỉ truyền được vài bit trước khi dừng lại không truyền tiếp; Vài bit này sẽ lan tỏa được đến A, nhưng chúng không tạo đủ năng lượng để A có thể phát hiện xung đột. Để đảm bảo A phát hiện được xung đột, B phải truyền thêm tín hiệu báo nhiễu dài khoảng 32 bit.

Trong trường hợp xảy ra xung đột, thuật toán exponetial backoff sẽ tính toán thời gian Card mạng sẽ phải chờ bao lâu trước khi nghe đường truyền để tiếp tục phát lại. Giả sử adapter lần đầu tiên truyền đi một frame và trong khi truyền phát hiện có xung đột. Sau đó Card mạng sẽ chọn K=0 với xác suất 0,5 và chọn K=1 với xác suất 0,5. Nếu Card mạng chọn K=0 thì ngay lập tức nó sẽ nhảy đến bước

2 sau khi truyền đi tín hiệu báo nhiễu. Nếu Card mạng chọn K=l thì nó sẽ đợi 51,2 microsecond trước khi quay lại bước 2.

Sau xung đột lần thứ hai, K được chọn ngẫu nhiên giữa các giá trị (0,1,2,3) với xác suất bằng nhau, sau ba xung đột. K sẽ được chọn ngẫu nhiên giữa các giá trị (0, 1, 2, . . . ,7) với xác suất bằng nhau, sau nhiều hơn mười xung đột K được chọn ngẫu nhiên giữa các giá trị (0, 1, 2, . . . , 1023) với xác suất bằng nhau.. Như vậy tổng số giá trị mà K có thể lựa chọn tăng theo luỹ thừa cơ số 2 với số mũ là số lần xung đột cho (cho đến khi N=10). Chuẩn Ethernet ấn định giới hạn khoảng cách giữa hai nút trên mạng. Giới hạn này bảo đảm rằng nếu Card mạng A chọn giá trị K thấp hơn giá trị K của tất cả các Card mạng khác liên quan đến xưng đột trong pha trước thì A có thể truyền đi khung dữ liệu mà không bị xung đột nữa.

Việc sưr dụng thuật toán backoff với hàm số mũ sẽ tạo ra những khoảng thời gian chờ không giống nhau. Sau khi Card mạng gặp xung đột lần đầu tiên, nó không hình dung được có bao nhiêu adapter liên quan đến xung đột đó. Nếu chỉ có một số lượng nhỏ Card mạng thì chắc chắn K sẽ được chọn trong một tập hợp hạn chế. Ngược lại, nếu có nhiều Card mạng liên quan thì K được chọn trong một tập hợp lớn hơn. Bằng cách tăng kích cỡ của tập hợp sau mỗi xung đột, adapter sẽ thích nghi được với nhiều hoàn cảnh. Mỗi lần Card mạng chuẩn bị khung dữ liệu mới để gửi đi, nó sử dụng thuật toán CSMA/CD nói trên, Card mạng không quan tâm từ bất kỳ xung đột nào trước đó, do vậy, rất có khả năng Card mạng với khung dữ liệu mới có thể truyền xen vào trong khi một vài Card mạng khác đang trong trạng thái chờ.

8.3.1.5 Hiệu suất Ethernet

Khi chỉ có một nút truyền số liệu, nút đó có thể truyền với tốc độ tối đa (10 Mbps, 100Mbps hoặc 1 Gbps). Tuy nhiên nếu nhiều nút cùng truyền thì tốc độ truyền thành công (effective rate) của kênh truyền có thể giảm đi đáng kể. Hiệu suất của Ethernet là tỉ lệ thời gian không có xung đột trên kênh truyền khi có nhiều nút truyền số liệu, mỗi nút cần truyền nhiều khung dữ liệu trong một khoảng thời gian nhất định. Để xác định hiệu suất gần đúng của Ethernet, giả sử t_{prop} là thời gian lớn nhất năng lượng tín hiệu lan tỏa giữa hai Card mạng. Giả sử t_{trans} là thời gian để truyền đi một frame Ethernet với độ lớn cực đại (xấp xỉ 1,2 ms với Ethernet 10 Mbps), ta sử dụng công thức:

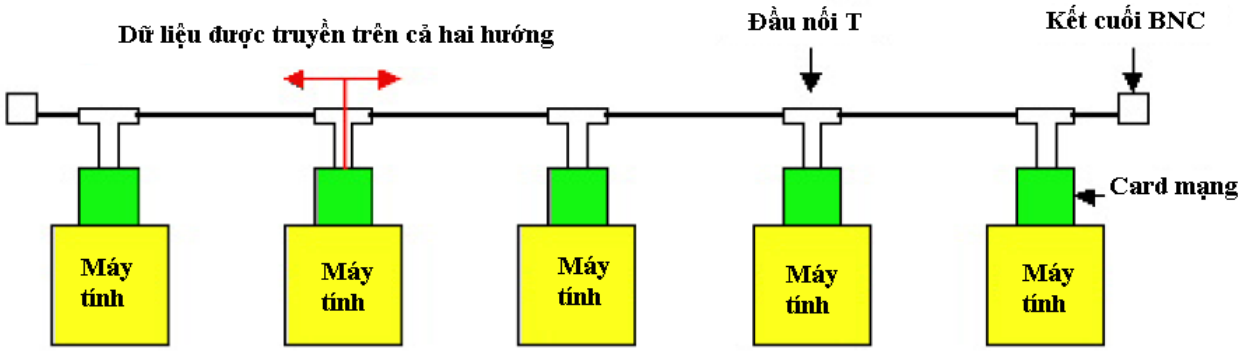
$$\text{Efficiency} = \frac{1}{1 + t_{prop}/t_{trans}}$$

Từ công thức này chúng ta thấy nếu t_{prop} đạt tới 0 thì hiệu suất đạt tới 1. Điều này cũng rất hợp lý: nếu thời gian trễ là 0 thì các nút sẽ phát hiện ra xung đột ngay lập tức và do đó không lãng phí kênh truyền. Khi t_{trans} trở lên rất lớn, hiệu suất đạt từ 1, điều này cũng hiển nhiên vì khi Card mạng có được kênh truyền nó sẽ chiếm dụng kênh truyền trong khoảng thời gian dài, như vậy kênh truyền hầu như lúc nào cũng trong trạng thái làm việc.

8.3.1.6 Các công nghệ Ethernet

Công nghệ Ethernet sử dụng cáp đồng trục (coaxial cable) có hình trạng dạng bus, tốc độ truyền là 10 Mbps. Sử dụng cặp dây xoắn, hình trạng hình sao, tốc độ truyền từ 10 đến 1000Mbps. Gigabyte Ethernet sử dụng cả sợi quang hay dây đồng xoắn, truyền với tốc độ 1 Gbps. Những công nghệ Ethernet này được chuẩn hoá bởi IEEE 802.3. Vì thế LAN Ethernet thường được gọi là 802.3 LAN. Tín hiệu truyền trên đường dây thường bị suy hao theo khoảng cách, do đó đến một khoảng cách nào đó sẽ mất tín hiệu, vì vậy cần phải có thiết bị tái tạo lại tín hiệu trước khi không nhận ra tín hiệu đó, thiết bị thực hiện chức năng này gọi là bộ lặp (Repeater). Bộ lặp là thiết bị tầng vật lý xử lý trên từng bit riêng lẻ chứ không phải trên khung dữ liệu. Khi tín hiệu (biểu diễn bit 0 hoặc 1) đến từ một cổng, bộ lặp tái tạo lại tín hiệu này bằng cách tăng cường độ năng lượng của tín hiệu và gửi tín hiệu đó qua tất cả các cổng còn lại. Bộ lặp được sử dụng rộng rãi trong LAN để mở rộng phạm vi mạng, bộ lặp không có khả năng cảm nhận sóng mang hay thực hiện bất kỳ một chức năng nào của CSMA/CD, nó chỉ tái tạo và gửi tín hiệu đến từ một cổng đến tất cả các cổng khác, kể cả trong trường hợp các cổng kia cũng đang có tín hiệu gửi đến.

Ethernet 10BASE2

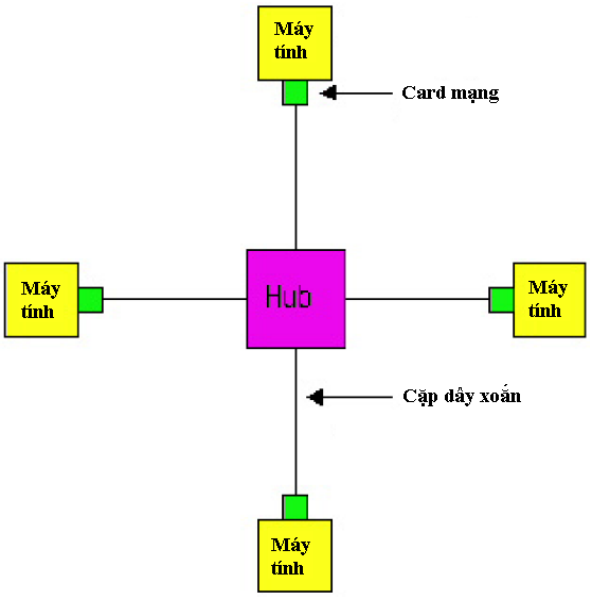


Hình 8.9 Ethernet 10Base2

10Base2 là một công nghệ Ethernet rất phổ biến trong những năm 90 của thế kỷ trước. Số 10 trong 10base2 có nghĩa là tốc độ truyền tối đa là 10 Mbit/s, số 2 có ý nghĩa khoảng cách tối đa giữa hai trạm không có bộ lặp ở giữa không vượt quá 200m. Hình 8.9 minh họa mạng Ethernet 10BASE2 có hình trạng dạng bus, các Card mạng được kết nối trực tiếp vào một môi trường dùng chung - cáp đồng trục. Khi Card mạng gửi đi một khung dữ liệu, khung đó sẽ được truyền qua đầu nối chữ T, sau đó sẽ lan tỏa theo cả hai hướng của dây dẫn. Trên đường đi, mỗi Card mạng sẽ thu được các tín hiệu của khung dữ liệu đó. Khi đến điểm cuối cùng của dây dẫn, tất cả các tín hiệu sẽ bị kết cuối BNC hấp thụ. Nếu không có bộ lặp, độ dài tối đa của bus là 185m. Nếu bus có độ đài lớn hơn, suy hao tín hiệu sẽ làm hệ thống hoạt động không chính xác. Ngoài ra nếu không có repeater, số lượng nút tối đa là 30. Người ta sử dụng bộ lặp để nối các đoạn 10base2 liên tiếp nhau, mỗi đoạn có thể có 30 máy và dài không quá 185m.

