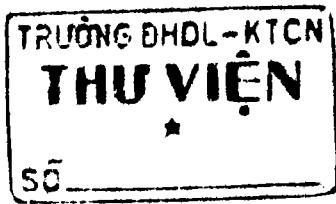


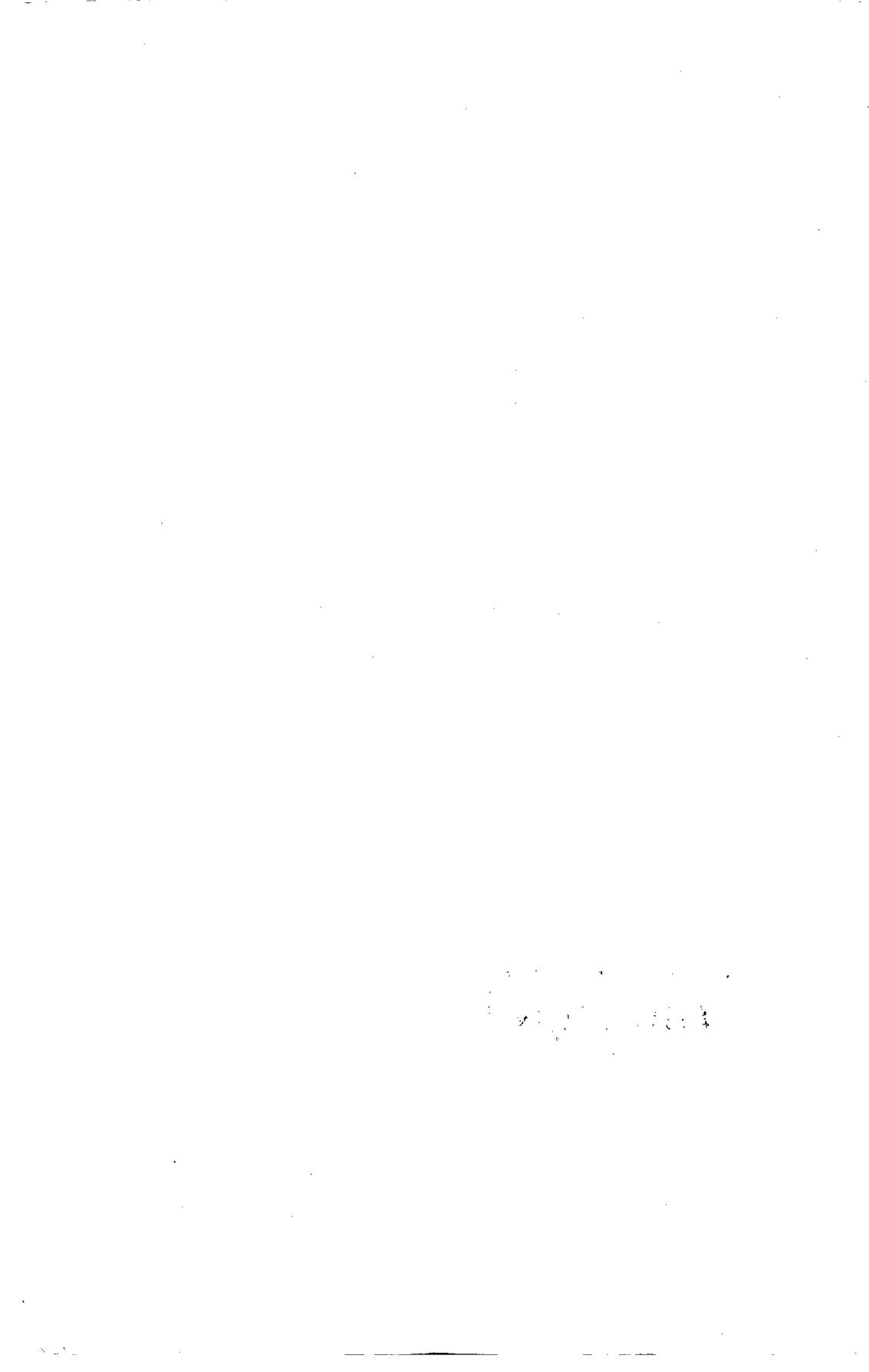


Nguyễn Hồng Sơn (chủ biên)
Hoàng Đức Hải

KỸ THUẬT TRUYỀN SỐ LIỆU



NHÀ XUẤT BẢN LAO ĐỘNG - XÃ HỘI
NĂM 2002



LỜI NÓI ĐẦU

Nhu cầu trao đổi số liệu máy tính xuất hiện không lâu sau khi máy tính cá nhân ra đời, tuy nhiên hệ thống thực hiện ở thời kỳ đầu còn đơn giản. Với sự phát triển nhanh chóng của xã hội, nhu cầu này cũng tăng theo và đa dạng hơn, vì thế hệ thống thực thi cũng càng phức tạp, luôn phải đương đầu với những thách thức mới. Kỹ thuật truyền số liệu đã ra đời trong bối cảnh này.

Mặc dù là học phần mới mẻ, nhưng kỹ thuật truyền số liệu là mảng kiến thức không thể thiếu đối với các sinh viên chuyên ngành điện tử, viễn thông và công nghệ thông tin. Có thể nói đây là nền tảng ứng dụng và nguồn đối tượng cho nghiên cứu chuyên sâu trong các chuyên ngành này. Mặc dù mang đậm giải pháp cho dịch vụ số liệu (data service) nhưng kỹ thuật truyền số liệu ngày nay lại là xuất phát điểm cho đa dịch vụ, một xu thế tất yếu trong mạng viễn thông hiện đại. Tiếp cận và lĩnh hội kỹ thuật truyền số liệu bằng cách nào cho hiệu quả đã và đang là mối quan tâm của nhiều người, đặc biệt là các sinh viên chuyên ngành liên quan. Thiết nghĩ, không có kiến thức căn bản vững chắc sẽ không có phát triển và ứng dụng. Vì vậy, trao cho bạn đọc tất cả những điều căn bản, vô cùng thiết thực của kỹ thuật truyền số liệu là mục tiêu của cuốn sách này. Công việc lựa chọn tài liệu tham khảo và biên soạn đều dựa trên tiêu chí đó. Đối tượng của cuốn sách là các sinh viên chuyên ngành điện tử, viễn thông và công nghệ thông tin, cuốn sách cũng rất hữu ích đối với các cán bộ kỹ thuật trước đây chưa có điều kiện được học một cách có hệ thống về truyền số liệu.

Cuốn sách gồm bảy chương được sắp xếp theo thứ tự những chủ đề liên quan từ mức vật lý đến mức giao thức. Chương 7 trình bày chi tiết phương pháp và hoạt động truyền số liệu trong mạng máy tính cục bộ, trong chương này cũng khái quát các giải pháp kỹ thuật liên quan đến mạng máy tính cục bộ không dây, gọi tắt là WLAN (Wireless LAN). Với

tiềm năng vốn có, WLAN đang được dự đoán sẽ rất phát triển trong thời gian tới.

Xin chân thành cảm ơn các đồng nghiệp đã đọc và đóng góp nhiều ý kiến quý báu cho nội dung cuốn sách này.

Trong khi biên soạn, mặc dù rất cố gắng song không thể tránh khỏi những thiếu sót nhất định, rất mong nhận được các ý kiến xây dựng tích cực của bạn đọc. Mọi ý kiến đóng góp xin gửi về địa chỉ e-mail: mk.pub@cinet.vnnews.com.

MK.PUB

www.minhkhai.com.vn

MỤC LỤC

LỜI NÓI ĐẦU	3
MỤC LỤC	5

CHƯƠNG 1 : MẠNG TRUYỀN SỐ LIỆU VÀ SỰ CHUẨN HÓA.....	11
1. KHÁI QUÁT THÔNG TIN SỐ LIỆU	11
2. MẠNG TRUYỀN SỐ LIỆU.....	12
3. SỰ CHUẨN HÓA VÀ MÔ HÌNH THAM CHIẾU ISO.....	17
3.1.Chuẩn hóa các mô hình truyền số liệu	17
3.2.Mô hình tham chiếu OSI	20
3.2.1. Các lớp hướng ứng dụng	23
3.2.2. Các lớp phụ thuộc mạng	25
4. CÁC CHUẨN HỆ THỐNG MỞ (OPEN SYSTEM STANDARDS)	26
CHƯƠNG 2 : GIAO TIẾP VẬT LÝ	29
1. Môi trường truyền.....	30
1.1. Các đường truyền 2 dây không xoắn	30
1.2. Các đường dây xoắn đôi	31
1.3. Cáp đồng trục	32
1.4. Cáp quang	32
1.5. Đường truyền vệ tinh	34
1.6. Đường truyền vi ba	35
1.7. Đường truyền vô tuyến tần số thấp	35
2. SỰ SUY GIẢM VÀ BIẾN DẠNG TÍN HIỆU	37
2.1. Sự suy giảm.....	38
2.2. Băng thông bị giới hạn	39
2.3. Sự biến dạng do trễ pha	43
2.4. Sự can nhiễu (tạp âm).....	43
3. CÁC LOẠI TÍN HIỆU	47
3.1. V.28	48
3.2. Dòng 20mA	48
3.3. RS-422A/V.11	49
3.4. Các tín hiệu cáp đồng trục.....	50
3.5. Các tín hiệu cáp quang	52
3.6. Tín hiệu vệ tinh và radio.....	53
4. TRÈ DO LAN TRUYỀN TÍN HIỆU	55
5. CÁC MẠCH TÀI CÔNG CỘNG	56
5.1. Các mạch PSTN analog	56

5.2. Mạch thuê riêng kỹ thuật số	67
5.2.1.Kỹ thuật số hóa	67
5.2.2.Kỹ thuật ghép kênh	72
6. CÁC CHUẨN GIAO TIẾP VẬT LÝ	82
6.1. Giao tiếp EIA -232D/V24	82
6.2. Modem rỗng (Null Modem).....	85
6.3. Giao tiếp EIA-530.....	86
6.4. Giao tiếp EIA-430/V.35	86
6.5. Giao tiếp X21	88
6.6. Giao tiếp ISDN	90
6.7. Tóm lược về giao tiếp	91

CHƯƠNG 3 : GIAO TIẾP KẾT NỐI SỐ LIỆU..... 93

1. CÁC KHÁI NIỆM CƠ BẢN VỀ TRUYỀN SỐ LIỆU	93
1.1. Các chế độ thông tin (Communication modes)	93
1.2. Các chế độ truyền (Transmission modes)	93
1.3. Kiểm soát lỗi (error control).....	96
1.4. Điều khiển luồng (flow control).....	97
1.5. Các giao thức liên kết dữ liệu.....	97
1.6. Các hình thức truyền	98
1.7. Mã truyền (transmission code).....	99
1.8. Các đơn vị dữ liệu (data unit).....	101
1.9. Giao thức (protocol).....	101
1.10. Hoạt động kết nối.....	102
1.11. Đường nối và liên kết	102
2. THÔNG TIN NỐI TIẾP BẤT ĐỒNG BỘ.....	102
2.1. Khái quát.....	102
2.2. Nguyên tắc đồng bộ bit	102
2.3. Nguyên tắc đồng bộ ký tự	105
2.4. Nguyên tắc đồng bộ frame	105
3. THÔNG TIN NỐI TIẾP ĐỒNG BỘ.....	106
3.1. Khái quát.....	106
3.2. Nguyên tắc đồng bộ bit	107
3.2.1. Mã hoá và giải mã tín hiệu đồng hồ	108
3.2.2. Mạch vòng khóa pha số (Digital Phase Lock-Loop)	110
3.2.3. Các lược đồ lai	115
3.3. Truyền đồng bộ thiên hướng ký tự	116
3.4. Truyền đồng bộ thiên hướng bit	118
4. MẠCH ĐIỀU KHIỂN TRUYỀN SỐ LIỆU	120
4.1. Khái quát.....	120
4.2. Giao tiếp truyền có thể lập trình UART 8250 của Intel	123
4.2.1. Giao tiếp bus	124
4.2.2. Xung đồng hồ và sự định thời	125
4.2.3. Cấu trúc bên trong và hoạt động của 8250	125



4.3. Giao tiếp truyền có thể lập trình USART 8251 của Intel	132
4.3.1. Giới thiệu.....	132
4.3.2. Kiến trúc và hoạt động của 8251 ở chế độ bắt đồng bộ	132
4.3.3. Hoạt động ở chế độ truyền đồng bộ của 8251A	139
4.3.4. Giao tiếp 8251A	142
5. Các thiết bị điều khiển truyền dữ liệu.....	143
5.1. Khái quát.....	143
5.2. Bộ ghép kênh phân thời	145
5.3. Bộ ghép kênh thống kê (statistical multiplexer).....	147
5.4. Các thiết bị làm việc theo chế độ khối (block-mode)	149
5.4.1. Các đường dây đa điểm.....	149
5.4.2. Poll-select.....	150

CHƯƠNG 4 : XỬ LÝ SỐ LIỆU TRUYỀN..... 153

1. MÃ HÓA SỐ LIỆU MỨC VẬT LÝ	153
2. PHÁT HIỆN LỖI VÀ SỬA SAI.....	153
2.1.Tổng quan.....	153
2.2. Phương pháp kiểm tra bit chẵn lẻ (parity bit).....	155
2.3. Kiểm tra tổng BSC (Block Sum Check)	157
2.4. Kiểm tra CRC (cyclic redundancy check)	158
2.5. Phát hiện và sửa sai theo Hamming	166
3. NÉN SỐ LIỆU	170
3.1.Khái quát.....	170
3.2. Nén nhờ đơn giản mã cho các chữ số (Packed decimal)	171
3.3. Nén theo mã hóa quan hệ (Relative coding).....	172
3.4. Nén bằng cách bỏ bớt các ký tự giống nhau (Character suppression)....	172
3.5. Nén theo mã hóa Huffman	172
3.6. Mã hóa Huffman động.....	178
3.7. Nén fax (facsimile)	182
4. MẬT MÃ HÓA SỐ LIỆU.....	189
4.1. Khái quát.....	189
4.2. Mật mã hóa cổ điển.....	189
4.3. Mật mã khóa công khai	191

CHƯƠNG 5 : CƠ SỞ CỦA GIAO THỨC 193

1. KIỂM SOÁT LỖI	194
2. Idle RQ.....	195
2.1. Kiến trúc phân lớp	199
2.2. Đặc tả giao thức.....	202
2.3. Đặc tả idle RQ	203
2.4. Hiệu suất sử dụng liên kết.....	213
3. RQ LIÊN TỤC (CONTINUOUS RQ).....	217

3.1. Truyền lại có chọn lựa (selective repeat).....	218
3.2. Truyền lại một nhóm (Go-back_N).....	223
3.3. Điều khiển luồng (flow control).....	227
3.3.1. X-ON/X-OFF	227
3.3.2. Cửa sổ trượt (<i>sliding window</i>)	228
3.4. Các chỉ số tuần tự.....	230
3.5. Đặc tả giao thức.....	232
3.6. Hiệu suất sử dụng liên kết.....	237
4. QUẢN LÝ LIÊN KẾT.....	240

CHƯƠNG 6 : CÁC GIAO THỨC ĐIỀU KHIỂN LIÊN KẾT SỐ LIỆU

.....	243
2. CÁC MÔI TRƯỜNG ỨNG DỤNG.....	244
3. CÁC GIAO THỨC THIÊN HƯỚNG KÝ TỰ	247
3.1. Các giao thức đơn công (simplex protocols)	248
3.2. Các giao thức bán song công	251
3.2.1. <i>Các dạng frame</i>	252
3.2.2. <i>Sự trong suốt dữ liệu</i>	255
3.2.3. <i>Hoạt động của giao thức</i>	255
3.2.4. <i>Giao tiếp user</i>	258
3.2.5. <i>Hiệu suất của giao thức</i>	258
3.3. Các giao thức song công hoàn toàn	261
3.4. Ví dụ về các giao thức thiên hướng ký tự thường gấp.....	263
3.4.1. <i>Giao thức XMODEM</i>	263
3.4.2. <i>Giao thức YMODEM</i>	267
3.4.3. <i>Giao thức kermit</i>	267
4. CÁC GIAO THỨC THIÊN HƯỚNG BIT	274
4.1. Giao thức điều khiển liên kết số liệu mức cao HDLC (High-level Data Link Control)	274
4.1.1. <i>Các khuôn dạng của frame</i>	276
4.1.2. <i>Các kiểu frame</i>	278
4.1.3. <i>Hoạt động của giao thức</i>	278
4.1.4. <i>Giao tiếp user</i>	286
4.2. Thủ tục truy xuất liên kết LAPB	288
4.3. Thủ tục đa truy xuất.....	289
4.4. Thủ tục truy xuất liên kết LAPM	291
4.5. Thủ tục truy xuất liên kết LAPD	293
4.6. Điều khiển liên kết luận lý	296
4.6.1. <i>Các dịch vụ user</i>	296
4.6.2. <i>Hoạt động của giao thức</i>	299
4.6.3. <i>Các dịch vụ MAC</i>	300



CHƯƠNG 7 : KỸ THUẬT TRUYỀN SỐ LIỆU TRONG MẠNG MÁY TÍNH CỤC BỘ	303
1. CÁC MẠNG LAN NỐI DÂY	303
1.1. Các Topo	303
1.2. Môi trường truyền dẫn	307
1.3. Các phương pháp điều khiển truy xuất môi trường	311
1.3.1. <i>Đa truy xuất cảm nhận sóng mang có phát hiện đụng độ CSMA/CD</i>	311
1.3.2. <i>Token (thẻ) điều khiển</i>	313
1.3.3. <i>Vòng được phân chia khe (slotted ring)</i>	314
1.4. Hệ thống Ethernet và Fast Ethernet (CSMA/CD)	316
1.4.1. <i>Điều khiển truy xuất IEEE 802.3</i>	316
1.4.2. <i>Hệ thống CSMA/CD</i>	319
1.4.3. <i>Cấu trúc frame và các tham số hoạt động</i>	322
1.4.4. <i>Hoạt động truyền frame</i>	324
1.4.5. <i>Tiếp nhận frame</i>	326
1.4.6. <i>Các đặc tả IEEE 802.3 có băng thông 10Mbps</i>	326
1.4.7. <i>Các đặc tả IEEE 802.3 có băng thông 100Mbps</i>	329
1.5. Hệ thống Token ring và FDDI	331
1.5.1. <i>Điều khiển truy xuất môi trường trong IEEE 802.5</i>	331
1.5.2. <i>Giao tiếp ring</i>	333
1.5.3. <i>Khuôn dạng frame IEEE 802.5</i>	335
1.5.4. <i>Ưu tiên trên vòng token</i>	337
1.5.5. <i>Đặc tả lớp vật lý IEEE 802.5</i>	341
1.5.6. <i>Điều khiển trung xuất FDDI</i>	341
1.6. Hệ thống mạng Token bus	347
1.6.1. <i>Khái quát</i>	347
1.6.2. <i>Hoạt động cơ bản</i>	349
1.6.3. <i>Chuyển token</i>	351
1.6.4. <i>Cửa sổ đáp ứng</i>	352
1.6.5. <i>Khởi động</i>	353
1.6.6. <i>Hoạt động ưu tiên</i>	353
1.7. Hệ thống 100 VG - AnyLAN	356
1.7.1. <i>Topo</i>	356
1.7.2. <i>Điều khiển truy xuất môi trường</i>	356
1.7.3. <i>Đặc tả lớp vật lý của 100VG -AnyLAN</i>	360
1.8. ATM LAN	361
1.9. Fibre Channel	365
1.9.1. <i>Khái quát</i>	365
1.9.2. <i>Các phần tử của fibre channel</i>	367
1.9.3. <i>Kiến trúc giao thức Fibre Channel</i>	368
1.9.4. <i>Môi trường và giao tiếp vật lý</i>	370
1.9.5. <i>Giao thức truyền</i>	370
1.9.6. <i>Giao thức đóng bộ frame</i>	370
1.9.7. <i>Các dịch vụ chung</i>	370

1.9.8. <i>Ánh xạ</i>	371
1.9.9. <i>Môi trường vật lý và cấu hình</i>	371
2. CÁC LAN KHÔNG DÂY	373
2.1. Khái quát.....	373
2.2. Đường truyền không dây	376
2.2.1. <i>Đường truyền bằng sóng radio</i>	376
2.2.2. <i>Đường truyền bằng sóng hồng ngoại</i>	379
2.3. Các lược đồ truyền.....	382
2.3.1. <i>Lược đồ truyền sóng radio</i>	382
2.3.2. <i>Lược đồ hồng ngoại (infrared)</i>	390
2.4. Các phương pháp điều khiển truy xuất môi trường.....	392
2.4.1. <i>CDMA (Code-Division Multiple Access)</i>	393
2.4.2. <i>CSMA/CD</i>	393
2.4.3. <i>CSMA/CA</i>	395
2.4.4. <i>TDMA</i>	397
2.4.5. <i>FDMA</i>	398
2.4.6. <i>Chức năng bổ sung</i>	398
2.5. Các chuẩn	399
3. CÁC GIAO THỨC	400
3.1. Các dịch vụ lớp MAC	402
3.2. Lớp LLC	403
3.3. Lớp mạng	405



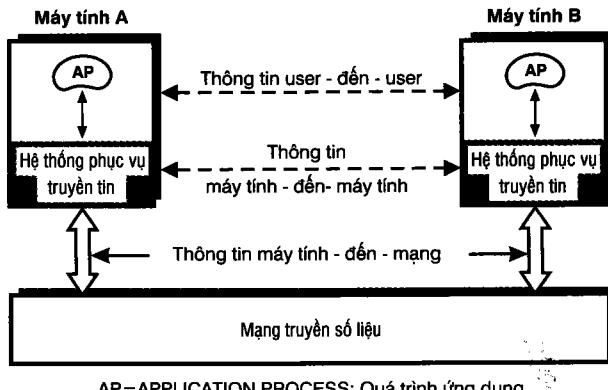
CHƯƠNG 1

MẠNG TRUYỀN SỐ LIỆU VÀ SỰ CHUẨN HÓA

1. KHÁI QUÁT THÔNG TIN SỐ LIỆU

Thông tin liên lạc đóng vai trò hết sức quan trọng trong cuộc sống, hầu hết chúng ta luôn gắn liền với một vài dạng trao đổi thông tin nào đó. Các dạng trao đổi tin có thể như: đàm thoại giữa người với người, đọc sách, gửi và nhận thư, nói chuyện qua điện thoại, xem phim hay truyền hình, xem triển lãm tranh, tham dự diễn đàn...

Có hàng nghìn ví dụ khác nhau về thông tin liên lạc, trong đó thông tin số liệu là một phần đặc biệt trong toàn bộ lĩnh vực thông tin.



Hình 1.1 Một hệ thống thông tin cơ bản.

Từ các ví dụ trên chúng ta nhận thấy rằng mỗi hệ thống truyền tin đều có các đặc trưng riêng nhưng có một số đặc tính chung cho tất cả các hệ thống. Đặc trưng chung có tính nguyên lý là tất cả các hệ thống truyền tin đều nhằm mục đích chuyển tải thông tin từ điểm này đến điểm khác. Trong các hệ thống truyền số liệu, thường gọi thông tin là dữ liệu hay thông điệp. Thông điệp có nhiều dạng khác nhau, để truyền thông điệp từ một điểm này đến điểm khác cần phải có sự tham gia của 3 thành phần của hệ thống: nguồn tin là nơi phát sinh và chuyển thông điệp lên môi trường truyền, môi trường truyền là phương tiện mang thông điệp và đích thu. Các phần tử này là yêu cầu tối thiểu trong bất cứ quá trình truyền tin nào. Nếu một trong các thành phần này không tồn tại, truyền tin không thể xảy ra. Một hệ thống truyền tin thông thường được miêu tả trên hình 1.1. Các thành phần cơ bản có thể xuất hiện dưới nhiều dạng khác nhau tùy thuộc

vào hệ thống. Khi xây dựng các thành phần của một hệ thống truyền tin, cần phải xác định một số các yếu tố liên quan đến phẩm chất hoạt động của nó:

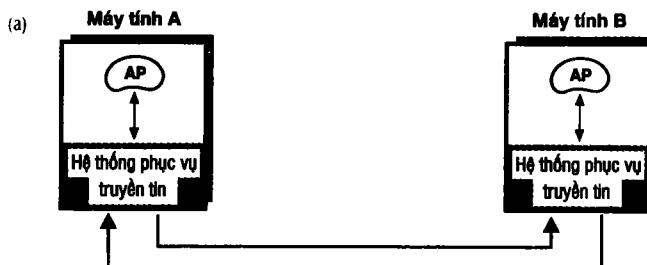
- Để truyền tin hiệu quả các chủ thể phải hiểu được thông điệp. Nơi thu nhận thông điệp phải có khả năng dịch thông điệp một cách chính xác. Điều này là hiển nhiên bởi vì trong giao tiếp hằng ngày nếu chúng ta dùng một từ mà người nghe không thể hiểu thì hiệu quả thông tin không đạt yêu cầu. Tương tự, nếu máy tính mong muốn thông tin đến với tốc độ chỉ định và ở một dạng mã nào đó nhưng thông tin lại đến với tốc độ khác và với dạng mã khác thì rõ ràng không thể đạt được hiệu quả truyền.
- Các đặc trưng toàn cục của một hệ thống truyền được xác định và bị giới hạn bởi các thuộc tính riêng của nguồn tin, của môi trường truyền và đích thu. Nhìn chung, dạng thông tin cần truyền quyết định kiểu nguồn tin, môi trường và đích thu.
- Trong một hệ thống truyền, hiện tượng nhiễu có thể xảy ra trong tiến trình truyền và thông điệp có thể bị ngắt quãng. Bất kỳ sự xâm nhập không mong muốn nào vào tín hiệu đều gọi là nhiễu. Có nhiều nguồn nhiễu và nhiễu dạng gây nhiễu khác nhau.

Hiểu biết được các nguyên tắc căn bản về truyền tin sẽ giúp chúng ta dễ dàng tiếp cận một lĩnh vực đặc biệt hấp dẫn đó là thông tin số liệu. Thông tin số liệu liên quan đến một tổ hợp nguồn thông tin, môi trường và máy thu trong các kiểu mạng truyền số liệu khác nhau.

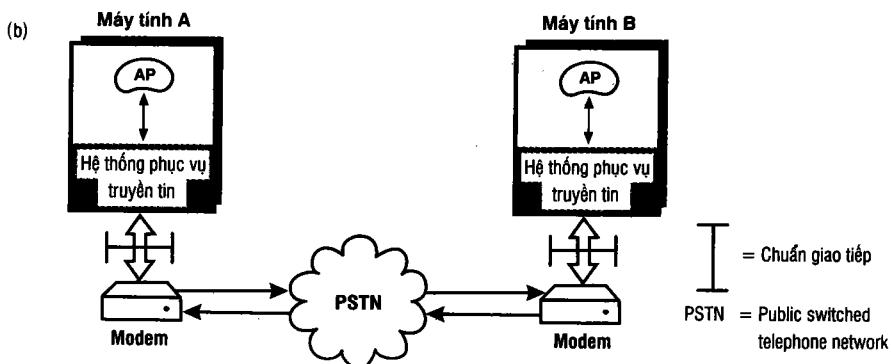
2. MẠNG TRUYỀN SỐ LIỆU

Dạng thức của phương tiện truyền số liệu được qui định bởi bản chất tự nhiên của ứng dụng, bởi số lượng máy tính liên quan và khoảng cách vật lý giữa chúng. Các dạng truyền số liệu được trình bày trên các hình 1.2 đến 1.6.

Nếu chỉ có hai máy tính và cả hai đều đặt ở một văn phòng, thì phương tiện truyền số liệu có thể chỉ gồm một liên kết điểm nối điểm đơn giản, như hình 1.2a. Tuy nhiên, nếu chúng tọa lạc ở những vị trí khác nhau trong một thành phố hay một quốc gia thì phải cần đến các phương tiện truyền tải công cộng. Mạng điện thoại công cộng được dùng nhiều nhất, trong trường hợp này sẽ cần đến bộ thích nghi gọi là Modem. Sắp xếp truyền theo dạng này được trình bày trên hình 1.2b.



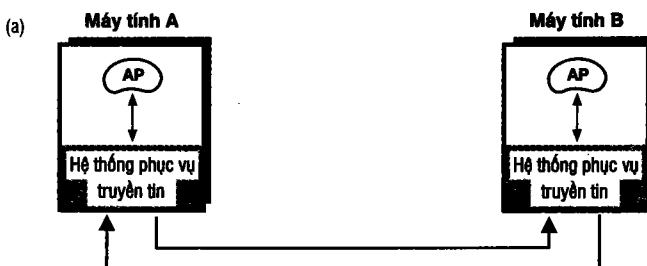
Hình 1.2 Truyền số liệu đơn giản giữa hai máy tính: (a) Nối dây trực tiếp



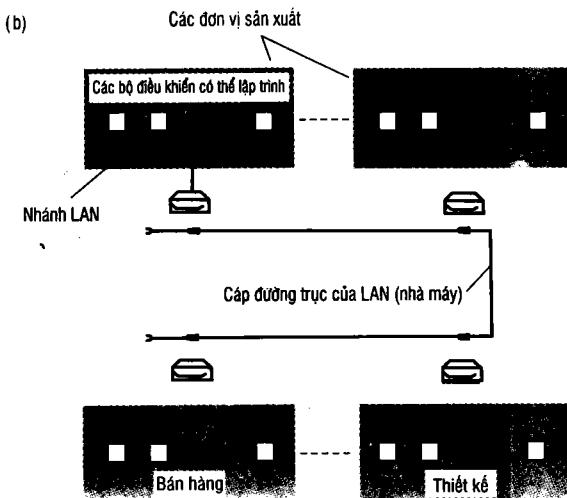
**Hình 1.2 (tiếp theo) Truyền số liệu đơn giản giữa hai máy tính:
(b) Nối qua mạng điện thoại công cộng dùng modem.**

Khi cần nhiều máy tính trong một ứng dụng, một mạng chuyển mạch sẽ được dùng cho phép tất cả các máy tính có thể liên lạc với nhau vào bất cứ thời điểm nào. Nếu tất cả các máy tính đều nằm trong một tòa nhà, có thể xây dựng một mạng riêng. Một mạng như vậy được xem như mạng cục bộ LAN (Local Area Network). Nhiều chuẩn mạng LAN và các thiết bị liên kết đã được tạo ra cho các ứng dụng thực tế. Hai hệ thống LAN cơ bản được trình bày trên hình 1.3.

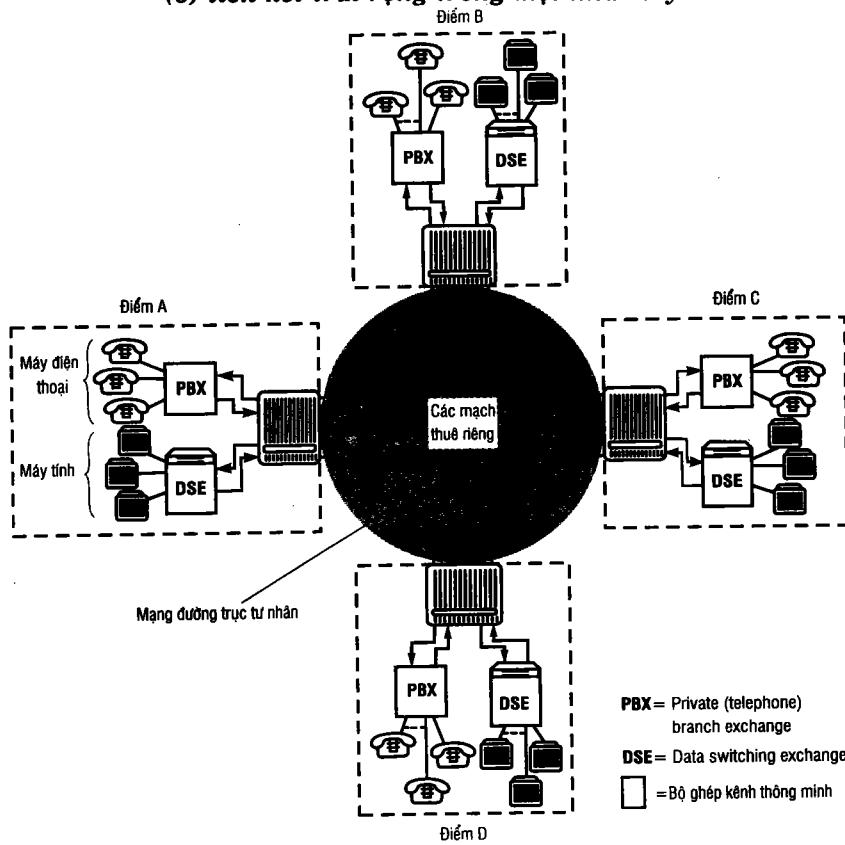
Khi nhiều máy tính tọa lạc ở nhiều nơi cách xa nhau cần liên lạc với nhau, phải dùng đến các phương tiện truyền dẫn công cộng. Việc liên kết các máy tính này tạo nên một mạng rộng lớn, được gọi là mạng điện rộng WAN (Wide Area Network). Kiểu mạng WAN được dùng phụ thuộc vào từng ứng dụng tự nhiên.



**Hình 1.3 Các hệ thống LAN cơ bản:
(a) Liên kết các LAN qua backbone trong một văn phòng**



**Hình 1.3 (tiếp theo) Các hệ thống LAN cơ bản:
(b) liên kết trải rộng trong một nhà máy.**

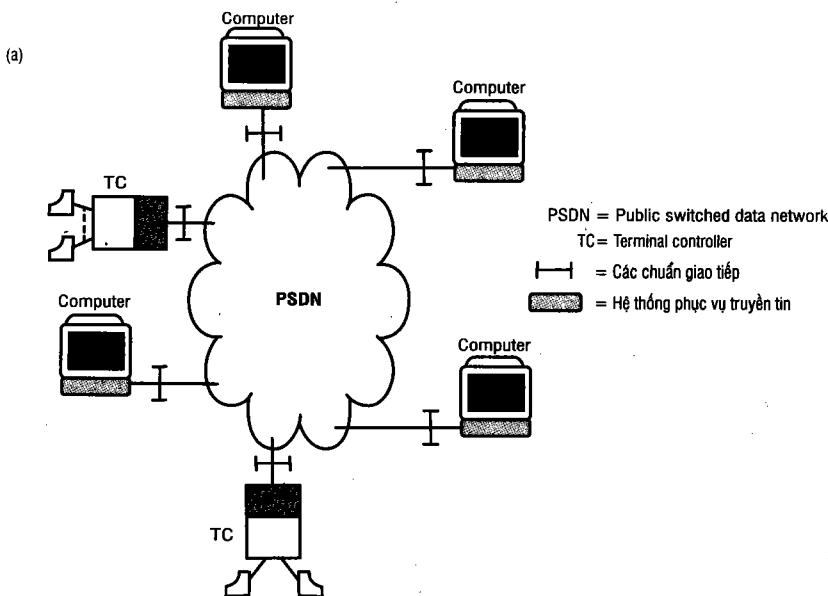


Hình 1.4 Mạng tư nhân tiêu biểu.

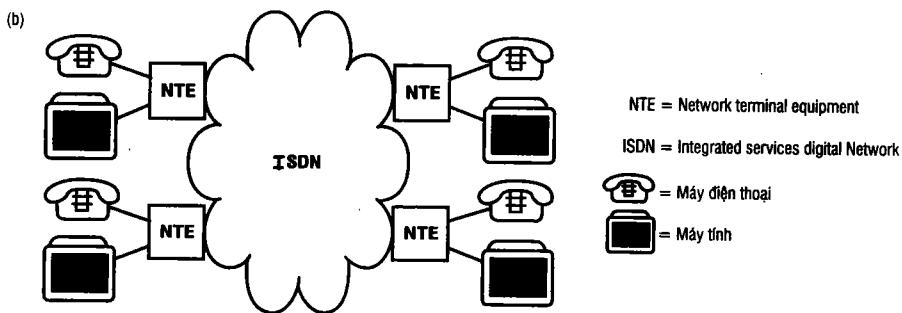


Ví dụ nếu tất cả các máy tính đều thuộc về một công ty và có yêu cầu truyền một số lượng dữ liệu quan trọng giữa các điểm, thì giải pháp đơn giản nhất cho vấn đề là thuê các đường truyền từ các nhà cung cấp phương tiện truyền dẫn và xây dựng hệ thống chuyển mạch riêng tại mỗi điểm để tạo thành mạng tư nhân. Nhiều công ty lớn đã thực hiện giải pháp này, các mạng như vậy thường kết hợp cả truyền số liệu và truyền thoại. Sơ đồ cơ bản của dạng này được minh họa trên hình 1.4.

Các giải pháp thuê kênh chỉ hiệu quả đối với các công ty lớn vì có tài hữu ích để cân đối với giá thuê kênh, giá lắp đặt và vận hành một mạng tư nhân. Trong hầu hết các trường hợp khác đều cần đến các mạng truyền dẫn công cộng. Bên cạnh việc cung cấp dịch vụ điện thoại công cộng, ngày nay hầu hết các nhà cung cấp dịch vụ truyền dẫn đều cung cấp một dịch vụ chuyển mạch số liệu mang tính công cộng. Thật ra các mạng này cũng tương tự như mạng PSTN (Public Switched Telephone Network) là được liên kết quốc tế, chỉ khác ở chỗ được thiết kế chuyên biệt cho truyền số liệu. Như vậy các ứng dụng liên quan đến máy tính được phục vụ bởi mạng số liệu chuyển mạch công cộng PSDN (Public Switched Data Network). Ngoài ra, các nhà khai thác dịch vụ cũng chuyển đổi các mạng PSTN có sẵn của họ sao cho có thể truyền được số liệu mà không cần dùng modem. Các mạng này hoạt động trong chế độ số (digital) hoàn toàn và được gọi là các *mạng số liệu kết đa dịch vụ ISDN* (Integrated Services Digital Network). Sơ đồ tiêu biểu được trình bày trên hình 1.5.



Hình 1.5 Các mạng số liệu công cộng: (a) PSDN

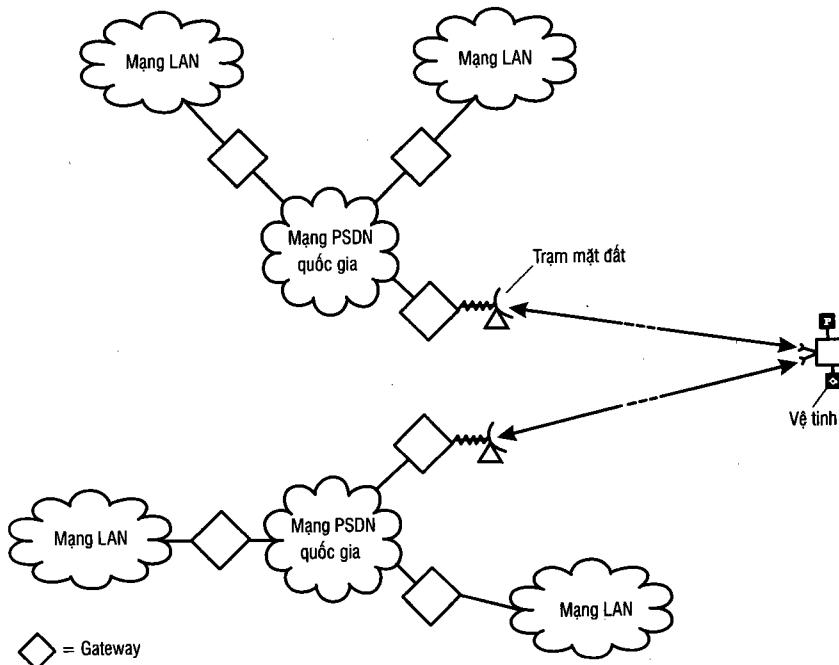
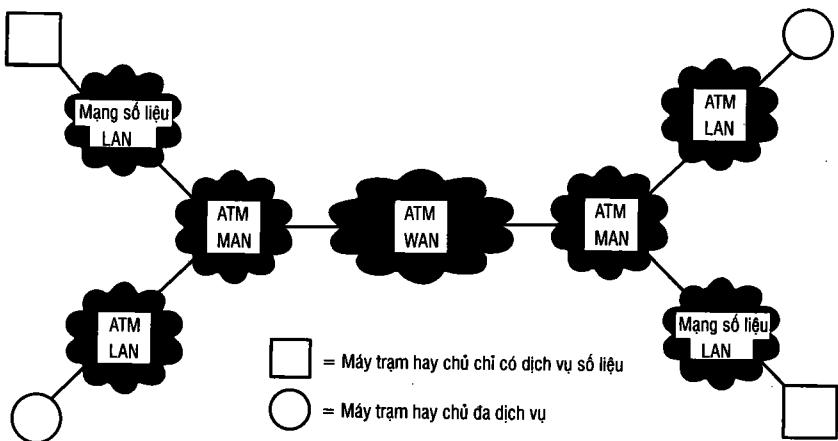


Hình 1.5 (tiếp theo) Các mạng số liệu công cộng: (b) ISDN.