Chỉ có thể sử dụng tối đa 4 bộ lặp, nếu nhiều hơn sẽ thường xuyên xảy ra xung đột.

Ethernet 10BaseT và 100BaseT



Hình 8.10 Hình trạng dạng sao 10BaseT và 100 BaseT

10BaseT và 100BaseT là hai công nghệ tương tự nhau, điểm khác biệt quan trọng nhất là tốc độ truyền của 10BaseT là 10Mbit/s trong khi tốc độ truyền của Ethernet 100BaseT là 100Mbit/s. 10BaseT và 100BaseT là công nghệ được sử dụng rất phổ biến hiện nay, chúng có topo dạng sao, như minh họa trên hình 8.10.

Trong topo hình sao có một thiết bị trung tâm được gọi là hub (đôi khi gọi là bộ tập trung – concentrator), thực chất đó là bộ lặp nhiều cổng. Card mạng trên mỗi nút có kết nối trực tiếp đến hub. Kết nối này gồm hai lặp dây đồng xoắn đôi, một để truyền và một để nhận. Tại mỗi đầu của kết nôi có một bộ nối theo chuẩn RJ-45 – tương tự như đầu nối chuẩn RJ-11 được sử dụng cho điện thoại thông thường. Chữ "T" trong 10BaseT và 100BaseT là viết tắt của "Twisted pair" nghĩa là cặp dây xoắn.

Đối với 10BaseT và 100BaseT, khoảng cách tối đa giữa Card mạng và hub là 100m, vì vậy độ dài lớn nhất giữa hai nút là 200m. Khoảng cách này có thể được tăng nếu sử dụng các thiết bị như hub, bridge, switch. Về bản chất, bộ tập trung và bộ lặp giống nhau vì khi nhận được tín hiệu từ Card mạng chúng đều gửi tín hiệu đó đến tất cả các Card mạng khác. Theo cách này, mỗi Card mạng có thể cảm nhận kênh truyền để xác định liệu kênh truyền có rỗi không và đồng thời phát hiện xung đột trong khi đang truyền dữ liệu.

Nhiều Card mạng Ethernet hỗ trợ các tốc độ 10/100 Mbps, tức là chúng ta có thể sử dụng được dùng cả hai kiểu Ethernet: 10BaseT và 100BaseT. 10BaseT, đặc trưng của nó là sử dụng cáp xoắn kiểu 5. Khác 10Base2 và 10BaseT, 100BaseT không sử dụng phương pháp mã hoá Manchester sử dụng phương pháp 4B5B có hiệu suất cao hơn: mỗi nhóm 5 chu kỳ đồng hồ được sử

dụng để mã hóa 4 bit và cung cấp đủ thông tin cho phép đồng bộ hoá đồng hồ. Cả hai công nghệ 10BaseT và 100BaseT đều có thể sử dụng cáp quang, tuy giá thành cao nhưng khả năng chống nhiễu tốt.

Gigabit Ethernet

Gigabit Ethernet là sự mở rộng của chuẩn Ethernet 10BaseT và 100BaseT. Với tốc độ truyền lên tới 1000 Mbit/s, Gigabit Ethernet vẫn duy trì khả năng tương thích hoàn toàn với các thiết bị Ethernet kiểu cũ. Chuẩn Gigabit Ethernet (IEEE802.3x), thực hiện các công việc sau:

- Sử dụng khuôn dạng khung Ethernet chuẩn, tương thích với công nghệ 10BaseT và 100BaseT . Điều này cho phép dễ dàng tích hợp Gigabit Ethernet vào các cơ sở đã cài đặt các thiết bị Ethernet.
- Cho phép đường truyền điểm-điểm cũng như kênh truyền quảng bá. Đường truyền điểm-điểm dùng switch trong khi kênh truyền quảng bá sử dụng hub giống 10BaseT và 100BaseT.
- Sử dụng CSMA/CD cho kênh truyền quảng bá dùng chung. Để đạt được hiệu suất mong muốn, khoảng cách lớn nhất giữa các nút bị hạn chế chặt chẽ.
- Kênh truyền điểm-điểm có đặc tính song công, mỗi hướng truyền với tốc độ 1 Gbps.

Giống như 10BaseT và 100BaseT, Ethernet Gigabit có hình trạng dạng sao, Gigabit Ethernet thường được sử dụng trên các trục chính kết nối nhiều mạng cục bộ Ethernet 10BaseT và 100BaseT. Gigabit Ethernet có thể sử dụng loại cáp 5UTP hoặc cáp quang.

8.3.2 Kết nối mạng diện rộng

8.3.2.1 Giao thức PPP

PPP là giao thức được sử dụng chủ yếu khi người dùng truy cập Internet từ nhà thông qua đường điện thoại quay số, do đó PPP là một trong những giao thức tầng nên kết dữ liệu được sử dụng nhiều nhất nhất ngày nay. Giao thức quan trọng thứ hai là HDLC (High Level Data Link Control). Giao thức PPP được trình bầy tương đối đơn giản với mục đích khảo sát một số tính năng quan trọng nhất của lớp giao thức điểm nối điểm ở tầng liên kết dữ liệu.

Giao thức PPP là giao thức tầng liên kết dữ liệu trên kênh truyền nối trực tiếp giữa hai nút - mỗi nút ở một đầu của đường truyền. Đường truyền PPP có thể là đường điện thoại quay số (ví dụ kết nối modem 56k), đường truyền SONET, kết nối X.25 hoặc mạng ISDN. Như đã nói trên, PPP chủ yếu được lựa chọn để kết nối máy tính gia đình đến ISP thông qua đường dây điện thoại. IETF đã đặt ra cho mọi thiết kế của PPP :

- Đóng gói gói tin (Framing): phía gửi trong giao thức PPP phải có khả năng lấy gói tin ở tầng mạng, đặt nó trong khung dữ liệu tầng liên kết dữ liệu. Phía nhận xác định được vị trí bắt đầu và kết thúc của frame cũng như vị trí gói tin tầng mạng trong frame.

- Tính trong suốt: Giao thức PPP không được đặt ra bất kỳ hạn chế nào trên gói dữ liệu tầng mạng. Tức là nó có khả năng chuyển đi bất kỳ gói dữ liệu tầng mạng nào.
- Hỗ trợ nhiều giao thức tầng mạng: Giao thức PPP phải có khả năng hỗ trợ nhiều giao thức tầng mạng (ví dụ, IP và DECnet) trên cùng đường truyền vật lý tại cùng một thời điểm. Điều này cũng giống như giao thức IP có khả năng phân kênh cho nhiều giao thức vận tải khác nhau (ví dụ, TCP và UDP). Như vậy PPP cũng cần có một cơ chế để khi nhận được một frame, thực thể PPP phía nhận xác định được cần chuyển gói dữ liệu cho thực thể tầng mạng nào.
- Hỗ trợ nhiều kiểu đường truyền: Ngoài khả năng hỗ trợ nhiều giao thức ở tầng cao hơn, PPP phải có khả năng vận hành trên nhiều kiểu đường truyền khác nhau bao gồm đường truyền tuần tự (truyền lần lượt từng bit một) hoặc song song (truyền nhiều bit cùng một lần), đồng bộ (truyền tín hiệu đồng hồ cùng với bit dữ liệu) hoặc dị bộ, truyền với tốc độ chậm hoặc cao, tín hiệu điện tử hoặc quang học.
- Phát hiện lỗi: PPP phía nhận có khả năng phát hiện liệu có lỗi bit trong frame nhận được hay không.
- Thời gian kết nối: PPP phải có khả năng phát hiện đường truyền bị lỗi ở mức link (ví dụ, không có khả năng để truyền dữ liệu từ phía gửi sang phía nhận) vả phải thông báo tình trạng lỗi này cho tầng mạng.
- Thoả thuận địa chỉ tầng mạng: PPP phải cung cấp cơ chế cho phép hai thực thể tầng mạng tham gia truyền thông (IP) có thể học hay đặt cấu hình địa chỉ tầng mạng cho nhau.
- Đơn giản: Người ta đòi hỏi PPP đáp ứng nhiều yêu cầu ngoài những yêu cầu nêu trên. một trong những yêu cầu quan trọng nhất là tính "đơn giản". Hiện nay hơn 50 RFC định nghiã những khía cạnh "đơn giản" của giao thức này.

Tuy vậy các đặc tả trong thiết kế PPP không yêu cầu:

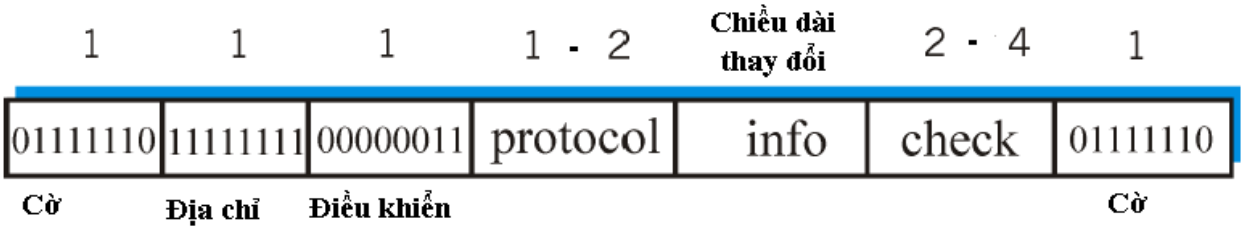
- Sửa lỗi: PPP cần phát hiện được lỗi bit nhưng không cần phải sửa lỗi.
- Kiểm soát lưu lượng: PPP phía nhận được hy vọng có khả năng nhận khung dữ liệu với tốc độ cao nhất của tầng vật lý phía dưới. Nếu tầng mạng không thể nhận với tốc độ này thì việc loại bỏ gói tin hay yêu cầu bên kia truyền chậm lại là trách nhiệm của các tầng cao hơn. Khi đó tầng cao hơn sẽ yêu cầu thực thể tương đương phía bên kia giảm tốc độ tạo ra dữ liệu gửi cho PPP.
- Đánh số thứ tự: PPP không được yêu cầu chuyển khung dữ liệu đến phía nhận theo đúng thứ tự gửi.
- Đường truyền đa điểm: PPP vận hành trên những đường truyền với một phía gửi và một phía nhận duy nhất. một số giao thức tầng liên kết dữ liệu khác (ví dụ, HDLC) cho phép nhiều nút nhận trên cùng một đường truyền (giống Ethernet).