Trên đây chúng ta đã già sử trong tất cả các ứng dụng, các máy tính đều được nối vào cùng một mạng LAN hay WAN. Tuy nhiên, có nhiều ứng dụng phải dùng phương tiện truyền gồm nhiều mạng kết hợp ví dụ như LAN-WAN-LAN. Ví dụ một máy tính nối vào một LAN cần liên lạc với một máy tính khác thuộc về mạng LAN khác, và hai LAN được nối qua PSDN. Dạng truyền số liệu này được gọi là liên mạng (Internetwork) hay Internet, Khi đó địa chỉ hóa là nhu cầu thiết yếu trong mỗi liên hệ cùng mạng và cả trong mỗi giao tiếp với các máy tính thuộc mạng khác. Ví dụ về liên mạng được trình bày trên hình 1.6.

Các mạng vừa đề cập ở trên được thiết kế chủ yếu để truyền số liệu giữa các máy trạm chỉ hỗ trợ dịch vụ số liệu. Gần đây, các máy trạm được cải tiến hỗ trợ các dịch vụ không chỉ liên quan đến số liệu mà còn bao gồm hàng loạt các dạng thông tin khác. Các máy trạm như vậy được cung cấp thêm các trang thiết bị về điện thoại thấy hình, hội nghị qua màn hình và nhiều dịch vụ đa phương tiện khác. Nhằm hỗ trợ nhiều dịch vụ đồng thời, một thế hệ mạng mới đã được phát triển được gọi là các mạng đa dịch vụ băng rộng BMN (Broadband Multiservice Network).

Vì sự khác nhau giữa truyền số liệu, truyền thoại cũng như truyền hình ảnh mà đã có nhiều công trình nghiên cứu nhằm tìm ra một giải pháp mạng hỗ trợ đồng thời các dịch vụ này. Một phương thức mới đã được tạm chấp nhận có khả năng truyền và chuyển mạch trong các mạng đa dịch vụ. Tiếp cận mới này được gọi là chế độ truyền bất đồng bộ ATM (Asynchronous Transfer Mode). Các ATM LAN hiện nay dựa trên chế độ hoạt động này. Một thế hệ ATM WAN mới đã được thiết kế nhằm liên kết các ATM LAN với nhau. Ngoài ra còn có một kiểu mạng mới được gọi là mạng đô thị (MAN_Metropolitan Area Network) được tạo ra để liên kết các ATM LAN, các LAN chỉ dùng cho số liệu phân bố trong nội đô và thành phố. Sơ đồ tiêu biểu của một mạng đa dịch vụ băng rộng được trình bày trên hình 1.7.

**Hình 1.6** Liên mạng.**Hình 1.7** Các mạng đa dịch vụ băng rộng.

3. SỰ CHUẨN HÓA VÀ MÔ HÌNH THAM CHIẾU ISO

3.1. Chuẩn hóa các mô hình truyền số liệu

Trước đây, các chuẩn được dùng trong công nghệ máy tính của các tổ chức quốc tế khác nhau để cập chủ yếu đến các hoạt động bên trong của máy tính hay

các kết nối thiết bị ngoại vi mang tính cục bộ. Kết quả là các hệ thống phần cứng và phần mềm truyền số liệu xuất hiện sớm từ mỗi nhà máy chỉ cho phép chạy trên các máy tính của chính nhà sản xuất đó để trao đổi thông tin với nhau. Các hệ thống như vậy được gọi là *hệ thống đóng*, vì các máy tính từ các nhà máy khác không thể trao đổi thông tin với chúng trừ khi tuân thủ các chuẩn của nhà máy có sản phẩm này.

Ngược lại, các tổ chức quốc tế khác liên quan đến các mạng truyền dẫn công cộng đã trải qua nhiều năm xây dựng nên có các tiêu chuẩn thống nhất mang tính quốc tế để kết nối thiết bị vào trong mạng. Các khuyến nghị họ V-series liên quan đến tiêu chuẩn kết nối thiết bị, thường là thiết bị đầu cuối số liệu DTE (Data Terminal Equipment) nối vào một modem để truyền qua mạng PSTN; các khuyến nghị X-series cho kết nối DTE vào mạng dữ liệu công cộng PSDN; và các khuyến nghị I-series cho kết nối DTE vào mạng ISDN. Các khuyến nghị là kết quả của sự tương thích giữa thiết bị từ các nhà cung cấp sản phẩm khác nhau, cho phép khách hàng có thể chọn thiết bị phù hợp từ nhiều nhà máy khác nhau.

Đầu tiên các dịch vụ được cung cấp bởi hầu hết các công ty truyền dẫn công cộng liên quan chủ yếu đến truyền số liệu, do đó các chuẩn liên kết chỉ đề cập đến phương thức giao tiếp giữa một thiết bị vào các mạng. Gần đây, các công ty truyền dẫn đã bắt đầu cung cấp nhiều dịch vụ thông tin phân bố mở rộng như trao đổi các bản tin điện tử và truy xuất vào các cơ sở dữ liệu. Để phục vụ cho các dịch vụ này, các tổ chức chuẩn hóa liên quan đến công nghệ viễn thông đã xây dựng không chỉ các chuẩn giao tiếp mạng mà còn xây dựng các chuẩn mức cao liên quan đến dạng thức, cú pháp và điều khiển trao đổi thông tin giữa các hệ thống. Trên cơ sở đó, thiết bị từ bất cứ nhà máy nào tuân thủ các chuẩn này có thể được dùng thay thế với thiết bị từ bất kỳ nhà máy nào khác cũng tuân thủ các tiêu chuẩn này. Một hệ thống được xây dựng trên nguyên tắc như vậy được gọi là *hệ thống mở* hay nói đầy đủ hơn là môi trường liên kết hệ thống mở (OSIE: Open System Interconnection Environment). Hình 1.8 trình bày tóm tắt sự phát triển của các chuẩn cùng với các tổ chức chuẩn hóa chủ yếu.

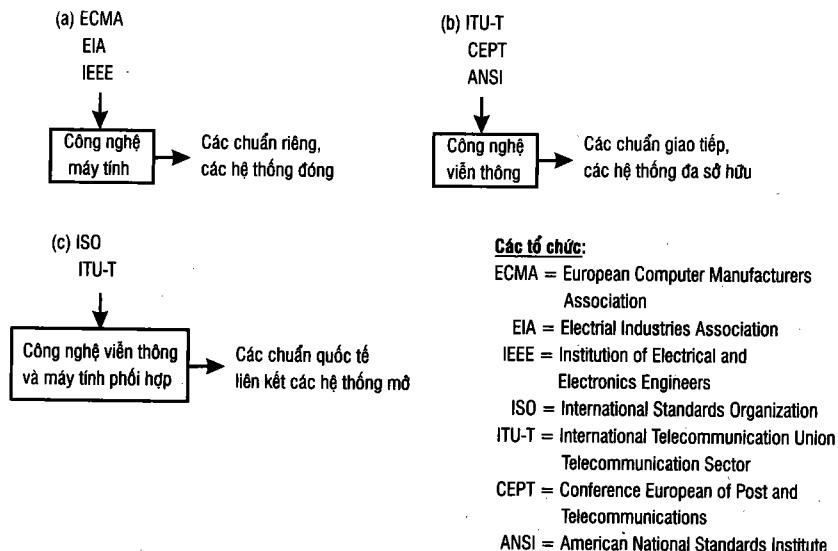
Vào khoảng giữa thập niên 70 của thế kỷ XX, khi mà các loại hệ thống phân bố khác nhau bắt đầu xuất hiện, các ưu điểm của hệ thống mở đã được minh chứng qua công nghệ máy tính. Kết quả là một loạt các tiêu chuẩn được công bố. Tiêu chuẩn đầu tiên đề cập đến kiến trúc tổng thể của hệ thống thông tin hoàn chỉnh trong mỗi máy tính là chuẩn được công bố bởi Tổ chức tiêu chuẩn quốc tế ISO (international standards organization) và được gọi là Mô hình tham chiếu của ISO cho liên kết các hệ thống mở (OSI -open systems interconnection).

Mục đích của mô hình OSI là cung cấp khuôn mẫu cho sự phối hợp phát triển các chuẩn và cho phép các chuẩn hiện có phù hợp khuôn mẫu này. Mục tiêu cụ thể là cho phép một quá trình ứng dụng trong bất kỳ máy tính nào được hỗ trợ một tập các tiêu chuẩn đặc biệt để thông tin với một quá trình ứng dụng trong một máy tính khác có hỗ trợ cùng một chuẩn, bất chấp nó có nguồn gốc từ nhà máy nào.

Ví dụ các quá trình ứng dụng cần liên lạc theo phương thức mở :

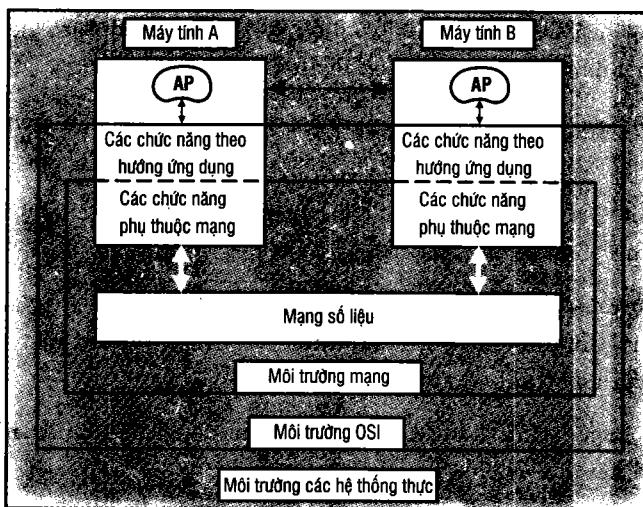
- Một quá trình thực thi bên trong máy tính và truy xuất một hệ thống tập tin ở xa.
- Một quá trình đóng vai trò dịch vụ tập tin trung tâm (server) đối với cộng đồng phân bố của quá trình (client).
- Một quá trình xảy ra trong máy trạm truy xuất đến dịch vụ thư điện tử.
- Một quá trình đóng vai trò hộp thư chủ (mail server) đối với cộng đồng phân bố của quá trình (client).
- Một quá trình trong một máy tính chủ kiểm soát cộng đồng phân bố máy tính hay các bộ điều khiển robo liên kết với một quá trình tự động hóa trong nhà máy.
- Một quá trình trong trang thiết bị tự động điều khiển nhận lệnh và trả kết quả về cho hệ thống chủ.
- Một quá trình trong một máy tính ngân hàng khởi động các thao tác ghi nợ và tín dụng trên một hệ thống ở xa.

OSI liên quan đến hoạt động trao đổi thông tin giữa các quá trình như trên. Nó cho phép các quá trình ứng dụng cùng nhau chia sẻ các nhiệm vụ xử lý thông tin đặc biệt mà không cần biết chúng đang chạy trên hệ thống máy tính nào.



Hình 1.8 Sự phát triển chuẩn và các Tổ chức chuẩn hóa chính.

3.2. Mô hình tham chiếu OSI



Hình 1.9 Các môi trường hoạt động.

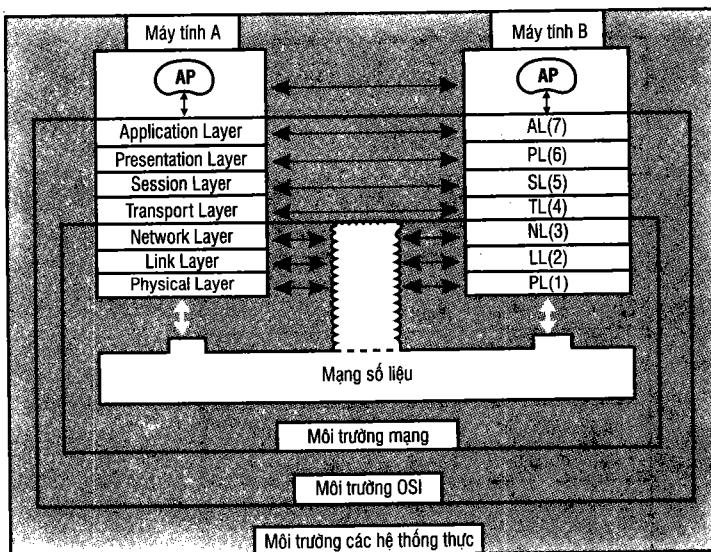
Một hệ thống truyền tin là một cơ cấu phức tạp gồm cả phần cứng và phần mềm. Các nỗ lực đầu tiên nhằm hiện thực phần mềm cho hệ thống như vậy thường dựa vào một chương trình đơn phức tạp không có cấu trúc (thường được viết bằng ngôn ngữ assembly) mang nhiều thành phần giao tác. Các phần mềm như vậy thường khó kiểm thử và sửa chữa.

Để khắc phục tình trạng này, ISO đã thừa nhận một tiếp cận phân lớp cho mô hình tham khảo. Trong đó, một hệ thống truyền thông hoàn chỉnh được phân thành một số lớp, mỗi lớp thực hiện một chức năng đặc biệt. Về mặt khái niệm có thể xem các lớp này thực thi một trong hai chức năng tổng quát: các chức năng phụ thuộc mạng và các chức năng thiên hướng ứng dụng (application-oriented). Điều này tạo ra ba môi trường hoạt động tách biệt:

1. **Môi trường mạng:** liên quan đến các giao thức và các tiêu chuẩn thuộc về các dạng khác nhau của hạ tầng cơ sở mạng truyền số liệu.
2. **Môi trường OSI:** bao gồm môi trường mạng và thêm vào các giao thức cũng như tiêu chuẩn hướng ứng dụng để cho phép các hệ thống đầu cuối liên lạc với đầu cuối khác theo phương thức mở.

3. **Môi trường hệ thống thực:** nó xây dựng nên môi trường OSI, liên quan đến các dịch vụ và phần mềm đặc trưng của nhà chế tạo, các dịch vụ và phần mềm đặc trưng này được phát triển nhằm thực hiện một nhiệm vụ xử lý thông tin phân tán đặc biệt nào đó.

Các môi trường được mô tả trong hình 1.9

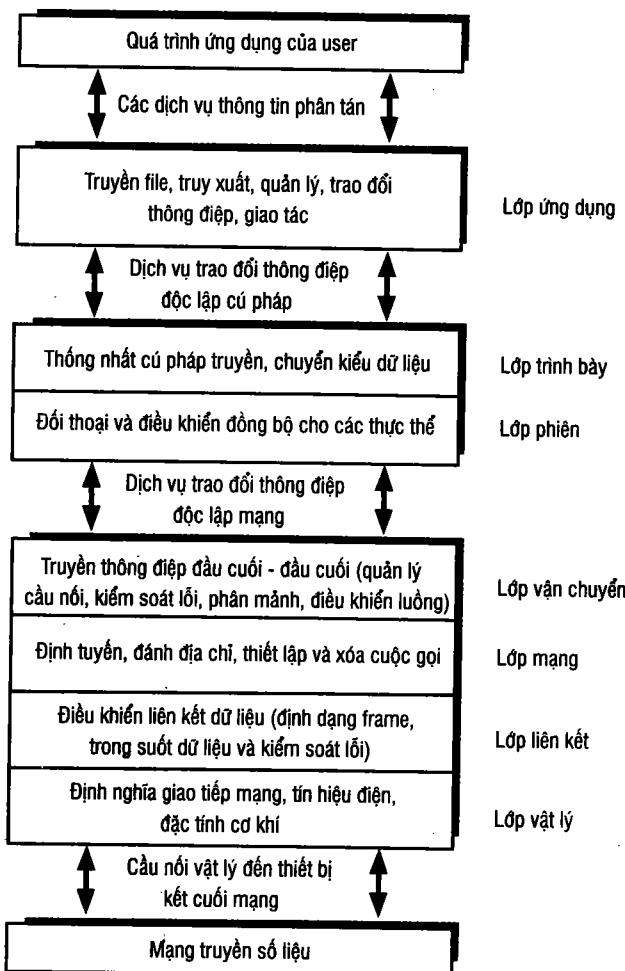


Hình 1.10 Kiến trúc toàn cục của mô hình tham chiếu OSI.

Cả hai thành phần phụ thuộc mạng và hướng ứng dụng của mô hình OSI được triển khai theo một số lớp. Ranh giới giữa mỗi lớp và các chức năng được thực hiện tại mỗi lớp được chọn trên cơ sở kinh nghiệm có được từ các hoạt động chuẩn hóa trước đây.

Mỗi lớp thực thi một chức năng đặc biệt theo ngữ cảnh của hệ thống thông tin toàn cục. Lớp sẽ hoạt động tùy vào một giao thức được định nghĩa_tập luật_nhờ trao đổi các thông điệp bao gồm thông điệp số liệu và thông điệp điều khiển với lớp ngang hàng trên hệ thống đầu xa. Mỗi lớp có một giao tiếp đặc biệt với các lớp kế trên và kế dưới. Công việc hiện thực một lớp giao thức đặc biệt là độc lập với tất cả các lớp khác.





Hình 1.11 Tóm tắt lớp giao thức.

Kiến trúc luận lý của mô hình tham chiếu của ISO được thiết lập thành 7 lớp giao thức, được minh họa trên hình 1.10. Ba lớp dưới cùng (1 đến 3) là phụ thuộc mạng và liên hệ mật thiết với các giao thức liên quan đến mạng truyền số liệu đang được dùng để liên kết hai hệ thống máy tính với nhau. Ba lớp trên cùng (5-7) là thiên hướng ứng dụng và liên quan đến các giao thức cho phép hai quá trình ứng dụng ở hai đầu cuối tương tác với nhau, thông thường phải trải qua một loạt các dịch vụ được cung cấp bởi hệ điều hành cục bộ. Lớp giữa (lớp 4) che các lớp hướng ứng dụng ở trên đối với các hoạt động cụ thể của các lớp phụ thuộc mạng bên dưới. Chủ yếu là đi xây dựng dịch vụ trao đổi thông điệp độc lập mạng trên cơ

sở các dịch vụ được cung cấp từ các lớp phụ thuộc mạng để cung cấp cho các lớp hướng ứng dụng phía trên.

Chức năng của mỗi lớp được đặc tả một cách hình thức như một giao thức, giao thức xác lập tập luật và tiêu chuẩn được dùng bởi lớp này để thông tin với lớp ngang hàng (tương tự) trên hệ thống khác (ở xa). Mỗi lớp cung cấp một tập xác định các dịch vụ cung cấp cho lớp kế trên. Cũng dùng các dịch vụ được cung cấp bởi lớp kế dưới để tải các đơn vị thông điệp liên hệ với giao thức đi đến lớp tương tự ở hệ thống đầu xa. Ví dụ, lớp 4 cung cấp dịch vụ tải thông điệp độc lập mạng cho các lớp phía trên nó và dùng các dịch vụ được cung cấp bởi lớp mạng bên dưới để tải các đơn vị thông điệp liên hệ với giao thức truyền tải đến lớp vận chuyển (lớp 4) ở đầu xa. Như vậy mỗi lớp sẽ liên lạc với lớp ngang hàng với nó ở hệ thống đầu xa tùy vào giao thức đã được định nghĩa. Tuy nhiên, trong thực tế các đơn vị thông điệp giao thức của lớp được chuyển thông qua các phương tiện dịch vụ được cung cấp bởi lớp kế dưới. Các chức năng cơ bản của mỗi lớp được tóm tắt trong hình 1.11.

3.2.1. Các lớp hướng ứng dụng

Lớp ứng dụng cung cấp giao tiếp user_thông thường dưới dạng một chương trình/quá trình_cho một dải các dịch vụ thông tin phân tán thông dụng. Các dịch vụ bao gồm truy xuất tập tin, quản lý cũng như các dịch vụ trao đổi thông điệp như thư điện tử. Hiện có một số giao thức chuẩn cho các loại dịch vụ này.

Việc truy xuất các dịch vụ ứng dụng thường phải dựa vào một tập hàm thực thể xác định, mỗi hàm thực thể có các tham số liên hệ, được hỗ trợ bởi hệ điều hành cục bộ. Các hàm thực thể truy xuất cũng giống như các lời gọi hàm của hệ điều hành (ví dụ được dùng để truy xuất hệ thống tập tin cục bộ) và kết quả là một thủ tục/chương trình con thích hợp của hệ điều hành được kích hoạt. Các thủ tục/chương trình con của hệ điều hành này sử dụng hệ thống truyền số liệu (gồm cả phần cứng và phần mềm) như là thiết bị cục bộ của nó. Do đó, việc hiện thực và hoạt động của hệ thống truyền số liệu là trong suốt đối với các quá trình ứng dụng (user). Khi quá trình ứng dụng tiến hành một lời gọi thủ tục và thủ tục được thực thi, thì một hay nhiều tham số trạng thái được trả về để chỉ ra sự thành công của giao tác mạng được yêu cầu.

Bên cạnh truyền tải thông tin, lớp ứng dụng còn cung cấp các dịch vụ như sau:

- Nhận dạng các chủ thể tham gia thông tin qua tên và địa chỉ

- Xác định khả năng hiện hành của một chủ thể tham gia thông tin
- Thiết lập thẩm quyền thông tin
- Thống nhất cơ cấu bảo mật
- Cấp quyền cho các chủ thể tham gia thông tin
- Chọn qui tắc đàm thoại, bao gồm các thủ tục khởi tạo và xóa bỏ
- Thống nhất trách nhiệm khắc phục lỗi
- Xác nhận các ràng buộc trên cú pháp dữ liệu (các tập ký tự, các cấu trúc dữ liệu...)

Lớp trình bày (presentation layer)

Lớp trình bày liên quan đến việc biểu diễn dữ liệu (cú pháp) trong quá trình truyền giữa hai quá trình ứng dụng. Để liên kết đạt hiệu quả, một số dạng cú pháp trùu tượng đã được định nghĩa dùng cho các quá trình ứng dụng cùng với các cú pháp truyền liên hệ. Lớp trình bày đàm phán và chọn cú pháp truyền thích hợp được dùng trong quá trình giao tác để duy trì cấu trúc của thông điệp đang được trao đổi giữa hai thực thể ứng dụng. Nếu hình thức trình bày này khác với dạng trùu tượng bên trong thì giao thức trình bày sẽ thực hiện các chuyển đổi cần thiết.

Để minh họa các dịch vụ được cung cấp bởi lớp trình bày chúng ta xem xét một cuộc nói chuyện giữa một người Việt nam và một người Nhật. Giả sử mỗi người đều dùng một thông dịch viên và hai thông dịch viên chỉ trao đổi với nhau bằng tiếng Anh. Mỗi thông dịch viên phải dịch từ ngôn ngữ địa phương sang tiếng Anh và ngược lại. Hai người trao đổi thông tin với nhau tương tự như hai quá trình ứng dụng với hai thông dịch viên đại diện các thực thể lớp trình bày. Tiếng Việt và tiếng Nhật là các cú pháp cục bộ và tiếng Anh là cú pháp truyền tải. Lưu ý rằng phải có một ngôn ngữ phổ biến được xác lập nhằm cho phép ngôn ngữ truyền tải phù hợp được trao đổi. Cũng lưu ý rằng các thông dịch viên không nhất thiết phải hiểu ngữ nghĩa của cuộc đàm thoại.

Một chức năng khác của lớp trình bày liên quan đến bảo mật số liệu. Trong một vài ứng dụng, trước khi truyền số liệu của một ứng dụng được mã hóa với một khóa nào đó, và chỉ có lớp trình bày của hệ thống thu hợp lệ mới biết được. Lớp trình bày thu giải mã bất kỳ dữ liệu nào thu được bằng cách dùng một khóa tương ứng trước khi chuyển nó cho quá trình ứng dụng. Vấn đề mật mã sẽ được trình bày trong chương 4.

Lớp phiên (session layer)

Lớp này cho phép hai thực thể giao thức lớp ứng dụng tổ chức và đồng bộ đối thoại cũng như quản lý việc trao đổi số liệu giữa chúng. Do đó, trách nhiệm của nó là thiết lập (hay xóa) một kênh truyền giữa hai thực thể giao thức lớp ứng dụng (thực tế là hai thực thể giao thức lớp trình bày) trong một quá trình giao tác mạng hoàn chỉnh. Ở đây có một số dịch vụ tùy chọn được cung cấp, bao gồm:

- Quản lý giao tác: hoạt động trao đổi dữ liệu có thể là song công hay bán song công. Trong trường hợp bán song công, giao thức lớp trình bày cung cấp các phương tiện để điều khiển công việc trao đổi dữ liệu theo phương thức đồng bộ.
- Đồng bộ: trong các giao tác mạng dài, user (through qua các dịch vụ được cung cấp bởi lớp phiên) có thể chọn thiết lập đồng bộ tại các thời điểm liên quan với hoạt động truyền một cách định kỳ. Nếu xuất hiện lỗi trong thời gian của một giao tác, việc đổi thoại có thể được khởi động lại tại một thời điểm đồng bộ đã thống nhất trước.
- Thông báo đặc biệt: các trường hợp cá biệt không thể khắc phục phát sinh trong tiến trình giao tác có thể được lớp phiên thông báo với lớp ứng dụng.

Lớp chuyển tải (transport layer)

Lớp chuyển tải đóng vai trò như một giao tiếp giữa các lớp hướng ứng dụng cao hơn với các lớp giao thức phụ thuộc mạng bên dưới. Nó cung cấp cho lớp phiên một phương tiện truyền độc lập với loại mạng đang dùng. Bằng cách cung cấp cho lớp phiên một tập các phương tiện truyền thông điệp mà lớp chuyển tải đã che được hoạt động chi tiết của mạng bên dưới đối với lớp phiên.

Lớp chuyển tải cung cấp một số các lớp dịch vụ để bù đắp vào phẩm chất dịch vụ (QoS) lớp mạng, do phẩm chất dịch vụ này phụ thuộc các kiểu mạng khác nhau. Có năm lớp dịch vụ bắt đầu từ lớp 0, lớp 0 chỉ cung cấp các chức năng cơ bản cần cho thiết lập cầu nối và truyền số liệu, đến lớp 4 sẽ cung cấp các thủ tục kiểm soát lỗi và điều khiển luồng (flow control) đầy đủ. Ví dụ lớp 0 có thể được chọn để dùng với PSDN trong khi lớp 4 có thể được dùng với PSTN.

3.2.2. Các lớp phụ thuộc mạng

Ba lớp thấp nhất trong mô hình tham chiếu OSI là phụ thuộc mạng, hoạt động chi tiết của chúng thay đổi từ mạng này sang mạng khác. Tuy nhiên, tổng quát thì lớp mạng (network layer) chịu trách nhiệm thiết lập và xóa một cầu nối giữa hai thực thể giao thức lớp chuyển tải. Nó bao gồm các chức năng như định tuyến mạng và trong một vài trường hợp cho phép điều khiển luồng xuyên qua

giao tiếp máy tính_mạng. Trong trường hợp liên mạng nó cung cấp các chức năng tương thích giữa các mạng liên kết.

Lớp liên kết (link layer) xây dựng nên cầu nối luận lý và vật lý lấy từ hệ thống mạng đường truyền đặc biệt nào đó để cung cấp cho lớp mạng các phương tiện truyền tin cậy. Nó chịu trách nhiệm thực hiện các chức năng như phát hiện lỗi, ngay cả các lỗi truyền dẫn và truyền lại các gói tin bị hỏng. Thông thường có hai loại dịch vụ cơ bản được cung cấp:

1. Không tạo cầu nối (connectionless): nó xem đơn vị thông tin như một cá thể hoàn chỉnh, và được truyền theo phương thức tổng lực (best-trty), nghĩa là nếu phát hiện các lỗi trong frame dữ liệu thì không còn lực để khắc phục, đành phải loại bỏ frame.
2. Tạo cầu nối (connection-oriented): cố gắng cung cấp một phương tiện truyền không bị lỗi.

Cuối cùng là lớp vật lý (physical layer) liên quan đến giao tiếp về điện giữa các thiết bị đầu cuối user và thiết bị kết cuối mạng. Nó cung cấp cho lớp liên kết các phương tiện truyền dòng bit nối tiếp giữa hai thiết bị.

4. CÁC CHUẨN HỆ THỐNG MỞ (OPEN SYSTEM STANDARDS)

Mô hình tham chiếu của ISO được kiến thiết đơn giản như là khuôn mẫu về kiến trúc của một hệ thống thông tin, trong đó đặt nền tảng cho hoạt động của các chuẩn liên quan đến mỗi lớp. Không phải chỉ có một chuẩn cố định tại mỗi lớp, thay vì vậy tại mỗi lớp có một tập chuẩn liên quan. Đối với các môi trường OSI đặc biệt, một tập chuẩn được định nghĩa để sử dụng cho tất cả các hệ thống trong môi trường này.

Ba tổ chức chính tạo ra các chuẩn cho truyền thông máy tính là ISO, IEEE (American Institution of Electrical and Electronic Engineers) và ITU-T (International Telecommunications Union-Telecommunications Sector) hay CCITT (International Telegraph and Telephone Consultative Committee). Đặc biệt, ISO và IEEE đã đưa ra các chuẩn sử dụng trong các nhà máy chế tạo máy tính trong khi đó ITU-T định nghĩa các chuẩn để kết nối thiết bị vào các mạng công cộng quốc gia và quốc tế khác nhau. Mức độ hội nhập giữa công nghệ máy tính và công nghệ viễn thông ngày càng tăng đã kéo theo sự gia tăng mức phối hợp và nhất quán giữa các chuẩn được tạo ra bởi các tổ chức này.

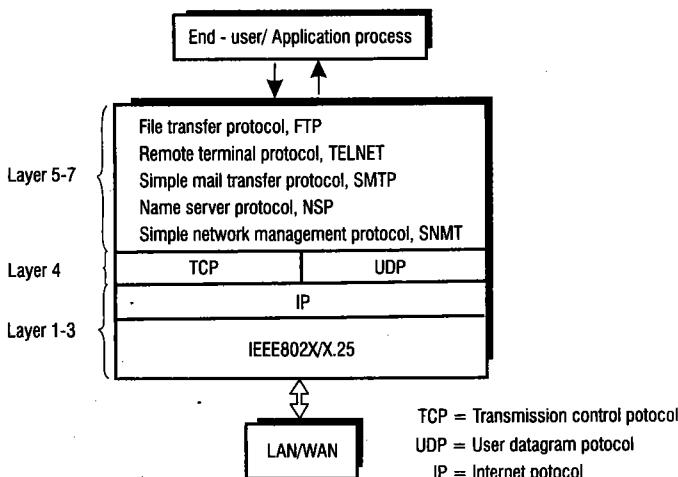
Ngoài ra, trước đây song song với hoạt động của ISO, cơ quan DARPA (Defense Advanced Research Projects Agency) của Bộ quốc phòng Hoa Kỳ cũng đã



nghiên cứu lĩnh vực máy tính và liên mạng. Như là một phần trong công việc nghiên cứu này, mạng máy tính của các trường Đại học và các Trung tâm nghiên cứu khác cũng được nối đến DARPA. Kết quả là ARPANET đã ra đời và ngày nay được mở rộng để kết nối các liên mạng của các tổ chức khác. Liên mạng phối hợp này, ngày nay gọi là Internet.

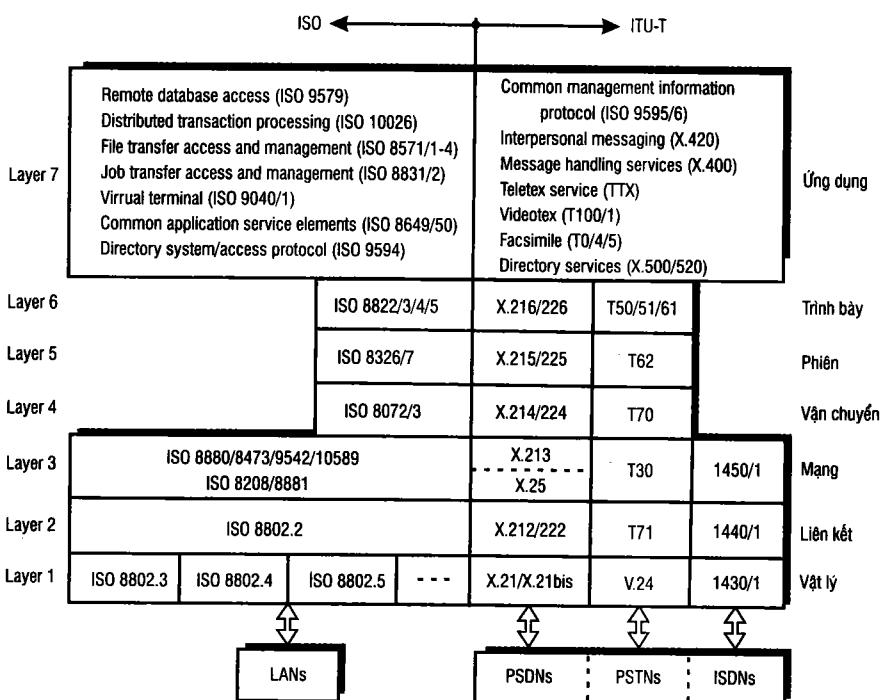
Bộ giao thức được dùng với Internet được gọi là TCP/IP (Transmission Control Protocol/Internet Protocol). Nó bao gồm cả các giao thức theo hướng mạng và các giao thức hỗ trợ ứng dụng. Bởi TCP/IP được dùng rộng rãi trong liên mạng có sẵn, nên nhiều giao thức TCP/IP đã được dùng như là giao thức cơ bản cho các chuẩn ISO. Hơn nữa, vì tất cả các đặc tả giao thức liên quan đến TCP/IP mang tính cộng đồng _do đó không cần bản quyền_chúng được dùng rộng rãi bởi các tổ chức thương mại và công cộng để tạo ra môi trường liên kết hệ thống mở. Trong thực tế có hai chuẩn hệ thống mở chủ yếu: là bộ giao thức TCP/IP và TCP/IP dựa trên các tiêu chuẩn của ISO.

Hình 1.12 trình bày vài chuẩn liên quan đến bộ giao thức TCP/IP. Như chúng ta thấy, vì TCP/IP được phát triển đồng thời với sự khởi đầu của ISO, nên nó không chứa các giao thức đặc biệt liên quan đến tất cả các lớp ISO. Hơn nữa, phương pháp đặc tả được dùng cho bộ giao thức TCP/IP khác với chuẩn được dùng cho các tiêu chuẩn ISO. Dầu sao đi nữa, hầu hết các chức năng liên quan với các lớp ISO cũng được nhúng vào trong bộ giao thức TCP/IP.



Hình 1.12 Bộ giao thức TCP/IP

Trong trường hợp của các chuẩn ISO/ITU-T, như trình bày trên hình 1.13, có một loạt các chuẩn liên quan đến mỗi lớp. Chúng cho phép chọn tập chuẩn phù hợp nhất cho ứng dụng trong quá trình thiết lập môi trường hệ thống mở. Tập giao thức kết quả là một **profile liên kết hệ thống mở** (open system interconnection profile). Một số các profile như vậy đã được định nghĩa, bao gồm : TOP là một tập giao thức dùng trong môi trường kỹ thuật và văn phòng; MAP là tập giao thức dùng trong nhà máy...



Hình 1.13 Tóm tắt các chuẩn

CHƯƠNG 2

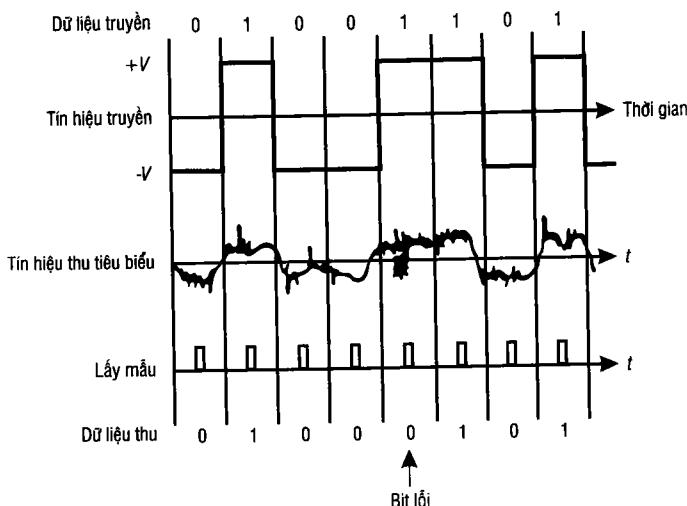
GIAO TIẾP VẬT LÝ

TỔNG QUAN

Để truyền dữ liệu nhị phân qua một đường dây, các chữ số nhị phân tạo nên mỗi phần tử truyền đi phải được chuyển thành các tín hiệu điện. Ví dụ có thể truyền một bit nhị phân 1 bằng cách đặt lên đường dây biên độ điện thế $+V$ và truyền bit nhị phân 0 với mức điện thế $-V$. Khi nhận các tín hiệu điện này, thiết bị thu sẽ dịch $+V$ thành 1 và $-V$ thành 0. Trong thực tế, các tín hiệu điện được truyền đi bị suy giảm và méo dạng bởi môi trường truyền, đôi khi bộ thu không thể phân tách đâu là tín hiệu 1 và đâu là tín hiệu 0, như minh họa trên hình 2.1. Mức độ suy giảm và méo dạng chịu ảnh hưởng nhiều nhất bởi:

- Loại môi trường truyền
- Tốc độ bit đang truyền
- Cự ly giữa hai thiết bị truyền.

Vì sự suy giảm và méo dạng trong các loại môi trường truyền và các thành phần vật lý khác nhau là khác nhau, nên các tiêu chuẩn quốc tế đã được định nghĩa cho giao tiếp điện giữa hai chủng loại thiết bị truyền dữ liệu. Các chuẩn này không chỉ định nghĩa các mức tín hiệu điện được dùng mà còn chỉ ra cách thức áp dụng và ý nghĩa của bất kỳ tín hiệu điều khiển nào cùng với các tiêu chuẩn được dùng tại giao tiếp vật lý. Hai tổ chức xây dựng các chuẩn liên kết các thiết bị truyền số liệu là ITU-T (International Telecommunication Union-Telecommunication Sector)-trước đây là CCITT-tại Châu Âu và EIA (Electrical Industry Association) tại Mỹ. Mặc dù các chuẩn được định nghĩa bởi cả hai tổ chức dùng kỹ thuật hơi khác nhau nhưng các tín hiệu cơ bản và ý nghĩa của chúng hoàn toàn giống nhau.



Hình 2.1 Ánh hưởng của môi trường truyền.

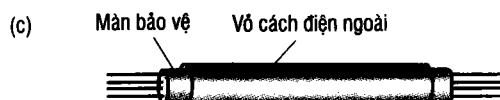
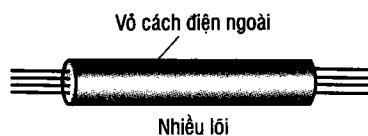
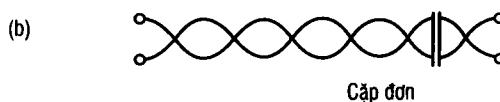
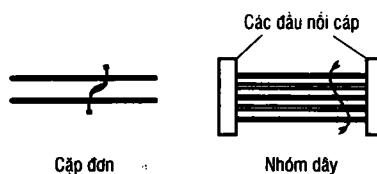
Cho dù hầu hết các trường hợp chúng ta sẽ xem xét là giao tiếp của một máy tính với các thành phần giao tiếp truyền số liệu khác nhau, nhưng thường dùng thuật ngữ 'thiết bị đầu cuối' DTE (Data Terminal Equipment) thay cho 'máy tính', đó là ngụ ý cho bất kỳ loại thiết bị đầu cuối nào.

1. Môi trường truyền

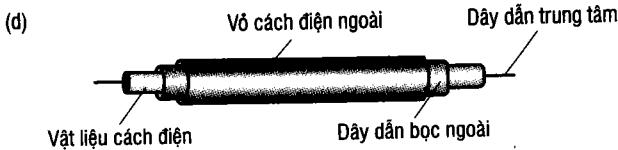
Việc truyền một tín hiệu điện đi xa cần đến một môi trường truyền mà thông thường sẽ dùng một đường dây truyền dẫn nào đó. Trong một số trường hợp sẽ là một cặp dây dẫn. Các trường hợp khác có thể là sóng ánh sáng được dẫn qua sợi thủy tinh hay sóng điện từ lan truyền qua không gian. Dạng của môi trường là rất quan trọng, vì chúng quyết định số bit tối đa có thể truyền trong một đơn vị thời gian _thường gọi là tốc độ bps (bits per second). Chúng ta sẽ khảo sát các môi trường thông dụng trong các phần sau đây.

1.1. Các đường truyền 2 dây không xoắn

Một đường truyền 2 dây không xoắn là môi trường truyền dẫn đơn giản nhất. Mỗi dây cách ly với dây kia và cả hai xuyên tự do (không xoắn nhau) qua môi trường không khí. Loại đường dây này thích hợp cho kết nối hai thiết bị cách xa nhau đến 50m dùng tốc độ bit nhỏ hơn 19,2kbps. Tín hiệu thường là mức điện thế hay cường độ dòng điện dựa vào tham chiếu điện thế đất (ground, không cân bằng) đặt lên một dây trong khi điện thế đất được đặt vào dây kia.



Hình 2.2 Môi trường truyền bằng dây dẫn đồng: (a) Đường hai dây không xoắn (b) Cáp xoắn từng cặp không được bảo vệ UTP (c) Cáp xoắn từng cặp có bảo vệ STP



Hình 2.2 (tiếp theo) Môi trường truyền bằng dây dẫn đồng:
(d) Cáp đồng trực.

Mặc dù một đường hai dây có thể được dùng để nối hai máy tính một cách trực tiếp, nhưng thường dùng nhất là cho kết nối một DTE đến một thiết bị kết cuối mạch dữ liệu cục bộ DCE (Data Circuit Terminating Equipment), ví dụ như Modem. Các kết nối như vậy thường dùng dây đa đường, cách tổ chức thông thường là cách ly riêng một dây cho mỗi tín hiệu và một dây nối đất (ground). Bộ dây hoàn chỉnh được bọc trong một cáp nhiều lõi được bảo vệ hay dưới dạng một hộp cáp, xem hình 2.2.

Với loại dây này cần phải cẩn thận tránh can nhiễu giữa các tín hiệu điện trong các dây dẫn kề nhau trong cùng một cáp. Hiện tượng này gọi là nhiễu xuyên âm. Ngoài ra cấu trúc không xoắn khiến chúng dễ bị thâm nhập bởi các tín hiệu nhiễu bắt nguồn từ các nguồn tín hiệu khác do bức xạ điện từ. Trở ngại chính đối với các tín hiệu truyền trên loại dây này là có thể chỉ một dây bị can nhiễu, ví dụ như dây tín hiệu, tạo thêm mức sai lệch tín hiệu giữa hai dây. Vì máy thu hoạt động trên cơ sở phân biệt mức chênh lệch điện thế giữa hai dây, nên điều này dẫn đến đọc sai tín hiệu gốc. Các yếu tố ảnh hưởng này đồng thời tạo ra giới hạn về cự ly cũng như tốc độ truyền.

1.2. Các đường dây xoắn đôi

Chúng ta có thể loại bỏ các tín hiệu nhiễu bằng cách dùng cáp xoắn đôi, trong đó một cặp dây xoắn lại với nhau. Sự xấp xỉ các đường dây tham chiếu đất và dây tín hiệu có ý nghĩa khi bất kỳ tín hiệu nhiễu nào thâm nhập thì sẽ vào cả hai dây, ảnh hưởng của chúng sẽ giảm đi bởi sự triệt lẫn nhau. Hơn nữa, nếu có nhiều cặp xoắn trong cùng một cáp thì sự xoắn của mỗi cặp trong cáp cũng làm giảm nhiễu xuyên âm. Hình 2.2b minh họa đường dây xoắn đôi.

Các đường dây xoắn đôi cùng với mạch phát và thu thích hợp lợi dụng các ưu điểm có được từ phương pháp hình học sẽ là đường truyền tốc độ xấp xỉ 1 Mbps qua cự ly ngắn (ngắn hơn 100m) và tốc độ thấp hơn qua cự ly dài hơn. Các mạch thu phát phức tạp cho phép tốc độ cao hơn qua cự ly dài hơn. Các đường dây này được gọi là cáp xoắn đôi không bảo vệ UTP (Unshielded Twisted Pair), được dùng rộng rãi trong mạng điện thoại và trong nhiều ứng dụng truyền số liệu. Đối với các cặp xoắn được bảo vệ STP (Shielded Twisted Pair), có dùng thêm một lưới bảo vệ để giảm hơn nữa ảnh hưởng của tín hiệu xuyên nhiễu (xem hình 2.2 c).

1.3. Cáp đồng trục

Các yếu tố giới hạn chính đối với cáp xoắn là khả năng và hiện tượng được gọi là 'hiệu ứng ngoài da'. Khi tốc độ bit truyền gia tăng, dòng điện chạy trên đường dây có khuynh hướng chỉ chạy trên bề mặt ngoài của dây dẫn, do đó dùng rất ít phần dây có sẵn. Điều này lại làm tăng trở kháng của đường dây đối với các tín hiệu có tần số cao, dẫn đến suy hao lớn đối với tín hiệu. Ngoài ra, với tần số cao thì năng lượng tín hiệu bị tiêu hao nhiều do ảnh hưởng bức xạ. Chính vì vậy trong các ứng dụng yêu cầu tốc độ bit cao hơn 1Mbps, chúng ta dùng các mạch thu phát phức tạp hơn hoặc dùng các môi trường truyền dẫn khác.

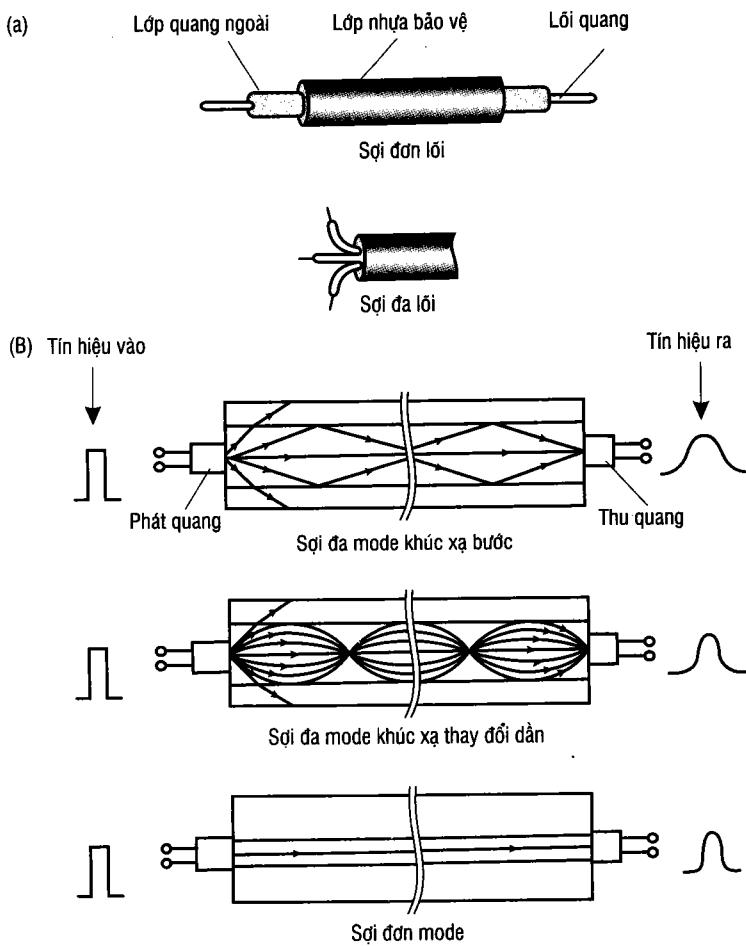
Cáp đồng trục tối thiểu được hai ảnh hưởng trên. Hình 2.2d mô tả các dây tín hiệu là một dây dẫn đặc (hay búi dây) và dây tham chiếu đất là dây dẫn viền quanh dây tín hiệu, cả hai được cách ly bởi một lớp cách điện. Lý tưởng nhất là không gian giữa hai dây dẫn phải là không khí, nhưng thực tế thường dùng một vật liệu cách điện có cấu tạo đặc hay dạng tổ ong.

Dây tín hiệu trung tâm được bảo vệ hiệu quả đối với các tín hiệu xuyên nhiễu từ ngoài nhờ lưới dây bao quanh bên ngoài. Chỉ suy hao lượng tối thiểu do bức xạ điện từ và hiệu ứng ngoài da do có lớp dây dẫn bao quanh. Cáp đồng trục có thể dùng với một số loại tín hiệu khác nhau, nhưng thông dụng nhất là dùng cho tốc độ 10Mbps trên cự ly vài trăm mét, nếu dùng điều chế tốt thì có thể đạt được thông số cao hơn.

1.4. Cáp quang

Mặc dù có nhiều cải tiến nhưng các loại cáp kim loại vẫn bị giới hạn về tốc độ truyền dẫn. Cáp quang khác xa với các loại cáp trước đây, cáp quang mang thông tin dưới dạng các chùm dao động của ánh sáng trong sợi thủy tinh. Sóng ánh sáng có băng thông rộng hơn sóng điện từ, điều này cho phép cáp quang đạt được tốc độ truyền khá cao lên đến hàng trăm Mbps. Sóng ánh sáng cũng "miễn dịch" đối với các nhiễu điện từ và nhiễu xuyên âm. Cáp quang cũng cực kỳ hữu dụng trong việc truyền các tín hiệu tốc độ thấp trong môi trường xuyên nhiễu nặng ví dụ như điện cao thế, chuyển mạch. Ngoài ra còn dùng trong các nơi có nhu cầu bảo mật, vì rất khó mắc xen rẽ (câu trộm) về mặt vật lý.

Một cáp quang bao gồm một sợi thủy tinh cho mỗi tín hiệu được truyền, được bọc bởi một lớp phủ bảo vệ ngăn ngừa bất kỳ nguồn sáng nào từ bên ngoài. Tín hiệu ánh sáng phát ra bởi một bộ phát quang, thiết bị này thực hiện chuyển đổi các tín hiệu điện thông thường từ một đầu cuối dữ liệu thành tín hiệu quang. Một bộ thu quang được dùng để chuyển ngược lại (từ quang sang điện) tại máy thu. Thông thường bộ phát quang là diode phát quang hay laser thực hiện chuyển đổi tín hiệu điện thành tín hiệu quang. Các bộ thu dùng các photodiode cảm quang hay photo transistor.



Hình 2.3 Nguyên lý cáp sợi quang: (a) Cấu trúc cáp; (b) Các chế độ truyền dẫn.

Bản thân sợi quang gồm hai phần: lõi thủy tinh và lớp phủ thủy tinh có hệ số khúc xạ thấp, xem hình 2.3(a). Ánh sáng lan truyền dọc theo lõi thủy tinh theo một trong ba cách phụ thuộc loại và bề rộng của vật liệu lõi được dùng. Ba chế độ truyền được mô tả trên hình 2.3(b). Trong chế độ đa mode khúc xạ bước _ *multimode stepped index* _vật liệu phủ và lõi khác nhau nhưng hệ số khúc xạ ổn định không thay đổi. Tất cả các ánh sáng phát ra bởi diode có góc phát nhỏ hơn góc tới hạn được phản xạ tại giao tiếp giữa lớp phủ và lõi và lan truyền trong lõi. Tùy vào góc phát mà ánh sáng sẽ mất một lượng thời gian để lan truyền dọc theo dây. Do đó tín hiệu nhận được có bề rộng xung rộng hơn xung gốc.



Sự phân tán có thể được hạn chế bằng cách dùng vật liệu lõi có hệ số khúc xạ thay đổi hay đa mode khúc xạ tăng dần *multimode graded index*, ánh sáng bị khúc xạ một lượng lớn khi di chuyển ra xa lõi. Điều này làm hẹp bề rộng xung của tín hiệu nhận, nhờ đó cho phép gia tăng tốc độ bit.

Một cải tiến cao hơn có thể đạt được bằng cách giảm đường kính lõi đến chiều dài bước sóng đơn ($3-10\mu\text{m}$) để tất cả các ánh sáng phát ra sẽ truyền theo một hướng dọc ống dẫn sóng (sợi quang cũng thường được gọi là ống dẫn sóng), và sợi quang dùng phương pháp này gọi là sợi đơn mode *monomode fiber*, nhờ vậy bề rộng xung nhận được sẽ xấp xỉ bề rộng xung gốc, nhờ đó tăng được tốc độ truyền.

1.5. Đường truyền vệ tinh

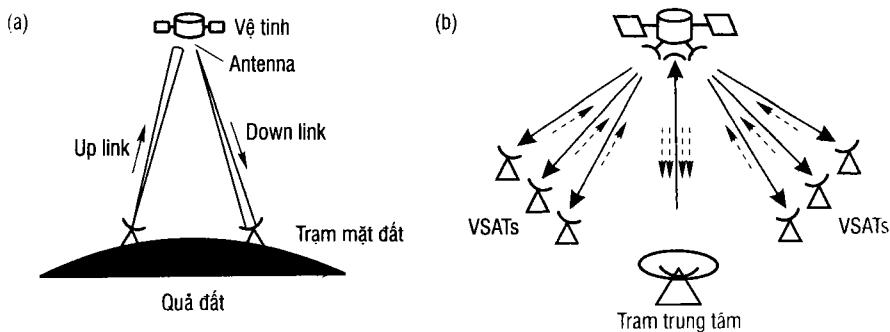
Tất cả các môi trường truyền được thảo luận ở trên đều dùng một đường dây vật lý để mang thông tin truyền. Số liệu cũng có thể được truyền bằng cách dùng sóng điện từ qua không gian tự do như trong các hệ thống thông tin vệ tinh. Một chùm sóng vi ba trực xạ trên đó mang số liệu đã được điều chế, được truyền đến vệ tinh từ trạm mặt đất. Chùm sóng này được thu và được truyền lại đến các đích xác định trước nhờ một mạch tích hợp thường được gọi là transponder. Một vệ tinh có nhiều transponder, mỗi transponder đảm trách một băng tần đặc biệt. Mỗi kênh vệ tinh thông thường đều có một băng thông cực cao (500MHz) và có thể cung cấp cho hàng trăm liên kết tốc độ cao thông qua kỹ thuật ghép kênh.

Các vệ tinh dùng cho mục đích liên lạc thường thuộc dạng địa tĩnh, có nghĩa là vệ tinh bay hết quí đạo quanh trái đất mỗi 24 giờ nhằm đồng bộ với sự quay quanh mình của trái đất và do đó vị trí của vệ tinh là đứng yên so với mặt đất. Quí đạo của vệ tinh được chọn sao cho đường truyền thẳng với trạm thu phát ở mặt đất, mức độ chuẩn trực của chùm sóng truyền lại từ vệ tinh có thể không cao để tín hiệu có thể được tiếp nhận trên một vùng rộng lớn, hoặc có thể hội tụ tốt để chỉ thu được trên một vùng giới hạn. Trong trường hợp thứ hai tín hiệu có năng lượng lớn cho phép dùng các bộ thu có đường kính nhỏ hơn thường gọi là chảo parabol, là các đầu cuối có độ mở rất nhỏ hay VSAT (Very Small Aperture Terminal). Các vệ tinh được dùng rộng rãi trong các ứng dụng truyền số liệu từ liên kết các mạng máy tính của quốc gia khác nhau cho đến cung cấp các đường truyền tốc độ cao cho các liên kết truyền tin giữa các mạng trong cùng một quốc gia.

Một hệ thống thông tin vệ tinh thông thường được trình bày trên hình 2.4, chỉ trình bày một đường dẫn đơn hướng nhưng là đường song công được sử dụng trong hầu hết các ứng dụng thực tế với các kênh đường lên (up link) và kênh đường xuống (down link) liên kết với mỗi trạm mặt đất hoạt động với tần số khác nhau. Các cấu hình thông dụng khác có liên quan đến trạm mặt đất trung tâm, trạm này liên lạc với một số trạm VSAT phân bố trên phạm vi quốc gia. Dạng tiêu biểu có một máy tính nối đến mỗi trạm VSAT và có thể truyền số liệu với máy tính trung tâm được nối đến trạm trung tâm như trên hình 2.4(b). Thông thường,

điểm trung tâm truyền rộng rãi đến tất cả các VSAT trên một tần số nào đó, trong khi ở hướng ngược lại mỗi VSAT truyền đến trung tâm bằng tần số khác nhau.

Để truyền thông tin với một VSAT nào đó, trung tâm sẽ quảng bá thông điệp với danh định của VSAT đặt tại đầu của thông điệp. Trong hoạt động thông tin giữa VSAT với VSAT, tất cả các thông điệp đều phải gửi đến trung tâm thông qua vệ tinh, tại đây sẽ quảng bá chúng đến các đối tác tham gia. Với thế hệ vệ tinh kế tiếp mạnh hơn, sẽ có khả năng định tuyến trên vệ tinh mà không cần đến trạm trung tâm. Và còn có thể liên lạc trực tiếp giữa các VSAT.



Hình 2.4 Truyền dẫn vệ tinh: (a) Điểm nối điểm (b) Đa điểm.

1.6. Đường truyền vi ba

Các liên kết vi ba mặt đất được dùng rộng rãi để thực hiện các liên kết thông tin khi không thể hay quá đắt tiền để thực hiện một môi trường truyền vật lý, ví dụ khi vượt sông, sa mạc, dồi núi hiểm trở v.v. Khi chùm sóng vi ba trực xạ đi xuyên ngang môi trường khí quyển, nó có thể bị nhiễu bởi nhiều yếu tố như địa hình và các điều kiện thời tiết bất lợi. Trong khi đối với một liên kết vệ tinh thì chùm sóng đi qua khoảng không gian tự do hơn nên ánh hưởng của các yếu tố này ít hơn. Tuy nhiên, liên lạc vi ba trực xạ xuyên môi trường khí quyển có thể dùng một cách tin cậy cho cự ly truyền dài hơn 50km.

1.7. Đường truyền vô tuyến tần số thấp

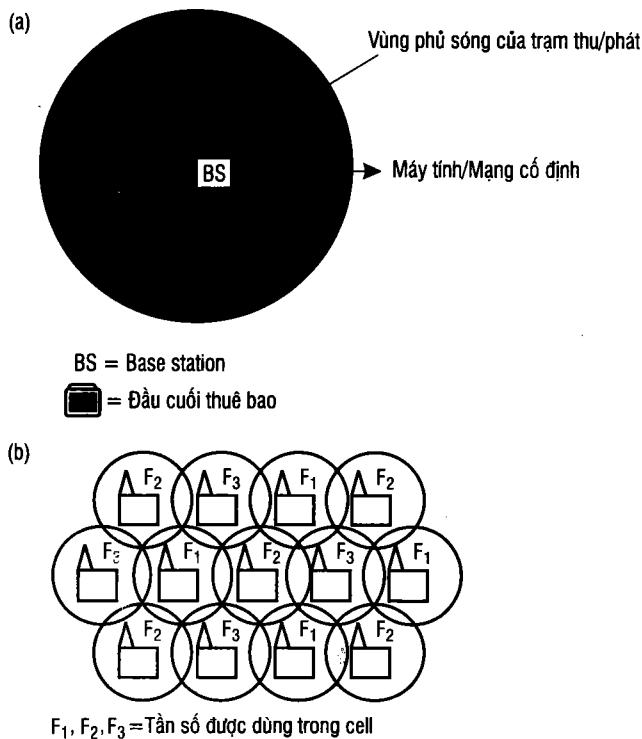
Sóng vô tuyến tần số thấp cũng được dùng để thay thế các liên kết hữu tuyến có cự ly vừa phải thông qua các bộ thu phát khu vực. Ví dụ kết nối một số lớn các máy tính thu thập số liệu bố trí trong một vùng đến một máy tính giám sát số liệu từ xa, hay kết nối các máy tính trong một thành phố đến máy cục bộ hay ở xa.

Có lẽ rất tốn kém khi lắp đặt các cáp dẫn hữu tuyến cho các ứng dụng như vậy. Sóng vô tuyến thường được dùng để thực hiện các liên kết không dây giữa một điểm kết cuối hữu tuyến và các máy tính phân tán. Một trạm phát vô tuyến được gọi là trạm cơ bản (base station) được đặt tại điểm kết cuối hữu tuyến, như

trên hình 2.5(a), cung cấp một liên kết không dây giữa mỗi máy tính và trung tâm.

Cần nhiều trạm cơ bản cho các ứng dụng yêu cầu phạm vi rộng và mật độ phân bố user cao. Phạm vi bao phủ của mỗi trạm cơ bản là giới hạn, do sự giới hạn nguồn phát của nó, nó chỉ đủ kênh để hỗ trợ cho toàn bộ tài trong phạm vi đó. Phạm vi rộng hơn có thể được thực hiện bằng cách tổ chức đa trạm theo cấu trúc tế bào (cell), xem hình 2.5(b). Trong thực tế, kích thước của mỗi tế bào thay đổi và được xác định bởi các yếu tố như mật độ đầu cuối và địa hình cục bộ.

Mỗi trạm cơ bản dùng một dải tần khác với trạm kế. Tuy nhiên, vì vùng phủ của mỗi trạm có giới hạn nên có thể dùng lại băng tần của nó cho các phần khác của mạng. Các trạm cơ bản được kết nối thành mạng hữu tuyến. Thông thường, tốc độ số liệu của mỗi máy tính trong một tế bào (cell) đạt được vài chục kbps.



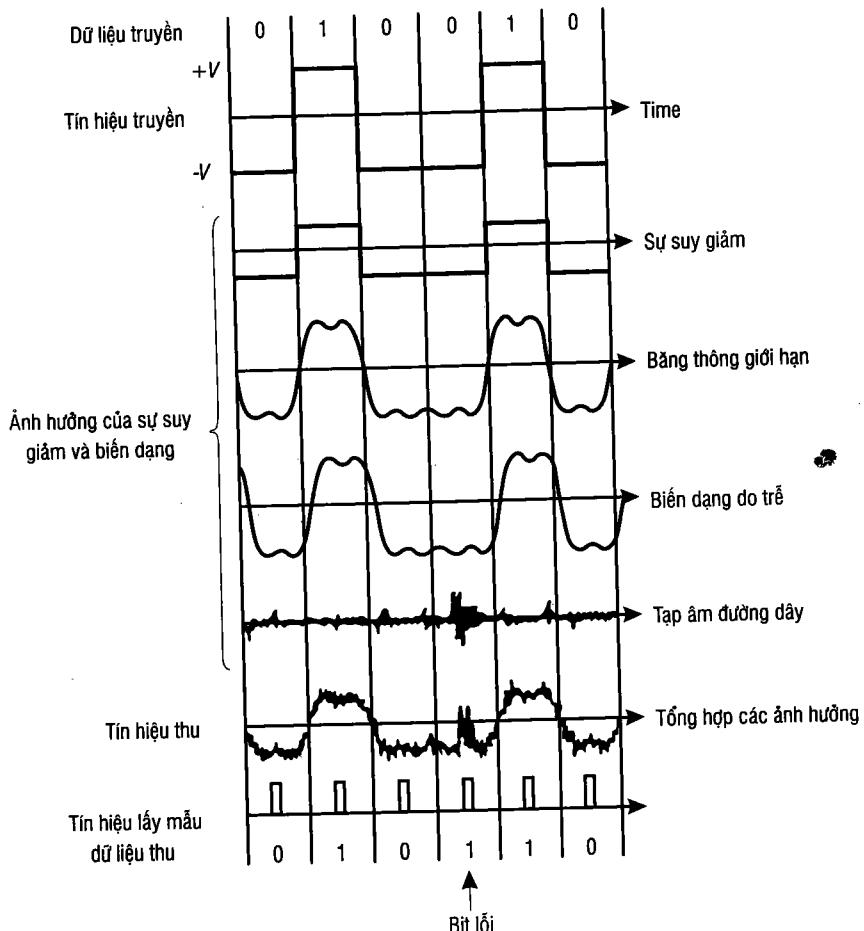
Hình 2.5 Truyền dẫn vô tuyến theo khu vực: (a) Một tế bào (b) Đa tế bào.

Dạng tổ chức tương tự có thể được dùng trong một tòa cao ốc để cung cấp các liên kết không dây cho thiết bị máy tính trong mỗi phòng. Trong các trường hợp như vậy, một hay nhiều trạm cơ bản sẽ tọa lạc trên mỗi tầng nhà và kết nối đến

mạng hữu tuyến. Mỗi trạm cơ bản cung cấp các liên kết không dây đến mạng hữu tuyến cho tất cả các máy tính thuộc phạm vi của nó. Nhờ vậy sẽ không còn phải bận tâm với việc chạy dây khi một máy tính được lắp đặt mới hay bỏ đi, nhưng cần phải cung cấp một đơn vị vô tuyến để chuyển số liệu sang dạng tín hiệu vô tuyến và ngược lại. Tốc độ truyền thường thấp hơn đường hữu tuyến.

2. SỰ SUY GIẢM VÀ BIẾN DẠNG TÍN HIỆU

Ảnh hưởng của suy giảm và biến dạng nói chung có thể làm thoái hóa một tín hiệu trong quá trình truyền. Hình 2.6 minh họa đơn giản nguyên nhân của quá trình thoái hóa tín hiệu dẫn đến sai dữ liệu ở máy thu.



Hình 2.6 Các nguyên nhân gây suy giảm và biến dạng tín hiệu

2.1. Sự suy giảm

Khi một tín hiệu lan truyền dọc dây dẫn vì lý do nào đó biên độ của nó giảm xuống được gọi là sự suy giảm tín hiệu. Thông thường mức độ suy giảm cho phép được qui định trên chiều dài cáp dẫn để đảm bảo rằng hệ thống nhận có thể phát hiện và dịch được tín hiệu ở máy thu. Nếu trường hợp cáp quá dài thì có một hay nhiều bộ khuếch đại (hay còn gọi là repeater) được chèn vào từng khoảng dọc theo cáp nhằm tiếp nhận và tái sinh tín hiệu.

Sự suy giảm tín hiệu tăng theo một hàm của tần số, trong khi đó tín hiệu lại bao gồm một dải tần vì vậy tín hiệu sẽ bị biến dạng do các thành phần suy giảm không bằng nhau. Để khắc phục vấn đề này, các bộ khuếch đại được thiết kế sao cho khuếch đại các tín hiệu có tần số khác nhau với hệ số khuếch đại khác nhau. Ngoài ra còn có thiết bị cân chỉnh gọi là **equalizer** được dùng để cân bằng sự suy giảm xuyên qua một băng tần được xác định.

Sự suy giảm và sự khuếch đại (độ lợi) được đánh giá và đo lường bằng đơn vị decibels (dB). Nếu gọi mức năng lượng của tín hiệu được truyền là P1 và mức năng lượng nhận là P2 thì:

$$\text{Sự suy giảm} = 10 \log_{10} P1/P2 (\text{dB})$$

và

$$\text{Sự khuếch đại} = 10 \log_{10} P2/P1 (\text{dB})$$

Vì cả P1 và P2 có cùng đơn vị là watts nên decibels là không thứ nguyên và đơn giản là đo lường độ lớn giữa hai mức năng lượng.

Ví dụ:

Một kênh truyền giữa hai DTE được thiết lập từ 3 phần. Phần thứ nhất có suy giảm 16dB, phần thứ 2 khuếch đại 20dB và phần 3 suy giảm 10dB. Giả sử mức năng lượng có ý nghĩa được truyền là 400mW. Hãy xác định mức năng lượng ở đầu ra của kênh.

Ta có :

$$\text{Phần 1: } 16 = 10 \log_{10} 400/P2 \Rightarrow P2 = 10,0475 \text{ mW.}$$

$$\text{Phần 2: } 20 = 10 \log_{10} P2/10,0475 \Rightarrow P2 = 1004,75 \text{ mW.}$$

$$\text{Phần 3: } 10 = 10 \log_{10} 1004,75/P2 \Rightarrow P2 = 100,475 \text{ mW.}$$

Vậy mức năng lượng ở đầu ra của kênh là 100,475mW

hoặc

$$\text{Tổng suy giảm} = (16+20) + 10 = 6 \text{ dB.}$$

$$\text{Do đó : } 6 = 10 \log_{10} 400/P2 \Rightarrow P2 = 100,475 \text{ mW.}$$

Khi truyền thông tin nhị phân qua kênh có băng thông giới hạn ví dụ như qua mạng điện thoại công cộng PSTN, chúng ta có thể dùng nhiều hơn hai trạng

thái tín hiệu. Điều này có nghĩa là mỗi trạng thái tín hiệu có thể đại diện cho nhiều ký số nhị phân. Tổng quát, nếu số trạng thái tín hiệu là M thì số bit trên một phần tử tín hiệu là m , và quan hệ giữa chúng được biểu diễn thông qua biểu thức:

$$m = \log_2 M$$

Ví dụ:

Nếu có 4 trạng thái tín hiệu được dùng để truyền thì mỗi trạng thái có thể được dùng để truyền cho 2 digit nhị phân.

Tốc độ thay đổi trạng thái của tín hiệu được xem như là tốc độ phát tín R_s (signaling rate) và được đo lường bằng đơn vị **baud**. Nó liên quan tới tốc độ truyền bit số liệu R qua đẳng thức :

$$R = R_s \log_2 M$$

2.2. Băng thông bị giới hạn

Bất kỳ một kênh hay đường truyền nào: cáp xoắn, cáp đồng trực, radio... đều có một băng thông xác định liên hệ với nó, băng thông chỉ ra các thành phần tần số nào của tín hiệu sẽ được truyền qua kênh mà không bị suy giảm. Do đó khi truyền dữ liệu qua một kênh cần phải đánh giá ảnh hưởng của băng thông của kênh đối với tín hiệu số liệu được truyền.

Thông thường phải dùng phương pháp toán học để đánh giá, công cụ thường được dùng nhất là phương pháp phân tích **Fourier**. Phân tích Fourier cho rằng bất kỳ tín hiệu tuần hoàn nào đều được hình thành từ một dãy xác định các thành phần tần số riêng biệt. Chu kỳ của tín hiệu xác định thành phần tần số cơ bản. Các thành phần tần số khác có tần số là bội số của tần số cơ bản gọi là các hài bậc cao (harmonics) của tần số cơ bản. Biểu diễn toán học của dạng sóng tuần hoàn theo Fourier như sau:

$$V(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos \omega_n t + \sum_{n=1}^{\infty} b_n \sin n \omega_n t$$

Trong đó:

$V(t)$ là tín hiệu điện áp, là hàm số theo thời gian t

ω_0 là tần số cơ bản đơn vị là radian/s.

$T = 2\pi/\omega_0$ là chu kỳ của dạng sóng tính bằng giây.

a_0, a_n và b_n là các hệ số Fourier được xác định từ 3 tích phân sau:

$$a_0 = 1/T \int_0^T V(t) dt$$

$$a_n = 2/T \int_0^T V(t) \cos(n\omega_0 t) dt$$

$$b_n = 2/T \int_0^T V(t) \sin(n\omega_0 t) dt$$

Từ tích phân đầu tiên có thể suy ra a_0 chính là thành phần một chiều DC trong tín hiệu.

Trong hệ thống số liệu, các tín hiệu truyền là các bit nhị phân tuần tự kế tiếp nhau. Trong thực tế sự xuất hiện các bit 0 và 1 là ngẫu nhiên. Để phân tích ta sẽ chọn trường hợp xấu nhất. Để dễ dàng nhận thấy chuỗi bit có tuần tự 101010 là dãy tín hiệu có chu kỳ ngắn nhất, chính vì thế sẽ có thành phần tần số cơ bản lớn nhất. Đây được xem là trường hợp xấu nhất và sẽ dùng để phân tích bằng thông.

Có hai dạng tín hiệu nhị phân cơ bản: đơn cực (unipolar) và lưỡng cực (bipolar) tham khảo hình 2.7(a). Với tín hiệu đơn cực, biên độ thay đổi giữa điện thế dương $+V$ và mức 0. Thường gọi là tín hiệu quay về zero (RZ). Với tín hiệu lưỡng cực biên độ điện áp thay đổi $+V$ và $-V$, các tín hiệu này gọi là không quay về zero (NRZ). Với tín hiệu đơn cực thì mức tín hiệu trung bình là $V/2$ trong khi tín hiệu lưỡng cực thì mức tín hiệu trung bình là không (0). Chuỗi Fourier cho hai tín hiệu này là:

Đơn cực :

$$V(t) = V/2 + 2V/\pi (\cos \omega_0 t - 1/3 \cos 3\omega_0 t + 1/5 \cos 5\omega_0 t - \dots)$$

Lưỡng cực:

$$V(t) = 4V/\pi (\cos \omega_0 t - 1/3 \cos 3\omega_0 t + 1/5 \cos 5\omega_0 t - \dots)$$

Từ biểu thức trên chúng ta có thể suy ra : bất cứ chuỗi bit nhị phân tuần hoàn nào đều được tạo nên từ dãy xác định các tín hiệu riêng biệt bao gồm thành phần tần số cơ bản f_0 , thành phần hài bậc 3, $3f_0$, thành phần hài bậc 5, $5f_0$.v.v. Lưu ý rằng chỉ có các hài bậc lẻ tồn tại và biên độ sẽ yếu dần theo chiều gia tăng tần số. Tham khảo hình 2.7(b).

Vì các kênh thông tin có băng thông bị giới hạn, nên khi tín hiệu nhị phân truyền qua kênh, chỉ những thành phần tần số trong dải thông sẽ được nhận bởi máy thu. Ảnh hưởng này trên dãy tín hiệu 101010... được mô tả bằng lược đồ hình thức ở hình 2.7(c).

Khi băng thông của kênh được đo lường bằng đơn vị Hertz thường được biểu diễn dưới dạng hàm của tần số và được mô tả như hình 2.7(d).

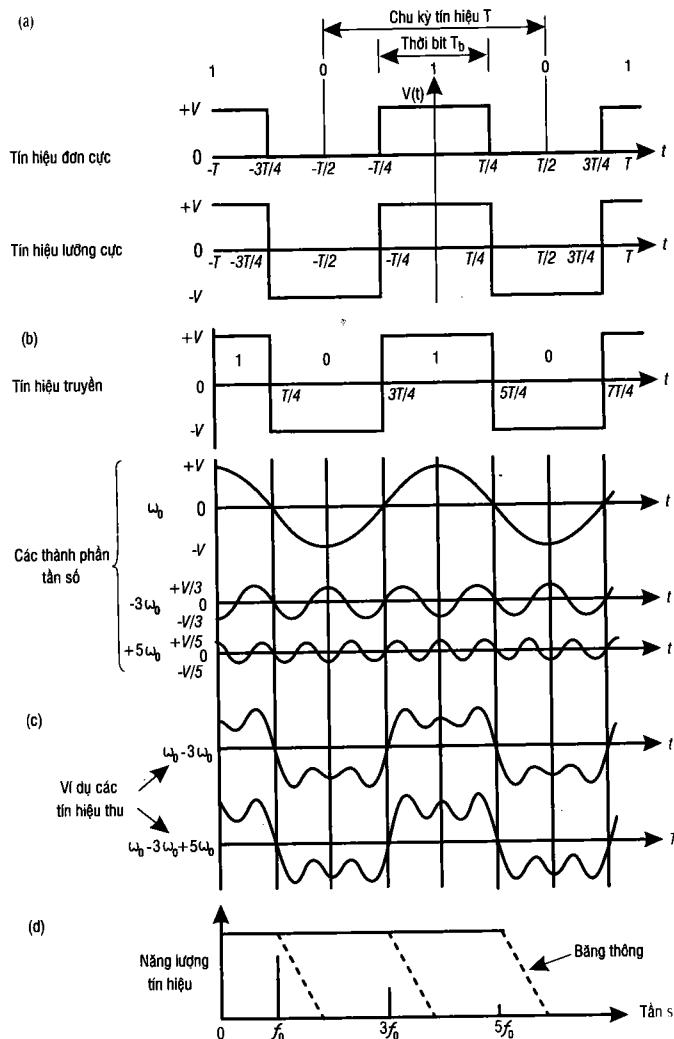
Ví dụ:

Một tín hiệu nhị phân được truyền với tốc độ 500bps qua kênh thông tin. Suy ra băng thông tối thiểu theo yêu cầu, giả sử:

- a/. Chỉ có tần số cơ bản.
- b/ Tần số cơ bản và hài bậc 3.
- c/ Tần số cơ bản, hài bậc 3 , và bậc 5.

Xét trường hợp xấu nhất.

Tóm tắt lời giải:



Hình 2.7 Ảnh hưởng do giới hạn của băng thông: (a) Tín hiệu nhị phân luân phiên (b) Các thành phần tần số của tuần tự nhị phân tuần hoàn (c) Một ví dụ về tín hiệu thu (d) Biểu diễn băng thông.

Trường hợp xấu nhất của tín hiệu là dạng 101010... tại 500bps sẽ có tần số cơ bản là 250Hz, do đó hài bậc 3 là 750Hz và hài bậc 5 là 1250Hz. Vậy băng thông yêu cầu trong mỗi trường hợp là:

- a/0 - 250 Hz
- b/ 0 - 750Hz
- c/ 0 - 1250 Hz

Như đã nói ở phần trước, có thể truyền nhiều hơn một bit với mỗi lần thay đổi trạng thái của tín hiệu nhờ đó có thể gia tăng tốc độ bit. Tuy nhiên băng thông của kênh luôn luôn là yếu tố hạn chế tốc độ tối đa có thể đạt được. Tiếp tục xem xét ta có công thức Nyquist xác định tốc độ tối đa của kênh không nhiễu được trình bày như sau:

$$C = 2 W \log_2 M$$

Trong đó W là băng thông của kênh tính bằng Hz và M là số trạng thái tín hiệu. Trong thực tế vì các bit mở rộng được thêm vào cho mục đích điều khiển truyền (overhead) nên tốc độ số liệu hiệu dụng thường nhỏ hơn tốc độ bit thực tế. Do đó, khi truyền thông tin qua kênh chúng ta có 3 tốc độ được liên hệ : tốc độ phát tín hiệu, tốc độ bit và tốc độ số liệu. Tất cả chúng có thể giống hay khác nhau.

Gọi thời gian của mỗi bit là T_b ta có sự tương ứng với tốc độ bit R như sau:

R liên hệ với chu kỳ tín hiệu T_s cho bởi :

$$R = (\log_2 M) / T_s = m/T_s \text{ (bps)}$$

Vì T_b có quan hệ hỗ trợ với R , do đó:

$$T_b = 1/R = T_s/m.$$

Kết hợp hai biểu thức trên suy ra đại lượng B của kênh truyền:

$$B = R / W = m / WT_s = 1 / WT_b \text{ (bps Hz}^{-1}\text{)}$$

B là hiệu suất băng thông của kênh.

Từ biểu thức trên ta suy ra tốc độ bit càng cao, ảnh hưởng của băng thông càng lớn. Giá trị tiêu biểu của B từ $0,25$ đến 3 bpsHz^{-1}

Ví dụ:

Dữ liệu được truyền qua mạng PSTN dùng lược đồ truyền với 8 trạng thái tín hiệu. Nếu băng thông của PSTN là 3000Hz. Hãy suy ra tốc độ Nyquist C và hiệu suất băng thông B .

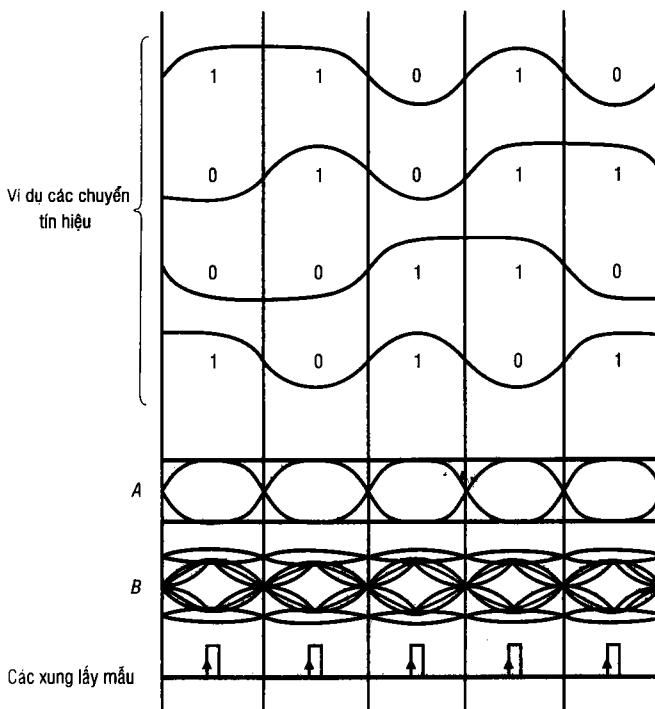
$$\begin{aligned} C &= 2 W \log_2 M \\ &= 2 \times 3000 \times \log_2 8 \\ &= 18.000 \text{ bps} \\ B &= 1 / WT_b \\ &= 18.000 / 3.000 = 6 \text{ bps Hz}^{-1} \end{aligned}$$

Trong thực tế cả hai giá trị sẽ nhỏ hơn do ảnh hưởng của một số nguyên nhân khác ví dụ như nhiễu.

2.3. Sự biến dạng do trễ pha

Tốc độ lan truyền của một tín hiệu thuần nhất dọc theo một đường truyền thay đổi tùy tần số. Do đó khi truyền một tín hiệu số, các thành phần tần số khác nhau tạo nên nó sẽ đến máy thu với độ trễ pha khác nhau, dẫn đến biến dạng do trễ của tín hiệu tại máy thu. Sự biến dạng sẽ gia tăng khi tốc độ bit tăng. Biến dạng trễ làm thay đổi các thời khắc của tín hiệu gây khó khăn trong việc lấy mẫu tín hiệu.

Có thể quan sát mức can nhiễu liên quan đến kênh truyền qua lược đồ mắt. Hình 2.8 là một ví dụ, lược đồ thu từ màn hình một oscilloscope.