Khuôn dạng khung dữ liệu

Hình 6.11 minh họa frame dữ liệu PPP giống khung dữ liệu của HDLC, PPP bao gồm những trường sau:

- Trường cờ: Mọi khung dữ liệu PPP bắt đầu và kết thúc bằng một byte cờ có giá trị 01111110.
- Trường địa chỉ: giá trị duy nhất của trường này là 11111111
- Trường điều khiển: giá trị duy nhất của trường này là 00000011.

Cả hai trường địa chỉ và điều khiển đều mang những giá trị cố định, điều này giải thích vì sao những trường này được định nghĩa đầu tiên. Khuyến nghị RFC 1662 ghi rõ rằng những giá trị này "có thể được định nghĩa sau này". Bởi vì những trường này mang giá trị cố định, PPP cho phép phía gửi không cần gửi byte địa chỉ và byte điều khiển, do đó tiết kiệm được hai byte tiêu đề trong khung dữ liệu PPP.



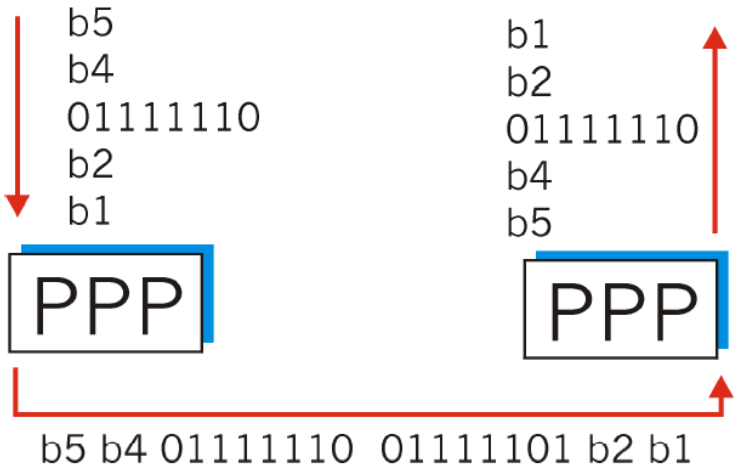
Hình 8.11 Khuôn dạng khung dữ liệu giao thức PPP

- Trường giao thức (protocol): Trường giao thức cho PPP xác định giao thức tầng trên sẽ nhận dữ liệu trong khung dữ liệu PPP. Khi nhận được khung dữ liệu PPP, bên nhận sẽ kiểm tra xem khung dữ liệu có lỗi không và sau đó chuyển phần dữ liệu trong gói tin cho giao thức thích hợp. RFC 1700 định nghĩa các mã 16 bit cho các giao thức được sử dụng cùng với PPP. Giao thức IP (dữ liệu trong khung dữ liệu PPP là gói dữ liệu IP) ứng với giá trị 21h, giao thức AppleTalk là 29h, DFCnet là 27h, giao thức điều khiển đường truyền PPP là C021h và giao thức điều khiển IP là 8021. Giao thức IP Control được PPP sử dụng khi kích hoạt kênh truyền lần đầu tiên để cấu hình mức IP giữa các thiết bị trên hai đầu kênh truyền.
- Thông tin: Trường này chứa gói tin được giao thức tầng mạng gửi đi trên đường truyền PPP (là gói tin IP). Độ dài lớn nhất của trường thông tin này là 1500 byte, mặc dù giá trị này có thể thay đổi lúc đặt cấu hình cho đường truyền.
- Checksum: Trường checksum được sử dụng để phát hiện các bit bị lỗi trong khung dữ liệu nhận được. Nó là mã CRC 2 hoặc 4 byte giống như trong giao thức HDLC.

Chèn Byte:

Một vấn đề phát sinh khi trong gói dữ liệu của giao thức tầng mạng lại có một byte có giá trị giống cờ đánh đấu điểm bắt đầu và kết thúc của khung dữ liệu (giá trị 01111110), bên nhận sẽ cho rằng đây là điểm kết thúc của khung dữ liệu PPP mặc dù trên thực tế không phải như vậy. Có thể cấm các giao thức tầng trên gửi dữ liệu chứa byte cờ, điều này vi phạm tính độc lập của các tầng, vì vậy PPP cũng như nhiều giao thức khác đã lựa chọn kỹ thuật chèn Byte (Byte stuff).

PPP định nghĩa một byte điều khiển đặc biệt, 01111101 làm nhiệm vụ đánh dấu. Nếu byte cờ 01111110 - xuất hiện trong khung dữ liệu (trừ vị trí mở đầu và kết thúc của khung dữ liệu), PPP đặt byte đánh dấu trước byte có giá trị giống cờ. Như vậy nó đã chèn thêm một byte điều khiển để đánh dấu rằng byte 01111110 không phải là cờ mà là dữ liệu thực. Bên nhận thấy 01111101 đứng trước 01111101 nên biết được 01111101 không phải là cờ mà là dữ liệu, nên nó sẽ tự động loại bỏ byte đánh dấu 01111110 được phía nhận chèn vào bên cạnh dòng dữ liệu thực. Hình 6.12 minh họa chèn byte trong PPP. Tương tự như vậy, nếu chính byte đánh dấu cũng xuất hiện trong dòng dữ liệu thực sự thì nó cũng cần được đánh dấu.



Hình 8.12 Chèn byte

Tóm lại, PPP là giao thức tầng liên kết dữ liệu cho hai thiết bị ở hai đầu của một đường truyền kiểu điểm-điểm, trao đổi các khung dữ liệu chứa gói dữ liệu của tầng mạng. Những chức năng chủ yếu của PPP là:

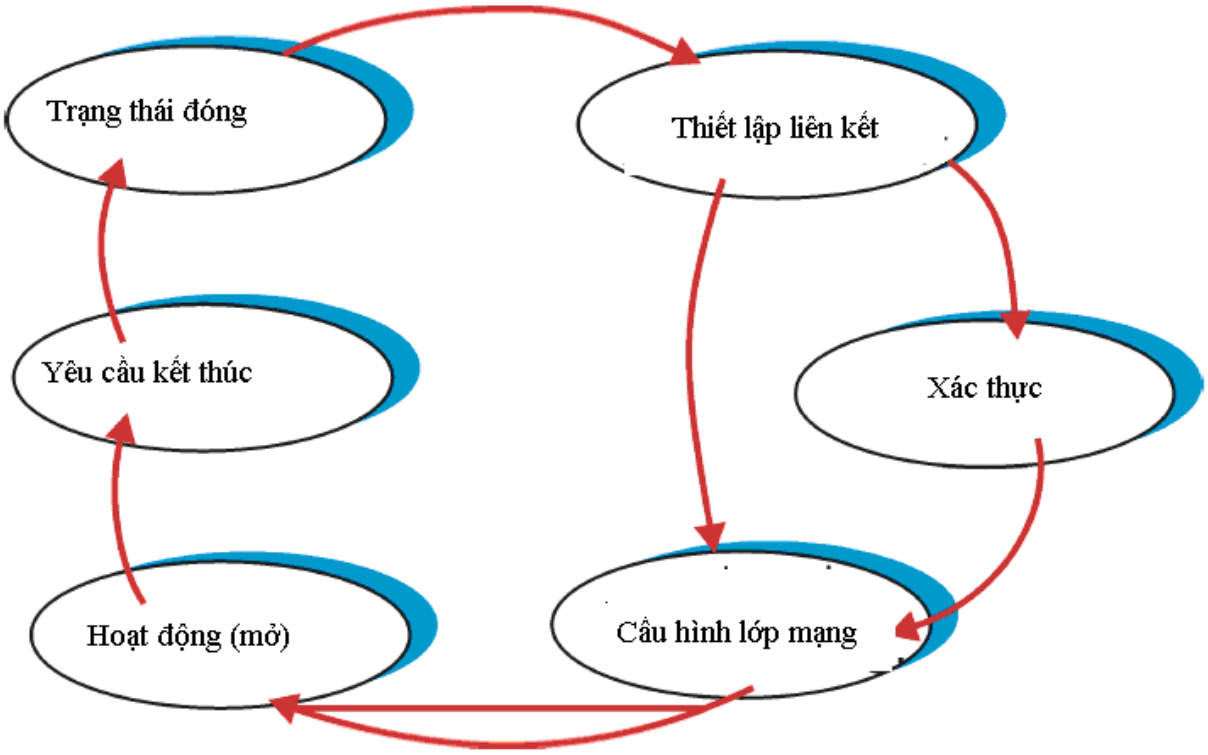
- Đóng gói dữ liệu: Phương thức đặt gói dữ liệu trong khung dữ liệu PPP
- Xác định vị trí bắt đầu và kết thúc của khung dữ liệu, phát hiện lỗi trong khung đó.
- Giao thức điều khiển đường truyền: khởi tạo, duy trì và kết thúc đường truyền PPP.

8.3.2.2 Giao thức điều khiển đường truyền PPP

Giao thức điều khiển mạng là một nhóm giao thức, mỗi giao thức ứng với một giao thức mạng ở tầng trên, cho phép module tầng mạng tự đặt cấu hình trước khi gói dữ liệu tầng mạng bắt đầu chuyển qua đường truyền. Quá trình khởi tạo, duy trì, báo lỗi và đóng đường truyền PPP được thực hiện nhờ giao thức điều khiển đường truyền (LCP -Linh Con trol Protocol) và các giao thức điều khiển mạng của PPP.

Trước khi trao đổi bất kì dữ liệu nào trên đường truyền PPP, hai phía (mỗi phía ở một đầu đường truyền) phải thực hiện nhiều công việc quan trọng để đặt cấu hình cho đường truyền, điều này cũng tương tự như thực thể TCP bên

gửi và bên nhận thực hiện bắt tay ba bước để đặt các tham số của kết nối TCP trước khi trao đổi TCP segment. Hình 8.13 minh họa biểu đồ chuyển trạng thái của giao thức LCP để đặt cấu hình, duy trì và kết thúc đường truyền PPP.



Hình 8.13 Sơ đồ chuyển trạng thái của giao thức LCP

Đường truyền PPP bắt đầu và kết thúc trong trạng thái đóng (dead). Khi phát sinh sự kiện như phát hiện sóng mang hay người quản trị mạng tác động để chỉ tầng vật lý sẵn sàng sử dụng, PPP bước sang trạng thái thiết lập đường truyền (link establishment). Trong trạng thái này, một phía của đường truyền gửi một số tùy chọn cấu hình mà nó mong muốn thông qua việc gửi khung dữ liệu yêu cầu cấu hình LCP (khung dữ liệu PPP có giá trị của trường protocol ứng với giao thức LCP và trường information chứa nội dung cấu hình yêu cầu). Sau đó phía bên kia trả lời với khung dữ liệu configure-nack (chấp nhận tất cả các lựa chọn), khung dữ liệu configure-nak (hiểu nhưng không chấp nhận các lựa chọn) hoặc khung dữ liệu configure-reject (không thể ghi nhận hoặc chấp nhận các lựa chọn để đàm phán). Tuỳ chọn cấu hình LCP bao gồm kích thước tối đa của khung dữ liệu trên đường truyền, giao thức kiểm chứng được sử dụng (nếu cần) và một tùy chọn xác định có bỏ qua việc sử dụng trường địa chỉ và trường điều khiển trong frame PPP hay không. Sau khi đường truyền được thiết lập, thỏa thuận xong các tuỳ chọn của đường truyền và kiểm chứng thành công, hai phía của đường truyền PPP sẽ trao đổi các gói tin kiểm soát của tầng mạng. Nếu IP chạy phía trên PPP, giao thức điều khiển IP được sử dụng để thiết lập cấu hình cho module giao thức IP tại mỗi đầu của đường truyền PPP.

Gói tin IPCP được đặt trong khung dữ liệu PPP. IPCP cho phép hai thực thể IP thay đổi hoặc đặt cấu hình địa chỉ IP hay thỏa thuận có nén gói dữ liệu IP

không. Những giao thức kiểm soát mạng tương tự được đưa ra cho những giao thức tầng mạng khác như DECNET [RFC 1762] và AppleTalk [RFC 1378]. Sau khi cấu hình xong tầng mạng, PPP có thể bắt đầu gửi gói tin của tầng mạng - đường truyền ở trạng thái mở và dữ liệu bắt đầu chuyển dọc theo đường truyền PPP. Các khung dữ liệu yêu cầu phản hồi và khung dữ liệu trả lời phản hồi LCP có thể được hai phía của đường truyền trao đổi để kiểm tra trạng thái đường truyền.

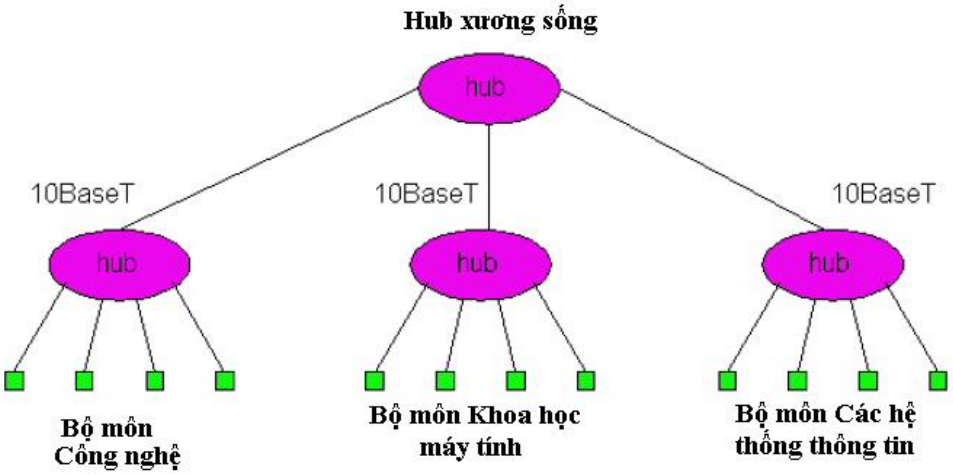
Đường truyền PPP được duy trì cho đến khi gói tin LCP yêu cầu kết thúc được gửi đi. Nếu frame LCP yêu cầu kết thúc (terminate-request) từ một phía của kết nối được trả lời bởi frame LCP chấp nhận kết thúc (terminate-ack) từ phía bên kia thì đường truyền bước vào trạng thái đóng.

8.4 Các thiết bị mạng nội bộ

Cơ quan các công ty, trường đại học - có đặc điểm gồm nhiều bộ phận, mỗi bộ phận có mạng cục bộ riêng. Tất nhiên, cơ quan muốn kết nối mạng cục bộ của các bộ phận, để làm được điều đó phải sử dụng các thiết bị như: bộ lặp (repeater), bộ tập trung (hub), cầu nối (bridge) và bộ chuyển mạch (switch). Với việc xuất hiện khái niệm mạng nội bộ ảo (VLAN), đôi khi người ta cũng xếp bộ định tuyến thuộc nhóm thiết bị trong mạng nội bộ.

8.4.1 Bộ tập trung

Cách đơn giản nhất để kết nối LAN là sử dụng bộ tập trung (Hub). Hub là một thiết bị đơn giản sao chép tín hiệu đến từ một cổng ra tất cả các cổng còn lại, bản chất của hub là bộ lặp thao tác trên bit, vì thế chúng là thiết bị ở tầng vật lý. Khi bit đi vào một cổng, hub sẽ truyền bit này qua tất cả các cổng khác.



Hình 8.14 Kết nối mạng nội bộ qua Hub

Hình 8.14 minh họa kết nối mạng LAN của ba bộ môn trong khoa Công nghệ thông tin qua hub. Mỗi bộ môn có một mạng Ethernet 10BaseT để cán bộ và sinh viên của bộ môn sử dụng. Mỗi máy tính của bộ môn kết nối điểm-điểm đến hub. Hub thứ tư, được gọi là hub xương sống kết nối điểm-điểm đến các hub của từng bộ môn để liên kết LAN của ba bộ môn. Thiết kế như trong hình 6.14 là thiết kế đa tầng vì các hub được sắp xếp trong hệ thống phân cấp. Có thể

tạo thiết kế nhiều tầng, ví dụ mạng LAN của khoa CNTT lại kết nối với các khoa khác trong Học viện. Trong thiết kế đa tầng, chúng ta coi toàn bộ mạng liên kết với nhau là mạng cục bộ LAN và coi mỗi phần mạng của của một bộ môn là một phân đoạn trong mạng LAN. Tất cả các phân đoạn trong mạng LAN trên hình 6.14 đều thuộc về cùng một vùng xung đột, nghĩa là chỉ cần 2 máy tính trong mạng gửi dữ liệu cùng một thời điểm sẽ xảy ra xung đột.

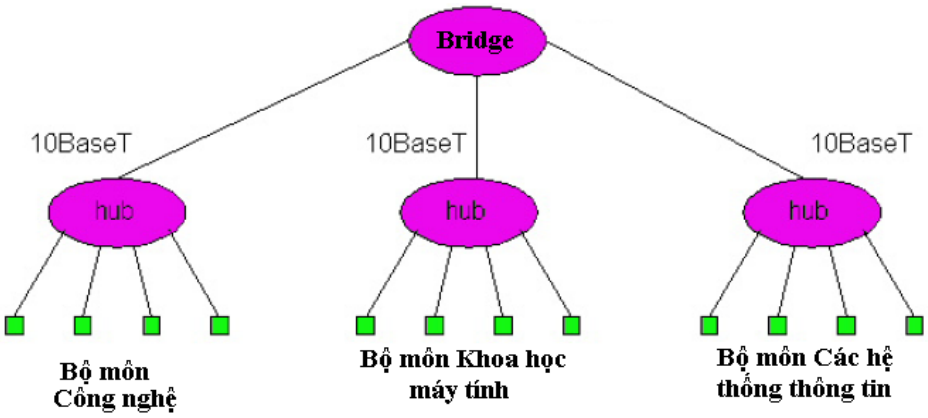
Việc kết nối các máy tính bằng Hub có đơn giản, nó mở rộng khoảng cách tối đa giữa bất cứ lặp nút nào trên nội bộ. Bằng kết nối qua hub, khoảng cách tối đa này có thể được mở rộng vì khoảng cách giữa các hub kết nối trực tiếp với nhau có thể là 100m khi sử dụng cáp xoắn đôi (và nhiều hơn khi dùng cáp quang). Việc thiết kế đa tầng giảm nguy cơ ngưng hoạt động của toàn bộ hệ thống. Giả sử nếu bất kỳ hub của bộ môn nào đó bị lỗi, hub trục chính có thể phát hiện vấn đề và phong tỏa kết nối tới hub bộ môn đó, như vậy các bộ môn còn lại vẫn có thể tiếp tục hoạt động và truyền thông trong khi hub bị lỗi không hoạt động.

Tuy vậy hub cũng có nhược điểm. Đầu tiên và có lẽ quan trọng nhất là khi sử dụng hub trung tâm, miền xung đột của mạng cục bộ của từng khoa trở thành miền xung đột chung của toàn bộ hệ thống. Xét ví dụ minh họa trên hình 6.14. Trước khi kết nối ba khoa, mạng cục bộ mỗi khoa có băng thông cực đại là 10mbps, vì vậy thông lượng toàn bộ tối đa của 3 LAN là 30mbps. Nhưng khi mạng LAN của ba khoa được kết nổi vào hub trung tâm, tất cả máy tính của ba khoa thuộc về cùng một miền xung đột, và thông lượng bị giảm xuống 10Mbps. Hạn chế thứ hai là nếu các khoa khác nhau sử dụng các công nghệ Ethernet khác nhau thì không có khả năng để kết nối chúng vào hub trung tâm. Ví dụ, nếu một vài khoa sử dụng 100BaseT, thì không thể kết nối chúng với nhau vì hub về bản chất là repeater. Hạn chế thứ ba là mỗi công nghệ Ethernet (10 Base2, 10 BaseT, 100 BaseT, . . .) có giới hạn về số nút, khoảng cách tối đa giữa hai máy tính trong miền xung đột và số tầng tối đa trong thiết kế nhiều tầng. Những hạn chế này hạn chế tổng số máy tính có thể kết nối đến mạng cục bộ cũng như phạm vi địa lý của mạng cục bộ nhiều tầng.

8.4.2 Cầu nối

Khác với hub (là thiết bị tầng vật lý), bridge có thể xử lý trên khung Ethernet, vì vậy nó là thiết bị tầng 2. Thực tế, cầu nối (Bridge) chính là thiết bị chuyển mạch thực hiện việc chuyển và lọc các khung dữ liệu căn cứ trên địa chỉ vật lý. Khi khung dữ liệu đến từ một cổng nào đó của bridge, nó không gửi khung đó đến tất cả các cổng khác. Bridge sẽ xác định địa chỉ vật lý đích của khung dữ liệu và chuyển đến cổng duy nhất dẫn về đích.

Hình 8.15 minh họa ba bộ môn trong ví dụ trước kết nối tới bridge. Ba chữ số bên cạnh bridge là số thứ tự các cổng của bridge. Khi các bộ môn được kết nối qua bridge với nhau tạo thành mạng LAN của khoa Công nghệ thông tin, mỗi bộ môn là một đoạn trong mạng nội bộ của khoa. Khác với việc kết nối bằng hub, mỗi đoạn mạng của từn bộ môn giờ đây là một vùng xung đột riêng biệt.