Hình 2.8 **Ví dụ lược đồ mắt: A lý tưởng; B thông thường**

2.4. Sự can nhiễu (tạp âm)

Khi không có tín hiệu, một đường truyền dẫn hay kênh truyền được xem là lý tưởng nếu mức điện thế trên đó là zero. Trong thực tế có những tác động ngẫu nhiên làm cho tín hiệu điện trên đường truyền vẫn khác zero, cho dù không có tín hiệu số liệu nào được truyền trên đó. Mức tín hiệu này được gọi là mức nhiễu đường dây. Khi một tín hiệu bị suy giảm thì biên độ của nó giảm đến mức nhiễu

đường (line noise). Tỉ số năng lượng trung bình của một tín hiệu thu được S so với năng lượng của mức nhiễu đường N được gọi là tỉ số tín hiệu trên nhiễu SNR (Signal_ to _ Noise Ratio), đây là tham số quan trọng liên quan đến đường truyền. Thông thường, SNR được biểu diễn qua đơn vị decibel (dB)

$$\text{SNR} = 10 \log_{10} (\text{S}/\text{N}) \text{ (dB)}$$

Rõ ràng nếu tỉ số SNR cao thì chất lượng tín hiệu thu sẽ cao. Ngược lại nếu SNR thấp có nghĩa là chất lượng tín hiệu thu thấp.

Tốc độ truyền dẫn tối đa theo lý thuyết của kênh truyền liên hệ chặt chẽ với tỉ số SNR và được xác định theo luật Shannon - Harley:

$$C = W \log_2 (1 + \text{S}/\text{N}) \text{ (bps)}$$

Trong đó C là tốc độ tính bằng bps, W là băng thông của đường dây hay kênh truyền tính bằng Hz, S và N tính bằng Watts.

Ví dụ:

Giả sử một PSTN có băng thông 3000Hz và tỉ số SNR là 20dB. Xác định tốc độ tối đa của thông tin truyền theo lý thuyết.

$$\text{SNR} = 10 \log_{10} (\text{S}/\text{N})$$

Do đó :

$$20 = 10 \log_{10} (\text{S}/\text{N})$$

$$\text{S}/\text{N} = 100$$

Từ đây :

$$C = W \log_2 (1 + \text{S}/\text{N})$$

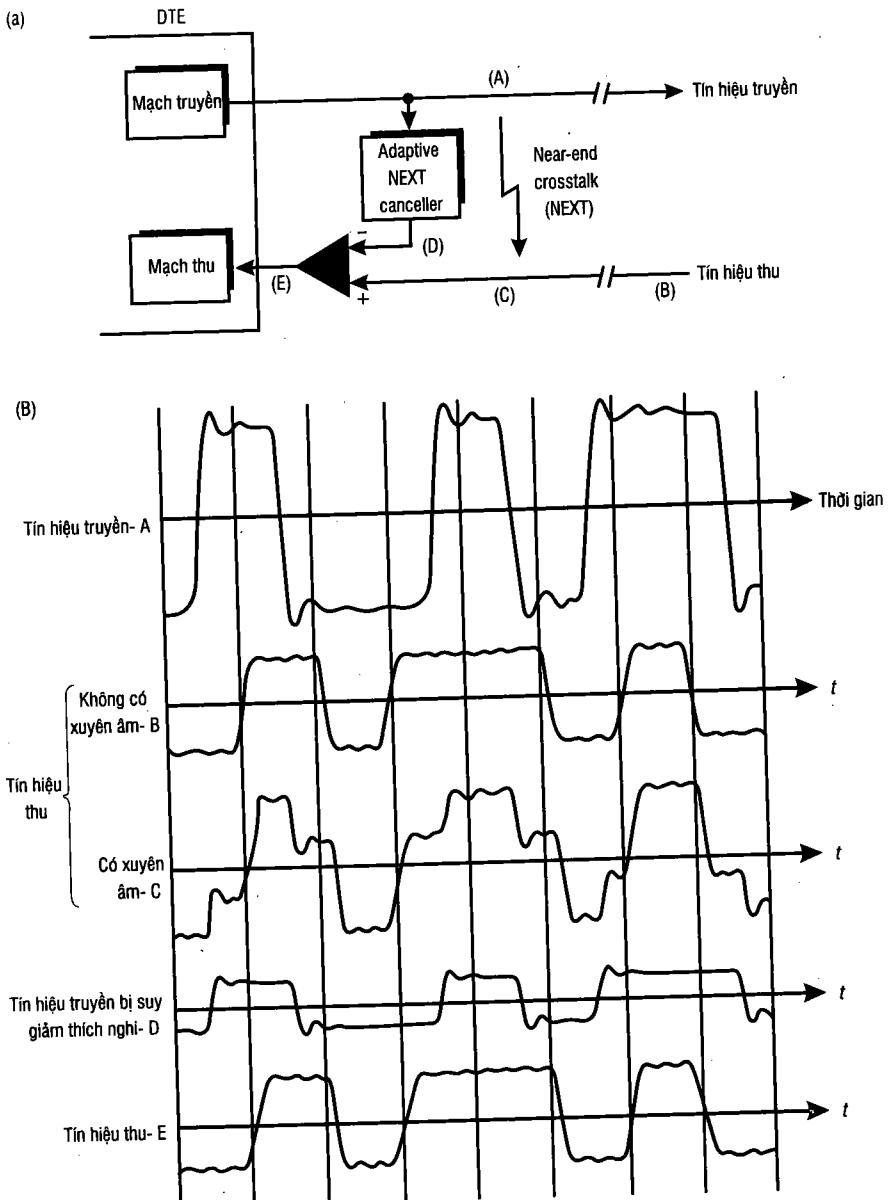
Vậy:

$$C = 3000 \log_2 (1 + 100) = 19963 \text{ bps}$$

Một nguyên nhân gây nhiễu được biết đến gọi là nhiễu xuyên âm (crosstalk). Nhiều hình thành do hai dây dẫn đặt kề nhau. Tín hiệu truyền trên dây này trở thành tín hiệu nhiễu vào dây kia. Ví dụ khi ta đang nói chuyện điện thoại có thể nghe tiếng của người khác xen vào ngay cả khi ta không đàm thoại với ai cả.

Có nhiều loại nhiễu xuyên âm nhưng trong hầu hết các trường hợp thì đáng quan tâm nhất là xuyên âm đầu cuối kề (near end crosstalk) hay còn gọi là **next**. Trường hợp này gọi là tự xuyên âm vì nó phát sinh bởi tín hiệu phát ở ngõ ra của mạch phát mạnh hơn so với tín hiệu thu ở ngõ vào mạch thu nội bộ, và tín hiệu phát sẽ gây nhiễu ở ngõ vào mạch thu.

Một mạch tích phân đặc biệt gọi là bộ triệt nhiễu thích nghi (adaptive next cancellers) được dùng để khắc phục vấn đề trên, xem hình 2.9.



Hình 2.9 Sử dụng bộ triệt nhiễu:
(a) Lược đồ mạch; (b) Ví dụ về dạng sóng.

Một dạng nhiễu khác là nhiễu xung điện, nguyên nhân gây ra bắt nguồn từ các tác nhân bên ngoài như nguồn điện năng, các thiết bị điện đang hoạt động. Cá

nhiều xuyên âm và nhiều xung điện đều phát sinh từ bên ngoài đường dây và tác động vào việc truyền trên dây.

Loại nhiễu thứ 3 được gọi là nhiễu nhiệt (therm noise), hiện diện trong tất cả các thiết bị điện và các phương tiện truyền bát chấp ảnh hưởng ngoại cảnh. Nó được sinh ra do tác động nhiệt của các electron liên kết với các nguyên tử của vật liệu điện chế tạo đường dây hay thiết bị. Ở mọi nhiệt độ trên mức tuyệt đối zero, tất cả các phương tiện truyền thông đều trải qua nhiễu nhiệt. Nó tạo nên các thành phần tần số ngẫu nhiên xuyên qua phổ tần đã hoàn chỉnh, có biên độ thay đổi liên tục và do đó còn gọi là nhiễu trắng (white noise).

Quay trở lại với luật Shannon - Harley. Khi xem xét ảnh hưởng của nhiễu trong thực tế, điều quan trọng là cần xác định cho được mức tín hiệu tối thiểu phải dùng có liên hệ đến mức nhiễu, nhằm đạt được tốc độ lỗi bit ở mức tối thiểu cho phép. Tốc độ lỗi bit qui định khoảng thời gian xác định một bit đơn có thể bị sai tại máy thu. Ví dụ tốc độ lỗi bit là 10^{-4} điều đó có nghĩa là trung bình 1 bit lỗi trong 10^4 bit nhận.

Goi E_b là năng lượng của 1 bit trong tín hiệu, đơn vị đó là joules (watts x seconds) - được tính theo công thức.

$$E_b = ST_b \text{ (watt-second)}$$

Trong đó S là năng lượng của tín hiệu (watts) và T_b là thời bit (second). Ta có:

$$E_b = S/R$$

Mức nhiễu trong chiều dài 1hz của băng thông (BW) trong bất kỳ đường truyền nào được cho bởi công thức:

$$N_o = kT \text{ (watt Hz}^{-1}\text{)}$$

N_o là mật độ năng lượng nhiễu, k là hằng số Boltzman bằng:

$1,3803 \times 10^{-23}$ jouleK $^{-1}$ và T là nhiệt độ Kelvin ($^{\circ}$ K)

Để đánh giá ảnh hưởng của nhiễu người ta dùng tỉ số :

$$E_b / N_o = (S/R) / N_o = (S/R) / kT.$$

Hay, băng decibels

$$E_b / N_o (\text{dB}) = 10 \log_{10} (S/R) - 10 \log_{10} (kT)$$

Có thể dễ dàng suy ra mức năng lượng của tín hiệu S để đạt được một tỉ lệ E_b/N_o có thể chấp nhận được và tốc độ bit lỗi tối thiểu. Hai đại lượng này sẽ tăng theo nhiệt độ và tốc độ bit.

Cũng có thể biểu diễn E_b/N_o theo băng thông W . Vì N_o là mật độ năng lượng nhiễu trên băng thông Watts Hz $^{-1}$, do đó năng lượng nhiễu trên băng thông là:

$$N = WN_o$$



Suy ra:

$$E_b / N_0 = S.W / N.R$$

Tính theo decibels

$$E_b / N_0 = 10 \log_{10} (S/N) + 10 \log_{10} W - 10 \log_{10} R$$

Để truyền số liệu thông qua mạng điện thoại công cộng cần phải chuyển dữ liệu số sang dạng tương tự (analog) bằng cách điều chế lên sóng mang hình sin, và phải thực hiện công việc ngược lại tại máy thu. Trong thực tế có nhiều phương pháp điều chế và giải điều chế (modulation/ demodulation), mỗi phương pháp cho phép tỉ số E_b/N_0 khác nhau.

Ví dụ:

Dữ liệu được truyền qua PSTN có băng thông 3000Hz. Nếu năng lượng nhiễu trung bình tại máy thu là 12dB, hãy xác định tốc độ dữ liệu tối đa có thể đạt được và hiệu suất băng thông trong hai trường hợp:

a/. $E_b/N_0 = 13\text{dB}$.

b/. $E_b/N_0 = 10\text{dB}$.

Tóm tắt giải:

$$10 \log_{10} R = S/N (\text{dB}) + 10 \log_{10} W - E_b / N_0 (\text{dB})$$

Hiệu suất băng thông $B = R/W$.

a/. $10 \log_{10} R = 12 + 10 \log_{10} 3000 - 13 = 33,77$

Suy ra : $R = 2382,32 \text{ bps}$ và $B = 0,79$

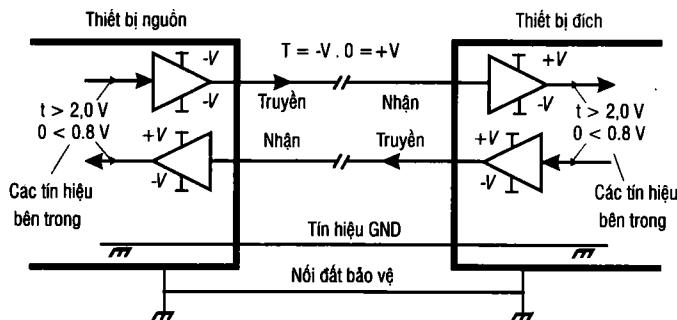
b/. $10 \log_{10} R = 12 + 34,77 - 10 = 36,77$

Suy ra : $R = 4753,35 \text{ bps}$ và $B = 1,58$

3. CÁC LOẠI TÍN HIỆU

Khi hai đầu cuối kết nối với nhau bằng tốc độ vừa phải, có thể truyền dữ liệu bằng các đôi dây không xoắn và các mạch giao tiếp đơn giản. Các mạch giao tiếp này thay đổi các mức tín hiệu được dùng bên trong thiết bị thành mức tín hiệu tương thích với cáp nối. Tuy nhiên, khi sự khác biệt giữa các đầu cuối và tốc độ bit gia tăng, cần phải dùng các kỹ thuật và mạch phức tạp hơn. Hơn nữa, nếu các đầu cuối nằm ở cách xa nhau trên phạm vi quốc gia (hay quốc tế) và không có các dịch vụ truyền số liệu công cộng, thì chỉ còn cách dùng các đường truyền được cung cấp bởi các nhà khai thác dịch vụ điện thoại và các dịch vụ viễn thông khác. Khi dùng môi trường này cần phải chuyển đổi các tín hiệu điện từ các DTE thành dạng tín hiệu analog mang các thông điệp đàm thoại. Tương tự, khi nhận cũng cần chuyển đổi trở về dạng tín hiệu phù hợp với dạng tín hiệu được dùng bởi DTE đích.

3.1. V.28

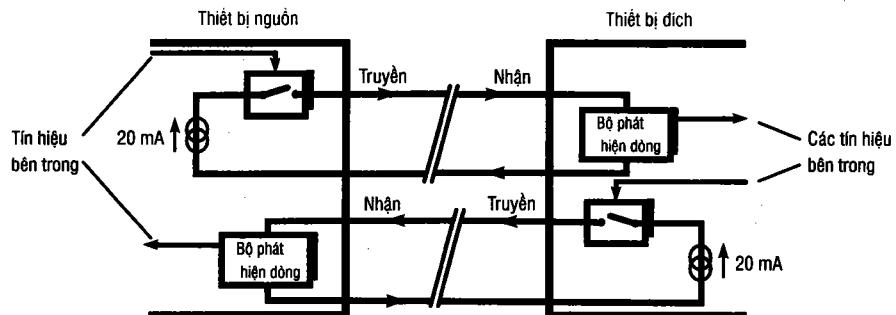


Hình 2.10 Các mức tín hiệu V.28

Các mức tín hiệu điện được qui định dùng cho một số giao tiếp EIA/ITU-T được chỉ ra trong khuyến nghị V.28. Các tín hiệu được trình bày trên hình 2.10. Chuẩn V.28 được xem là giao tiếp điện không cân bằng. Các tín hiệu điện áp được dùng trên đường dây là đối xứng so với mức tham chiếu gốc (ground) và ít nhất là mức 3vdc, +3vdc cho bit 0 và -3vdc cho bit 1. Trong thực tế nguồn cung cấp cho mạch giao tiếp có mức điện thế là ± 12 vdc hay ± 15 vdc, các mạch truyền cần chuyển mức tín hiệu điện áp thấp trong các thiết bị sang mức điện áp cao ngoài đường dây. Các mức tín hiệu được dùng ở đây cao hơn so với mức của TTL (2.0v-5.0 v là mức 1 và 0.2v-0.8v là mức 0) có tác dụng chống suy giảm và loại nhiễu tốt.

3.2. Dòng 20mA

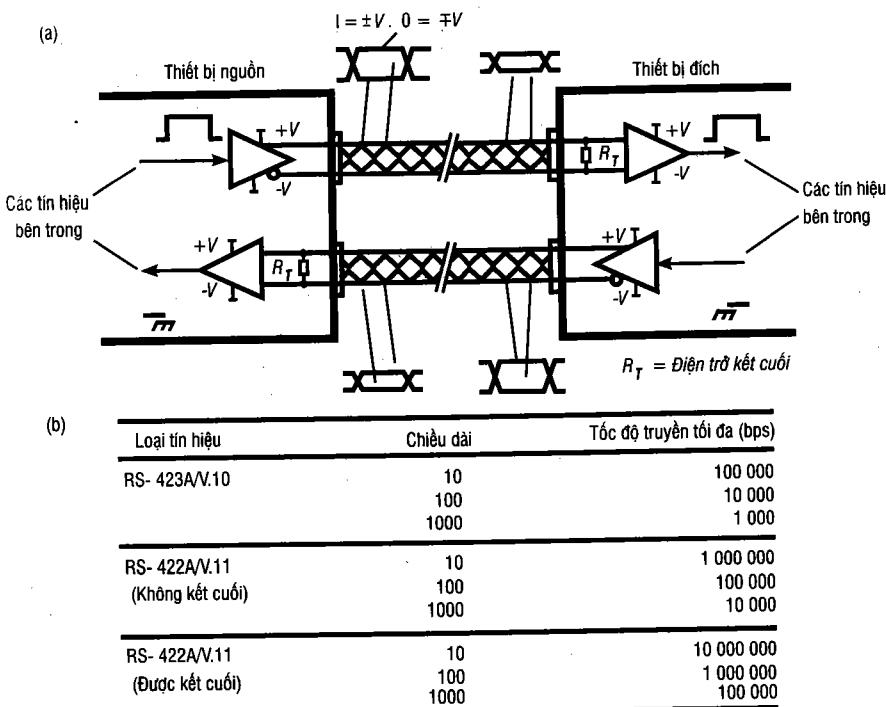
Một dạng tín hiệu khác có thể chọn bên cạnh EIA-232D/V.28 là giao tiếp vòng 20mA. Tên của giao tiếp này ngụ ý rằng dùng tín hiệu là dòng điện thay cho điện áp. Mặc dù không mở rộng tốc độ nhưng nó tăng khoảng cách vật lý giữa hai thiết bị thông tin. Tiếp cận cơ bản được trình bày trên hình 2.11. Hoạt động chính là trạng thái của chuyển mạch được điều khiển bởi luồng bit dữ liệu truyền: chuyển mạch đóng tương ứng với bit 1, do đó cho dòng 20mA qua, và ngược lại chuyển mạch mở cho bit 0, do đó không cho dòng 20mA qua. Tại đầu thu dòng điện được phát hiện bởi mạch cảm biến dòng và các tín hiệu nhị phân được tái tạo lại. Giao tiếp này loại bỏ nhiễu tốt hơn so với giao tiếp điều khiển bằng điện áp, phù hợp với đường dây dài (đến 1km), nhưng tốc độ vừa phải.



Hình 2.11 Vòng 20mA.

3.3. RS-422A/V.11

Nếu muốn tăng khoảng cách vật lý và tốc độ, chúng ta sẽ dùng RS-422A/V.11. Chuẩn này cơ bản dựa trên cáp xoắn đôi và mạch thu phát vi phân và được xem như giao tiếp điện cân bằng. Một sắp xếp tiêu biểu được trình bày trên hình 2.12. Một mạch phát vi phân tạo ra tín hiệu sinh đôi bằng nhau và ngược cực theo mỗi tín hiệu nhị phân 0 hay 1 khi được truyền. Tương tự, mạch thu chỉ cảm nhận sự khác nhau của hai tín hiệu trên hai ngõ nhập của chúng, nhờ đó nhiễu tác động đồng thời lên cả hai dây sẽ không ảnh hưởng đến tín hiệu cần thu. Một dẫn xuất của RS-422A/V.11 là RS-423/V10 có thể được dùng cho các ngõ ra điện áp không cân bằng bởi các giao tiếp EIA-232D/V.24 với một bộ thu vi phân. RS-422A/V.11 thích hợp trong trường hợp dùng cáp xoắn đôi truyền ở cự ly 10m với tốc độ 10Mbps và 1km với tốc độ 100kbps.



Hình 2.12 Các mức tín hiệu vi phân RS-422A/V.11

Một tham số quan trọng của bất kỳ đường truyền nào là đặc tính trở kháng Z_0 bởi vì một bộ thu chỉ hấp thụ hết các tín hiệu (năng lượng) đến chỉ khi đường dây được kết thúc bởi một điện trở R_t bằng với Z_0 _sự phối hợp trở kháng.

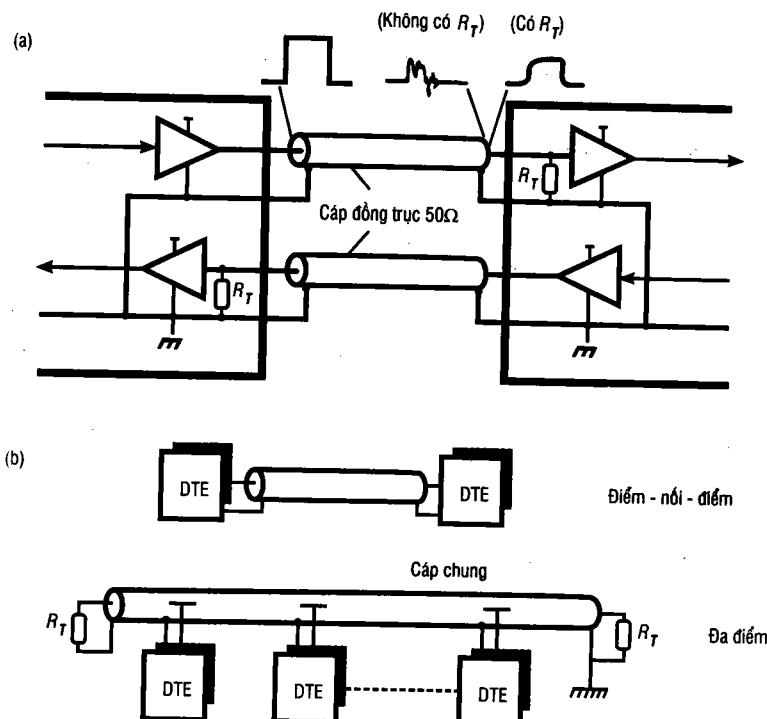
3.4. Các tín hiệu cáp đồng trục

Không như băng thông thấp sẵn có trong kết nối qua mạng chuyển mạch điện thoại, băng thông hữu dụng trong cáp đồng trục có thể lên đến 350MHz (hay cao hơn). Có thể dùng băng tần cao này theo một trong hai cách:

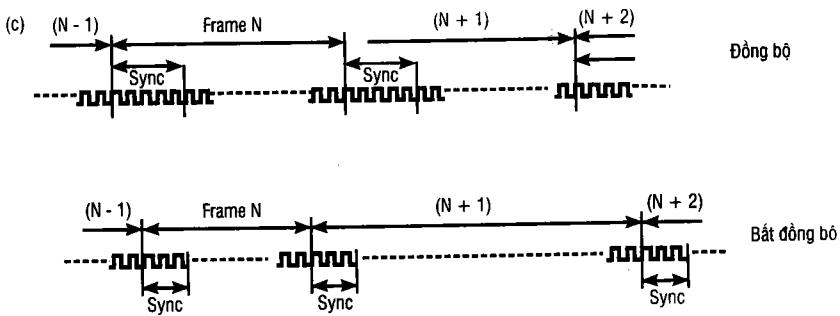
1. **Chế độ băng cơ bản:** trong đó tất cả băng thông có sẵn được dùng để tiếp nhận một kênh tốc độ cao (10Mbps hay cao hơn).
2. **Chế độ băng rộng:** trong đó băng thông có sẵn được chia thành một số các kênh có tốc độ nhỏ hơn trên một cáp.

Chế độ băng cơ bản

Trong chế độ này, cáp được điều khiển bởi một nguồn điện áp tại một đầu. Nhờ hình dạng của cáp nên hạn chế được nhiễu từ ngoài. Hiện đang có một số mạch giao tiếp cho ứng dụng cáp đồng trục. Một kết nối tiêu biểu được trình bày trên hình 2.13, trên đó cũng trình bày ảnh hưởng của kết cuối đường dây với trở kháng kết cuối chính xác Z_0 . Tổ chức này phù hợp với truyền số liệu tốc độ lên đến 10Mbps qua khoảng cách vài trăm mét.



Hình 2.13 Các nguyên lý băng cơ bản:
(a) Các tín hiệu cáp (b) Các phương pháp kết nối



**Hình 2.13 (tiếp theo) Các nguyên lý băng cơ bản:
(c) Các phương pháp TDM.**

Trong một vài ứng dụng, cáp này được dùng để truyền giữa hai hệ thống và được gọi là điểm-nối-diểm (point-to-point), trong một số ứng dụng khác thì kênh truyền tốc độ cao được chia sẻ cho một số các hệ thống và được gọi là cấu hình đa điểm (multipoint). Cả hai dạng được trình bày trên hình 2.13b.

Ghép kênh phân thời TDM (Time Division Multiplexing) được dùng để chia sẻ khả năng có sẵn của một kênh truyền băng cơ bản. Có hai loại TDM được dùng:

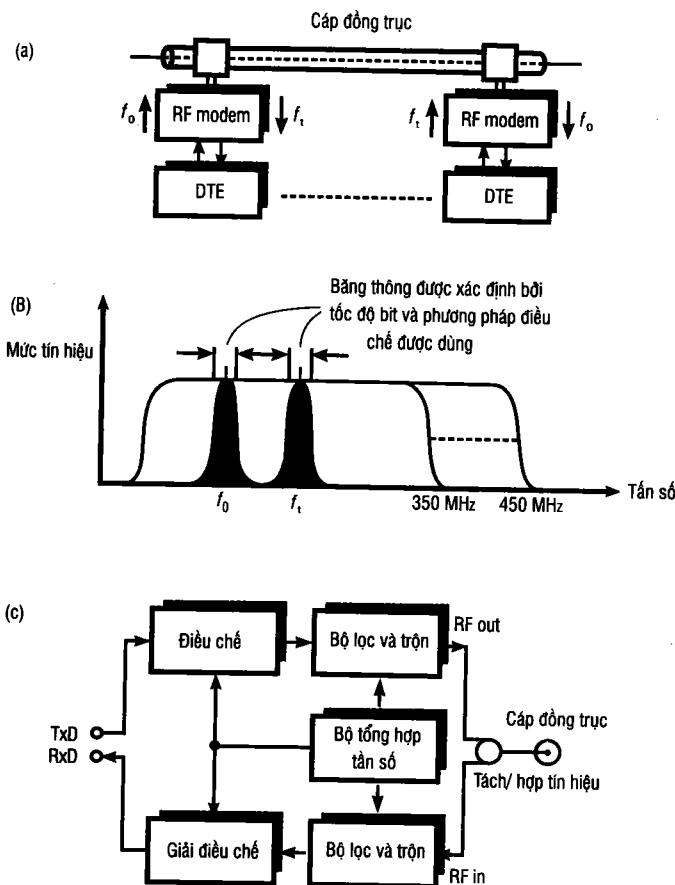
1. TDM đồng bộ. Mỗi đầu cuối truy xuất kênh tại những khoảng thời gian xác định trước.
2. TDM bất đồng bộ (hay theo yêu cầu). Mỗi đầu cuối truy xuất ngẫu nhiên kênh theo nhu cầu truy xuất.

Hai dạng TDM được trình bày trên hình 2.13(c).

Chế độ băng rộng

Dùng chế độ băng rộng, các kênh truyền được thực hiện trên một cáp nhờ kỹ thuật ghép kênh phân tần FDM (Frequency Division Multiplexing). FDM yêu cầu một modem RF (Radio Frequency) giữa mỗi thiết bị và cáp. Dùng thuật ngữ RF vì mỗi kênh dùng tần số thuộc phổ tần RF. Sóng mang truyền được điều chế bằng dữ liệu truyền và sóng thu được giải điều chế để suy ra số liệu.

Băng thông yêu cầu cho mỗi kênh được xác định bởi tốc độ số liệu mong muốn và phương pháp điều chế. Hiệu suất băng thông của modem RF là 0,25 đến 1 bHz⁻¹. Nguyên lý của chế độ hoạt động này được trình bày trên hình 2.14. Thông thường điều chế trong modem được thực hiện theo hai pha. Trước hết một tín hiệu đã chọn được điều chế bởi số liệu truyền dùng FSK hay PSK (xem phần điều chế phía sau). Sau đó tín hiệu đã điều chế được trộn với một tần số thứ hai để tín hiệu nằm trong băng tần được phân phối. Các bộ lọc trình bày trong hình 2.14 có vai trò chỉ cho phép truyền/nhận các tín hiệu thuộc băng tần đã phân phối.

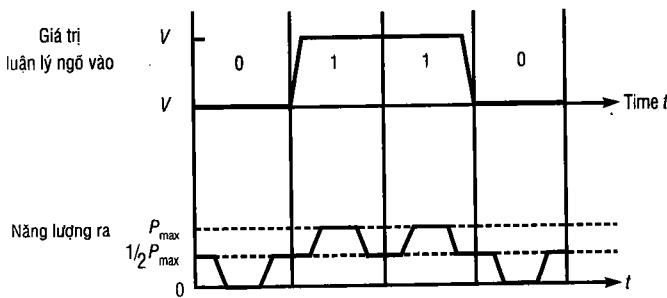
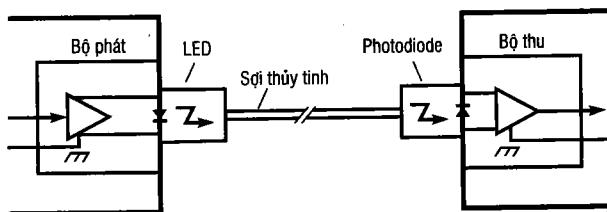


Hình 2.14 Các nguyên lý băng rộng:
(a) Sơ đồ cáp (b) Băng thông (c) Lược đồ modem RF.

3.5. Các tín hiệu cáp quang

Có một số dạng mã hóa tín hiệu quang. Một dựa trên lược đồ mã hóa luồng cực, được trình bày trong hình 2.15. Loại này tạo ra ngõ ra quang 3 mức, phù hợp với hoạt động của cáp từ DC đến 50Mbps. 3 mức năng lượng quang là zero, một nửa mức tối đa và mức tối da. Module truyền thực hiện chuyển từ các mức điện áp nhị phân bên trong sang tín hiệu quang 3 mức đặt lên cáp nhờ các bộ nối đặc biệt và một LED tốc độ cao.

Tại bộ thu, cáp được kết cuối với một bộ nối đặc biệt đi đến diode thu quang tốc độ cao ngũ trong một module thu đặc biệt. Module này chứa các mạch điện tử cần cho việc chuyển đổi tín hiệu điện tạo ra bởi diode quang ti lệ với mức ánh sáng, thành các mức điện áp bên trong tương ứng với bit 1 và 0.



Hình 2.15 Các tín hiệu cáp quang

3.6. Tín hiệu vệ tinh và radio

Trong mục 1 chúng ta đã đề cập đến các kênh truyền trong các hệ thống vệ tinh và radio được tạo ra nhờ ghép kênh phân chia tần số (FDM: Frequency Division Multiplexing). Bên cạnh đó, dung lượng có sẵn của mỗi kênh còn được chia nhỏ hơn nhờ kỹ thuật ghép kênh phân thời đồng bộ (TDM: Time Division Multiplexing).

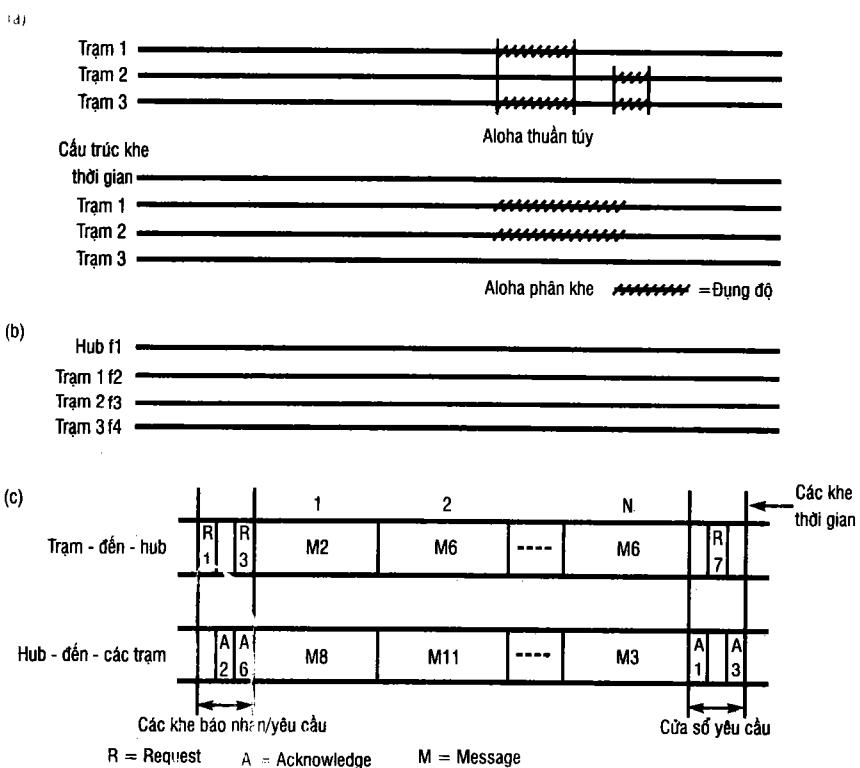
Có một số phương pháp điều khiển truy xuất khác nhau được dùng để điều khiển truy xuất vào phần dung lượng có sẵn:

- Truy xuất ngẫu nhiên: tất cả các trạm tranh chấp kênh truyền theo ngẫu nhiên (không có điều khiển)
- Gán cố định: cả khe thời gian cũng như tần số được gán trước cho mỗi trạm
- Gán theo yêu cầu: khi một trạm muốn truyền số liệu, trước hết nó yêu cầu dung lượng kênh từ trung tâm, trung tâm có chức năng phân phối dung lượng truyền cho các trạm yêu cầu.

Truy xuất ngẫu nhiên là phương pháp điều khiển truy xuất cổ điển nhất và được dùng lần đầu tiên để điều khiển truy xuất một kênh vệ tinh dùng chung (chia sẻ). Nó chỉ dùng với các ứng dụng trong đó, dạng thứ nhất là toàn bộ tải được cung

cấp chỉ là phần nhỏ của dung lượng kênh có sẵn và dạng thứ hai là tất cả các hoạt động truyền phân bố ngẫu nhiên. Kỹ thuật này còn được gọi là Aloha vì nó được dùng lần đầu tiên bởi Đại học Hawaii để liên kết một nhóm các máy tính phân bố rải rác trên các đảo đến máy tính trung tâm đặt tại đảo Oahu. Hai phiên bản của lược đồ này được trình bày trên hình 2.16 (a).

Với Aloha, khi một trạm có thông điệp cần truyền, nó chỉ làm công việc đơn giản là truyền nó. Nếu một trạm thứ hai bắt đầu truyền trong khi trạm thứ nhất đang truyền, sẽ có một sự đụng độ xảy ra và cả hai cuộc truyền đều bị hỏng. Vì vậy lược đồ này chỉ thích hợp nếu xác xuất hai trạm truyền cùng lúc là nhỏ. Giả sử các thông điệp được phát ngẫu nhiên, thông lượng trung bình đạt được của lược đồ này thấp hơn 20% khả năng. Điều này có thể được cải thiện bằng cách thiết lập một cấu trúc khe thời gian (time slot) đồng bộ và ràng buộc tất cả các hoạt động truyền vào các khe thời gian này. Lược đồ này được gọi là **slotted Aloha**, và vì một hoạt động truyền có thể làm hỏng một hoạt động truyền khác chỉ khi ở trong cùng một khe thời gian nên hiệu suất có thể đạt được hơn 30%.



Hình 2.16 Các phương pháp điều khiển truy xuất kênh vệ tinh và radio



Với phương pháp gán cố định, cả khe thời gian và kênh tần số được gán trước cho mỗi trạm. Nhìn chung việc gán trước các kênh tần số dễ hơn gán khe thời gian. Ví dụ, trong các ứng dụng vệ tinh dựa vào hub trung tâm, một kênh tần số cố định được gán trước cho mỗi VSAT và sau đó trung tâm phát quảng bá (broadcast) lên các kênh tần số được gán trước khác. Nhìn chung, vì chỉ có một kênh từ hub đến VSAT, nên băng tần của kênh này rộng hơn so với kênh được dùng cho hoạt động truyền từ VSAT đến hub. Thông thường tốc độ bit là 64Kbps cho mỗi kênh VSAT đến hub và đến 2Mbps cho kênh broadcast từ hub đến VSAT. Dạng lược đồ diều khiển truy xuất này được gọi là đa truy xuất phân tần được gán trước (preassigned frequency-division multiple access hay preassigned FDMA), được trình bày trên hình 2.16(b).

Chúng ta có thể đạt được hiệu suất kênh tốt hơn bằng cách dùng phương pháp diều khiển truy xuất gán theo yêu cầu. Lược đồ này cung cấp một số các khe thời gian theo yêu cầu_ gọi tắt là *khe thời gian yêu cầu* (request time slot)_trong đó VSAT và các trạm di động có thể gửi yêu cầu đến hub hay trạm cơ bản (base station) để lấy một hay nhiều *khe thời gian thông điệp* (message time slot). Nếu có sẵn, các điểm trung tâm sẽ gán các khe thời gian thông điệp đặc biệt cho hoạt động truyền đó và thông báo với trạm đã yêu cầu bằng *khe thời gian báo nhận* (acknowledgment time slot). Hình 2.16(c) mô tả lược đồ này, lược đồ này được gọi là đa truy xuất phân thời được gán theo yêu cầu (demand-assigned time-division multiple access hay demand-assigned TDMA)

Từ hình 2.16(c) chúng ta có thể thấy rằng khe yêu cầu ngắn hơn nhiều so với các khe thông điệp thông thường. Một thông điệp yêu cầu bao gồm danh định của trạm yêu cầu và giả sử có nhiều khe thời gian trên mỗi thông điệp thì phải có số lượng khe thời gian thông điệp yêu cầu. Sau đó thông điệp báo nhận tương ứng chỉ ra khe thời gian nào sẽ được dùng. Tất cả các hoạt động truyền từ điểm trung tâm được thực hiện trực tiếp với danh định của trạm đích được đặt tại đầu của thông điệp. Để giảm xác xuất dụng độ xảy ra khi yêu cầu khe, một trạm chọn ngẫu nhiên một khe để dùng. Nếu có dụng độ xảy ra (biểu hiện khi không có đáp ứng) thì trạm cố gắng lần nữa trong cửa sổ yêu cầu (request window) kế tiếp.

Trước khi kết thúc phần này cần nhấn mạnh rằng BER của liên kết radio nhìn chung cao hơn nhiều so với liên kết dây hữu tuyến. Hậu quả là thường phải dùng các khối thông điệp nhỏ và sử dụng các phương pháp phát hiện lỗi và sửa sai phức tạp hơn.

4. TRỄ DO LAN TRUYỀN TÍN HIỆU

Luôn luôn tồn tại một thời gian trễ tuy ngắn nhưng xác định khi một tín hiệu lan truyền từ một điểm này đến điểm khác qua môi trường truyền. Thời gian trễ này được gọi là trễ lan truyền bởi môi trường, thường được ký hiệu là T_p . Các tín hiệu lan truyền trong không gian tự do với tốc độ bằng tốc độ ánh sáng (3×10^8 m/s). Tốc độ lan truyền trong cáp xoắn hay cáp đồng trực thấp hơn. Thông thường, vào khoảng 2×10^8 m/s. Mặc dù điều này hầu như không có ý nghĩa gì, nhưng trong một vài trường hợp thời gian trễ này lại rất quan trọng.

Dữ liệu thường được truyền dưới dạng khối. Khi tiếp nhận một khối, máy thu gửi một báo nhận cho máy phát. Một tham số quan trọng của liên kết số liệu là thời gian trễ liên quan đến liên kết, đó chính là thời gian kể từ khi bit đầu tiên của khối được truyền bởi máy phát cho đến bit cuối cùng của báo nhận thu được. Rõ ràng, tham số này không chỉ phụ thuộc vào thời gian truyền frame dữ liệu T_x , mà còn phụ thuộc vào thời gian trễ do lan truyền T_p . Tỉ lệ giữa hai tham số thời gian này thay đổi theo các kiểu liên kết số liệu khác nhau.

$$a = T_p/T_x$$

Với T_p và T_x được tính như sau:

$$T_p = \text{khoảng cách(m)}/\text{tốc độ lan truyền (m/s)}$$

$$T_x = \text{số bit được truyền}/\text{tốc độ bit của liên kết (bps)}$$

Nếu $a < 1$ thì thời gian cần truyền được xác định chủ yếu theo thời gian truyền khối số liệu.

Nếu $a=1$ thì cả hai đều có ảnh hưởng như nhau.

Nếu $a>1$ thì trễ do lan truyền chiếm ưu thế.