Hình 8.15 Kết nối mạng bằng cầu nối

Bridge có thể khắc phục nhiều vấn đề của hub. Bridge cho phép truyền thông giữa các bộ môn trong khi cô lập miền xung đột của mỗi bộ môn, có thể kết nối các công nghệ LAN khác nhau và không bị giới hạn về khoảng cách tối đa trong mạng cục bộ khi sử dụng bridge để kết nối các phân đoạn mạng cục bộ.

8.4.2.1 Nguyên lý lọc và chuyển tiếp

Lọc là khả năng xác định liệu sẽ chuyển tiếp khung dữ liệu đến cổng nào đó hay loại bỏ. Chuyển tiếp là khả năng xác định cổng kế tiếp để chuyển khung dữ liệu đi. Bridge thực hiện hai chức năng này nhờ bảng chuyển mạch. Mỗi hàng trong bảng ứng với một nút đích trên mạng LAN. Tuy vậy bảng chuyển mạch không nhất thiết phải chứa tất cả các hàng cho mọi nút trong mạng. Mỗi hàng trong bảng chuyển mạch gồm địa chỉ vật lý của nút và cổng bridge có thể dẫn đến nút đó, thời gian thiết lập hàng đó trong bảng. Ví dụ về bảng bridge cho LAN trong hình 6.15 được minh họa trong hình 8.16. Mặc dù quá trình chuyển khung dữ liệu có vẻ tương tự quá trình chuyển mạch gói của tầng mạng, nhưng chúng hoàn toàn khác nhau: Địa chỉ bridge sử dụng là địa chỉ vật lý trong khi đó chuyển mạch ở tầng mạng sử dụng địa chỉ địa chỉ logic, bảng chuyển mạch và bảng định tuyến cũng khác rất nhiều.

Địa chỉ	Giao diện	Thời
62-FE-F7-11-89-	1	9:32
7C-BA-B2-B4-91-	3	9:36
“““	“.	“.

Hình 8.16 Cấu trúc bảng chuyển mạch

Giả sử khung dữ liệu với địa chỉ đích DD-DD-DD-DD-DD-DD đến bridge từ cổng X. Bridge tìm kiếm trên bảng lọc hàng ứng với địa chỉ vật lý DD-DD-DD-DD-DD-DD để tìm ra cổng Y tương ứng - là cổng sẽ dẫn đến nút có địa chỉ đích

DD-DD-DD- DD-DD-DD. Chúng ta sẽ thấy điều gì sẽ xảy ra nếu không có giao diện y như thế trong bảng:

- Nếu X = Y, thì frame đến từ đoạn chứa Card mạng có địa chỉ DD-DD-DD-DD-DD-DD. Không cần chuyển khung dữ liệu đến bất kỳ cổng nào khác, bridge thực hiện chức năng lọc bằng cách loại bỏ khung dữ liệu này.
- Nếu X # Y thì frame cần được gửi đến đoạn nào đó qua cổng Y, Bridge thực hiện chức năng chuyển tiếp bằng cách đặt khung dữ liệu vào bộ đệm ra của cổng Y.

Những quy tắc đơn giản trên cho phép bridge cô lập các miền xung đột của các đoạn mạng khác nhau kết nối tới các cổng của nó. Những quy tắc này cũng cho phép thiết bị trên hai phân đoạn khác nhau truyền đồng thời mà không xảy ra xung đột. Giả sử khung dữ liệu với địa chỉ đích 62-EF-F7-11-89-A3 được gửi đến bridge qua cổng 1. Bridge kiểm tra bảng và thấy rằng đích nằm trên phân đoạn được kết nối đến cổng 1 (là mạng LAN của bộ môn Công nghệ), điều này có nghĩa là khung dữ liệu thực sự đã được quảng bá trên phân đoạn này, do vậy bridge loại bỏ khung dữ liệu này. Giả sử khung dữ liệu trên đến từ cổng 2, Bridge kiểm tra bảng và thấy rằng đích nằm ở trên hướng ứng với cổng 1, do đó bridge chuyển khung dữ liệu ra cổng 1. Rõ ràng rằng nếu bảng bridge đầy đủ và chính xác, bridge cho phép truyền thông giữa các khoa nhưng cô lập các miền xung đột.

Khi có khung dữ liệu để gửi chuyển tiếp, hub gửi khung dữ liệu lên trên đường truyền mà không quan tâm xem có thiết bị nào khác đang chiếm dụng đường truyền không. Trái lại, bridge sẽ sử dụng thuật toán CSMA/CD, nó không truyền ngay nếu như có nút khác trên phân đoạn mạng đang truyền. Bridge cũng sử dụng thuật toán exponential backoff khi việc truyền của nó bị xung đột, tức là cổng của bridge hoạt động tương tự như Card mạng của một máy tính. Tuy nhiên về mặt kỹ thuật, bridge không phải là Card mạng vì chúng không có địa chỉ vật lý. Card mạng của máy tính hoặc thiết bị định tuyến luôn luôn chèn địa chỉ vật lý của nó vào trường địa chỉ nguồn trong tất cả các khung dữ liệu nó gửi đi nhưng bridge không thay đổi địa chỉ nguồn của mọi khung dữ liệu.

Một tính năng quan trọng của bridge là chúng có thể được dùng để nối các phân đoạn mạng sử dụng những công nghệ Ethernet khác nhau, Hub không đảm bảo tính năng này. Ví dụ nếu trong hình 6.15, bộ môn Công nghệ sử dụng Ethernet 10BaseT, bộ môn Khoa học máy tính sử dụng Ethernet 100BaseT và bộ môn Hệ thống thông tin sử dụng Ethernet 10BaseT thì bridge có thể kết nối cả 3 đoạn mạng trên.

Khi sử dụng bridge làm thiết bị kết nối, thì về lý thuyết mạng LAN không bị giới hạn bởi phạm vi địa lý. Trên lý thuyết chúng ta có thể xây dựng mạng LAN trải rộng toàn cầu bằng kết nối các hub qua các bridge. Theo thiết kế này, mỗi hub là một miền xung đột và do đó LAN không bị giới hạn. Tuy nhiên nó lại tạo ra vùng quảng bá lớn và do đó người ta sẽ kết nối qua thiết bị định tuyến, chứ không sử dụng bridge.

8.4.2.2 Xây dựng bảng chuyển mạch

Một đặc tính quan trọng của bridge là khả năng tự học. Đó là bảng chuyển mạch của bridge được xây dựng tự động, khả năng này được thực hiện như sau:

- Bảng bridge khởi đầu là rỗng.
- Khi khung dữ liệu đến cổng nào đó và địa chỉ đích của khung không có trong bảng, thì bridge sẽ chuyển khung dữ liệu đến bộ đệm ra của tất cả các cổng ngoại trừ cổng thông tin đi đến.
- Khi nhận được khung dữ liệu, bridge lưu trữ địa chỉ vật lý trong trường địa chỉ nguồn của khung, cổng nhận được và thời gian hiện hành. Như vậy bridge ghi nhớ được vị trí phân đoạn mạng LAN của nút gửi. Nếu nút nào đó trong LAN gửi khung dữ liệu đến bridge thì nó sẽ xác định được cổng để đi đến nút đó.
- Khi khung dữ liệu đến một trong các cổng và địa chỉ đích của khung có trong bảng, thì bridge chuyển đó đến cổng thích hợp.
- Sau một khoảng thời gian nhất định, nếu Bridge không nhận được khung dữ liệu từ Card mạng có địa chỉ vật lý được lưu trong bảng chuyển mạch thì Bridge sẽ xoá bản ghi đó. Như vậy, nếu một máy tính được thay thế bởi máy tính khác (với Card mạng khạc) thì địa chỉ vật lý của PC trước sẽ bị bridge xoá.

Xét quá trình tự học của bridge trong hình 8.17 và bảng bridge tương ứng trong hình 8.16. Giả sử rằng tại thời điểm 9:39 bridge nhận được một khung dữ liệu có địa chỉ gửi là 01-12-23-45- 56 đến cổng 2. Giả sử, địa chỉ này chưa có trong bảng, bridge sẽ bổ sung một hàng mới trong bảng như chỉ ra trên hình 6.17.

Địa chỉ	Giao diện	Thời gian
01-12-23-34-45-65	2	9:39
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:6

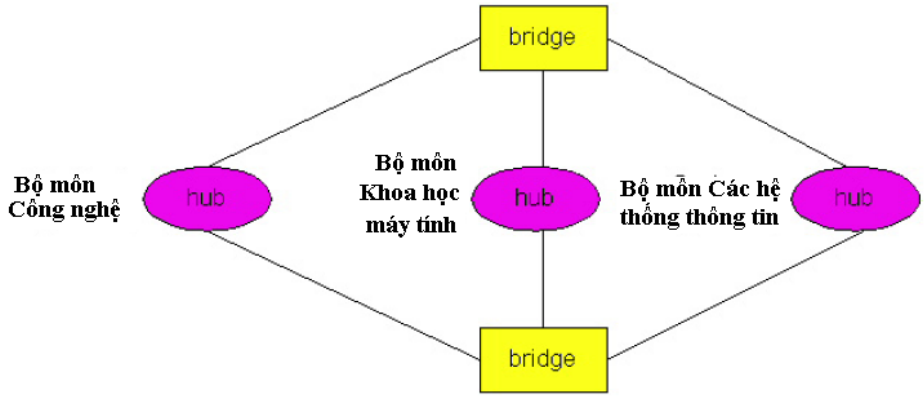
Hình 8.17 Bảng chuyển mạch của bridge

Giả thiết thời gian sống của mỗi hàng trong bảng là 60 phút và máy tính với địa chỉ 62-FE-F7-11-89-A3 không gửi đi bất kỳ khung dữ liệu nào qua bridge trong khoảng thời gian tớ 9:32 đến 10:32 thì lúc 10:32, bridge sẽ xóa địa chỉ này khỏi bảng. Bridge là thiết bị theo kiểu plug and lay bởi vì nó không cần sự can thiệp của người quản trị mạng. Người quản trị mạng chỉ cần nối đầu dây mạng vào các cổng của bridge mà không cần thiết lập cấu hình cho bảng bridge trong thời gian cài đặt hay khi máy tính tách khỏi phân đoạn mạng.