5. CÁC MẠCH TẢI CÔNG CỘNG

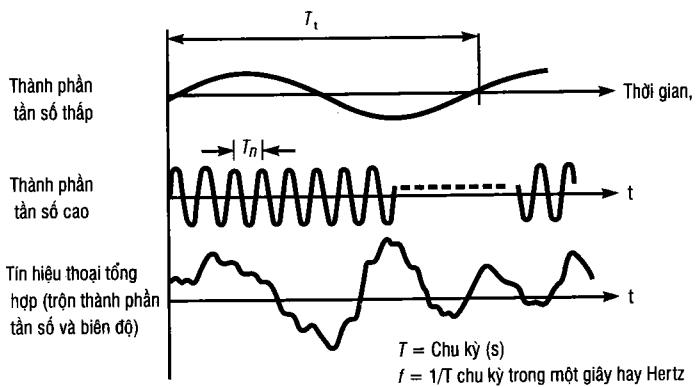
Khi muốn truyền số liệu giữa hai DTE trong một tòa nhà thì việc đơn giản là lắp đặt cáp dẫn cho nó. Thông thường, dùng cáp xoắn, cáp đồng trực hay sợi quang. Trong một vài trường hợp cũng có thể dùng sóng vô tuyến. Tuy nhiên trong các trường hợp muốn truyền số liệu giữa hai DTE cách xa nhau, chỉ có thể dùng các liên kết vi ba, vệ tinh hoặc dây hữu tuyến cung cấp từ các công ty truyền dẫn công cộng. Giải pháp thuê đường truyền rất phổ biến, có hai dạng: kênh chuyển mạch và kênh thuê riêng (leased line).

Chúng ta có thể xây dựng các cầu nối chuyển mạch dùng PSTN hay ISDN. Mặc dù PSTN được thiết kế để truyền thoại nhưng cũng có thể truyền dữ liệu thông qua modem. Trong trường hợp dùng ISDN, chúng ta có thể thiết lập cuộc gọi và truyền một cách trực tiếp và tốc độ đạt được cao hơn.

Với mạch thuê riêng, cho dù trong một vài trường hợp chúng ta cũng phải dùng các đường thuê riêng trên PSTN, đa số là các kênh truyền dẫn số.

5.1. Các mạch PSTN analog

Khi muốn truyền dữ liệu dùng các đường truyền PSTN analog có sẵn, cần phải chuyển đổi các tín hiệu điện đi ra từ các DTE sang dạng phù hợp với PSTN. Đường truyền PSTN được thiết kế cho thông tin tiếng nói có băng thông trong khoảng 300-3400Hz, như mô tả trên hình 2.17. Điều này có nghĩa là một đường dây điện thoại sẽ không truyền các tín hiệu có tần số thấp khi có một chuỗi liên tiếp các bit 1 hay 0. Chính vì lý do này mà không thể đặt 2 mức điện áp từ DTE trực tiếp lên đường dây. Thay vì vậy, cần phải chuyển đổi dữ liệu nhị phân sang một dạng tương thích với tín hiệu thoại tại DTE phát và chuyển đổi trở lại thành dạng nhị phân tại đầu thu. Mạch điện thực hiện hoạt động chuyển đổi thuận ban đầu gọi là bộ điều chế (modulator), và mạch thực hiện chuyển đổi ngược lại được gọi là bộ giải điều chế (demodulator). Vì mỗi đầu liên kết đều cần cả hai mạch, chúng kết hợp lại thành một thiết bị gọi chung là modem.

**Hình 2.17** Các thành phần tần số sóng âm

Điều chế

Có ba phương pháp điều chế cơ bản để chuyển tín hiệu nhị phân sang dạng thích hợp để truyền qua mạng điện thoại công cộng PSTN, đó là: phương pháp điều chế biên độ, phương pháp điều chế tần số và phương pháp điều chế pha (phase). Dữ liệu nhị phân được truyền chỉ yêu cầu hai mức tín hiệu. Sự chuyển mạch tín hiệu giữa hai mức mang ý nghĩa khóa (key). Vì vậy, ba loại điều chế trên lần lượt được gọi là ASK (Amplitude - Shift - Keying), FSK (Frequency - Shift - Keying) và PSK (Phase - Shift - Keying).

(1). ASK (amplitude-shift keying)

Nguyên lý hoạt động cơ bản của ASK được minh họa trong hình 2.18(a), sóng dạng đơn giản ở hình 2.18(b).

Bản chất của phương pháp là biên độ của sóng âm đơn tần được chuyển đổi giữa hai mức với tốc độ được xác định bởi tốc độ bit của tín hiệu nhị phân được truyền. Sóng âm tần hay còn gọi là sóng mang có tần số thuộc băng thông PSTN. Kích thước băng thông yêu cầu được xác định bởi tốc độ bit của tín hiệu, tốc độ bit càng cao thì kích thước băng thông yêu cầu càng lớn. Trong thực tế các phương pháp điều chế khác nhau đòi hỏi độ rộng băng thông khác nhau, nên việc đánh giá mức băng thông yêu cầu ứng với mỗi phương pháp là rất cần thiết.

Về mặt toán học, hoạt động điều chế ASK, FSK và PSK tương đương với việc nhân tín hiệu sóng mang với tín hiệu dữ liệu nhị phân. Sóng mang có tần số riêng ω_c và biên độ không đổi có thể biểu diễn dưới dạng biểu thức điện áp.

$$V_c(t) = \cos \omega_c t \quad \omega_c : \text{radian /second}$$

Ở phần trước, ta đã biểu diễn tín hiệu số liệu tuần hoàn đơn cực với biên độ không đổi và tần số cơ bản ω_0 theo chuỗi Fourier như sau:

$$V_d(t) = 1/2 + 2/\pi (\cos \omega_0 t - 1/3 \cos 3\omega_0 t + 1/5 \cos 5\omega_0 t - \dots)$$

Suy ra ASK có thể biểu diễn :

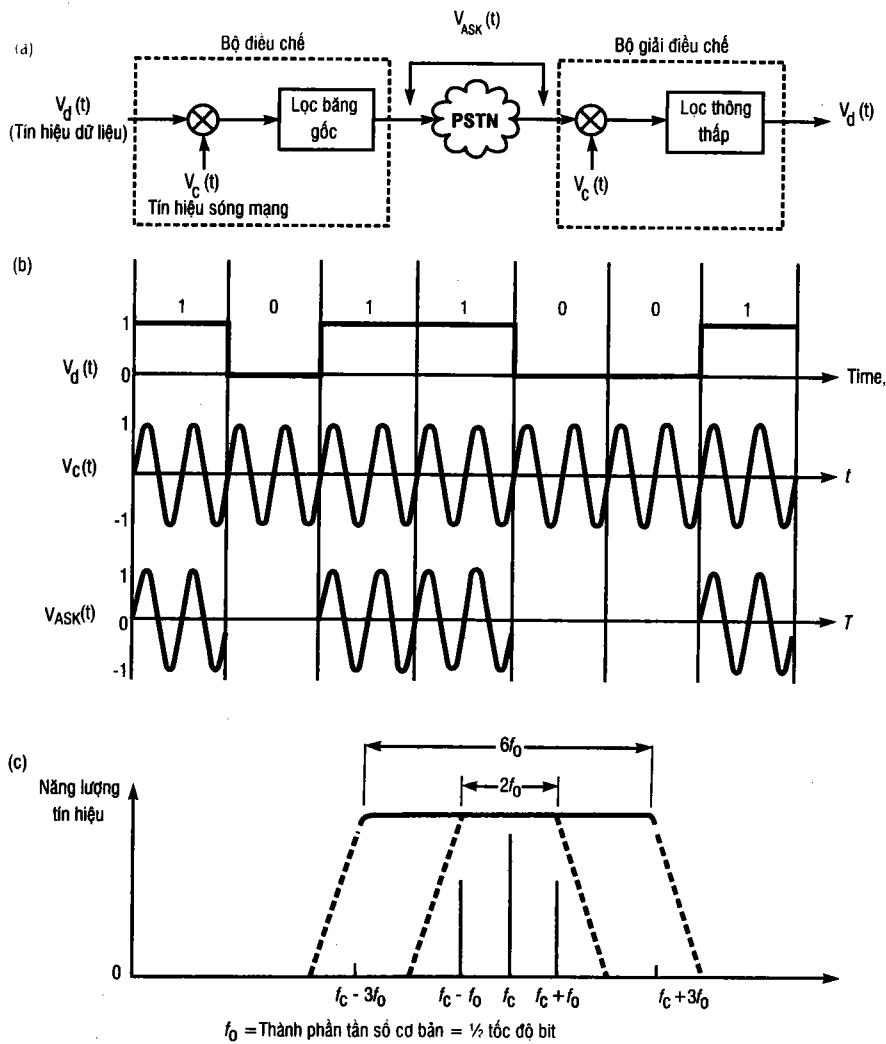
$$V_{ASK}(t) = V_c(t) \times V_d(t)$$

Do đó : $V_{ASK}(t) = 1/2 \cos \omega_c t + 2/\pi [\cos(\omega_c - \omega_0)t + \cos(\omega_c + \omega_0)t - 1/3 \cos 3\omega_c t - 1/3 \cos(3\omega_c - 3\omega_0)t - 1/3 \cos(\omega_c + 3\omega_0)t + ...]$

Suy ra :

$$V_{ASK}(t) = 1/2 \cos \omega_c t + 1/\pi [\cos(\omega_c - \omega_0)t + \cos(\omega_c + \omega_0)t - 1/3 \cos(3\omega_c - 3\omega_0)t - 1/3 \cos(\omega_c + 3\omega_0)t + ...]$$

Chúng ta thấy tín hiệu ASK tương đương với tín hiệu số liệu nguồn chuyển dịch sang dạng tần số của sóng mang, bên cạnh đó còn có hai thành phần tần số cơ bản ($\omega_c - \omega_0$) và ($\omega_c + \omega_0$) và hai thành phần hài bậc cao ($\omega_c - 3\omega_0$) và ($\omega_c + 3\omega_0$). Tất cả hình thành nên biên tần cơ bản, do đó băng thông của ASK được trình bày trên hình 2.18(c).



Hình 2.18 ASK:

(a) Lược đồ mạch; (b) Dạng sóng hoạt động; (c) Băng thông.

Băng thông càng lớn thì tín hiệu thu càng trung thực. Tuy nhiên, tổng quát mà nói thì hiệu quả hoạt động có thể đạt được nếu băng thông của kênh đủ chỗ để cho qua thành phần tần số cơ bản của dòng dữ liệu 101010... vì băng thông yêu cầu trong các dòng dữ liệu có thứ tự khác đều nhỏ hơn. Gọi f_0 là tần số cơ bản của tín hiệu dữ liệu theo tuần tự 101010... f_0 bằng nửa tốc độ bit (bps) suy ra băng thông tối thiểu của ASK thì bằng với tốc độ bit tính sang đơn vị Hz, $2f_0$, hoặc để thu được thành phần hài bậc 3 thì phải gấp 3 lần tốc độ bit tính sang Hz, $6f_0$.

Từ hình 2.18 có thể thấy rằng đối với ASK tín hiệu sóng mang hiện diện trong tín hiệu thu ngay cả khi không có tín hiệu thông tin f_0 , $3f_0$, .v.v. trong đó. Từ công thức Nyquist ở trên ta suy ra với tín hiệu nhị phân thì tốc độ số liệu tối đa đạt được bằng hai lần băng thông. Vì thế, giả sử băng thông $2f_0$, tốc độ Nyquist sẽ gấp hai lần. Lý do chính ở đây là vì cả hai biên tần đều được dùng để xác định băng thông tối thiểu. Trên hình 2.18(c) cả hai biên đều chứa tín hiệu yêu cầu f_0 . Để dùng băng thông hiệu quả hơn có thể dùng bộ lọc thông như trong lược đồ hình 2.18(a) để giới hạn băng tần đến $f_c + (f_c + f_0)$ do đó loại bỏ biên thấp, $f_c - f_0$. Điều đó làm giảm băng thông yêu cầu đến f_0 và có thể đạt được tốc độ Nyquist. Tuy nhiên điều này sẽ cắt đôi năng lượng của tín hiệu biên tần chính liên hệ đến tín hiệu sóng mang, mà năng lượng giảm dẫn đến giảm SNR và do đó làm tăng tốc độ lỗi bit (BER).

Việc khôi phục lại tín hiệu số liệu được thực hiện bởi mạch giải điều chế. Tại đây tín hiệu thu lại được nhân một lần nữa với sóng mang cùng dạng. Từ đó sinh ra hai phiên bản của tín hiệu thu: thành phần bao quanh tần số $2f_c$ ($f_c + f_c$) và thành phần bao quanh zero ($f_c - f_c$). Cả hai phiên bản đều chứa thông tin trong hai biên tần, nhưng ta chọn thành phần sau cùng bằng cách cho đi qua bộ lọc thông thấp. Bộ lọc thông này được thiết kế để chỉ cho qua các tần số từ 0 đến f_0 hoặc nếu hài bậc 3 được nhận thì từ 0 đến $3f_0$. Do vậy, ở đầu ra của bộ lọc chính là phiên bản của tín hiệu dữ liệu truyền bị giới hạn băng tần.

Mặc dù ASK là phương pháp thực hiện đơn giản, tuy nhiên không được dùng trong các modem tốc độ thấp thế hệ đầu. Vào thời điểm đó hệ thống truyền tin hầu hết dùng kỹ thuật tương tự (analog) vì vậy dẫn đến suy giảm tín hiệu trong nhiều mức khác nhau. Gần đây tất cả hệ thống truyền dẫn và chuyển mạch đều được thực hiện bởi kỹ thuật số (digital) và ASK có cơ hội tham gia, thường được dùng kết hợp với PSK để thiết kế các modem tốc độ cao.

Ví dụ:

Giả sử dùng ASK để điều chế, hãy ước lượng băng thông yêu cầu của kênh truyền ở hai trường hợp: thành phần tần số cơ bản của chuỗi dữ liệu 101010... được nhận và thành phần tần số cơ bản cộng với hài bậc 3. Xét ở 3 tốc độ 300bps, 1200bps và 1800 bps

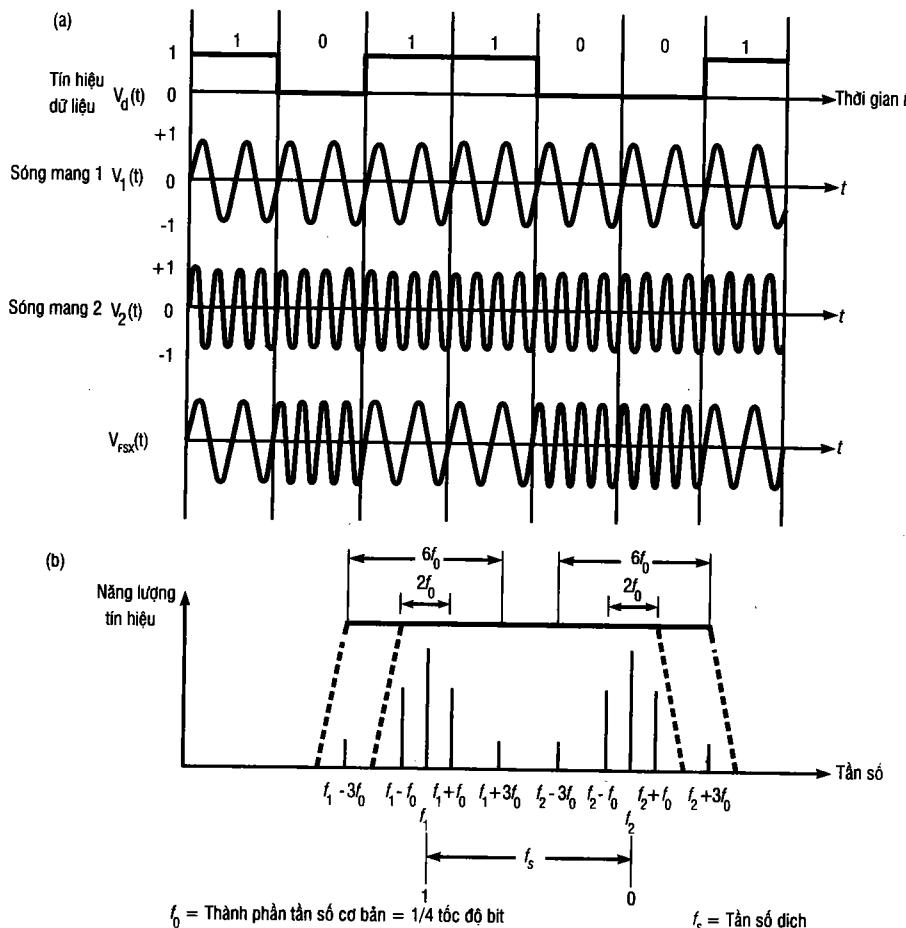
Tốc độ bit	300bps	1200bps	4800bps
Thành phần tần số cơ bản	150Hz	600Hz	2400Hz
Thành phần hài bậc 3	450Hz	1800Hz	7200Hz

Băng thông với tần số cơ bản	300Hz	1200Hz	4800Hz
Băng thông với tần số cơ bản và hài bậc 3	900Hz	3600Hz	14400Hz

Lưu ý rằng băng thông hữu dụng của mạng PSTN là 3000Hz. Suy ra chỉ 300bps có thể đạt được với hài bậc 3. Tốc độ 1200 chỉ đạt được với thành phần cơ bản. Tốc độ 4800 không thể truyền nếu chỉ dùng phương pháp ASK.

(2). FSK (frequency-shift keying)

FSK là phương pháp điều chế được dùng trong tất cả các modem tốc độ thấp thế hệ đầu. Nguyên lý hoạt động cơ bản được minh họa trong hình 2.19. Để tránh vấn đề thay đổi biên độ, với FSK dùng hai tín hiệu sóng mang có cùng biên độ và cố định, một cho bit nhị phân 0 và một cho bit nhị phân 1.



Hình 2.19 FSK : (a) Nguyên lý hoạt động; (b) Băng thông



Sự khác biệt giữa hai sóng mang là tần số. Như trình bày trên hình vẽ, hoạt động điều chế tương đương với sự tổng hợp các ngõ ra của hai bộ điều chế ASK riêng biệt: một thực hiện trên sóng mang thứ nhất dùng phần gốc của tín hiệu dữ liệu (mức cao) và một thực hiện trên sóng mang thứ hai dùng phần bù của tín hiệu dữ liệu (mức 0). Về mặt toán học có thể suy ra băng thông yêu cầu đối với phương pháp FSK thông qua biểu thức sau :

$$V_{FSK}(t) = \cos \omega_1 t \cdot V_d(t) + \cos \omega_2 t \times V_d(t)$$

Trong đó ω_1 và ω_2 là tần số góc của hai sóng mang

$V_d(t)$ là phần bù của tín hiệu dữ liệu gốc $V_d(t)$.

Ta có: $V_d(t) = 1 - V_d(t)$, vì vậy nếu giả sử tín hiệu dữ liệu tuần hoàn với tần số cơ bản W_0 thì :

$$V_{FSK}(t) = \cos \omega_1 t [1/2 + 2/\pi (\cos \omega_0 t - 1/3 \cos 3\omega_0 t + \dots)]$$

$$+ \cos \omega_2 t [1/2 + 2/\pi (\cos \omega_0 t - 1/3 \cos 3\omega_0 t + \dots)]$$

Do đó:

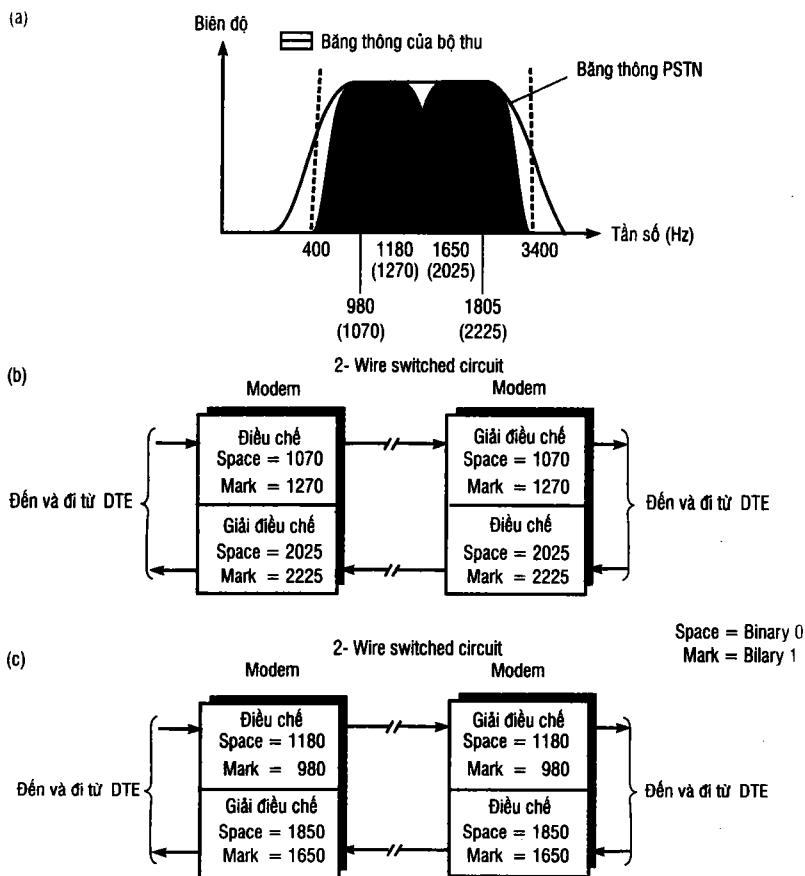
$$\begin{aligned} V_{FSK}(t) &= 1/2 \cos \omega_1 t + 1/\pi [\cos(\omega_1 - \omega_0)t + \cos(\omega_1 + \omega_0)t \\ &\quad - 1/3 \cos(\omega_1 - 3\omega_0)t - 1/3 \cos(\omega_1 + 3\omega_0)t + \dots] \\ &\quad + 1/2 \cos \omega_2 t + 1/\pi [\cos(\omega_2 - \omega_0)t + \cos(\omega_2 + \omega_0)t \\ &\quad - 1/3 \cos(\omega_2 - 3\omega_0)t - 1/3 \cos(\omega_2 + 3\omega_0)t + \dots] \end{aligned}$$

Có thể suy ra băng thông của FSK một cách đơn giản là tổng hai sóng mang điều chế ASK riêng biệt ở tần số góc ω_1 và ω_2 . Băng thông của FSK như hình 2.19(b).

Như đã đề cập ở phần trước, tín hiệu có tần số cao nhất tạo ra từ dãy bit 101010... Với FSK vì tín hiệu nhị phân 0 và 1 điều chế lên các sóng mang riêng rẽ, nên băng thông tối thiểu cho mỗi sóng mang bằng một nửa tốc độ bit, nghĩa là thành phần tần số cơ bản lớn nhất cho mỗi sóng mang, f_0 , bằng một nửa so với ASK. Do đó nếu giả sử chỉ thu thành phần tần số cơ bản thì băng thông tổng bằng $4f_0$ cộng với khoảng dịch tần, f_s . Tuy nhiên vì f_0 bằng một nửa so với ở ASK nên băng thông tổng bằng với băng thông tổng cho ASK cộng với khoảng dịch tần. Tương tự nếu cặp hài bậc 3 được thu thì băng thông bằng $6f_0$ cộng với khoảng dịch tần.

Ví dụ, nếu tốc độ bit lớn nhất là 600bps thì tốc độ bit lớn nhất trên mỗi sóng mang là 300bps, có thành phần tần số cơ bản là 150Hz. Do đó, phổ tần sẽ chứa các biên tần chính cách nhau 150Hz trên mỗi biên của cả hai sóng mang. Do đó nếu chúng ta chọn khoảng dịch tần (f_s) là 400Hz thì sẽ tạo ra khoảng 100Hz giữa hai biên tần chính của cả hai sóng mang. Điều này có nghĩa là băng thông tổng là 700Hz. Rõ ràng vì băng thông của PSTN là 3000Hz do đó có thể lấy ra hai kênh như vậy từ một cuộc nối đơn qua PSTN - một cho mỗi hướng truyền.

Hình 2.20 mô tả sự phân phối tần số được dùng cho hai loại modem FSK hỗ trợ truyền song công 300bps giữa 2 DTE. Một tập phân bố tần số được qui định bởi EIA cho modem Bell 103 và tập khác bởi ITU-T cho modem V.21. Lưu ý rằng mỗi modem như vậy thành phần tần số cơ bản cách với mỗi sóng mang là 75Hz. Suy ra khoảng dịch tần 200Hz tạo ra khoảng cách 50Hz giữa hai biên tần chính.



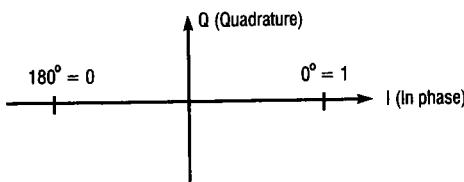
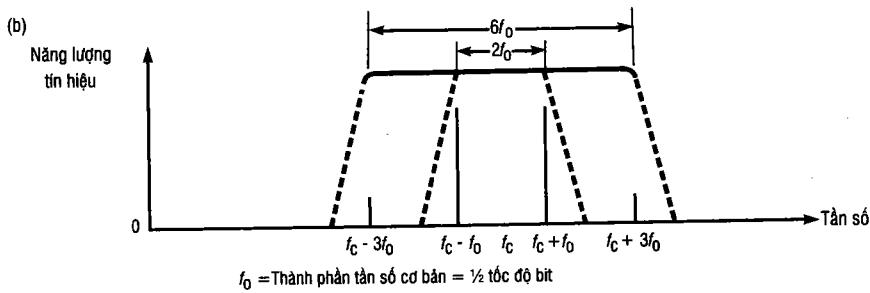
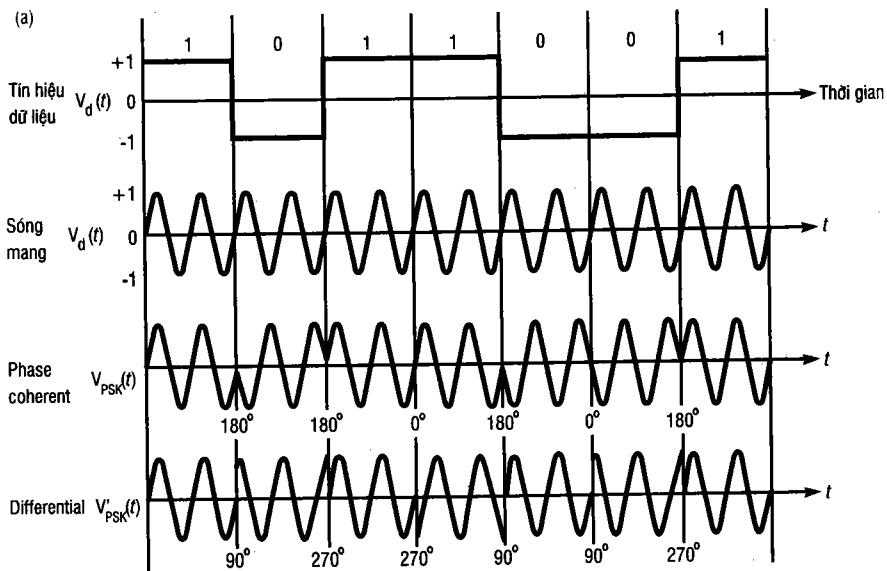
Hình 2.20 Phân phối tần số của modem Bell 300bps song công hoàn toàn:

(a) Phổ tần (b) Phân bố tần số theo EIA (c) Phân bố tần số theo ITU-T

(3). PSK (phase-shift keying)

Trong phương pháp điều chế PSK tần số và biên độ của sóng mang được giữ không đổi trong khi pha của nó được dịch theo mỗi bit của dòng dữ liệu truyền. Nguyên lý hoạt động được trình bày ở hình 2.21(a). Có hai loại PSK được dùng, loại thứ nhất dùng hai tín hiệu sóng mang cố định đại diện cho bit 0 và 1, hai sóng mang khác pha nhau 180° . Vì tín hiệu này chỉ là nghịch đảo của tín hiệu kia

nên loại này còn được gọi là phase-coherent PSK (PSK phối hợp phase). Điều bất tiện của loại này là tại máy thu đòi hỏi phải có sóng mang tham chiếu để so phase với tín hiệu thu, do đó cần thực hiện đồng bộ pha giữa máy thu và phát. Kết quả dẫn đến mạch giải điều chế phức tạp hơn.



Hình 2.21 PSK: (a) Nguyên lý (b) Băng thông (c) Lược đồ phase

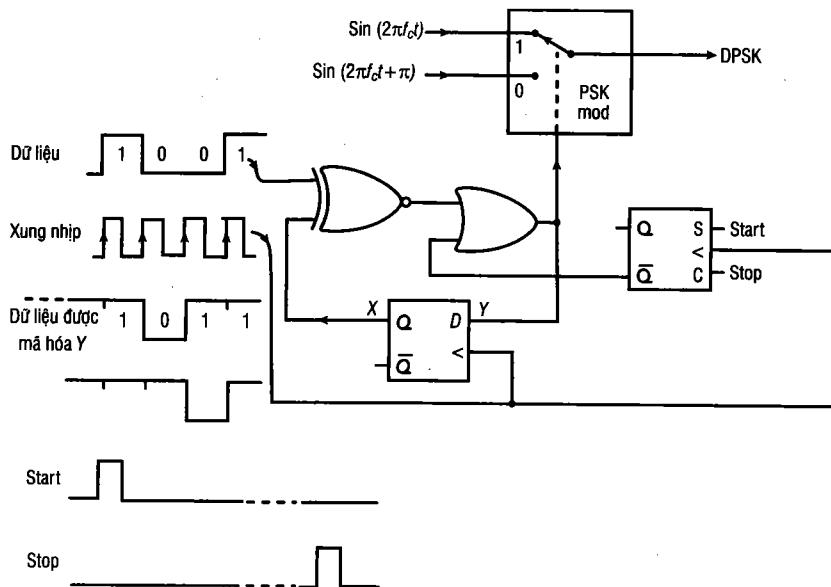
Loại PSK thứ hai gọi là PSK vi phân (differential PSK). Với loại này sự dịch pha xảy ra tại mỗi bit không cần quan tâm tới chuỗi bit 0 hay bit 1 đang được

truyền. Một sự dịch pha 90° tương ứng với tín hiệu hiện hành chỉ định 0 là bit kế tiếp trong khi sự dịch pha 270° chỉ bit 1. Như vậy mạch giải điều chế chỉ cần xác định độ lớn của sự dịch pha thay vì phải xác định giá trị tuyệt đối của từng pha. Ở mạch điều chế chỉ khi nào thay đổi trạng thái của dữ liệu mới đổi pha của sóng mang.

Để hiểu rõ hơn về DPSK (differential PSK) ta tham khảo mạch điều chế DPSK đơn giản ở hình 2.22.

Trong mạch điện cổng OR và flip-flop SC dùng để giữ ngõ ra ở một trạng thái duy nhất trong chu kỳ nhàn rỗi (không có dữ liệu). Kết thúc một ký tự có stop =1 và start =0 nên ngõ ra Q của flip-flop SC là 1, ngõ ra cổng OR luôn được giữ là 1, ngõ ra mạch chốt D cũng là 1. Tín hiệu X ở ngõ ra mạch chốt chính là tín hiệu Y trễ đi một thời bit.

Sóng mang DPSK sẽ đổi phase theo sự điều khiển của Y, như vậy khi bit dữ liệu hiện tại khác với bit dữ liệu đã mã hóa (Y) ở thời điểm trước đó thì sóng mang sẽ có góc pha là 180° , nếu giống nhau thì góc pha là 0° . Nhờ vậy, máy thu chỉ cần tách được các giá trị bit 1 và 0 chứ không cần biết giá trị pha tuyệt đối của sóng mang.



Hình 2.22 Sơ đồ mạch điều chế pha vi phân đơn giản

Về mặt toán học ta có thể xác định bằng thông của PSK. Ở đây chúng ta trình bày tín hiệu số liệu nhị phân dưới dạng lưỡng cực vì mức tín hiệu âm sẽ là kết quả đổi pha 180° của sóng mang. Như trình bày ở phần trước tín hiệu số liệu tuần hoàn có biên độ tổng hợp và thành phần tần số cơ bản là ω_0 được biểu diễn dưới dạng chuỗi Fourier

$$V_d(t) = 4/\pi [\cos \omega_0 t - 1/3 \cos 3\omega_0 t + 1/5 \cos 5\omega_0 t \dots]$$

Suy ra:

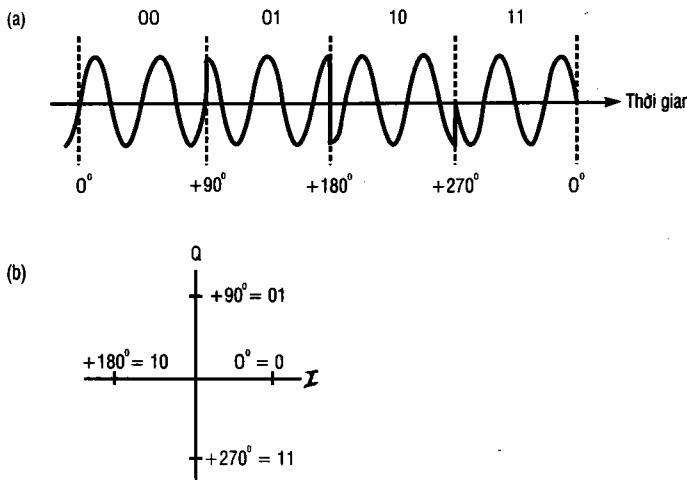
$$\begin{aligned} V_{PSK} &= 4/\pi [\cos \omega_c t \cdot \cos \omega_0 t - 1/3 \cos \omega_c t \cdot \cos 3\omega_0 t + \dots] \\ &= 2/\pi [\cos (\omega_c - \omega_0) t + \cos (\omega_c + \omega_0) t - 1/3 \cos (\omega_c - 3\omega_0) t \\ &\quad - 1/3 \cos (\omega_c + 3\omega_0) t + \dots] \end{aligned}$$

Từ trên cho thấy phô tần giống như ASK chỉ khác là không có thành phần sóng mang. Do đó, băng thông của tín hiệu PSK như hình 2.21. Vì vậy nếu giả sử chỉ thành phần tần số cơ bản của dãy 101010...được nhận, thì băng thông tối thiểu bằng $2f_0$, bằng với giá trị của tốc độ bit. Tuy nhiên, do vắng mặt thành phần sóng mang nên có nhiều năng lượng trong biên tần hơn (data), điều đó giúp PSK kháng nhiễu tốt hơn ASK hay FSK. Giới hạn băng của tín hiệu từ f_c đến $f_c + f_0$, có nghĩa là băng thông bằng f_0 và đạt được tốc độ Nyquist. Hầu hết năng lượng nhận được đều thuộc về tín hiệu mang thông tin, $f_c + f_0$ (do không có f_c).

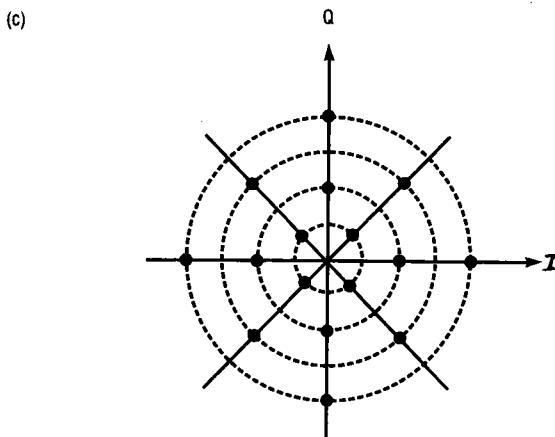
Thường biểu diễn PSK dưới dạng lược đồ pha như hình 2.21(c). Một lược đồ pha biểu diễn sóng mang bằng đường thẳng đơn gọi là véc tơ, chiều dài của véc tơ biểu diễn biên độ sóng. Véc tơ quay xung quanh gốc theo chiều ngược kim đồng hồ với tần số góc ω_c . Bit 1 được đặc trưng bởi véc tơ cùng pha với sóng mang trong khi bit 0 nhanh pha hơn sóng mang 180° . Hai trục được gọi là I (in phase) và Q (quadrature).

(4). Điều chế đa mức

Như đã đề cập từ đầu, các thiết bị truyền dẫn và chuyển mạch kỹ thuật số đã và đang được dùng trong mạng PSTN hiện đại. Kết quả ứng dụng đó, tạo điều kiện đạt được tốc độ bit vượt xa tốc độ có được theo các phương pháp điều chế cơ bản nhờ sử dụng các phương pháp điều chế phức tạp hơn. Trong các phương pháp nhằm gia tăng tốc độ, tồn tại hai khuynh hướng: hoặc nhiều mức tín hiệu hoặc trộn lẫn các phương pháp điều chế cơ bản, đặc biệt là ASK và PSK.



Hình 2.23 Kỹ thuật điều chế đa mức :
(a) Dạng sóng của 4-PSK (b) Lược đồ 4-PSK;



Hình 2.23 (tiếp theo) Kỹ thuật điều chế đa mức : (c) Lược đồ 16-QAM.

Trong các ví dụ thảo luận phía trước thì tốc độ bit bằng với tốc độ baud. Tuy nhiên có thể thực hiện tốc độ bit cao hơn nhiều so với tốc độ baud. Có nghĩa là mỗi phần tử tín hiệu có thể đại diện cho 2 bit (4 giá trị) hay 3 bit (8 giá trị) của thông tin. Và do đó tốc độ bit sẽ gấp 2 hay 3 lần tốc độ baud. Ví dụ một modem PSK thực hiện thay đổi theo 4 pha khác nhau ($0^\circ, 90^\circ, 180^\circ, 270^\circ$) thay vì chỉ 2 pha. Điều đó cho phép mỗi thay đổi pha mang thông tin của 2 bit như hình 2.23. Vì dùng 4 pha nên còn gọi là QPSK hay 4-PSK.

Tốc độ bit sẽ đạt cao hơn bằng cách dùng 8 hay 16 pha. Tuy nhiên trong thực tế cần hạn chế số lượng pha có thể dùng vì sự khác biệt pha giảm xuống thì tiến trình có xu hướng gia tăng nhiều và pha dễ bị sai trong khi truyền. Do đó để gia tăng tốc độ bit thì dùng biên độ để phân biệt giữa các vec tơ, điều này cũng mang ý nghĩa tương tự sự khác biệt pha. Loại điều chế kết hợp ASK và PSK gọi là Quadrature Amplitude Modulation (QAM). Trên hình 2-23(c) là lược đồ pha có 16 phần tử tín hiệu được viết là 16-QAM, mỗi mức tín hiệu đại diện cho 4 bit thông tin.

Như trên hình đã chỉ ra, ta thấy điểm mạnh của lược đồ này là tránh lỗi do các trạng thái quá gần nhau. Do đó với lược đồ này 8 pha được dùng nhưng các mức biên độ liên quan đến hai pha kề nhau là khác nhau. Nói chung điều đó tạo cho hoạt động tại máy thu ít có nguy cơ dẫn đến lỗi nhưng lại có phần dư thừa ở đây bởi vì không phải tất cả 4 mức biên độ đều được dùng cho mỗi pha. Trong tất cả các lược đồ điều chế, trước khi thực hiện điều chế dòng bit được cho qua mạch xáo trộn *scrambler*, tại đây mạch sẽ đổi dòng bit thành thứ tự giả một cách ngẫu nhiên. Điều này có tác dụng làm giảm xác suất các bit liên tiếp trong tuần tự thuộc những vị trí bit kề nhau. Tại đầu thu sau khi thực hiện giải điều chế, dòng bit sẽ được chuyển qua mạch *descrambler* để phục hồi lại tuần tự bit nguyên thủy. Loại điều chế này được sử dụng trong modem V.29 đạt được tốc độ 9600bps.

Phương pháp khác dùng phiên 32 trạng thái pha và biên độ hỗn hợp, trong đó bit thứ 5 được dùng cho mục đích sửa lỗi và được tạo ra bằng cách dùng bộ mã hóa chập (convolutional encoder). Loại này được dùng trong modem V.32

đạt được tốc độ 9600bps, V.32bis đạt đến 14400 bps. Các tốc độ 19200, 24000, 28800 bps, và 56Kbps đã có trong các modem V.34/V-fast, V.90/X2, V.92.