8.4.2.3 Spanning Tree

Nếu bridge bị hỏng, thì các phân đoạn mạng sẽ không kết nối được với nhau, để dự phòng rủi ro, người ta thường xây dựng mạng với nhiều đường nối

giữa các phân đoạn mạng. Trên hình 8.18, nhiều đường dư thừa giữa các phân đoạn mạng làm giảm khả năng sụp đổ của toàn bộ hệ thống. Nhưng việc có nhiều đường dẫn giữa các đoạn mạng cũng sẽ phát sinh ra nhiều vấn đề - một khung dữ liệu có thể di chuyển vòng quanh hay được nhân bản lên nhiều lần trong mạng cục bộ. Giả sử bảng chuyển mạch trong hình 6.18 rỗng và máy tính của bộ môn Khoa học máy tính gửi khung dữ liệu đến máy tính trong bộ môn Công nghệ. Khi khung dữ liệu đến hub của bộ môn Khoa học máy tính, hub sẽ sinh ra hai bản sao của khung và gửi mỗi bản đến hai bridge. Khi mỗi bridge nhận được khung, nó sẽ tạo ra 2 bản sao của khung, một bản gởi đến hub của bộ môn Khoa học máy tính và bản kia đến hub của bộ môn Công nghệ. Vì cả hai bridge cùng làm như vậy, sẽ có 4 khung dữ liệu giống hệt nhau trong LAN. Khung dữ liệu có thể được nhân bản liên tục nếu bridge không biết nút nhận nằm ở đâu. Trong trường hợp này, số bản sao của khung dữ liệu gốc tăng theo hàm số mũ, hàm tràn ngập toàn bộ mạng.



Hình 8.18 Spanning tree

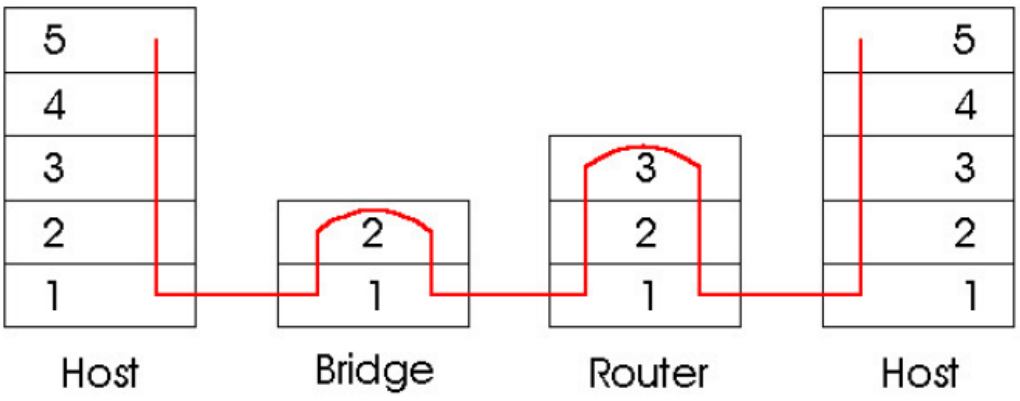
Để ngăn ngừa những tình huống nêu trên, bridge sử dụng giao thức spanning tree. Trong giao thức spanning tree, bridge liên lạc với bridge khác trên mạng nội bộ để xác định tập con của hình trạng ban đầu không có vòng lặp. Sau khi xác định được spanning tree, bridge chỉ kết nối với các cổng phù hợp để tạo spanning tree từ hình trạng ban đầu. Ví dụ trong hình 6.18, spanning tree được tạo nên nếu bridge phía trên phong tỏa kết nối cổng kết nối đến bộ môn Công nghệ và bridge phía dưới phong tỏa cổng kết nối đến bộ môn Hệ thống thông tin. với các cổng bị phong tỏa và loại bỏ được các vòng lặp, frame sẽ không lặp và nhân bản. Nếu khi nào đó một liên kết trong spanning tree bị mỗi, bridge có thể kết nối lại giao diện đã bị phong tỏa, kích hoạt thuật toán spanning tree lần nữa và xác định spanning tree mới.

8.4.2.4 So sánh cầu nối và thiết bị định tuyến

Thiết bị định tuyến hoạt động theo kiểu kiểu lưu và chuyển, nó chuyển tiếp gói tin dựa trên địa chỉ tầng mạng. Mặc dù cũng là thiết bị chuyển mạch kiểu lưu và chuyển, nhưng điểm khác biệt cơ bản giữa bridge và thiết bị định tuyến nằm ở vấn đề sử dụng địa chỉ để thực hiện nhiệm vụ chuyển mạch. Như vậy thiết bị định tuyến là chuyển mạch gói ở tầng mạng trong khi bridge là

chuyển mạch gói ở tầng liên kết dữ liệu. Dù bridge và thiết bị định tuyến khác nhau, song người quản trị mạng sẽ phải lựa chọn giữa chúng khi cài đặt thiết bị kết nối.Ví dụ, với hệ thống mạng trong hình 6.19, người quản trị mạng có thể lựa chọn thiết bị định tuyến thay vì lựa chọn bridge. Thiết bị định tuyến cũng sẽ cô lập ba miền xung đột trong khi vẫn cho phép truyền thông giữa các khoa. Như vậy cả bridge và router đều có thể làm thiết bị kết nối.

Bridge là thiết bị kiểu "cắm vào là chạy" - tính năng được tất cả các nhà quản trị mạng ưa thích. Bridge cũng có tốc độ lọc và chuyển gói dữ liệu cao - như minh họa trên hình 6.19, nó chỉ phải xử lý gói dữ liệu của tầng liên kết dữ liệu trong khi thiết bị định tuyến phải xử lý gói dữ liệu của tầng mạng. mặt khác, giao thức spanning tree hạn chế topo của toàn bộ mạng. Điều này có nghĩa là tất cả các khung dữ liệu chỉ được chuyển trên spanning tree, thậm chí khi có nhiều đường dẫn trực tiếp nhưng bị phong tỏa giữa nguồn và đích. Sự hạn chế của spanning tree cũng tập trung vào khả năng tải trên đường truyền spanning tree khi nó có thể đã được lan truyền đến tất cả các đường truyền khác của mạng cũ. Hơn nữa bridge không đưa ra bất cứ sự bảo vệ nào để chống lại sự phát ra hàng loạt - nếu máy tính bị lỗi và truyền đi một luồng dữ liệu liên tục, bridge sẽ chuyển tất cả những khung dữ liệu này khiến toàn bộ mạng bị tắc nghẽn.



Hình 8.19 Xử lý gói dữ liệu trong máy tính, cầu nối và thiết bị định tuyến

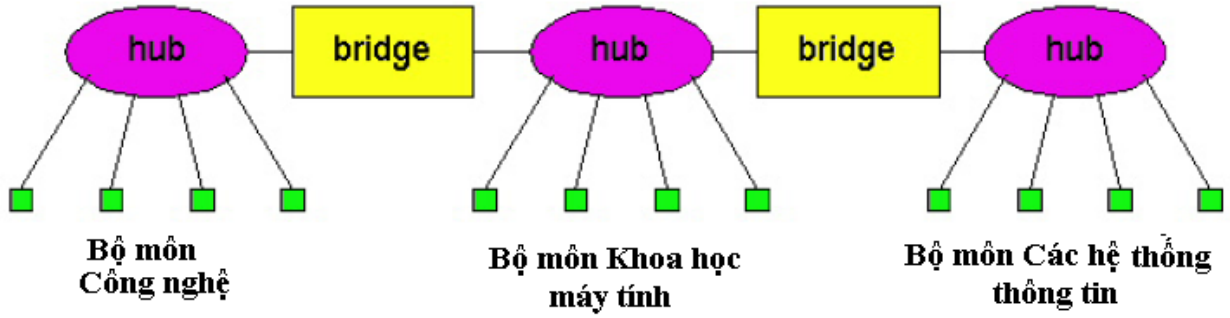
Nói chung địa chỉ mạng thường được tổ chức phân cấp, gói dữ liệu chắc chắn không vòng lại qua thiết bị định tuyến ngay cả khi có nhiều đường đi (Thực sự gói dữ liệu có thể quay vòng nếu bảng định tuyến của router bị đặt cấu hình sai, nhưng IP sử dụng trường TTL trong tiêu đề gói dữ liệu để loại bỏ các gói tin loại này). Vì vậy, gói dữ liệu không bị giới hạn chuyển trong spanning tree, nó có thể sử dụng đường dẫn tốt nhất giữa nguồn và đích. Một đặc tính quan trọng khác của thiết bị định tuyến là không quảng bá chống lại sự phát tán liên tục ở tầng liên kết dữ liệu. Yếu điểm duy nhất của thiết bị định tuyến thời gian xử lý gói tin của router thường lâu hơn bridge vì chúng phải xử lý các trường tiêu đề của tầng 3, việc quản trị thiết bị định tuyến cũng phức tạp hơn.

Thông thường một mạng nhỏ gồm vài trăm máy tính với vài đoạn mạng thì Bridge đủ để đáp ứng cho loại mạng nhỏ này mà không yêu cầu bất kì cấu

hình địa chỉ IP. Với những mạng lớn gồm hàng nghìn máy tính sẽ cần tới nhiều thiết bị định tuyến bên trong mạng. Những thiết bị định tuyến này cung cấp khả năng cô lập các vùng xung đột và các vùng quảng bá.

8.4.2.5 Kết nối các đoạn mạng qua đường trục

Một thiết kế khác về mạng của khoa Công nghệ thông tin được minh họa trong hình 8.20. Thiết kế này sử dụng hai bridge, mỗi bridge có hai cổng. Bridge thứ nhất kết nối hai bộ môn Công nghệ và Khoa học máy tính, Bridge kia kết nối bộ môn Khoa học máy tính với bộ môn Hệ thống thông tin. Mặc dù bridge hai cổng rất phổ biến do giá rẻ và đơn giản, nhưng mô hình thiết kế trong hình 6.20 không được ưa chuộng. Có hai lý do, thứ nhất, nếu hub của Computer Science bị hỏng thì máy tính ở hai bôn môn công nghệ và Hệ thống thông tin không thể trao đổi được với nhau. Thứ hai truyền thông giữa hai bộ môn đó phải thông qua bộ môn Khoa học máy tính, dễ gây xung đột trong đoạn mạng của bôn môn này.