5.2. Mạch thuê riêng kỹ thuật số

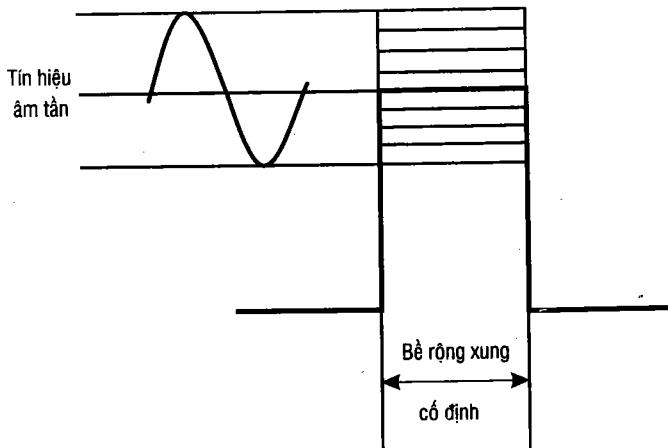
Các mạch thuê riêng kỹ thuật số không chỉ được dùng để kết nối trực tiếp giữa hai DTE mà còn dùng như là một mạng số liệu cơ bản. Tất cả các thông tin liên hệ đến cuộc gọi được truyền giữa các hệ thống chuyển mạch trong một mạng dưới dạng số. Mạng số còn được mở rộng đến từng nhà khách hàng và hình thành nên mạng số liên kết đa dịch vụ ISDN (Integrated Services Digital Network), vì thuê bao có thể truyền dữ liệu ngay tức thời không cần modem. Một ưu điểm của truyền dẫn số là có thể thuê các kênh truyền có tốc độ từ vài chục Kbps đến hàng trăm Mbps.

5.2.1. Kỹ thuật số hóa

5.2.1.1. Các loại điều chế và ứng dụng

Để cải thiện chất lượng truyền dẫn người ta lợi dụng sự chuyển đổi từ ghép kênh analog sang ghép kênh số. Việc khảo sát các phương pháp điều chế ra tín hiệu số nhằm xác định phương pháp nào tốt nhất trong mọi môi trường truyền là điều hết sức quan trọng. Có 3 phương pháp điều chế tín hiệu số cơ bản:

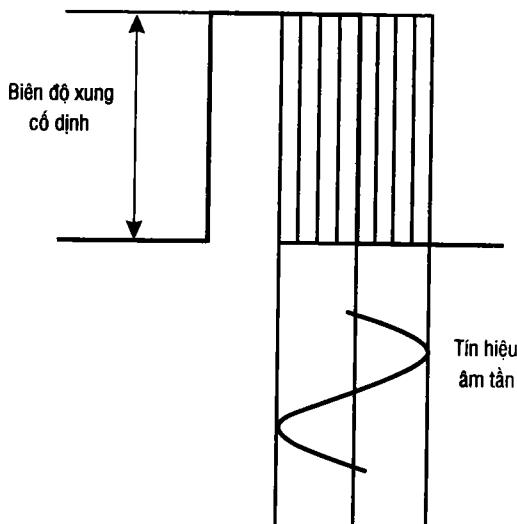
Điều chế biên độ xung PAM (Pulse Amplitude Modulation): Dạng điều chế này lấy xung có bề rộng cố định nhưng có chiều cao thay đổi theo tín hiệu điều chế. Điều chế biên độ xung rất nhạy cảm với nhiễu nhiệt và nhiễu xuyên âm, nhiễu sẽ tác động lên đỉnh xung và hình thành tạp âm trên kênh khi giải điều chế. Xem hình 2.24



Hình 2.24 Điều chế biên độ xung

Điều chế bề rộng xung (Pulse Width Modulation): Trong dạng điều chế này, giữ biên độ xung cố định, nhưng bề rộng xung thay đổi. Cảnh của xung dịch theo tín hiệu điều chế làm bề rộng xung thay đổi theo. Khi nhiễu làm thay đổi biên độ xung sẽ không ảnh hưởng đến tín hiệu gốc, vì biên độ xung không mang thông tin nào. Tuy vậy, nhiễu cũng làm xáo trộn cạnh của xung bởi hiện tượng méo dạng khi truyền.

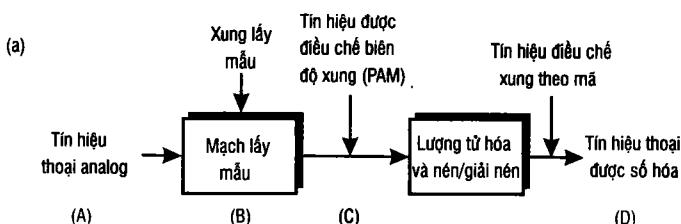
Phương pháp điều chế này có thể xem như tương đương với điều tần, chất lượng có tốt hơn điều biến, song cũng vướng phải một số khuyết điểm. Xem hình 2.25.



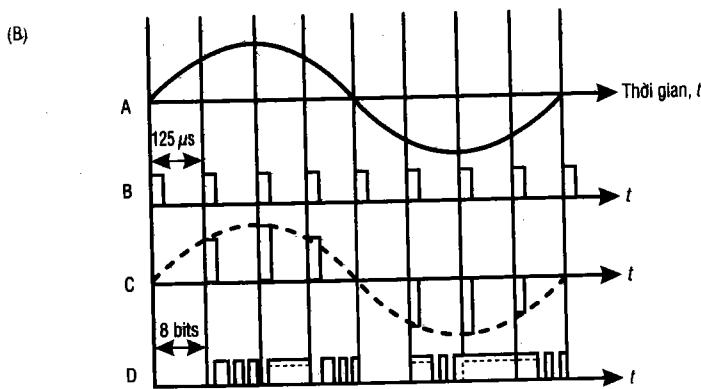
Hình 2.25 **Điều chế bề rộng xung.**

Điều chế xung theo mã PCM (Pulse Code Modulation): Phương pháp này thay thế trị số chính xác của mẫu bằng một chỉ số điện áp cơ bản, và chuyển đổi chỉ số này thành mã số nhị phân. Việc chuyển đổi tín hiệu thành mã nhị phân đã tạo ra một hệ thống thông tin không bị can nhiễu bởi môi trường. Phương pháp này sẽ được trình bày cụ thể hơn trong các mục tiếp theo đây.

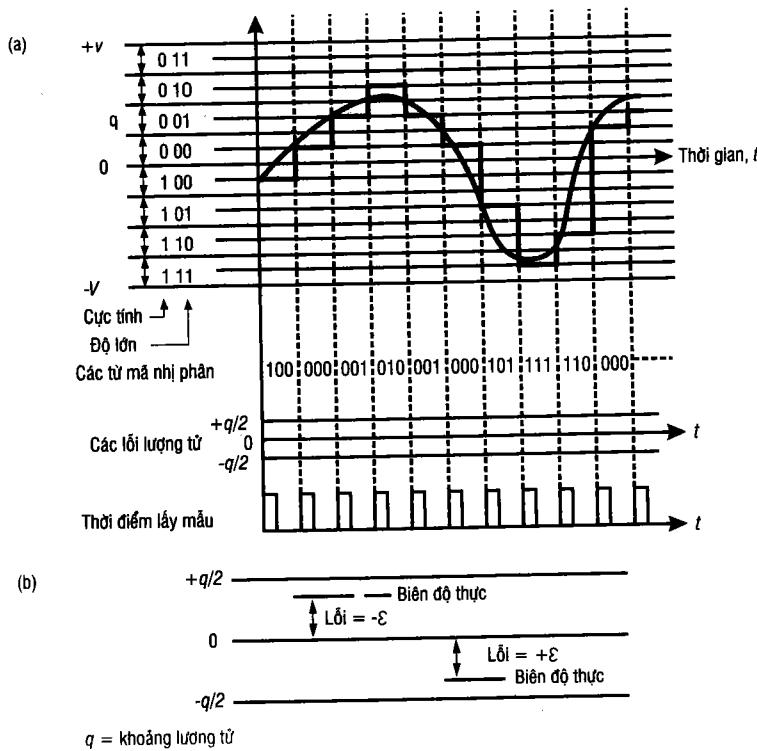
5.2.1.2. Sự số hóa tiếng nói



Hình 2.26 **Các nguyên lý số hóa:** (a) **Lược đồ mã hóa**



Hình 2.26 (tiếp theo) Các nguyên lý số hóa: (b) Các tín hiệu mã hóa.



Hình 2.27 Quá trình lượng tử hóa: (a) Nguồn gốc phát sinh lỗi (b) Dãy nhiễu lượng tử.

Các mạch số cũng liên quan đến nhu cầu truyền tiếng nói dưới dạng số. Như đã biết, tiếng nói có tần số nằm trong dải âm tần nhỏ hơn 4kHz. Để chuyển đổi các tín hiệu này sang dạng số, theo lý thuyết lấy mẫu Nyquist thì biên độ của tín hiệu phải được lấy mẫu với tốc độ tối thiểu lớn hơn hai lần thành phần tần số cao nhất của tín hiệu. Do đó, để chuyển đổi tín hiệu 4kHz sang dạng số cần tốc độ lấy mẫu là 8000 lần trong một giây. Hình 2.26 trình bày lược đồ tổng quát của sự lấy mẫu. Như chúng ta thấy, tín hiệu lấy mẫu trước hết được chuyển sang một dòng xung, biên độ của mỗi xung bằng với biên độ của tín hiệu analog gốc tại thời điểm lấy mẫu. Vì vậy các tín hiệu này được gọi là tín hiệu PAM (Pulse Amplitude Modulated). Tín hiệu PAM vẫn còn là analog, chúng sẽ được chuyển sang dạng số hoàn toàn bằng cách lượng tử hóa mỗi PAM theo mức tương ứng với một dạng nhị phân. 8 bit nhị phân được dùng để lượng tử hóa các tín hiệu PAM, trong đó có một bit chỉ dấu của tín hiệu (âm và dương). Điều này có nghĩa là có 256 mức khác nhau được dùng. Tín hiệu sau cùng được gọi là tín hiệu điều chế xung theo mã PCM (pulse code modulated signal) và có tốc độ bit là 64kbps (8000 mẫu trên một giây, mỗi mẫu 8 bit), đây là tốc độ tối thiểu có sẵn của các kênh số (digital channel).

Để biểu diễn biên độ các mẫu analog một cách chính xác cần một số vô hạn các ký số nhị phân. Do đó, nếu chỉ dùng 8 bit có nghĩa là mỗi mẫu chỉ được biểu diễn bởi một trong các mức rời rạc tương ứng. Sai biệt giữa các mức kề nhau được gọi là *khoảng cách lượng tử*, ký hiệu là q , nó xác định tính chính xác của quá trình lượng tử hóa. Chúng ta có thể thấy được điều này qua xem xét ví dụ lượng tử hóa ở hình 2.27 (a).

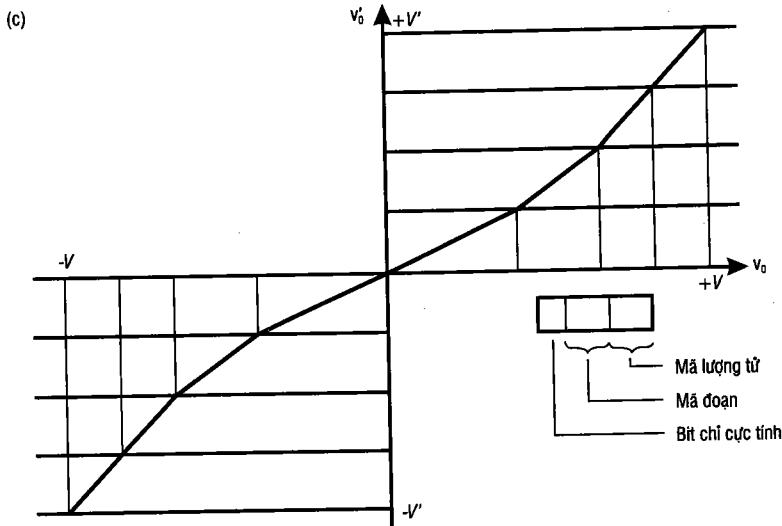
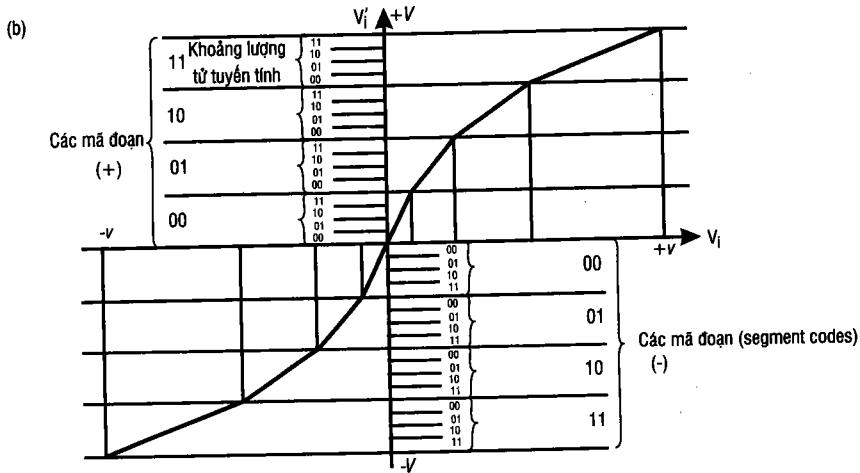
Trong ví dụ này, chúng ta dùng 3 ký số nhị phân để biểu diễn mỗi xung, bao gồm bit dấu, là kết quả trong 8 mức lượng tử hóa. Thông thường bit nhị phân 0 chỉ giá trị dương và bit nhị phân 1 chỉ giá trị âm. Các bit chỉ độ lớn được xác định bởi khoảng lượng tử đặc biệt mà tín hiệu analog thuộc về ngay tại thời điểm của mỗi mẫu.

Hình 2.27(a) mô tả bất kỳ một vị trí nào của tín hiệu ở trong cùng một khoảng lượng tử đều được biểu diễn cùng một từ mã nhị phân, mỗi một từ mã tương ứng với một điện áp nhập danh nghĩa ngay tại khoảng giữa của khoảng lượng tử tương ứng và điện áp nhập thực sự có thể sai biệt bằng một khoảng cộng hay trừ $q/2$. Sai biệt giữa biên độ tín hiệu thực sự và biên độ mẫu tương ứng được gọi là *sai số lượng tử* (*quantization error*) và giá trị có dấu của nó được miêu tả rõ hơn ở hình 2.27(b). Đối với tín hiệu đàm thoại, các giá trị sai số lượng tử thay đổi một cách ngẫu nhiên theo mẫu và được gọi là *nhiều lượng tử* (*quantization noise*).

Như chúng ta mới vừa đề cập, với các khoảng lượng tử tuyến tính, các tín hiệu có biên độ nhỏ sẽ chịu mức nhiễu lượng tử lớn hơn các tín hiệu có biên độ lớn. Tuy nhiên, tai của chúng ta nhạy cảm hơn đối với nhiều tác động vào các tín hiệu thoại có biên độ thấp so với nhiều tác động vào tín hiệu có biên độ cao. Để giảm bớt ảnh hưởng này, trong các hệ thống PCM thực tế, các khoảng lượng tử được thực hiện không tuyến tính bằng cách thay đổi dãy của biên độ tín hiệu nhập liên hệ với mỗi khoảng lượng tử, nghĩa là khi biên độ tín hiệu nhập gia tăng, thì các từ



mã tương ứng biểu diễn một dây tín hiệu lớn hơn. Nguyên lý của lược đồ được trình bày trên hình 2.28(a).



Hình 2.28

Các nguyên lý companding: (a) Lược đồ mạch (b) Lược đồ nén (c) Đặc tính giải nén.

Hình 2.28(a) cho biết rằng trước khi tín hiệu nhập được lấy mẫu và được chuyển sang dạng số, nó được chuyển qua một mạch được gọi là mạch nén (compressor). Tương tự, tại đích thủ tục ngược lại được thực hiện tại ngõ ra của bộ chuyển đổi số sang analog _DAC (Digital-to-Analog Converter) bởi một mạch gọi là mạch giải nén (expander). Toàn bộ tiến trình này vì thế được gọi là **companding**. Mỗi quan hệ xuất nhập của cả hai mạch này được trình bày trên hình 2.28(b) và (c); phần (b) được gọi là đặc tính nén và phần (c) được gọi là đặc tính giải nén.

Để mô tả nguyên lý, chúng ta chỉ dùng từ mã 5 bit. Tổ hợp 5 bit bao gồm một bit dấu, 2 bit mã đoạn (segment code) và 2 bit mã lượng tử. Các mã lượng tử và phân đoạn của một dãy tín hiệu hoàn chỉnh được trình bày như trên hình. Hoạt động số hóa được thực hiện qua 2 tầng như trên hình 2.28. Trong tầng chứa mạch nén, trước tiên tín hiệu nhập được nén một lượng được xác định bởi đoạn mà tín hiệu nhập rơi vào đó. Tín hiệu đã nén được đưa qua bộ chuyển đổi analog sang số _ADC (Analog-to-Digital Converter), tại đây đến lượt ADC thực hiện lượng tử hoá tuyến tính trên tín hiệu đã nén.

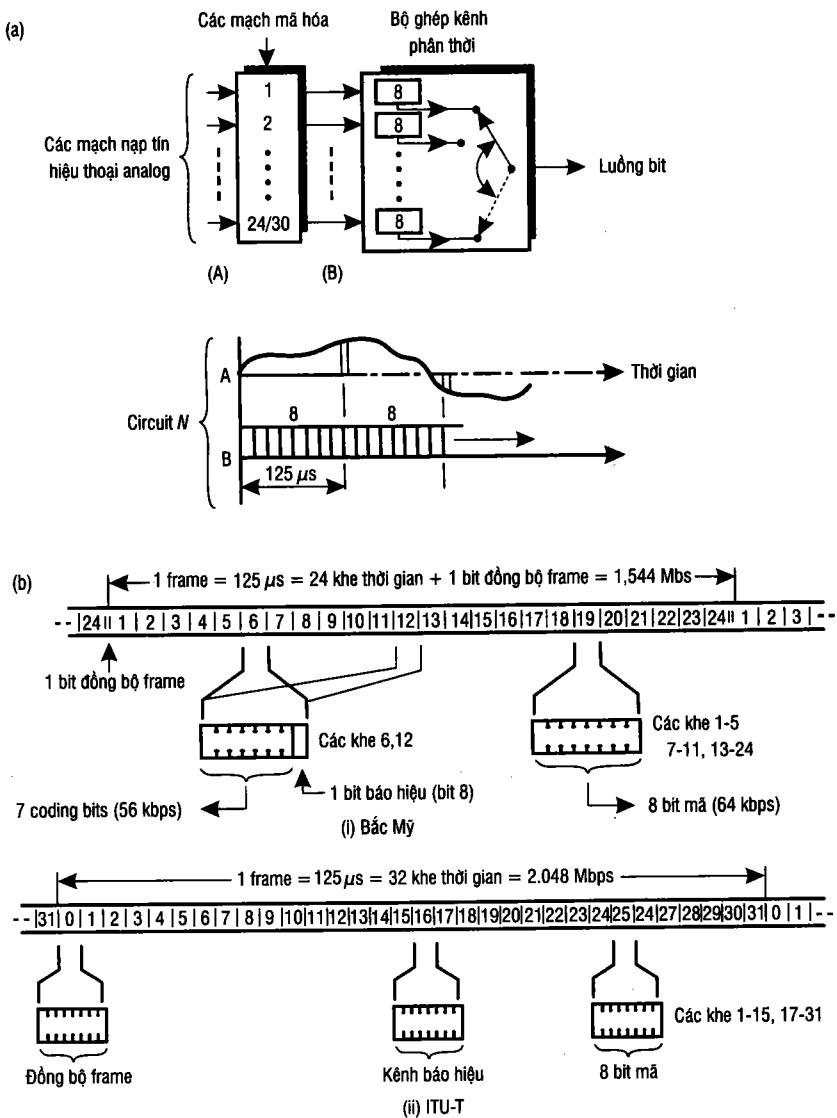
Tương tự, tại đích mỗi từ mã được chuyển đổi ngược lại dạng analog ban đầu dùng một quá trình hai tầng. Trước hết mỗi từ mã được nạp vào DAC tuyến tính và từ ngõ ra của DAC từ mã được chuyển qua mạch giải nén (expander), tại đây hoạt động ngược với mạch nén được thực hiện.

Trong thực tế, mặc dù các mã PCM đầu tiên hoạt động theo phương pháp này, nhưng hầu hết các mã PCM hiện nay đều thực hiện nén và giải nén số (digital). Có hai đặc tính nén khác nhau được dùng là: luật μ , được dùng rộng rãi ở Bắc Mỹ và Nhật bản và luật A là khuyến nghị của ITU-T. Do sử dụng hai luật khác nhau nên cần phải chuyển đổi cho các mạch nối giữa các châu lục khác nhau. May mắn thay, điều này chỉ cần thiết cho kênh thoại, còn đối với kênh số liệu thì không.

5.2.2. Kỹ thuật ghép kênh

5.2.2.1. Tổng quan

Các mạch liên kết giữa các tổng đài thực hiện nhiều cuộc gọi đồng thời dưới dạng số (digital) nhờ vào kỹ thuật ghép kênh phân thời TDM (Time Division Multiplexing). Với TDM, các tín hiệu số xuất phát từ nhiều nguồn khác nhau được gán vào các khe thời gian thích hợp trên liên kết tổng hợp tốc độ cao. Vì mỗi tín hiệu analog được lấy mẫu 8000 lần trong một giây, do đó sẽ tạo ra một mẫu 8 bit trong thời gian 125 micro-giây (μ s). Vì vậy, tốc độ bit trung bình của đường ghép kênh bằng tổng tốc độ của số kênh mà chúng chuyển tải. Ở Bắc Mỹ và Nhật có 24 kênh được nhóm lại với nhau, trong khi đó ở Châu Âu thì dùng 30 kênh. Tốc độ bit tương ứng lần lượt là 1,544Mbps, gọi là T1 và 2,048 Mbps, gọi là E1. Lược đồ tổng quát được trình bày trên hình 2.29 (a).



Hình 2.29 Phân cấp ghép kênh: (a) Lược đồ TDM (b) Hai dạng cấu trúc khung.

Cũng có thể thêm các bit (hay các kênh) cho các mục đích khác nhau (hình 2.29(b)), ví dụ như các bit chỉ bắt đầu của frame _đồng bộ frame_ và các bit cho mục đích thiết lập gọi (báo hiệu). Trong hệ thống ở Bắc Mỹ việc đồng bộ frame dùng một bit tại đầu của mỗi frame, bit này lật trạng thái giữa 1 và 0 cho

các frame liên tiếp nhau. Thông tin báo hiệu được tải trong bit đầu tiên của các khe 6 và 12 chỉ để lại 7 bit user trong các khe này. Do đó tốc độ bit trung bình là:

$$(24 \times 8 + 1) \text{bit} / 125\mu\text{s} = 1,544 \text{Mbps}$$

Mạch như vậy gọi là các liên kết DS1 hay T1.

Trong các hệ thống theo khuyến nghị của ITU-T, khe thời gian 0 (zero) được dùng cho sự đồng bộ frame (frame synchronization) _đôi khi cũng dùng thuật ngữ **frame alignment** vì nó cho phép máy thu dịch các khe thời gian trong mỗi frame theo các ranh giới đã được xếp hàng. Thông tin báo hiệu được tải trong khe thời gian 16 cho ra tốc độ bit trung bình là:

$$32 \times 8 \text{bit} / 125\mu\text{s} = 2,048 \text{Mbps}$$

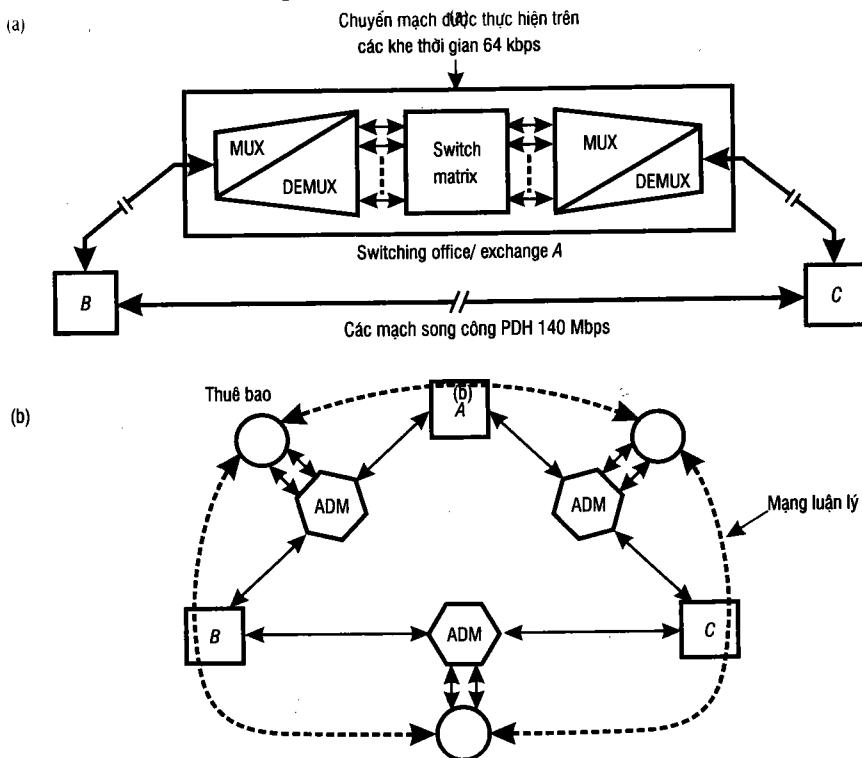
Các mạch như vậy được gọi là các liên kết E1. Trong cả hai hệ thống T1 và E1, các tốc độ bit nhỏ hơn được gọi là T1/E1 phân đoạn.

Bảng 2.1 Các cấp ghép kênh được dùng trong hệ thống truyền dẫn số của Bắc Mỹ và châu Âu (ITU-T)

	Mạch	Tốc độ bit (Mbps)	Số kênh thoại/dữ liệu
Bắc Mỹ	DS1	1,544	24
	DS1C	3,152	48
	DS2	6,312	96
	DS3	44,736	672
	DS4E	139,264	1920
	DS4	274,176	4032
Châu Âu	E1	2,048	30
	E2	8,448	120
	E3	34,368	480
	E4	139,264	1920
	E5	565,148	7680

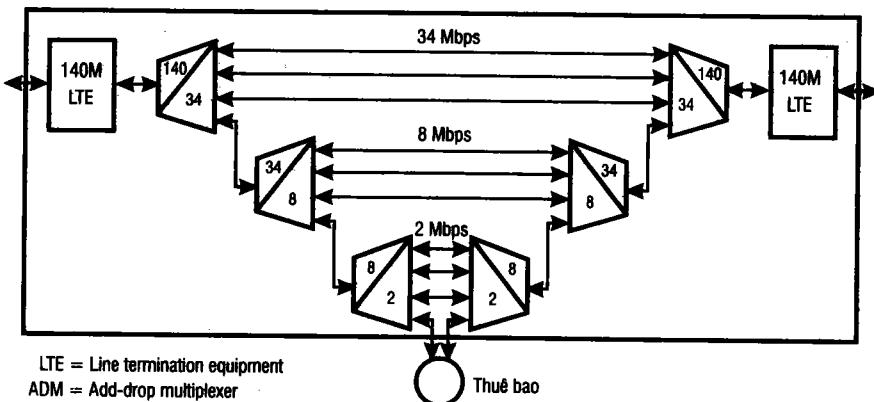
Chúng ta có thể ghép nhiều liên kết T1 (hay E1) lại với nhau để tạo ra các liên kết có tốc độ cao hơn. Bảng 2.1 trình bày dải tốc độ bit cho hai hệ thống, kèm theo tên gọi của nó. Các liên kết có tốc độ bit cao hơn yêu cầu các bit thêm vào để đồng bộ frame và điều khiển. Ví dụ $4 \times 2,048 = 8,192$ và do đó $0,256 \text{Mbps}$ được dùng cho các chức năng điều khiển. Các tốc độ bit khác nhau thường được viết gọn là 1,5; 3 ; 6 ; 44 ; 274 và 2 ; 8 ; 34 ; 140; 565. Tuy nhiên, việc ghép này làm nảy sinh một số vấn đề phức tạp. Hầu hết các hệ thống truyền dẫn số trước đây đều được xem như là các hệ thống analog được nâng cấp, kết quả là mỗi luồng TDM dùng một nguồn định thời riêng. Khi kết hợp hai hay nhiều luồng TDM lại với nhau cần phải cân chỉnh sự sai biệt định thời giữa chúng. Để khắc phục, dùng tốc độ bit ngõ ra lớn hơn một ít so với tổng tốc độ bit lối vào. Bất kỳ bit nào không dùng sẽ được làm đầy bằng các bit gọi là bit cân chỉnh. Loại ghép kênh tốc độ cao này

được gọi là kỹ thuật ghép kênh phân cấp số cân đồng bộ (PDH-Plesiochronous Digital Hierarchy). Mặc dù dùng bit cân chỉnh đối với bản thân nó không tạo ra vấn đề gì, nhưng sự hiện diện của nó không giúp xác định chính xác điểm bắt đầu của luồng bit tốc độ thấp trong luồng bit mức cao hơn. Dễ dàng thấy được ảnh hưởng này khi xem xét một nhu cầu hoạt động thông thường. Giả sử có 3 tổng đài được đặt ở 3 thành phố khác nhau được kết nối thông qua các mạch trung kế 140Mbps(PDH) được trình bày trên hình 2.30(a). Có một khách hàng có các cơ sở ở đâu đó giữa chúng yêu cầu liên kết các cơ sở này bằng các mạch thuê riêng 2Mbps để tạo ra mạng tư nhân. Điều này được trình bày dưới dạng lược đồ như hình 2.30(b). Vì không thể nhận dạng một kênh tốc độ thấp từ luồng bit tốc độ cao, nên người điều hành phải phân tách kênh đầy đủ cho luồng 140Mbps xuống mức 2Mbps trước khi có thể phân phối cho khách hàng. Luồng này phải được ghép trở lại thành luồng 140Mbps để truyền đi tiếp. Hoạt động ghép kênh/phân giải kênh kiểu này được thực hiện bởi thiết bị gọi là **drop-and-insert** hay **add-drop multiplexer** (ADM) và như chúng ta có thể thấy trên hình 2.30(c), thiết bị được yêu cầu cho nhu cầu khá đơn giản này lại rất phức tạp.



Hình 2.30 SỰ CUNG ỨNG CHO MẠNG TƯ NHÂN TRONG PDH:
(a) MẠNG ĐANG TỒN TẠI (b) MẠNG ĐƯỢC SỬA CHỮA

(c)



Hình 2.30 (tiếp theo) Sơ cung ứng cho mạng tư nhân trong PDH:
(c) Nguyên lý ADM

Mặc dù không được trình bày trên hình nhưng mỗi tổng đài được phân phối mạch thuê riêng 2Mbps phải được nhận dạng tương tự và cho qua để tạo nên liên kết trực tiếp giữa các cơ sở của khách hàng. Do đó, khi các mạch thuê riêng được cung cấp cho các khách hàng theo cách này, phải lưu giữ cẩn thận các record chỉ các mạch và thiết bị đang phục vụ cho mỗi khách hàng, đề phòng nếu có lỗi thì có thể xúc tiến các hành động phù hợp. Trong thực tế, chỉ giám sát phẩm chất cơ bản trong các dạng frame của PDH, nghĩa là thông thường khách hàng phải thông báo với nhà cung cấp các lỗi xảy ra.

Nhu cầu các mạch thuê riêng tốc độ cao ngày càng tăng khi xuất hiện các mạng mới tốc độ cao. Tự các nhà điều hành cũng đang phải cung cấp mạng truyền dẫn linh hoạt hơn khi các dịch vụ mới ra đời yêu cầu các phương tiện truyền dẫn giữa nhiều vị trí thay đổi khác nhau. Để đáp ứng các nhu cầu này một kiểu truyền dẫn mới hoàn chỉnh hơn khắc phục được tất cả các mặt hạn chế của PDH đã được công bố. Loại này được gọi là ghép kênh phân cấp số đồng bộ SDH (Synchronous Digital Hierachy). Hầu hết các mạch thuê riêng yêu cầu tốc độ ghép kênh mức cao hiện nay đều xuất phát từ các mạch SDH.

5.2.2.2.Ghép kênh phân cấp số đồng bộ SDH

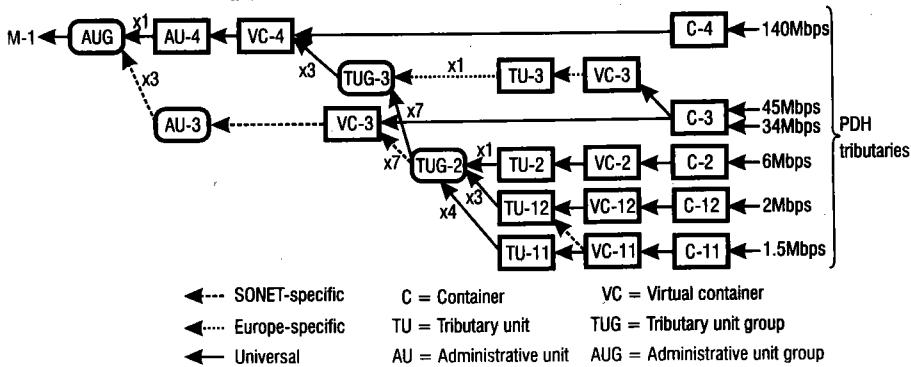
SDH được phát triển bởi Bellcore (USA) dưới tiêu đề SONET (Synchronous Optical Network). Tất cả các thiết bị được đồng bộ vào một đồng hồ chủ (master clock). Tốc độ truyền dẫn cơ bản được định nghĩa trong SDH là 155,52 Mbps thường gọi tắt là 155Mbps và được gọi là module truyền tải đồng bộ mức 1 (STM-1: Synchronous Transport Module level-1). Các tốc độ cao hơn có STM-4 (622Mbps) và STM-16(2,4Gbps). Trong phân cấp SONET thuật ngữ tín hiệu truyền tải đồng bộ STS (Synchronous Transport Signal) hay đôi khi còn gọi là tín hiệu quang OC (Optical signal) được dùng để định nghĩa tín hiệu tương đương STM. Trong SONET tốc độ 51,84Mbps hình thành nên mức tín hiệu đầu tiên STS-1/OC-



1. Tín hiệu STM-1 được tạo ra bằng cách ghép 3 tín hiệu như vậy lại với nhau và do đó tương đương với tín hiệu STS-3/OC-3.

Giống với PDH, tín hiệu STM-1 bao gồm một tập liên tục các frame 125μs. Nội dung thông tin của mỗi frame có thể được dùng để mang nhiều luồng PDH 1,5/2/6/34/45 hay 140Mbps. Mỗi luồng này được mang trong một đơn vị chứa (container) khác nhau, nó cũng chứa các bit nhồi thêm để cho phép các thay đổi theo tốc độ thực tế. Trong đó cũng được thêm vài thông tin điều khiển được gọi là **path overhead**, cho phép những chỉ tiêu như BER liên quan đến đơn vị chứa được giám sát từ đầu cuối này đến đầu cuối kia bởi quản lý mạng. Đơn vị chứa cùng với path overhead của nó hình thành nên một VC (virtual container) và một frame STM-1 có thể chứa nhiều VC cùng loại hay khác loại. Một số ví dụ ghép kênh dàn xen được trình bày trên hình 2.31. Lưu ý rằng ký số đầu tiên của đơn vị chứa mức thấp nhất và do đó là VC_là 1 và ký số thứ 2 chỉ định nó chứa tín hiệu PDH 1,5Mbps (1) hay 2Mbps (2).

SONET	SDH	Bit rate (Mbps)
STS-1/OC-1		51.84
STS-3/OC-3	STM-1	155.52
STS-9/OC-9		466.56
STS-12/OC-12	STM-4	622.08
STS-18/OC-18		933.12
STS-24/OC-24		1244.16
STS-36/OC-36		1866.24
STS-48/OC-48	STM-16	2488.32



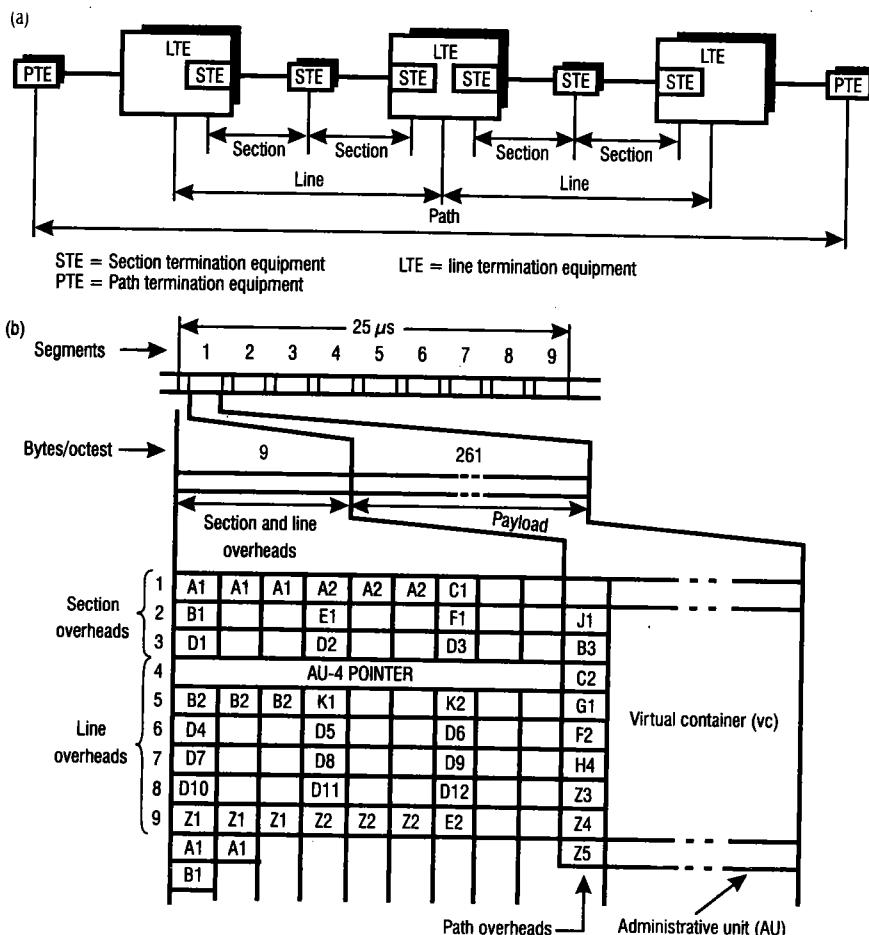
Hình 2.31 Phân cấp ghép kênh SDH/SONET và thuật ngữ.

Các tốc độ truyền dẫn bậc cao hơn được tạo ra bằng cách ghép nhiều STM-1 (STS-3/OC-3) lại với nhau. Ví dụ, một STM-16 (STS-48/OC-48) được tạo ra bằng cách ghép 16 STM-1 (STS-3/OC-3) hoặc ghép 4 STM-4 (STS-12/OC-12). Để tạo ra tính linh hoạt cần thiết cho mỗi tín hiệu bậc cao hơn, ngoài thông tin overhead đặt tại đầu của mỗi frame STM mức thấp hơn, còn có một con trỏ (pointer) được dùng để chỉ vị trí của frame STM mức thấp trong frame mức cao hơn.

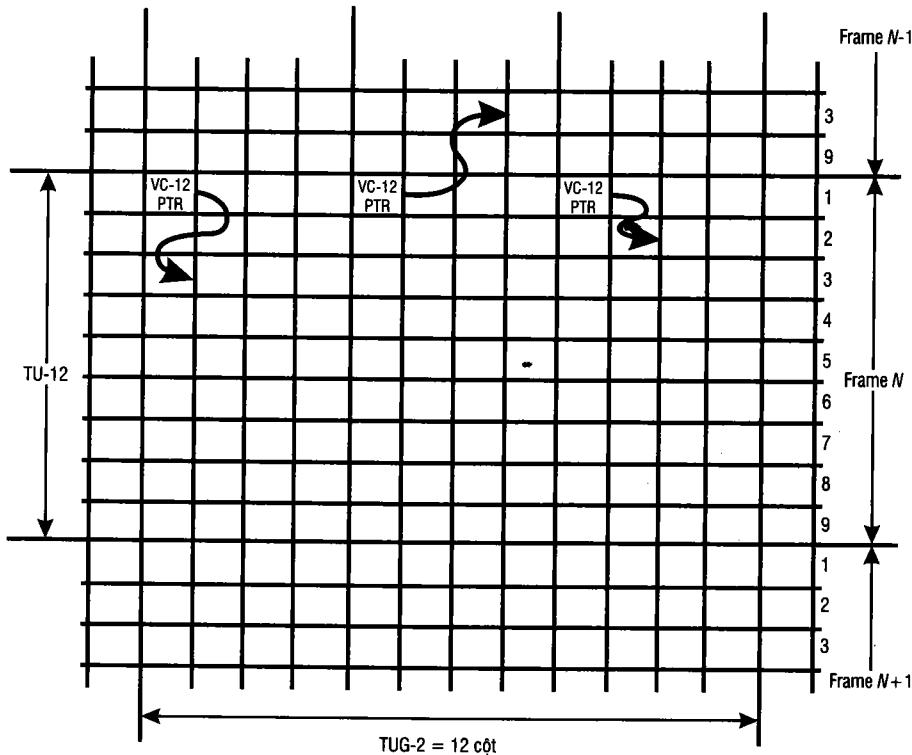
Chúng ta có thể mô tả cấu trúc của một frame SDH/SONET tốt nhất trong mối liên hệ với một hệ thống truyền dẫn đồng bộ hoàn chỉnh vì mỗi frame chứa

thông tin quản lý liên quan đến mỗi thành phần cấu thành. Các thành phần này bao gồm các đoạn (sections), các đường (lines) và các đường dẫn (paths) và các mối quan hệ giữa chúng được trình bày trên hình 2.32(a).

Một đoạn (section) là một chiều dài của một cáp truyền dẫn. Cả hai đầu cáp được kết thúc bằng một thiết bị kết cuối đoạn STE (Section Termination Equipment). Các repeater (bộ lặp) là ví dụ về STE, chúng tái sinh các tín hiệu quang/diện đang được truyền trên đoạn này. Một đường mở rộng qua nhiều đoạn cáp và được kết thúc bởi một thiết bị LTE (Line Termination Equipment). Các bộ ghép kênh và các node chuyển mạch là các ví dụ về LTE. Một đường dẫn là một đường truyền từ đầu cuối đến đầu cuối xuyên qua một hệ thống truyền hoàn chỉnh. Mỗi đầu của đường dẫn được kết thúc bởi PTE (Path Termination Equipment).



Hình 2.32 Chi tiết về SDH
(a) Các thực thể quản lý (b) Khuôn dạng của frame



Hình 2.32 (tiếp theo) Chi tiết về SDH (c) Ví dụ về ánh xạ VC.

Cấu trúc của một frame STM-1 (STS-3/OC-3) được trình bày trên hình 2.32(b). Mỗi frame bao gồm 2430 bytes/octets và lặp lại mỗi $125\mu s$ tạo ra tốc độ bit 155,52Mbps. Một frame bao gồm chín đoạn 270 byte, mỗi đoạn có một header 9 byte liên hệ với nó và một vùng chứa tải 261 byte. Các header còn được gọi là overhead. Thông thường, các phân đoạn tạo nên một frame được trình bày theo lối đoạn này nằm trên đoạn kia vì các byte đặc biệt trong mỗi header liên hệ đến một đoạn khác.

. Các byte overhead của đoạn liên quan đến sự quản lý của một đoạn đặc biệt. Như chúng ta thấy trong hình 2.32, một vài byte được sao ra vì các mục đích bảo vệ chống lỗi. Việc sử dụng các byte như sau:

A1-A2 Luôn là các byte được truyền trước tiên; được dùng để định dạng frame.

B1 Một byte kiểm tra chẵn lẻ được dùng để giám sát lỗi bit trên đoạn.

C1 Nhận dạng một frame STM-1 trong một frame bậc cao hơn (STM-n)

D1-D3 Hình thành một kênh truyền số liệu cho các thông điệp quản lý mạng liên quan đến đoạn này.

E1 Được dùng cho các kênh nghiệp vụ, là các kênh thoại được dùng bởi nhân viên bảo trì.

F1 Các kênh khách hàng, sẵn sàng cho quản lý thiết bị tại thuê bao.

Các byte overhead liên quan đến sự quản lý một đường hoàn chỉnh. Việc dùng mỗi byte như sau:

B2 Một byte kiểm tra chẵn lẻ được dùng để giám sát lỗi bit trên đường.

D4-D12 Hình thành một kênh truyền số liệu cho các thông điệp quản lý mạng liên hệ đến đường.

E2 Các kênh nghiệp vụ liên quan đến đường.

K1-K2 Hình thành kênh báo hiệu cho chuyển mạch bảo vệ tự động của một đường hoàn chỉnh.

Z1-Z2 Được dành riêng cho sử dụng trong quốc gia.

Các cột trong vùng chứa thông tin (payload field) có thể được gán theo nhiều cách khác nhau để mang các tín hiệu tốc độ thấp. Để vận chuyển các luồng PDH tốc độ thấp hơn thường gọi là các nhánh (tributary), dung lượng tải trong mỗi đơn vị tải được phân bổ thành một số cột. Mỗi đơn vị chứa có một cột chứa các byte thuộc path overhead được gán cho nó và tập hợp lại thành VC.

Sự nhận dạng mỗi byte được trình bày trên hình 2.32 và sử dụng chúng như sau:

J1 Byte này xác nhận kết nối đường dẫn VC

B3 Byte kiểm tra chẵn lẻ để giám sát tốc độ lỗi bit của đường dẫn

C2 Chỉ định thành phần của tải trong VC

G1 Được máy thu sử dụng để phản hồi trạng thái của tín hiệu thu về cho máy phát

F2 Cung cấp một kênh truyền số liệu cho user

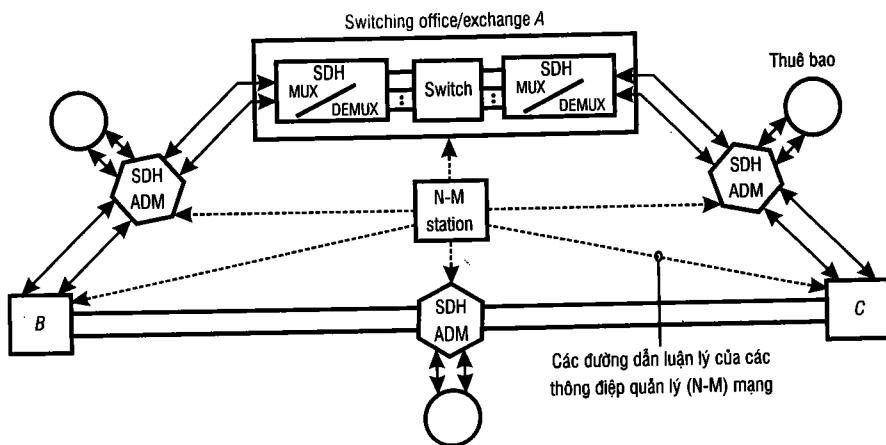
H4 Chỉ cho biết tải có phải là thành phần của một đa frame hay không

Z3_Z5 Dành cho sử dụng trong quốc gia

Con trỏ được đặt tại đầu của mỗi VC và được dùng để chỉ điểm bắt đầu của VC liên hệ với đầu của frame. Lưu ý rằng nếu đơn vị chứa chứa một luồng PDH mức thấp hơn thì giá trị con trỏ có thể thay đổi theo frame bởi các sai biệt định thời có thể. Các tổ hợp khác nhau của các VC có thể được dùng để làm đầy vùng tải của frame bằng cách cho các VC lớn mang các VC nhỏ hơn. VC và con trỏ của nó được gọi là các đơn vị nhánh TU (Tributary Unit) nếu VC mang một nhánh bậc thấp hơn hoặc được gọi là nhóm nhánh TUG (TribUtary Group) nếu nó mang một số nhánh bậc thấp hơn. VC lớn nhất trong frame STM-1 được gọi là

đơn vị quản lý AU(Administrative Unit), và có thể thấy trên hình 2.32(b), con trỏ chỉ vào đầu của nó được ghi vào vị trí của octet đầu tiên của overhead đường.

Ví dụ được trình bày trên hình 2.32(c) mô tả một frame TUG-2 mang 3 VC-12 như thế nào. Mỗi VC-12 gồm có 4 cột tải STM-1 và do đó TUG-2 bao gồm 12 cột. VC-12 và con trỏ của nó hình thành nên TU-12. Con trỏ luôn chiếm vị trí của byte đầu tiên, nhưng nếu định thời của nội dung VC-12 liên quan đến frame STM-1 thay đổi, thì vị trí của VC-12 được phép trượt theo để chứa nó, giá trị trong con trỏ thay đổi sao cho nó luôn chỉ đến byte đầu tiên trong VC. Một VC-12 sẽ chứa $4 \times 9 = 36$ byte. Do đó, vì một VC-12 gồm có 33 byte_32 byte cho frame E1 cộng với 1 byte con trỏ_các byte còn lại được làm đầy với những gì gọi là phần nhồi tạo kích thước cố định (fixed stuff).



Hình 2.33 Cung ứng dịch vụ với SDH dùng quản lý mạng

Để mang các dạng tín hiệu khác mà không có một đơn vị chứa nào được qui định để chấp nhận chúng, thì hai hay nhiều TU có thể được kết hợp lại dùng kỹ thuật gọi là **nối ghép** (concatenation). Ví dụ, 5 TU-2 có thể nối ghép để mang tín hiệu 32Mbps. Thay vì 3, 4 tín hiệu như vậy có thể được mang trong một VC-4 nếu dùng một C-3 chuẩn. Kỹ thuật này cũng được dùng để vận chuyển tải ATM. Các đơn vị tải của ATM được gọi là các cell và chúng ta sẽ khảo sát trong một chương phía sau.

Tất cả các thiết bị SDH đều có phần mềm đi kèm được gọi là chương trình đại diện quản lý mạng NM (Network Management) và các kênh thông tin trong các byte overhead được dùng bởi phần mềm để thông báo bất kỳ hoạt động sai lệch nào của các đoạn (sections), các đường(lines) hay các đường dẫn (paths) cho một trạm quản lý mạng trung tâm. Chúng cũng được trạm quản lý dùng để tải các lệnh nhằm thay đổi sự phân phối vùng tải liên quan đến mỗi frame STM-1. Ví dụ, các SDH ADM có thể được cấu hình (hay cấu hình lại) từ xa để cung cấp bất kỳ

một băng thông mong muốn nào mà không cần phải phân giải kênh. Nguyên lý tổng quát được trình bày trên hình 2.33. Các liên kết dự phòng được dùng giữa mỗi cặp ghép kênh SDH và chúng có thể được đưa vào sử dụng thông qua các chỉ thị từ một trạm quản lý mạng.

6. CÁC CHUẨN GIAO TIẾP VẬT LÝ

6.1. Giao tiếp EIA -232D/V24

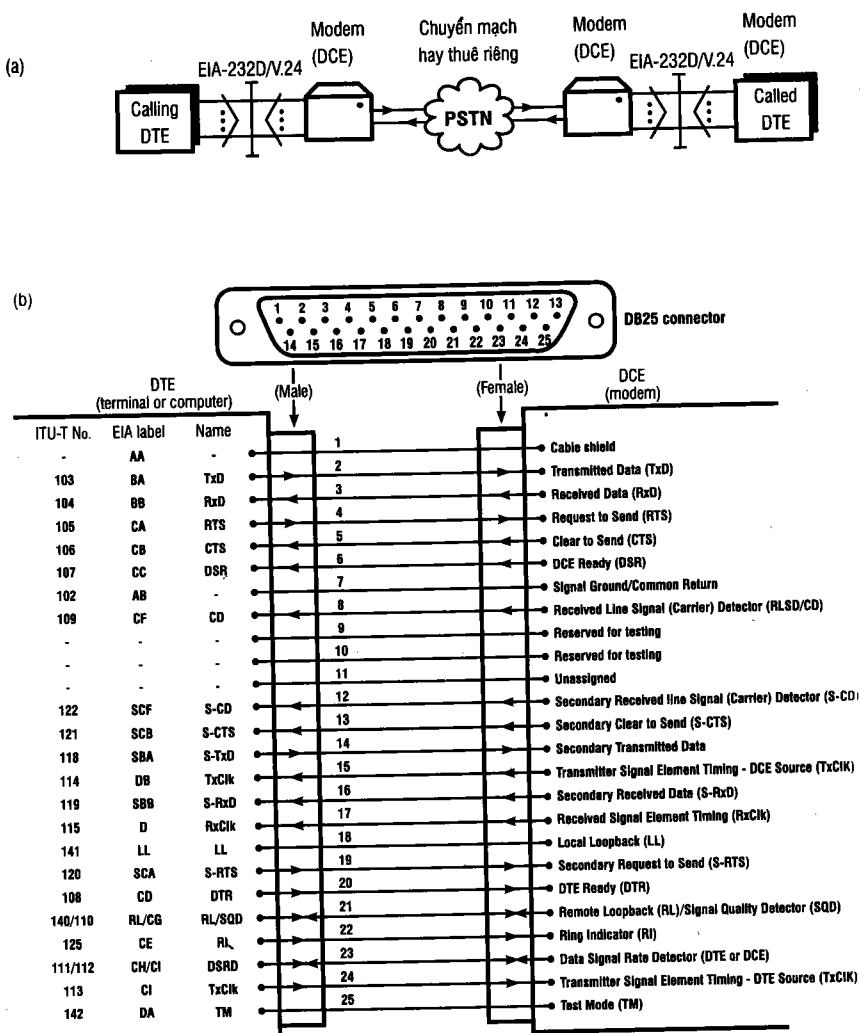
Giao tiếp EIA -232D/V24 được định nghĩa như là một giao tiếp chuẩn cho việc kết nối giữa DTE và modem. ITU-T gọi là V.24. Thông thường modem được đề cập đến như một DCE (Data Connect Equipment). Lược đồ hình thức ở hình 2.34(a) chỉ ra vị trí của giao tiếp trong kết nối điểm nối điểm giữa hai DTE (Data Terminal Equipment). Đầu nối giữa DTE và modem là đầu nối 25 chân như trên hình 2.34(b), nó được qui định trong chuẩn ISO 2110 và được gọi là bộ nối DB-25 (Data Bus-25). Trên hình cũng trình bày toàn bộ các tín hiệu liên quan đến giao tiếp.

Các đường dữ liệu truyền TxD (Transmitted Data) và dữ liệu nhận RxD (Received Data) là các đường được DTE dùng để truyền và nhận dữ liệu. Các đường khác thực hiện các chức năng định thời và điều khiển liên quan đến thiết lập, xóa cuộc nối qua PSTN (Public Switching Telephone Network) và các hoạt động kiểm thử tùy chọn.

Các tín hiệu định thời TxClk và RxClk có liên quan tới sự truyền và nhận của dữ liệu trên đường truyền nhận dữ liệu. Như đã biết, dữ liệu được truyền theo chế độ đồng bộ hoặc chế độ bất đồng bộ. Trong chế độ truyền bất đồng bộ cả hai đồng hồ truyền và thu đều được thực hiện độc lập ở hai đầu máy phát và máy thu. Trong chế độ này chỉ các đường dữ liệu truyền/nhận là được nối đến modem (và các đường điều khiển cần thiết khác). Các đường tín hiệu đồng hồ vì vậy không cần dùng và không nối đến modem. Tuy nhiên trong chế độ truyền đồng bộ, số liệu truyền và nhận được truyền nhận một cách đồng bộ với tín hiệu đồng hồ tương ứng và thường được tạo ra bởi modem. Các modem làm việc trong chế độ thứ hai này gọi là modem đồng bộ, khi tốc độ baud nhỏ hơn tốc độ bit thì các tín hiệu đồng hồ được tạo ra bởi modem hoạt động với một tần số thích hợp so với tốc độ thay đổi tín hiệu trên đường truyền.

Chúng ta sẽ dễ hiểu hơn về các đường điều khiển với các chức năng và tuần tự hoạt động của nó trong quá trình thiết lập hay xóa cuộc nối qua mạng điện thoại công cộng (PSTN). Hình 2.35 sẽ mô tả tiến trình một cuộc gọi qua bước thiết lập đầu tiên, rồi số liệu được trao đổi trong chế độ bán song công và sau cùng là cầu nối sẽ bị xóa. Giả sử DTE khởi xướng gọi là một máy tính cá nhân và modem của nó có dịch vụ gọi tự động. Các dịch vụ này được định nghĩa trong khuyến nghị V.25

Khi DTE sẵn sàng thực hiện yêu cầu truyền nhận dữ liệu, tín hiệu trên DTR được đặt ở mức tích cực, và modem nội bộ sẽ đáp ứng bằng tín hiệu tích cực được đặt trên DSR.



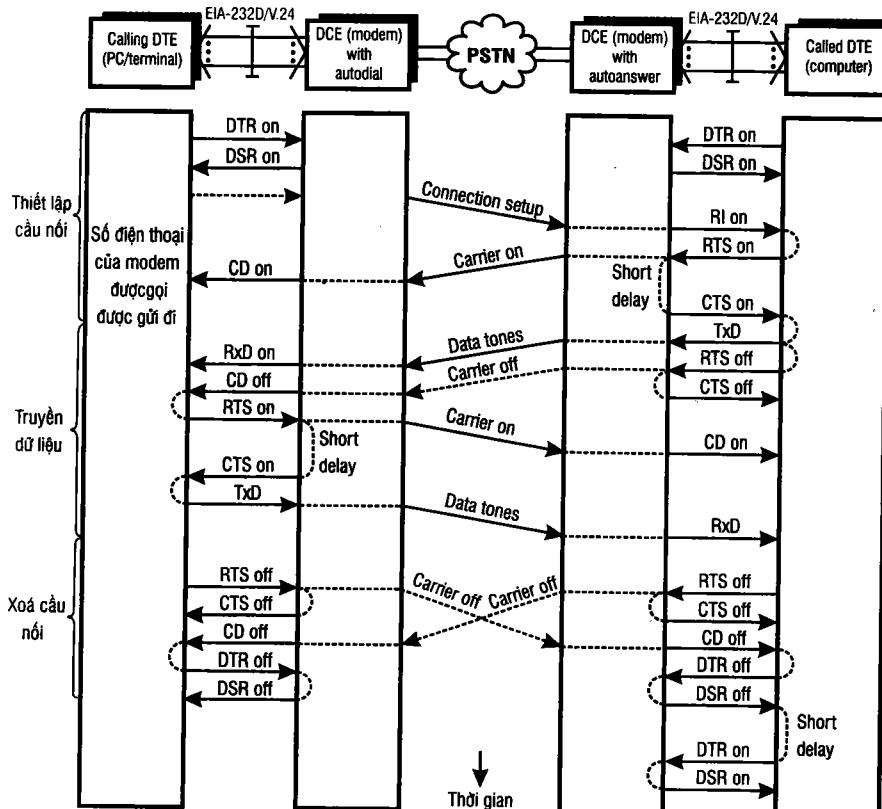
Hình 2.34 Chuẩn giao tiếp EIA -232D/V.24:

(a) *Chức năng giao tiếp* (b) *Socket, qui định chân và tín hiệu*.

Cuộc nối được thiết lập bởi DTE phát cuộc gọi gửi số điện thoại của đầu xa ra modem để thực hiện quay số (trường hợp quay qua PSTN) đến modem thu. Khi nhận được tín hiệu chuông từ tổng đài gọi đến, modem được gọi sẽ đặt RI lên mức tích cực và DTE được gọi đáp ứng lại bằng cách đặt RTS vào mức tích cực. Trong sự đáp ứng này modem được gọi đồng thời gửi sóng mang (âm hiệu dữ liệu của bit 1) đến modem gọi để báo rằng cuộc gọi đã được chấp nhận. Sau một thời khắc, gọi là thời gian trì hoãn, thời trễ này cho phép modem nơi gọi chuẩn bị nhận dữ liệu,

modem được gọi đặt CTS ở mức tích cực để thông báo cho DTE được gọi rằng nó có thể bắt đầu truyền số liệu. Khi phát hiện được sóng mang từ đầu xa gửi đến, modem gọi đặt CD ở mức tích cực, lúc này cầu nối đã được thiết lập, cung đoạn chuyển tin có thể bắt đầu.

14



Hình 2.35 EIA-232D/V.24: kết nối, truyền dữ liệu bán song công và tuần tự xóa cầu nối.

DTE được gọi bắt đầu với việc gửi một thông điệp ngắn mang tính thăm dò qua cầu nối. Khi thông điệp đã được gửi đi, nó lập tức chuẩn bị nhận đáp ứng từ DTE gọi bằng cách đặt RTS về mức không tích cực (off), phát hiện được điều này modem được gọi ngưng gửi tín hiệu sóng mang và trả CD về mức không tích cực. Ở phía gọi, modem gọi phát hiện sóng mang từ đầu xa đã mất, sẽ đáp ứng bằng cách trả CD về off. Để truyền thông điệp đáp ứng, DTE gọi đặt RTS lên mức tích cực và modem sẽ đáp ứng bằng mức tích cực trên CTS và bắt đầu truyền số liệu. Thủ tục này sau đó được lặp lại khi một bản tin được trao đổi giữa hai DTE.

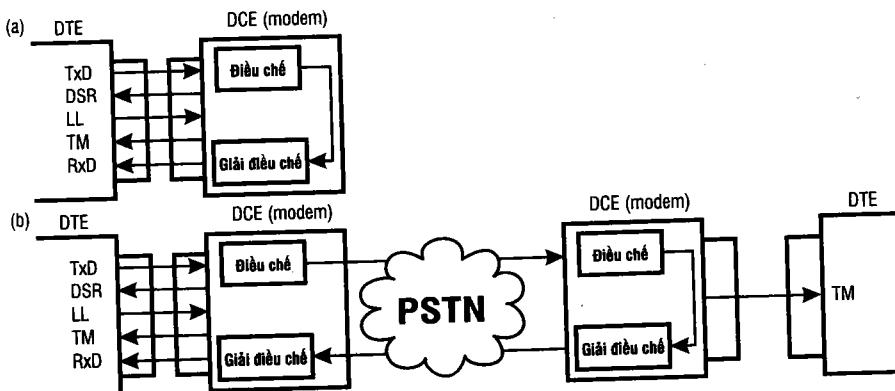
Cuối cùng, sau khi đã truyền xong cuộc gọi sẽ bị xóa, công việc này đều có thể thực hiện bởi cả hai DTE bằng cách đặt RTS của chúng về mức không tích cực, lần lượt khiến hai modem cắt sóng mang. Điều này được phát hiện ở cả hai

modem và chúng sẽ đặt CD về off. Cả hai DTE sau đó sẽ đặt DTR của chúng về off và hai modem sẽ đáp ứng với mức off trên DSR, do đó cầu nối bị xóa. Sau đó một khoảng thời gian, DTE được gọi chuẩn bị nhận cuộc gọi mới bằng cách đặt DTR lên mức tích cực.

Trên đây là mô tả hoạt động trao đổi số liệu theo chế độ bán song công để minh họa việc sử dụng các đường điều khiển của chuẩn. Trong thực tế sự trì hoãn trong chế độ bán song công làm cho ý nghĩa truyền nhận số liệu giảm sút, vì vậy thường phải dùng chế độ song công hoàn toàn. Trong chế độ này cả hai RTS được duy trì ở mức tích cực trên DTE và cả hai modem cũng duy trì mức tích cực trên CTS và phát sóng mang thường xuyên đến đầu xa.

Khi hai DTE đang truyền và một lỗi xảy ra, rất khó biết chắc nguyên nhân nào gây ra và nằm ở đâu trong số: modem nội bộ, modem đầu xa, đường truyền dẫn hay DTE ở xa. Để giúp nhận dạng nguyên nhân gây ra lỗi, giao tiếp còn cung cấp 3 đường điều khiển: LL (local loopback), RL (remote loopback) và TM (test mode). Chức năng của chúng được minh họa trên hình 2.36

Modem luôn đặt DSR ở mức tích cực khi sẵn sàng truyền nhận dữ liệu. Để thực hiện kiểm tra trên modem cục bộ, DTE sẽ đặt LL ở mức tích cực, ngay tức thì đáp ứng bên trong modem sẽ thực hiện một kết nối nội liền ngõ ra mạch điều chế với ngõ vào mạch giải điều chế. Sau đó nó sẽ đặt TM ở mức tích cực, khi DTE phát hiện được điều này nó sẽ truyền số liệu _ mẫu thử_ lên TxD, đồng thời nhận dữ liệu về từ RxD. Nếu số liệu nhận giống với số liệu truyền thì modem nội bộ hoạt động đạt yêu cầu, ngược lại modem nội bộ có vấn đề.



Hình 2.36 Kiểm thử : (a) Nội bộ; (b) đầu xa.

Nếu modem nội bộ coi như tốt, tiếp theo DTE tiến hành kiểm thử modem đầu xa bằng cách đặt RL ở mức tích cực, phát hiện được điều này modem nội bộ phát lệnh đã qui định trước đến modem đầu xa và tiến hành kiểm thử. Modem đầu xa sau đó đặt TM ở mức tích cực để báo DTE nội bộ biết đang bị kiểm thử (không thể truyền số liệu lúc này) và gửi trả lại một lệnh thông báo chấp nhận đến modem thử. Modem thử sau khi nhận lệnh đáp ứng sẽ đặt TM lên mức tích cực và DTE khi phát hiện điều này sẽ gửi mẫu thử. Nếu số liệu truyền và nhận như nhau thì cả hai modem hoạt động tốt và lỗi chỉ có thể ở DTE đầu xa. Nếu không có tín hiệu nhận được thì đường dây có vấn đề.

6.2. Modem rỗng (Null Modem)

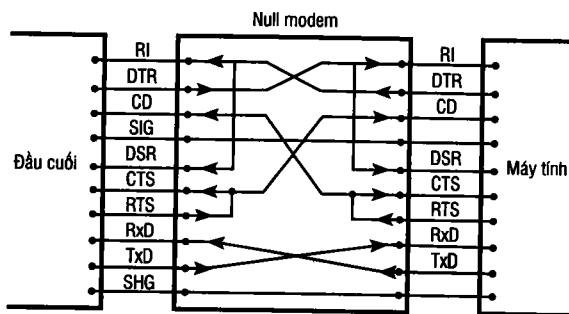
Với tín hiệu được phân bố như hình 2.37 thì cả truyền và nhận số liệu từ đầu cuối đến máy tính đều trên cùng một đường, vì modem có cùng chức năng ở hai phía. Tuy nhiên, theo định nghĩa nguyên thủy, chuẩn EIA-232D/V24 là giao tiếp chuẩn nối các thiết bị ngoại vi vào máy tính nên để dùng được cần quyết định thiết bị nào sẽ là máy tính và thiết bị nào sẽ là thiết bị ngoại vi, vì cả hai thiết bị không thể truyền và nhận số liệu trên cùng một đường dây. Có 3 khả năng lựa chọn :

(1) Đầu cuối mô phỏng modem và định nghĩa các đường một cách thích hợp để hoàn chỉnh hoạt động.

(2) Máy tính mô phỏng modem.

(3) Cá đầu cuối và máy tính đều không thay đổi và các dây dẫn được nối lại.

Bất tiện của hai lựa chọn đầu là không có đầu cuối nào hay máy tính nào có thể được dùng trực tiếp với một modem. Từ đó tiếp cận tổng quát cho vấn đề là bằng cách nối lại tín hiệu trên cổng giao tiếp EIA-232D/V24 để mô phỏng một modem, cho phép đầu cuối và máy tính nối trực tiếp vào modem. Lựa chọn thứ ba được dùng rộng rãi, yêu cầu một modem rỗng (null modem) chèn vào giữa đầu cuối và máy tính, các đường kết nối như mô tả ở hình 2.37.



Hình 2.37 Kết nối modem rỗng.

Như chúng ta đã thấy, các đường truyền nhận trao đổi với nhau từng đôi một, các đường điều khiển cũng được đổi lại. Ví dụ, vì thông thường đầu cuối và máy tính hoạt động ở chế độ song công hoàn toàn, các đường RTS và CTS được nối với nhau tại đầu đường dây và sau đó tín hiệu này được nối đến ngõ vào CD của thiết bị kia. Tương tự, DSR và RI được nối với nhau và nối đến ngõ vào DTR. Tín hiệu signal ground và shield ground được nối trực tiếp.

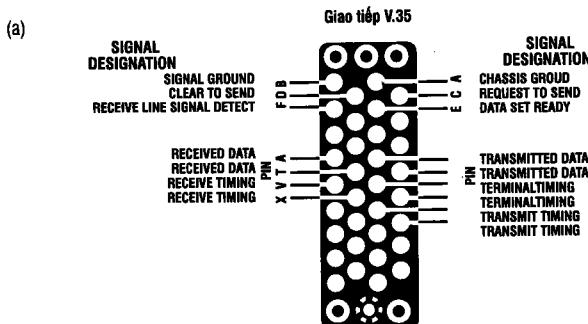
Khi hai thiết bị liên lạc với nhau qua một liên kết số liệu đồng bộ thì đồng hồ truyền từ mỗi thiết bị thường được nối đến và được dùng như đồng hồ thu tại thiết bị kia. Trong vài trường hợp không có thiết bị nào có đồng hồ và đồng hồ cho cả hai thiết bị được tạo ra trong modem rỗng, thành phần này gọi là bộ **modem eliminator**.

6.3. Giao tiếp EIA-530

Chuẩn EIA-530 là giao tiếp có tập tín hiệu giống giao tiếp EIA-232D/V24. Điều khác nhau là giao tiếp EIA-530 dùng các tín hiệu điện vi sai theo RS-422A/V11 để đạt được cự ly truyền xa hơn và tốc độ cao hơn. Dùng bộ nối 37 chân cùng với bộ nối tăng cường 9 chân nếu tập tín hiệu thứ hai cũng được dùng.



6.4. Giao tiếp EIA-430/V.35

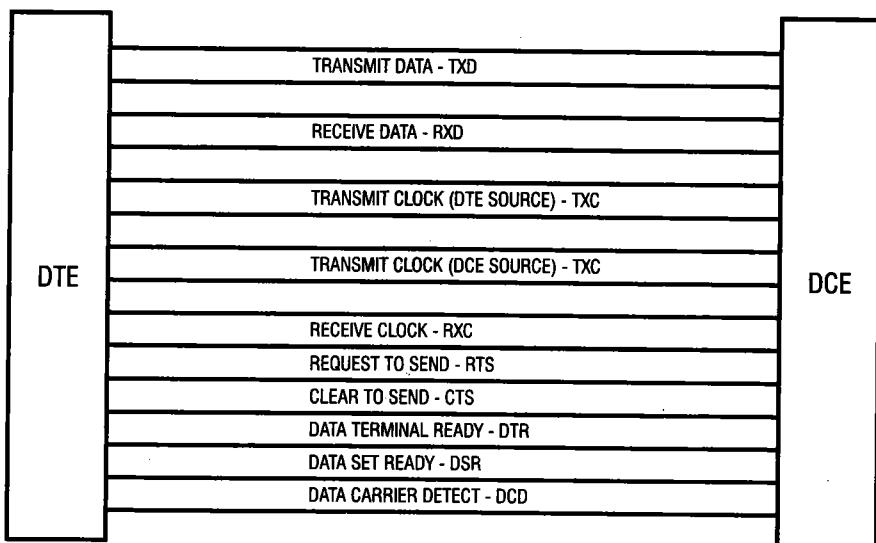


Chân	Tên	Đến DTE	Đến DCE	Chức năng
A	FG			Frame (or protective) ground
B	SG			Signal (or reference) ground
C	RTS	→		Request to send
D	CTS	←		Clear to send
E	DSR	←		Data set ready
F	RLSD	←		Received line signal
H	DTR	→		Data terminal ready
J	RI	←		Ring indicator
K	LT	→		Local test
R	RD	←		Received data (Sig. A)
T				Received data (Sig. B)
V	SCR	←		Serial clock receive (Sig. A)
T				Serial clock receive (Sig. B)
P	SD	→		Send data (Sig. A)
S				Send data (Sig. B)
U	SCTE	→		Serial clock xmit ext (Sig. A)
W				Serial clock xmit ext (Sig. B)
Y	SCT	←		Serial clock transmit (Sig. A)
a				Serial clock transmit (Sig. B)
h, i, j, k, m, n				Unused
L, M, N, Z, b, c, d, f, g				Unused

Hình 2.38**Giao tiếp V.35 (a) Các tín hiệu**

(b)

Giao tiếp DTE - DCE qua V.35



**Hình 2.38 (tiếp theo) Giao tiếp V.35
(b) Ứng dụng giao tiếp giữa DTE và DCE.**

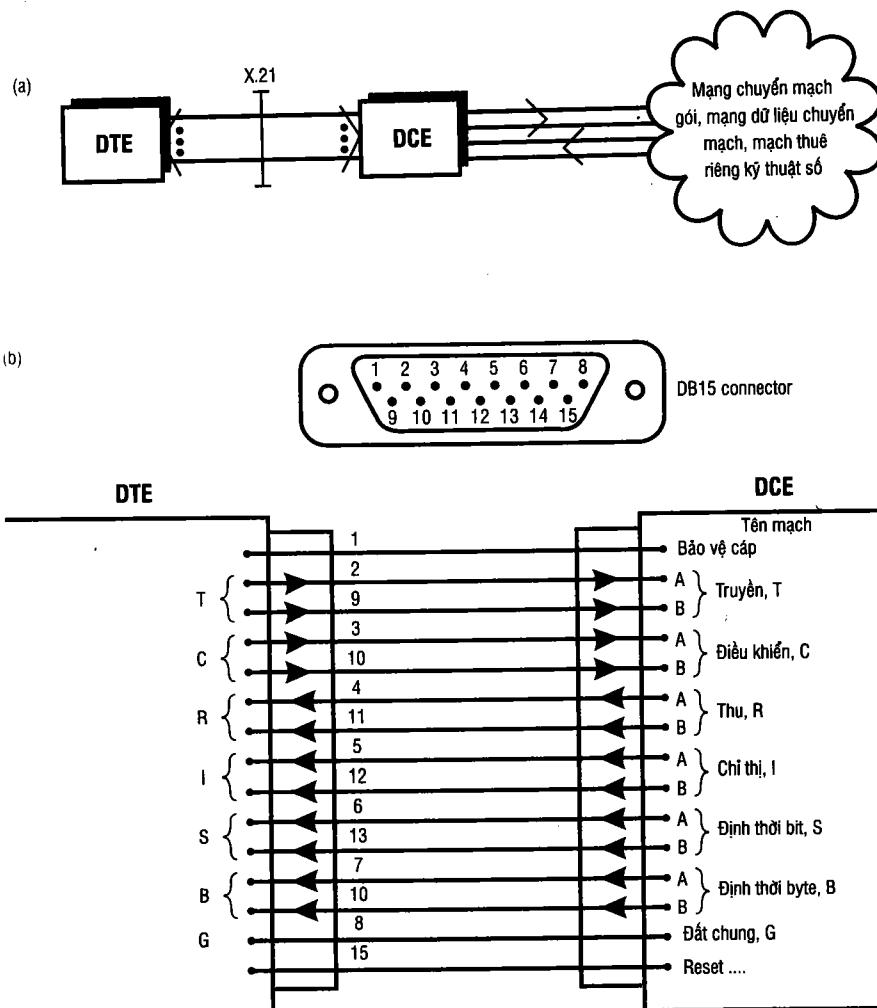
Giao tiếp EIA-430/V.35 được định nghĩa cho việc giao tiếp giữa một DTE với một modem đồng bộ bằng rộng hoạt động với tốc độ từ 48Kbps đến 168 Kbps. Giao tiếp này dùng tập tín hiệu giống với giao tiếp EIA-232D/V.24 ngoại trừ không có các đường thuộc kênh thứ hai hay kiểm thử. Các tín hiệu điện là một tập hợp theo lối không cân bằng (V28) và cân bằng (RS 422A/V11). Các đường tín hiệu không cân bằng dùng cho các chức năng điều khiển; các đường tín hiệu cân bằng dùng cho dữ liệu và tín hiệu đồng hồ. Vì tất cả các đường tín hiệu dữ liệu và đồng hồ là cân bằng nên trong các trường hợp truyền với đường cáp dài thường hay sử dụng các đường truyền nhận EIA-430/V.35. Giao tiếp EIA-430/V.35 dùng bộ nối 34 chân (xem hình 2.38), nhưng với các áp dụng chỉ dùng các đường truyền số liệu và đồng hồ thì có bộ kết nối nhỏ hơn. Các tín hiệu và giao tiếp của V.35 được mô tả trên hình 2.38.

6.5. Giao tiếp X21

Giao tiếp X21 được định nghĩa cho giao tiếp giữa một DTE với một DCE trong một mạng dữ liệu công cộng. Giao tiếp X21 cũng được dùng như một giao tiếp kết cuối cho các mạch thuê riêng số tốc độ là bội số của 64Kbps. Đầu nối và các đường tín hiệu được trình bày trên hình 2.39.

Tất cả các đường tín hiệu dùng bộ phát và thu cân bằng (RS-422A/V11). Là giao tiếp đồng bộ, bên cạnh cặp tín hiệu truyền (T) và nhận (R) còn có tín hiệu định thời phần tử bit (S) và định thời byte (B). Các tín hiệu điều khiển (C) và (I) được dùng kết hợp với các đường truyền và thu thiết lập nên cầu nối xuyên qua một mạng dữ liệu chuyển mạch số hóa hoàn toàn.

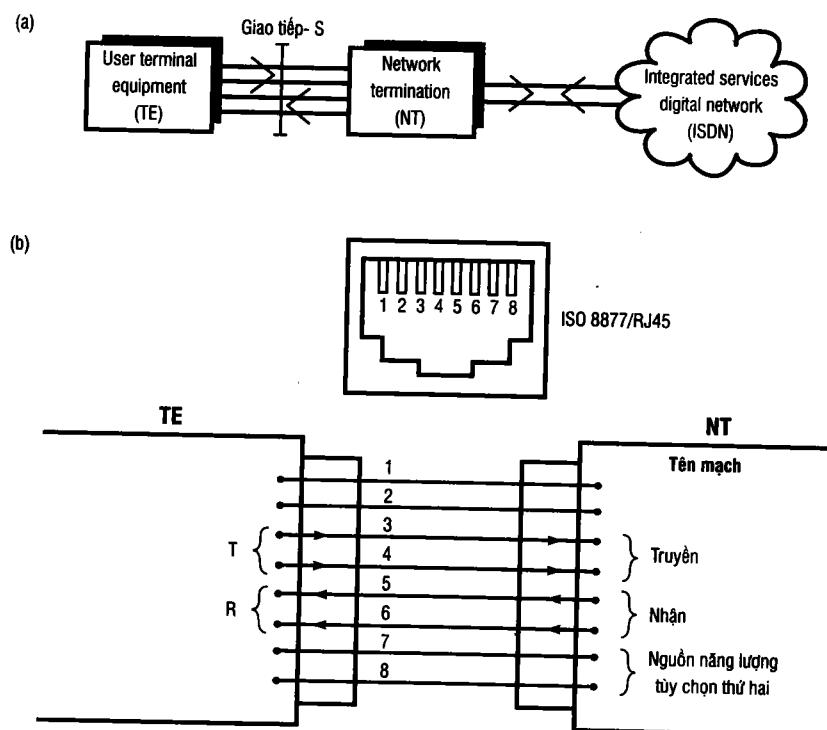
Một dẫn xuất của X21 được gọi là X21bis đôi khi cũng được dùng để giao tiếp một DTE được thiết kế dùng với một modem đồng bộ nối vào mạng dữ liệu công cộng. Dùng các đường điều khiển thông thường tương tự giao tiếp V.24/V.35.



Hình 2.39 Giao tiếp chuẩn X.21: (a) Chức năng giao tiếp (b) Các tín hiệu.

6.6. Giao tiếp ISDN

Giao tiếp ISDN là giao tiếp thay thế được số hóa hoàn toàn vào PSTN. Mạch thoại được số hóa hoạt động tại tốc độ 64kbps và một kết cuối tốc độ cơ bản cung cấp hai mạch như vậy cùng với một mạch 16Kbps cho mục đích thiết lập và xóa cuộc gọi. Ba mạch riêng biệt này được ghép kênh cho mục đích truyền đến và đi từ một tổng đài gần nhất lên một cặp dây. Thiết bị kết cuối mạng NT (Network Termination) tách biệt các đường dẫn đi và đến lên hai cặp dây riêng biệt. Năng lượng cũng có thể được cấp từ NT cho các DTE nếu có nhu cầu. Giao tiếp giữa user và NT trên hai cặp giây được gọi là giao tiếp S, xem hình 2.40. Nguồn năng lượng chính từ NT đến thiết bị đầu cuối được dẫn xuất từ các cặp truyền/nhận. Một nguồn năng lượng thứ hai cũng có sẵn qua chân 7 và 8. Nhằm kết nối thiết bị có tốc độ thấp vào giao tiếp S tốc độ cao này cần dùng thiết bị có tên là '**bộ thích nghi đầu cuối**' TA (Terminal Adapter).



Hình 2.40 Giao tiếp S - ISDN: (a) Chức năng (b) Các tín hiệu.