Hình 8.20 Kết nối mạng không có đường trục

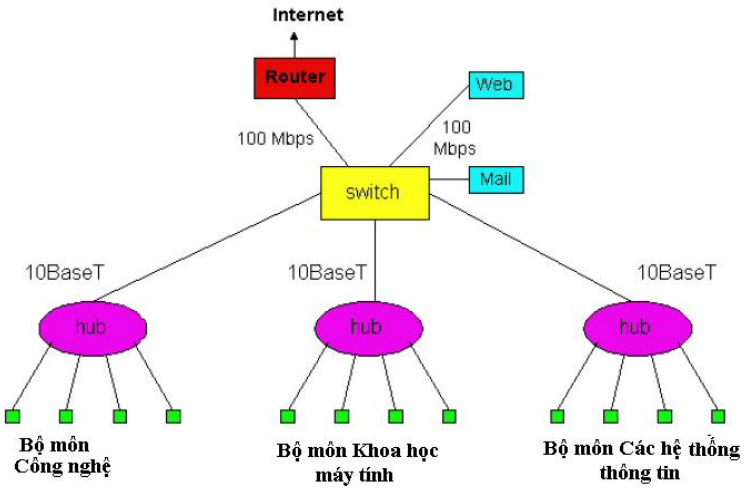
Một nguyên tắc quan trọng định hướng dẫn việc kết nối các phân đoạn mạng là sử dụng đường trục chính, đó là một mạng có kết nối trực tiếp đến tất cả các đoạn mạng khác. Khi có một trục chính thì hai phân đoạn mạng có thể truyền tin trực tiếp cho nhau mà không cần thông qua phân đoạn thứ ba.

8.4.3 Switch

Đầu thập kỷ 90, ba loại thiết bị kết nối mạng cục bộ được sử dụng chủ yếu là hub (repeater), bridge, router. Từ giữa những năm 90, một thiết bị trở nên rất thông dụng là switch (thực chất đó là bridge nhiều cổng. Giống như bridge, switch chuyển và lọc khung dữ liệu dựa trên địa chỉ vật lý đích, tự động xây dựng bảng chuyển mạch khi có một khung dữ liệu đi qua. Có thể mua switch có các cổng tốc độ khác nhau 10 Mbps, 100 Mbps và 1Gbps. Ví dụ, một người có thể mua switch có bốn cổng 100 Mbps, hai mươi cổng 10 Mbps hoặc switch có bốn cổng 100 Mbps và một cổng 1 Gbps. Nhiều switch vận hành trong chế độ song công, chúng có thể gửi và nhận khung dữ liệu tại cùng một thời điểm trên cùng một cổng.

Ưu điểm của switch nhiều cổng và ở chỗ dễ dàng kết nối trực tiếp giữa các máy tính với switch. Khi một máy tính có đường kết nối trực tiếp song công với switch, nó có thể truyền và nhận dữ liệu ở tốc độ truyền tối đa của Card mạng và

không bao giờ xảy ra xung đột. Khi máy tính có kết nối trực tiếp đến switch thì nó được xem như có đường dùng riêng.



Hình 8.21 Mô hình ví dụ mạng nội bộ có kết nối Internet

Hình 8.21 minh họa cách tổ chức mạng nội bộ của khoa Công nghệ thông tin, mỗi bộ môn là một phân đoạn mạng sử dụng bộ tập trung tốc độ 10 Mbit/s. Vì mỗi bộ tập trung đều kết nối đến switch nên các máy tính của tất cả các bộ môn đều có khả năng trao đổi dữ liệu với nhau. Máy chủ cho dịch vụ Web và thư điện tử đều có đường dùng riêng 100 Mbps đến switch. Cuối cùng thiết bị định tuyến sẽ kết nối toàn bộ hệ thống mạng nội bộ ra ngoài mạng Internet.

Các loại chuyển mạch:

Giả sử khung dữ liệu đến switch từ một cổng nào đó, switch chỉ cần phải đọc 6 byte đầu tiên là đã có thể xác định được cần phải chuyển dữ liệu đó đến cổng nào. Dựa trên đặc điểm của khung dữ liệu Ethernet, một khung hợp lệ tối thiểu phải có độ dài 64 Bytes. Như vậy có thể thiết kế ba loại: chuyển mạch lưu và chyển tiếp, chuyển mạch xuyên suốt và chuyển mạch Fragment-Free.

Với chuyển mạch loại lưu và chyển tiếp, thời gian trễ tương đối lớn vì switch phải nhận đầy đủ dữ liệu của một khung sau đó mới xác định cần phải chuyển khung dữ liệu đó đi đâu, như vậy đảm bảo độ chính xác và không chuyển tiếp những khung dữ liệu lỗi. Chuyển mạch xuyên suốt có tốc độ cao hơn, nhưng nó chuyển tất cả các khung dữ liệu mà không cần biết khung dữ liệu đó có bị lỗi hay không, điều này làm tăng lưu lượng mạng, nhất là trong trường hợp xảy ra xung đột. Loại thứ ba dung hòa hai loại trên, swich chỉ chuyển tiếp các khung dữ liệu sau khi đã nhận được ít nhất 64 bytes..

8.5 Kết nối không dây

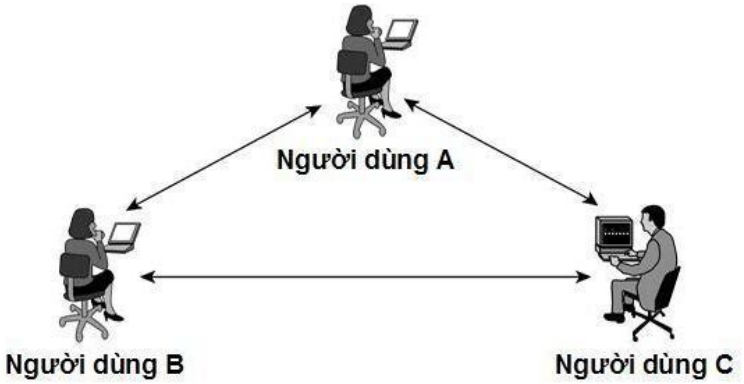
Kết nối không dây là hình thức kết nối giữa các thành phần trong mạng không sử dụng các loại cáp như một mạng thông thường, môi trường truyền thông của các thành phần trong mạng là không khí. Các thành phần trong mạng sử dụng sóng điện từ để truyền thông với nhau. Công nghệ mạng không dây lần đầu tiên xuất hiện vào cuối năm 1990, khi những nhà sản xuất giới thiệu những

sản phẩm hoạt động trong băng tần 900Mhz. Những giải pháp này cung cấp tốc độ truyền dữ liệu 1Mbps, thấp hơn nhiều so với tốc độ 10Mbps của hầu hết các mạng Ethernet tại thời điểm đó.

Năm 1992, những nhà sản xuất bắt đầu bán những sản phẩm phục vụ cho mạng không dây sử dụng băng tần 2.4Ghz. Mặc dù những sản phẩm này đã có tốc độ truyền dữ liệu cao hơn nhưng chúng vẫn là những giải pháp riêng của mỗi nhà sản xuất không được công bố rộng rãi. Sự cần thiết cho việc hoạt động thống nhất giữa các thiết bị ở những dãy tần số khác nhau dẫn đến một số tổ chức bắt đầu phát triển ra những chuẩn mạng không dây chung. Năm 1997, IEEE (Institute of Electrical and Electronics Engineers) đã đưa ra chuẩn 802.11, và cũng được biết với tên gọi WIFI (Wireless Fidelity) cho các mạng không dây. Chuẩn 802.11 hỗ trợ ba phương pháp truyền tín hiệu, trong đó có bao gồm phương pháp truyền tín hiệu vô tuyến ở tần số 2.4Ghz. Năm 1999, IEEE thông qua hai sự bổ sung cho chuẩn 802.11 là các chuẩn 802.11a và 802.11b (định nghĩa ra những phương pháp truyền tín hiệu). Những thiết bị dựa trên chuẩn 802.11b đã nhanh chóng trở thành công nghệ không dây vượt trội. Các thiết bị WLAN 802.11b truyền phát ở tần số 2.4Ghz, cung cấp tốc độ truyền dữ liệu có thể lên tới 11Mbps. IEEE 802.11b được tạo ra nhằm cung cấp những đặc điểm về tính hiệu dụng, thông lượng và bảo mật để so sánh với mạng có dây. Năm 2003, IEEE công bố thêm một sự cải tiến là chuẩn 802.11g mà có thể truyền nhận thông tin ở cả hai dãy tần 2.4Ghz và 5Ghz và có thể nâng tốc độ truyền dữ liệu lên đến 54Mbps. Thêm vào đó, những sản phẩm áp dụng 802.11g cũng có thể tương thích ngược với các thiết bị chuẩn 802.11b, hiện nay chuẩn 802.11g đã đạt đến tốc độ 108Mbps - 300Mbps.

8.5.1 Các mô hình kết nối mạng không dây

Mô hình mạng độc lập:

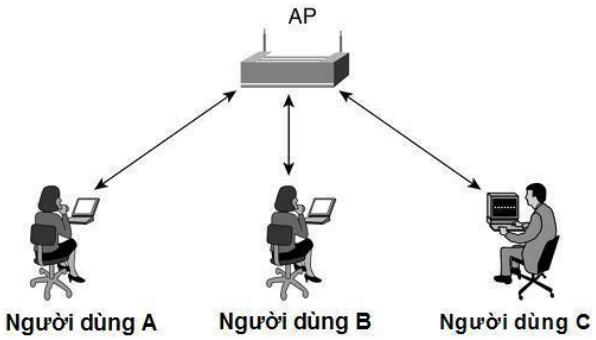


Hình 6.22 Kết nối theo mô hình ngang hàng

Các nút di động tập trung lại trong một không gian nhỏ để hình thành nên kết nối ngang hàng, chúng có thể trao đổi thông tin trực tiếp với nhau. Vì các mạng ad-hoc này có thể thực hiện nhanh và dễ dàng nên chúng thường được thiết lập mà không cần một công cụ hay kỹ năng đặc biệt nào vì vậy nó rất thích

hợp để sử dụng trong các hội nghị thương mại hoặc trong các nhóm làm việc tạm thời. Tuy nhiên chúng có thể có những nhược điểm về vùng phủ sóng bị giới hạn, mọi người sử dụng đều phải nghe được lẫn nhau.

Mô hình mạng cơ sở:



Hình 8.23 Mô hình kết nối mạng không dây cơ sở

Bao gồm các điểm truy nhập (AP - Access Point) gắn với mạng đường trục hữu tuyến và giao tiếp với các thiết bị di động trong vùng phủ sóng, nó đóng vai trò chuyển tiếp cho các thiết bị di động của người sử dụng. Một điểm truy nhập nằm ở trung tâm có thể điều khiển và phân phối truy nhập cho các nút tranh chấp, cung cấp truy nhập phù hợp với mạng đường trục, ấn định các địa chỉ và các mức ưu tiên, giám sát lưu lượng mạng, quản lý chuyển đi các gói và duy trì theo dõi cấu hình mạng. Tuy nhiên giao thức đa truy nhập tập trung không cho phép các nút di động truyền trực tiếp tới nút khác nằm trong cùng vùng với điểm truy nhập như trong cấu hình mạng không dây độc lập. Trong trường hợp này, mỗi gói tin sẽ phải được phát đi 2 lần (từ nút phát gốc và sau đó là điểm truy nhập) trước khi nó tới nút đích, quá trình này sẽ làm giảm hiệu quả truyền dẫn và tăng trễ truyền dẫn.

Mô hình mạng mở rộng:

Chuẩn 802.11 mở rộng phạm vi di động tới một phạm vi bất kì thông qua mô hình mở rộng giao tiếp giữa các điểm truy nhập từ mạng này sang mạng khác. AP thực hiện việc giao tiếp thông qua hệ thống phân phối, đó là hệ thống được tiếp sóng trở lại một đích trong cùng một mạng cơ sở, chuyển tiếp trên hệ thống phân phối tới một AP khác, hoặc gởi tới một mạng có dây tới đích không nằm trong mạng mở rộng. Các thông tin nhận bởi AP từ hệ thống phân phối được truyền tới BSS sẽ được nhận bởi trạm đích.

8.5.2 Ưu và nhược điểm của kết nối không dây

So với các hình thức kết nối hữu tuyến, kết nối không dây có những ưu điểm sau:

- Sự tiện lợi: Kết nối không dây cũng như hệ thống cho phép người dùng truy xuất tài nguyên mạng ở bất kỳ nơi đâu trong khu vực được triển khai.
- Hiệu quả: Người dùng có thể duy trì kết nối mạng khi họ đi từ nơi này đến nơi khác.

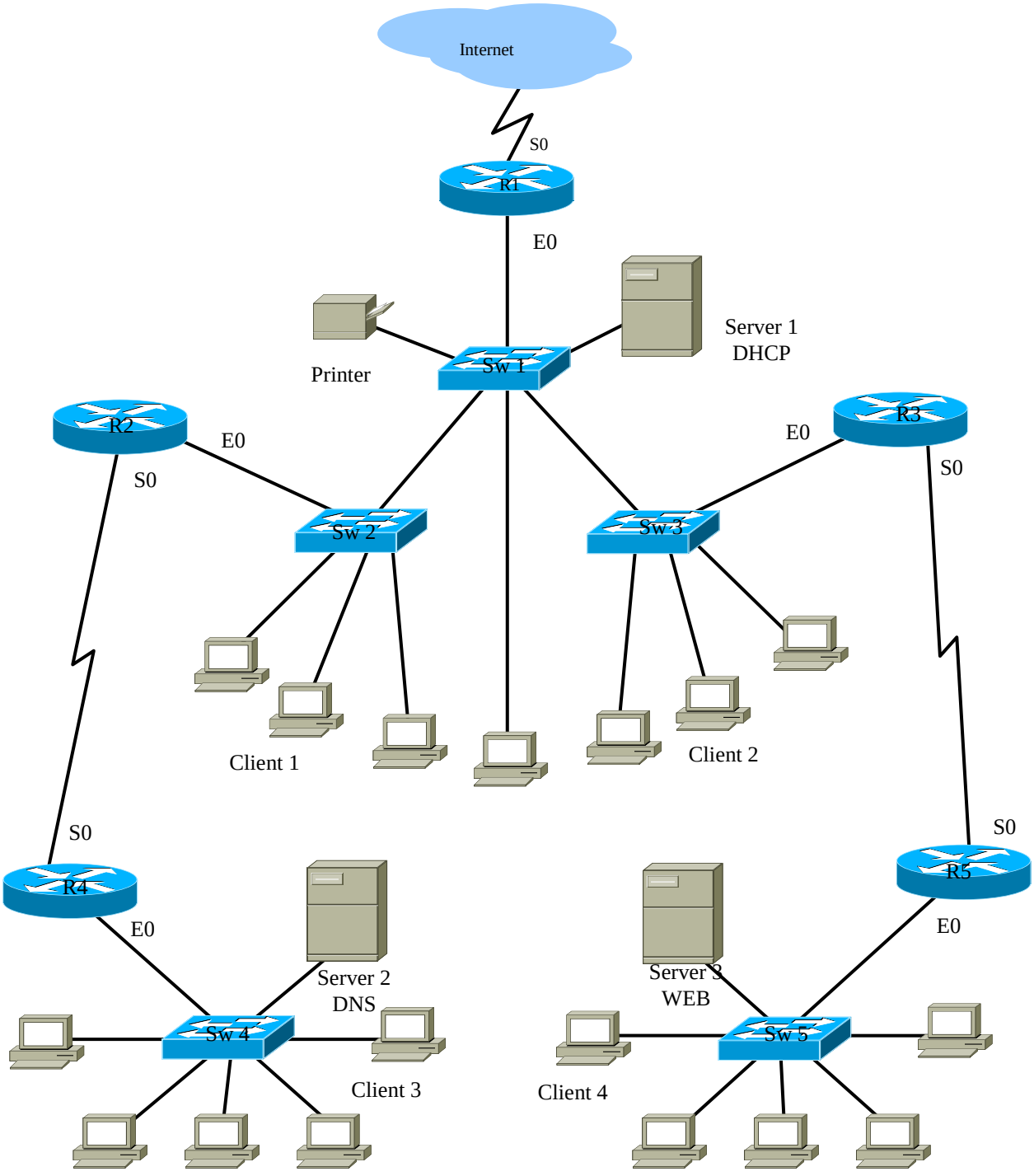
- Triển khai: Việc thiết lập hệ thống mạng không dây ban đầu chỉ cần ít nhất 1 AP. Với mạng dùng cáp, phải tốn thêm chi phí và có thể gặp khó khăn trong việc triển khai hệ thống cáp ở nhiều nơi trong tòa nhà.
- Khả năng mở rộng: Mạng không dây có thể đáp ứng tức thì khi gia tăng số lượng người dùng. Với hệ thống mạng dùng cáp cần phải gắn thêm cáp.

Mặc dù có những ưu điểm trên, kết nối không dây có những nhược điểm sau:

- Bảo mật: Môi trường kết nối không dây là không khí nên khả năng bị tấn công hoặc đánh cắp thông tin của người dùng rất cao.
- Phạm vi: Một mạng chuẩn 802.11g với các thiết bị chuẩn chỉ có thể hoạt động tốt trong phạm vi vài chục mét. Nó phù hợp trong 1 căn nhà, nhưngvới một tòa nhà lớn thì không đáp ứng được nhu cầu.
- Độ tin cậy: Vì sử dụng sóng vô tuyến để truyền thông nên việc bị nhiễu, tín hiệu bị giảm do tác động của các thiết bị khác.
- Tốc độ: Tốc độ của mạng không dây (1- 300 Mbps) rất chậm so với mạng sử dụng cáp (10 Mbps đến 10 Gbps), tốc độ này phụ thuộc vào khoảng các từ máy tính của người sử dụng đến AP.

BÀI TẬP TỔNG HỢP

Cho sơ đồ mạng như sau:



Giả thiết:

- Toàn bộ mạng sử dụng chung địa chỉ mạng **172.18.0.0/16**
- Máy chủ Server 1 đã được cấu hình để cấp phát địa chỉ IP động bằng giao thức DHCP.

- Trang thông tin nội bộ được đặt trên máy chủ Server 3, cổng mặc định
- Máy chủ Server 2 cài đặt dịch vụ tên miền, chứa địa chỉ máy chủ Server 3.

Nhiệm vụ 1: Xác định số lượng vùng quảng bá, số lượng mạng con, số lượng bit cần mượn và điền vào bảng sau**:**

Mạng con	Địa chỉ mạng	Phạm vi địa chỉ hợp lệ	Địa chỉ quảng bá	Có thể sử dụng?	Có dùng ?
#0					
#1					
#2					
#3					
#4					
#5					
#6					
#7					

Nhiệm vụ 2: Gán địa chỉ IP cho các thiết bị và điền vào bảng sau**:**

Giao diện	Địa chỉ IP	Mặt nạ mạng	Cổng mặc định	Địa chỉ mạng
Router 1 – E0				
Router 2 – S0				
Router 2 – E0				
Router 3 – S0				
Router 3 – E0				
Router 4 – S0				
Router 4 – E0				
Router 5 – E0				
Router 5 – S0				
Switch 1				
Switch 2				
Switch 3				
Switch 4				

Switch 5				
Printer				
Server 1				
Server 2				
Server 3				

Nhiệm vụ 3: Phân tích quá trình cấp phát địa chỉ chỉ IP động khi máy trạm Client 1 khởi động.

Nhiệm vụ 4: Ngay sau khi được cấp phát địa chỉ IP, người sử dụng dùng trình duyệt để truy nhập trang thông tin điện tử đặt trên máy Server 3, hãy phân tích quá thực hiện trên các thiết bị.

Nhiệm vụ 5: Thiết lập mạng trên và thực hiện các công việc sau:

- Thiết lập địa chỉ IP cho các thiết bị như đã chuẩn bị ở nhiệm vụ 2.
- Cấu hình định tuyến trên các Router.
- Cấu hình máy chủ Server 1 để cấp phát địa chỉ IP động.
- Cài đặt trang thông tin nội bộ trên máy chủ Server 3.
- Cài đặt dịch vụ DNS trên máy chủ Server 2 và cấu hình để trỏ tên miền của trang thông tin nội bộ về địa chỉ IP của máy chủ Server 3.
- Kiểm thử quá trình cấp phát địa chỉ IP động và quá trình truy nhập trang thông tin nội bộ
- Trên máy trạm, thực hành các tiện ích mạng và ghi lại và tìm hiểu ý nghĩa các thông số trong quá trình thực hiện:
    - Ipconfig, Ipconfig/all, Ipconfig/renew, Ipconfig/release
    - Route Print, Route Add, Route Delete
    - Ping, Tracert, Nslookup, Arp

TÀI LIỆU THAM KHẢO

[1]. J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet (2nd edition), Addison-Wesley, 2002.

[2] CNAP: Networking Fundamentals, version 4.0, 2004

[3] Andrew S. Tanenbaum Computer Networks, Prentice Hall, 4th edition, 2003.

[4]. Alberto Leon-Garcia and Indra Widjaja, Communication Networks: Fundamental Concepts and Key Architectures, McGraw-Hill, 2000.

[5] Larry L. Peterson and Bruce S. Davie Computer Networks: A Systems Approach (2nd ed.), Mogran-Kaufmann, 1999.