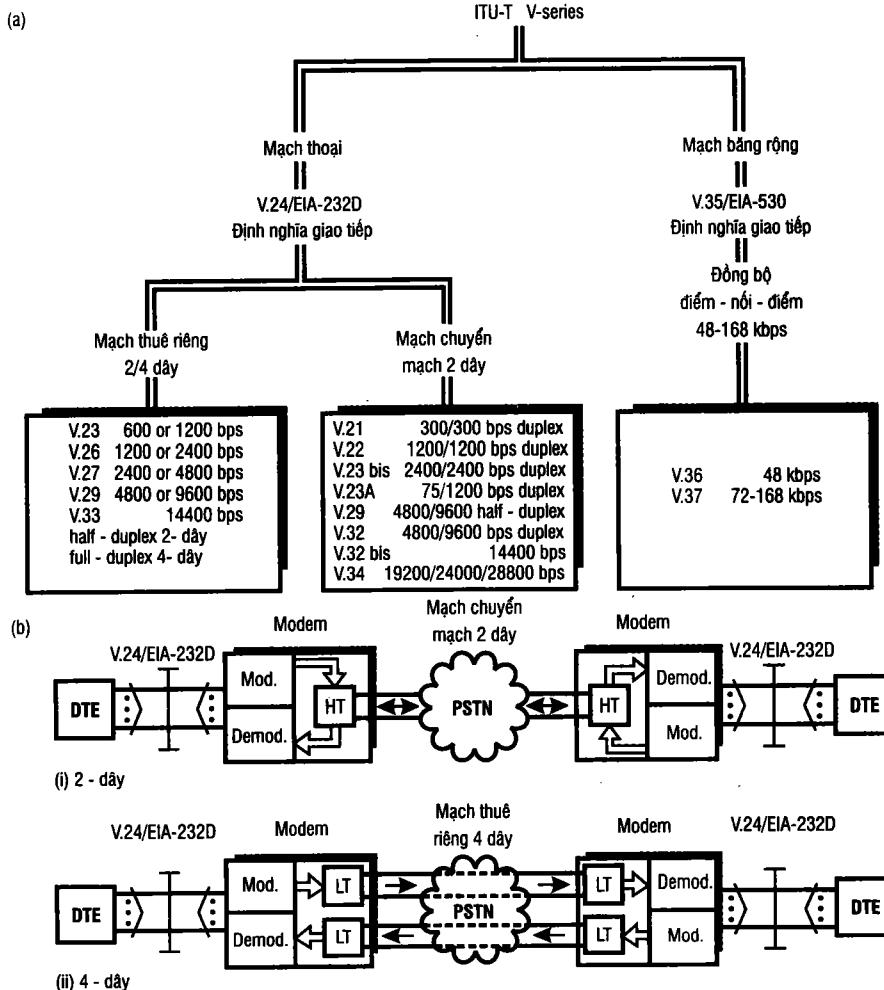
6.7. Tóm lược về giao tiếp

Các chuẩn đã trình bày ở trên chỉ là một phần của các chuẩn được định nghĩa bởi ITU-T dùng với mạng PSTN và thường được gọi là các chuẩn họ V (V-series). Một vài chuẩn được tóm tắt trên hình 2.41(a).

Hai chuẩn giao tiếp vật lý thông dụng là V.24 (EIA-232D) và V.35 (EIA-430). Dạng V.24 có khuynh hướng phổ dụng với PSTN và V.35 thì áp dụng cho các mạch băng rộng tốc độ cao hơn. Các chuẩn khác nhau được định nghĩa một cách cung nhắm và bao gồm sự xác định chính xác lược đồ điều chế phải dùng.

Một vài modem được thiết kế để dùng với hệ thống chuyển mạch 2 dây, trong khi có những loại modem có thể dùng với cả hệ thống chuyển mạch 2 dây và 4 dây (xem hình 2.41(b)). Các mạch cố định 4 dây được dùng cho các kết nối cố định giữa hai DTE, trong đó một cặp dây dùng để truyền và một cặp dùng để thu. Vì giá thành của dạng này khá cao nên chỉ được sử dụng trong các áp dụng có mật độ lưu lượng tải lớn.

Với các áp dụng có lưu lượng tải không cao đến mức phải dùng đến mạch cố định thì một kết nối chuyển mạch sẽ được sử dụng. Các kết nối như vậy chỉ dùng hệ thống 2 dây, và hoạt động song công đạt được bằng cách dùng biến áp sai động (hybird transformer). Tuy nhiên việc dùng biến áp sai động sẽ phát sinh vấn đề là một phần tín hiệu truyền phản hồi trở vào bộ giải điều chế nội bộ, thường gọi là tín hiệu dội lại (echo signal). Do đó, trước đây mạch này chỉ thường dùng trong hoạt động bán song công. Ngày nay bằng cách dùng các kỹ thuật điều chế tinh vi hơn và thêm vào một mạch gọi là bộ triệt dội (echo canceler) người ta có thể dùng chế độ truyền song công hoàn toàn trên hệ thống 2 dây với tốc độ cao. Có thể thấy được điều đó trong các modem V.32/V.33, trong đó sẽ bao gồm các dịch vụ khắc phục lỗi.

**Hình 2.41 Tóm tắt về giao tiếp:****(a) Các chuẩn họ V của ITU-T (b) Các trường hợp dùng modem.**

CHƯƠNG 3

GIAO TIẾP KẾT NỐI SỐ LIỆU

1. CÁC KHÁI NIỆM CƠ BẢN VỀ TRUYỀN SỐ LIỆU

1.1. Các chế độ thông tin (Communication modes)

Khi một người đang diễn thuyết thì thông tin được truyền đi chỉ theo một chiều. Tuy nhiên, trong một cuộc đàm thoại giữa hai người thì thông điệp được trao đổi theo hai hướng. Các thông điệp này thường được trao đổi lần lượt nhưng cũng có thể xảy ra đồng thời. Tương tự, khi truyền số liệu giữa hai thiết bị, có thể dùng một trong 3 chế độ thông tin sau:

- **Đơn công (one way hay simplex):** được dùng khi dữ liệu được truyền chỉ theo một hướng, ví dụ trong một hệ thống thu thập số liệu định kỳ.
- **Bán song công (either way hay half-duplex):** được dùng khi hai thiết bị kết nối với nhau muốn trao đổi thông tin một cách luân phiên, ví dụ một thiết bị chỉ gửi dữ liệu đáp lại khi đáp ứng một yêu cầu từ thiết bị kia. Rõ ràng hai thiết bị phải có thể chuyển đổi qua lại giữa truyền và nhận sau mỗi lần truyền.
- **Song công hoàn toàn (both way hay full-duplex):** được dùng khi số liệu được trao đổi giữa hai thiết bị theo cả hai hướng một cách đồng thời.

1.2. Các chế độ truyền (Transmission modes)

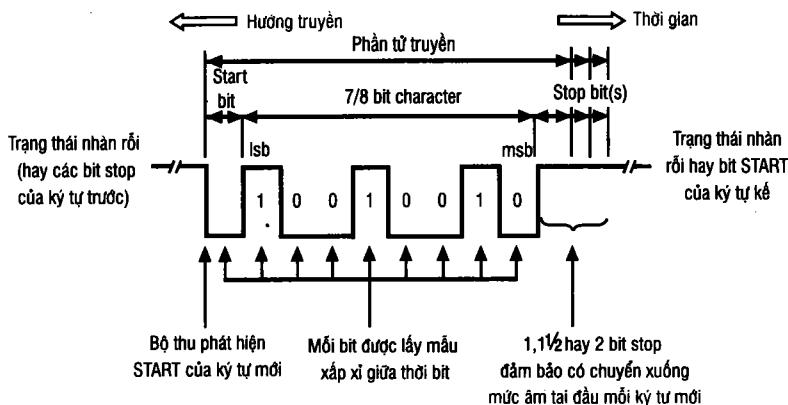
- Truyền bất đồng bộ (asynchronous transmission)

Cách thức truyền trong đó các ký tự dữ liệu mã hóa thông tin được truyền đi tại những thời điểm khác nhau mà khoảng thời gian nối tiếp giữa hai ký tự không cần thiết phải là một giá trị cố định.

Ở chế độ truyền này hiểu theo bản chất truyền tín hiệu số thì máy phát và máy thu độc lập nhau trong việc sử dụng đồng hồ _đồng hồ chính là bộ phát xung clock cho việc dịch bit dữ liệu (shift) _và như vậy không cần kênh truyền tín hiệu đồng hồ giữa hai đầu phát và thu. Tất nhiên, để có thể nhận được dữ liệu, máy thu buộc phải đồng bộ theo từng ký tự một.

Phương pháp truyền này thường được dùng khi truyền dạng dữ liệu phát sinh theo những khoảng thời gian ngẫu nhiên, ví dụ : user nhập ký tự từ bàn phím vào máy tính. Rõ ràng, với dạng thông tin này, user gõ bàn phím theo một tốc độ không xác định và khoảng thời gian giữa hai ký tự nhập thành công là ngẫu nhiên. Điều này có nghĩa là tín hiệu trên đường dây sẽ ở trạng thái nhàn rỗi (marking) trong một khoảng thời gian dài giữa hai ký tự. Do vậy, với dạng thông

tin này buộc máy thu phải đồng bộ trở lại tại thời điểm đầu của mỗi ký tự mới đến. Để thực hiện điều này, trước khi truyền mỗi ký tự đều được đóng gói giữa một start bit và một hay nhiều stop bit, xem hình 3.1.



Hình 3.1 Cơ sở cho truyền bắt đồng bộ.

Mặc dù được dùng chủ yếu để truyền ký tự giữa một bàn phím và một máy tính, truyền bắt đồng bộ cũng còn được dùng để truyền các khối ký tự giữa hai máy tính. Trong trường hợp này, mỗi ký tự kế tiếp đi ngay sau stop bit của ký tự trước đó vì các ký tự trong một khối được truyền tức thời ngay sau ký tự trước mà không có khoảng thời gian trì hoãn nào giữa chúng.

Như trình bày trên hình 3.1 cực tính của start bit và stop bit là khác nhau. Sự khác biệt này nhằm đảm bảo có tối thiểu một sự chuyển trạng thái ($1 \rightarrow 0 \rightarrow 1$) trong khoảng mỗi ký tự kế tiếp, bất chấp tuần tự của các bit truyền như thế nào. Sự chuyển trạng thái từ 1 đến 0 đầu tiên sau một chu kỳ nhàn rỗi được máy thu xác nhận bắt đầu của một ký tự mới. Ngoài ra, bằng cách dùng một đồng hồ có tần số gấp N lần tần số truyền bit, máy thu có thể xác định (xấp xỉ khá tốt) trạng thái của mỗi bit truyền thông qua hoạt động lấy mẫu (gần đúng) tín hiệu đến tại trung tâm của thời bit. Điều này cũng được minh họa trong hình 3.1 và chúng ta sẽ xem xét chi tiết hơn trong một mục phía sau.

Có thể suy ra rằng mỗi đơn vị dữ liệu của user dùng 10 (1 start bit và 1 stop bit) hay 11 bit (1 start bit và 2 stop bit). Giả sử có một start bit và 2 stop bit trên một phần tử 8 bit và giả sử tốc độ truyền là 1200 bps thì tốc độ dữ liệu là $1200/11$ hay vào khoảng 110 byte trong một giây. Thực tế, tốc độ dữ liệu hiệu quả thấp hơn nhiều vì các lý do mà chúng ta sẽ bàn đến sau.

Khi xác định tốc độ truyền trên đường dây, người ta thường dùng thuật ngữ **baud**. Tuy nhiên, khi dùng chính xác thì thuật ngữ này chỉ ra số lần chuyển đổi trạng thái của tín hiệu trong một đơn vị thời gian. Do đó, nếu tín hiệu truyền chỉ tồn tại một trong hai trạng thái thì tốc độ baud và tốc độ bit/s là tương đương



nhau. Tuy nhiên, như đã trình bày trong chương trước, đôi khi tín hiệu của đường dây có thể tồn tại ở một trong nhiều hơn 2 trạng thái, và do đó mỗi sự chuyển tín hiệu trên đường dây có thể được dùng để truyền nhiều hơn một bit dữ liệu. Vì vậy, trong trường hợp này tốc độ baud và tốc độ bit/s không còn tương đương nhau nữa. Để tránh nhầm lẫn, chúng ta nên dùng thuật ngữ *tốc độ phát tín* để xác định số lần chuyển trạng thái tín hiệu trong một đơn vị thời gian và dùng thuật ngữ *tốc độ số liệu* (hay thông tin) để xác định số bit thực truyền trong một đơn vị thời gian.

Cuối cùng, khi đang truyền các khối ký tự, mỗi khối sẽ được đóng gói giữa hai ký tự điều khiển mở và đóng khống nhằm đạt được sự đồng bộ theo khống. Điều này đảm bảo rằng máy thu có thể nhận biết một khống ký tự đến khi phát hiện ra *ký tự mở* dẫn đầu sau một chu kỳ nhàn rồi, tương tự khi nhận một *ký tự đóng* máy thu biết rằng đã kết thúc khống ký tự. Một khống ký tự cũng được gọi là khung, có thể gọi theo từ tiếng Anh là frame.

- Truyền đồng bộ (synchronous transmission)

Cách thức truyền trong đó khoảng thời gian cho mỗi bit là như nhau, và trong hệ thống truyền ký tự khoảng thời gian tính từ bit cuối của ký tự này đến bit đầu của ký tự kế tiếp thì bằng không hoặc bằng bội số tổng thời gian cần thiết truyền hoàn chỉnh một ký tự.

Về góc độ truyền tín hiệu số thì máy phát và máy thu sử dụng một đồng hồ chung, nhờ đó máy thu có thể đồng bộ được với máy phát trong hoạt động dịch bit để thu dữ liệu. Như vậy, cần phải có kênh (cần hiểu hoặc là cặp dây dẫn hoặc là một kênh trên đường ghép kênh hay kênh do mã hóa) thứ hai cho tín hiệu đồng hồ chung.

Tuy nhiên, khi xét đến các mức thông tin cao hơn mức vật lý trong mô hình hệ thống mở thì việc đồng bộ này được thực hiện theo từng khống dữ liệu và đặc tính truyền đồng bộ hiểu theo nghĩa hẹp trong một khống.

Như chúng ta đã biết, truyền bất đồng bộ thường được sử dụng trong các trường hợp mà tốc độ phát sinh dữ liệu không xác định được hoặc trong các trường hợp truyền các khống dữ liệu tốc độ thấp. Trong truyền bất đồng bộ việc dùng các bit thêm vào để đồng bộ ký tự và cả việc dùng phương pháp đồng bộ bit không chắc chắn đều được chấp nhận. Tuy nhiên, khi muốn truyền các khống dữ liệu lớn hơn có tốc độ cao hơn, phải dùng đến truyền đồng bộ.

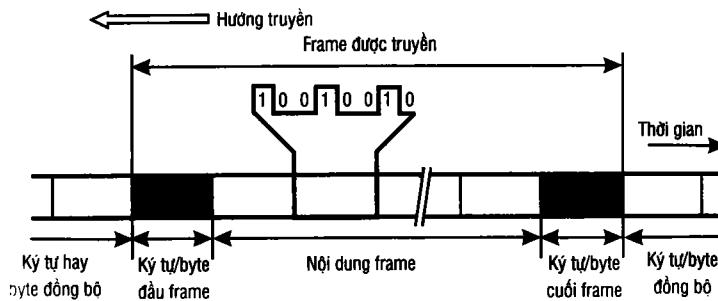
Với truyền đồng bộ, khống dữ liệu hoàn chỉnh được truyền như một luồng bit liên tục không có trì hoãn giữa mỗi phần tử 8 bit. Để cho phép thiết bị thu đạt được các mức đồng bộ khác nhau, cần có các đặc trưng sau :

- i. Luồng bit truyền được mã hóa một cách thích hợp để máy thu có thể duy trì trong cơ cấu đồng bộ bit.
- ii. Tất cả các frame được dẫn đầu bởi một hay nhiều byte điều khiển nhằm đảm bảo máy thu có thể dịch luồng bit đến theo các ranh giới byte hay ký tự một cách chính xác và tin cậy.

- iii. Nội dung của mỗi frame được đóng gói giữa một cặp ký tự điều khiển để đồng bộ frame.

Trong trường hợp truyền đồng bộ, khoảng thời gian giữa hai frame truyền liên tiếp có các byte nhàn rỗi được truyền liên tiếp để máy thu duy trì cơ cấu đồng bộ bit và đồng bộ byte hoặc mỗi frame được dẫn đầu bởi hai hay nhiều byte đồng bộ đặc biệt cho phép máy thu thực hiện tái đồng bộ. Điều này được minh họa trên hình 3.2.

Chúng ta sẽ bàn đến các phương pháp mã hóa bit được dùng để đồng bộ bit trong phần sau. Cũng giống như trong truyền bất đồng bộ, chúng ta cần đảm bảo rằng byte khởi đầu cũng như byte kết thúc frame phải là duy nhất cho chức năng này, do đó chúng nhất thiết không được xuất hiện như là một thành phần trong nội dung của dữ liệu frame. Rõ ràng, nếu nội dung của frame là các tập tin nhị phân thì điều này không thể đảm bảo và buộc chúng ta phải có giải pháp cho vấn đề này. Khía cạnh này sẽ được bàn đến trong các mục phía sau.



Hình 3.2 Cơ sở cho truyền đồng bộ.

1.3. Kiểm soát lỗi (error control)

Trong quá trình truyền luồng bit nối tiếp giữa hai DTE, rất thường xảy ra sự sai lạc thông tin, có nghĩa là mức tín hiệu tương ứng với bit 0 bị thay đổi làm cho máy thu dịch ra thành bit 1 và ngược lại, đặc biệt khi khoảng cách vật lý truyền khá xa ví dụ như dùng mạng PSTN để truyền. Vì thế, khi truyền số liệu giữa hai thiết bị cần phải có phương tiện phát hiện các lỗi có thể xảy ra và khi lỗi xảy ra nên có phương tiện sửa chữa chúng.

Chúng ta có thể dùng một số các lược đồ, nhưng việc chọn loại nào là tùy vào phương pháp truyền được dùng. Khi dùng phương pháp truyền bất đồng bộ, vì mỗi ký tự được chăm sóc như một thực thể riêng biệt, nên thường thêm một ký số nhị phân vào mỗi ký tự được truyền. Ký số nhị phân thêm vào này được gọi là bit chẵn lẻ – parity bit – chúng ta sẽ xem xét chi tiết trong phần sau.

Ngược lại, khi dùng phương pháp truyền đồng bộ, chúng ta thường xác định các lỗi xảy ra trên một frame hoàn chỉnh. Hơn thế nữa, nội dung của một frame có



thể rất lớn và xác suất nhiều hơn một bit lỗi gia tăng. Vì vậy cần dùng tuần tự kiểm tra lỗi phức tạp hơn. Cũng có một số dạng kiểm tra lỗi khác nhau, nhưng nhìn chung thiết bị sẽ tính toán ra tuần tự các ký số kiểm tra dựa vào nội dung của frame đang được truyền và gắn tuần tự này vào cuối của frame sau ký tự dữ liệu hay trước byte báo hiệu kết thúc frame.

Trong quá trình truyền frame, máy thu có thể tính toán lại một tuần tự kiểm tra mới dựa vào những gì nhận được từ frame hoàn chỉnh và so sánh với các ký số kiểm tra nhận được từ máy phát. Nếu hai chuỗi ký số này không giống nhau, coi như có một lỗi truyền xảy ra.

Cả hai lược đồ nói trên chỉ cho phép máy thu phát hiện lỗi truyền. Chúng ta cần máy thu lấy được một bản copy khác từ nguồn khi bản truyền bị lỗi. Có một số lược đồ cho phép điều này. Ví dụ xem xét trường hợp một đầu cuối và một máy tính truyền số liệu dùng phương pháp truyền bất đồng bộ. Khi user gõ vào bàn phím, ký tự đã mã hóa được truyền đến máy tính dưới dạng in được. Ngay sau đó, ký tự tương ứng với luồng bit vừa thu được máy tính dội trả lại (echo) đầu cuối và hiện lên màn hình. Nếu ký tự xuất hiện không giống như ký tự đã truyền trước đó, user có thể gửi một ký tự đặc biệt để thông báo với máy tính bỏ qua ký tự vừa nhận. Điều này được gọi là *kiểm soát lỗi*. Một phương thức có chức năng tương tự cũng phải được dùng khi truyền các khối ký tự. Chúng ta sẽ quay trở lại ở phần sau.

1.4. Điều khiển luồng (flow control)

Nếu số lượng dữ liệu truyền giữa hai thiết bị là nhỏ, thiết bị phát có thể truyền tất cả dữ liệu ngay tức thời vì máy thu có đủ tài nguyên để tiếp nhận dữ liệu. Tuy nhiên, trong nhiều tình huống truyền tin điều kiện này không thể có. Do đó chúng ta phải dùng một phương pháp điều khiển luồng dữ liệu để đảm bảo máy thu không bỏ qua bất kỳ phần dữ liệu nào do không đủ tài nguyên để lưu giữ. Điều này là hết sức quan trọng khi hai thiết bị đang truyền tin thông qua mạng số liệu, khi mà rất nhiều mạng sẽ đệm dữ liệu trong các bộ đệm có kích thước giới hạn. Nếu hai thiết bị hoạt động với tốc độ khác nhau, chúng ta thường phải điều khiển số liệu ngõ ra của thiết bị tốc độ cao hơn để ngăn chặn trường hợp tắc nghẽn trên mạng. Điều khiển luồng thông tin giữa hai thiết bị truyền thường được gọi ngắn tắt là *điều khiển luồng* (flow control). Chúng ta cũng sẽ quay trở lại trong một mục phía sau.

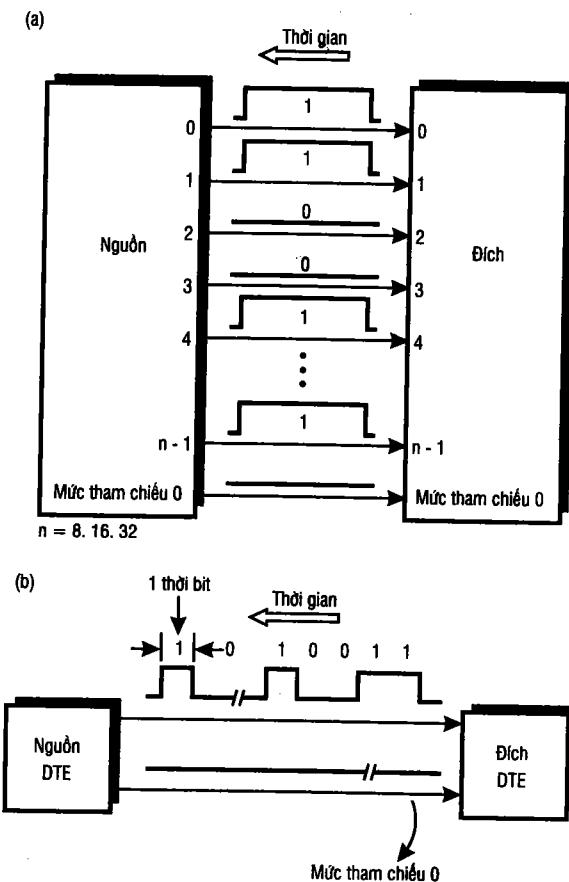
1.5. Các giao thức liên kết dữ liệu

Kiểm soát lỗi và điều khiển luồng là hai thành phần thiết yếu của một chủ đề tổng quát hơn đó là các *giao thức điều khiển liên kết số liệu*. Về cơ bản, một giao thức là một tập các tiêu chuẩn hay qui định phải tuân theo bởi cả hai đối tác truyền tin ở hai đầu, nhằm đảm bảo thông tin đang trao đổi xuyên qua một liên kết số liệu nối tiếp được tiếp nhận và được biên dịch ra một cách chính xác. Bên cạnh kiểm soát lỗi và điều khiển luồng, giao thức liên kết số liệu cũng định nghĩa những chi tiết sau:

- Khuôn dạng của mẫu số liệu đang trao đổi, nghĩa là số bit trên một phần tử thông tin và dạng lược đồ mã hóa đang được dùng.
- Dạng và thứ tự các thông điệp được trao đổi để đạt được độ tin cậy giữa hai đối tác truyền.

Ví dụ, thông thường trước khi truyền bất kỳ bản tin nào từ một đầu cuối đến đầu cuối khác, một cầu nối được thiết lập giữa chúng để đảm bảo rằng đầu cuối thu ở trong trạng thái sẵn sàng thu cho đầu cuối truyền này. Điều này được thực hiện bởi thiết bị truyền gởi cho thiết bị thu một thông điệp điều khiển đặc biệt_ví dụ call request_và máy thu sẽ đáp ứng bằng một thông điệp phúc đáp_ví dụ call accept hay reject. Chúng ta sẽ tìm hiểu kỹ hơn trong một mục phía sau.

1.6. Các hình thức truyền



Hình 3.3 Các hình thức truyền: (a) Song song (b) Nội tiếp



- Truyền song song

Là phương thức truyền trong đó các bit của một hay nhiều ký tự có thể được truyền đồng thời, và do đó tương ứng với một lần dịch bit là một hay nhiều ký tự được truyền. Mỗi bit của ký tự cần một kênh truyền (xem hình 3.3(a)). Ký tự được tạo ra trước sẽ được truyền trước.

- Truyền nối tiếp

Là phương thức truyền trong đó các bit dữ liệu từ một nguồn được truyền tuân tự nối tiếp nhau qua một kênh thông tin (xem hình 3.3(b)), ký tự được tạo ra trước trong khối dữ liệu thống nhất muốn truyền sẽ được truyền trước.

1.7. Mã truyền (transmission code)

Trong hệ thống thông tin số liệu, thường muốn truyền dòng các văn bản, các giá trị số, hình ảnh, âm thanh, .v.v. từ nơi này đến nơi khác. Các thông tin thì có nhiều dạng, tuy nhiên máy tính hay các thiết bị đầu cuối chỉ biết các bit 1 hay 0 vì chúng là các hệ thống nhị phân. Cần phải chuyển các thông tin về dạng nhị phân để thực hiện vấn đề phù hợp dữ liệu cho máy tính, đồng thời cũng cần phải có dấu hiệu nào đó cho phép con người hiểu được hay chuyển về dạng hiểu được khi nhận thông tin nhị phân. Nhu cầu này là nguyên nhân cho việc ra đời các bộ mã. Các bộ mã là tập hợp một số giới hạn các tổ hợp bit nhị phân, mỗi tổ hợp bit nhị phân mang ý nghĩa của một ký tự nào đó theo qui định của từng bộ mã. Số lượng bit nhị phân trong một tổ hợp bit nói lên qui mô của một bộ mã hay lượng ký tự chứa trong bộ mã. Nếu gọi n là số bit trong một tổ hợp bit thì số ký tự có thể mã hóa là 2^n . Có một số bộ mã thông dụng như Baudot, BCD, EBCDIC, ASCII.

Hai mã được sử dụng rộng rãi nhất là mã EBCDIC (Extended Binary Decimal Interchange Code) và mã ASCII (American Standards Committee for Information Interchange). EBCD là một mã 8 bit được dùng trong hầu hết các thiết bị của IBM. Các định nghĩa từ mã dùng với EBCDIC được trình bày trên hình 3.4 (a).

Mã ASCII giống như mã IA5 (International Alphabet Number 5) được định nghĩa bởi ITU-T và cũng được dùng bởi International Organization for Standardization gọi là ISO 645. ASCII nguyên là một mã 7 bit; các định nghĩa từ mã của ASCII được trình bày trên hình 3.4(b).

Hình 3.4 Các mã chuẩn: (a) EBCDIC (b) ASCII/IA5

Cá hai mã đều đáp ứng cho tất cả các ký tự thông thường, chữ số và các dấu chấm câu_tập hợp các ký tự này được gọi là ký tự in được (printable character).



cùng với một số các ký tự điều khiển (control character)_cũng được gọi là các ký tự không thể in (non-printable character). Ví dụ một vài ký tự điều khiển:

- Các ký tự điều khiển định dạng: BS (backspace), LF (line feed), CR(carriage return), SP (space), DEL (delete), ESC (escape), và FF (form feed).
- Các ký tự phân tách: FS (file separator) và RS (record separator).
- Các ký tự điều khiển truyền: SOH (start-of-heading), STX (start-of-text), ETX (end-of-text), ACK (acknowledge), NAK (negative acknowledge), và SYN (synchronous idle).

Mặc dù các mã này được dùng để xuất nhập, nhưng một khi dữ liệu được nhập vào trong máy tính nó được chuyển đổi và được lưu giữ dưới dạng số nhị phân tương đương có số bit cố định, thông thường 8, 16 hay 32 bit. Chúng ta gọi mẫu nhị phân 8 bit là một byte và mẫu dài hơn là từ. Vì một dãy bit được dùng để biểu diễn cho mỗi từ, nên thường dùng nhiều phần tử 8 bit khi truyền dữ liệu giữa hai DTE. Do đó trong vài trường hợp 8 bit được truyền qua một liên kết số liệu có thể đại diện cho một ký tự có thể in được mã hóa nhị phân (7 bit cộng với một bit kiểm tra) trong khi ở trường hợp khác nó có thể đại diện cho thành phần 8 bit của một giá trị lớn hơn. Trong trường hợp sau chúng ta xem phần tử như là byte hoặc như là octet cho các mục đích truyền tin.

1.8. Các đơn vị dữ liệu (data unit)

Theo đơn vị đo lường dung lượng thông tin thì đơn vị cơ bản byte, một byte là một tổ hợp gồm 8 bit (có giá trị bất kỳ_không được nhầm lẫn 1byte = 8bit).

$$1\text{Kb} = 2^{10} \text{ byte} = 1024 \text{ byte}$$

$$1\text{Mb} = 2^{10} \text{ Kb} = 1024 \text{ Kb}$$

$$1\text{Gb} = 2^{10} \text{ Mb} = 1024 \text{ Mb}$$

$$1\text{Tb} = 2^{10} \text{ Gb} = 1024 \text{ Gb}$$

Trong kỹ thuật truyền số liệu đôi khi xem các đơn vị dữ liệu truyền dưới dạng một ký tự hay một khối gồm nhiều ký tự. Việc nhóm các ký tự lại thành một khối gọi là đóng gói dữ liệu, và khối dữ liệu được xem như một đơn vị dữ liệu truyền trong một giao thức nào đó. Một khối dữ liệu như vậy được gọi là một gói (packet) hay một khung (frame).

1.9. Giao thức (protocol)

Giao thức truyền là tập hợp các qui định liên quan đến các yếu tố kỹ thuật truyền số liệu, cụ thể hóa các công tác cần thiết và qui trình thực hiện việc truyền nhận số liệu từ đầu đến cuối.

Tùy vào việc lựa chọn các giải pháp kỹ thuật và thiết kế qui trình làm việc mà sẽ có các giao thức khác nhau. Mỗi giao thức sẽ được sử dụng trong những trường hợp tương ứng với thiết kế của nó.

1.10. Hoạt động kết nối

Điểm nối điểm (point-to-point) là dạng kết nối trao đổi thông tin trong đó một đầu cuối số liệu chỉ làm việc với một đầu cuối khác tại một thời điểm.

Đa điểm (multipoint) là dạng kết nối trao đổi thông tin trong đó một đầu cuối số liệu có thể thông tin với nhiều đầu cuối khác một cách đồng thời.

1.11. Đường nối và liên kết

Đường nối là đường kết nối thực tế xuyên qua môi trường truyền, vì vậy nó là đối tượng truyền dẫn cố định mang tính vật lý.

Liên kết là kết nối giữa các đầu cuối dựa trên các đường nối và tồn tại trong một khoảng thời gian nhất định, mỗi đường nối có thể chứa nhiều liên kết, ngoài ra một liên kết có thể được kết hợp từ nhiều liên kết hay một liên kết có thể phân thành nhiều liên kết. Do đó liên kết là đối tượng truyền dẫn phụ thuộc mang tính luận lý.

2. THÔNG TIN NỐI TIẾP BẤT ĐỒNG BỘ

2.1. Khái quát

Như đã đề cập trong phần khái niệm, thông thường số liệu được truyền giữa hai DTE dưới dạng chuỗi liên tiếp các bit gồm nhiều phần tử 8-bit gọi là byte hay ký tự, dùng chế độ truyền hoặc đồng bộ hoặc bất đồng bộ. Trong các DTE, mỗi phần tử như vậy được lưu trữ, xử lý và truyền dưới dạng thức song song. Do đó, các mạch điều khiển truyền trong DTE hình thành nên giao tiếp giữa thiết bị và liên kết dữ liệu nối tiếp, và phải thực thi các chức năng sau:

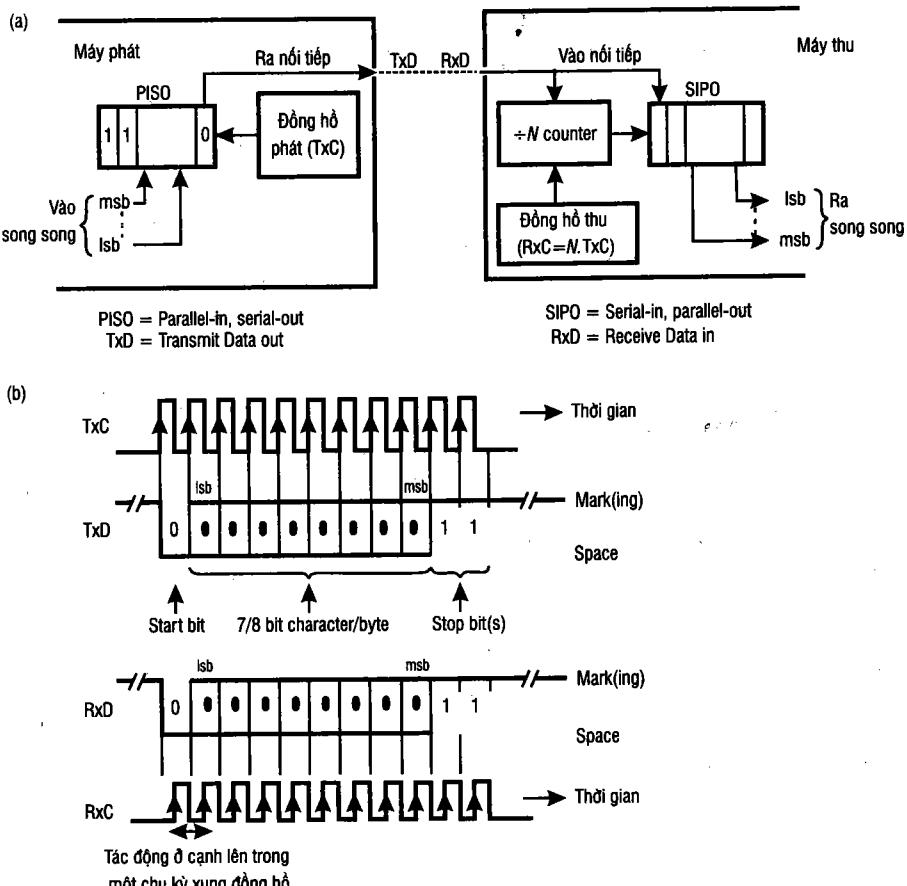
- Chuyển từ song song sang nối tiếp cho mỗi ký tự hay byte để chuẩn bị truyền chúng ra liên kết.
- Chuyển từ nối tiếp sang song song cho mỗi ký tự hay byte để chuẩn bị lưu trữ và xử lý bên trong thiết bị.
- Tại máy thu phải đạt được sự đồng bộ bit, byte và frame.
- Thực hiện cơ cấu phát sinh các ký số kiểm tra thích hợp để phát hiện lỗi và khả năng phát hiện lỗi ở máy thu phải khả thi.

Việc chuyển từ song song sang nối tiếp thực hiện bởi thanh ghi PISO (Parallel Input Serial Out) và việc chuyển ngược lại do SIPO (Serial Input Parallel Output).

2.2. Nguyên tắc đồng bộ bit

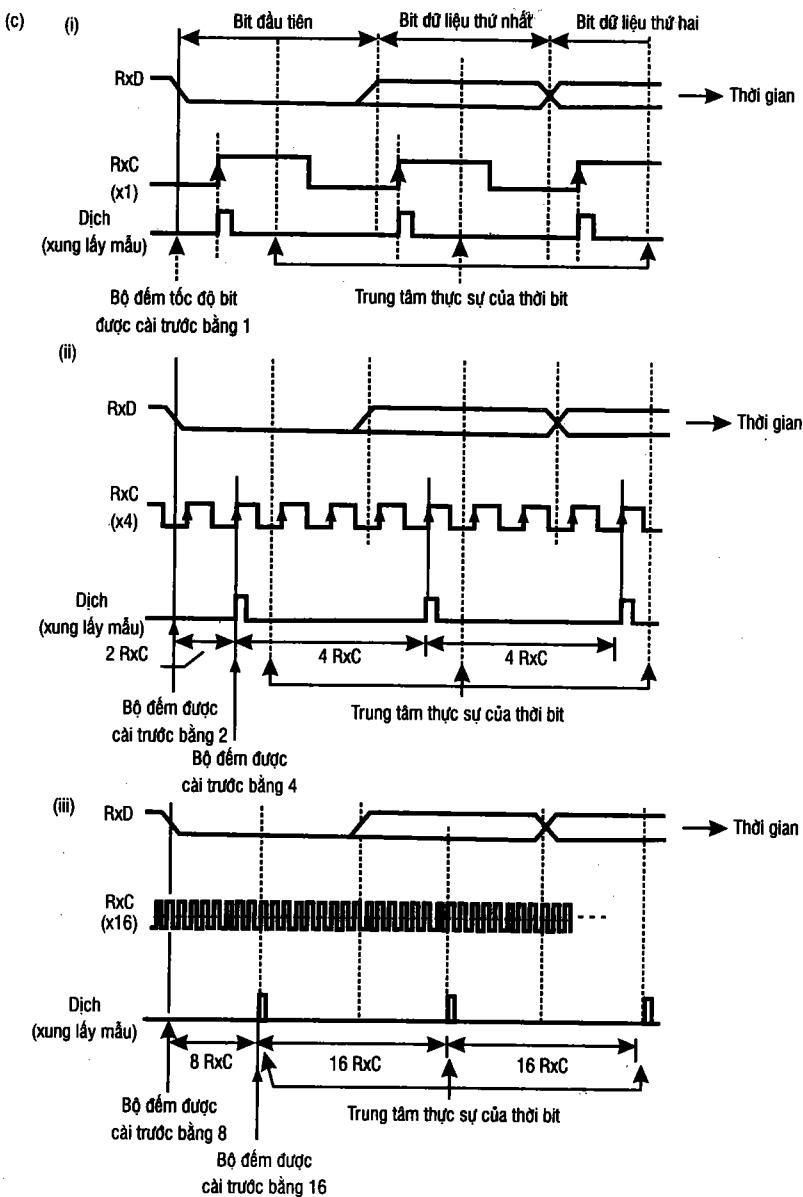
Trong truyền bất đồng bộ, đồng hồ thu chạy một cách bất đồng bộ với tín hiệu thu. Để xử lý thu hiệu quả, cần phải có kế hoạch dùng đồng hồ thu để lấy mẫu tín hiệu đến, ngay điểm giữa thời bit của dữ liệu.

Để đạt được điều này, **tín hiệu đồng hồ thu nhanh gấp N lần đồng hồ phát và mỗi bit được dịch vào SIP** (Serial In, Parallel Out) **chu kỳ xung đồng hồ**. Sự chuyển trạng thái từ 1 xuống 0 là dấu hiệu của bit start có ý nghĩa bắt đầu của một ký tự và chúng được dùng để khởi động bộ đếm xung clock ở máy thu. Mỗi bit bao gồm cả bit start, được lấy mẫu tại khoảng giữa của thời bit. Ngay sau khi phát hiện, bit start được lấy mẫu sau $N/2$ chu kỳ xung clock, tiếp tục lấy mẫu sau mỗi N xung clock tiếp theo cho mỗi bit trong ký tự. Lược đồ tổng quát được trình bày trên hình 3.5(a), các nguyên lý định thời trong hình 3.5(b), và 3 ví dụ về tỉ lệ tốc độ xung đồng hồ khác nhau trong phần 3.5(c).



Hình 3.5 Truyền bất đồng bộ:

(a) Nguyên lý hoạt động (b) Các nguyên tắc định thời



**Hình 3.5 (tiếp theo) Truyền bất đồng bộ:
(c) Ví dụ về tốc độ xung đồng hồ khác nhau.**

Cần lưu ý rằng, đồng hồ thu chạy bất đồng bộ với tín hiệu đến, các vị trí tương đối của hai tín hiệu có thể ở bất kỳ vị trí nào trong một chu kỳ của xung

đồng hồ thu, với N càng lớn thì vị trí lấy mẫu có khuynh hướng gần giữa thời bit hơn. Do vậy ở chế độ truyền này tốc độ truyền không thể cao được.

Ví dụ:

Một khối dữ liệu được truyền qua một liên kết số liệu nối tiếp. Nếu một đồng hồ có tần số 19,2kHz tại máy thu, hãy suy ra tỉ lệ tốc độ đồng hồ thích hợp và ước lượng độ lệch trong trường hợp xấu nhất từ trung tâm của thời bit danh định, biểu diễn theo phần trăm của thời bit cho các tốc độ truyền số liệu sau đây:

- a) 1200bps
- b) 2400bps
- c) 9600bps

Tóm tắt lời giải:

Có thể dễ dàng suy ra từ hình (c) ở trên rằng độ lệch trong trường hợp xấu nhất là xấp xỉ một chu kỳ của xung đồng hồ thu. Do đó:

- a) Với tốc độ 1200bps thì tỉ lệ Rx C cực đại có thể là x16. Độ lệch tối đa là 6,25%.
- b) Với tốc độ 2400bps thì tỉ lệ Rx C cực đại có thể là x8. Độ lệch tối đa là 12,5%.
- c) Với tốc độ 9600bps thì tỉ lệ Rx C cực đại có thể là x2. Độ lệch tối đa là 50%.

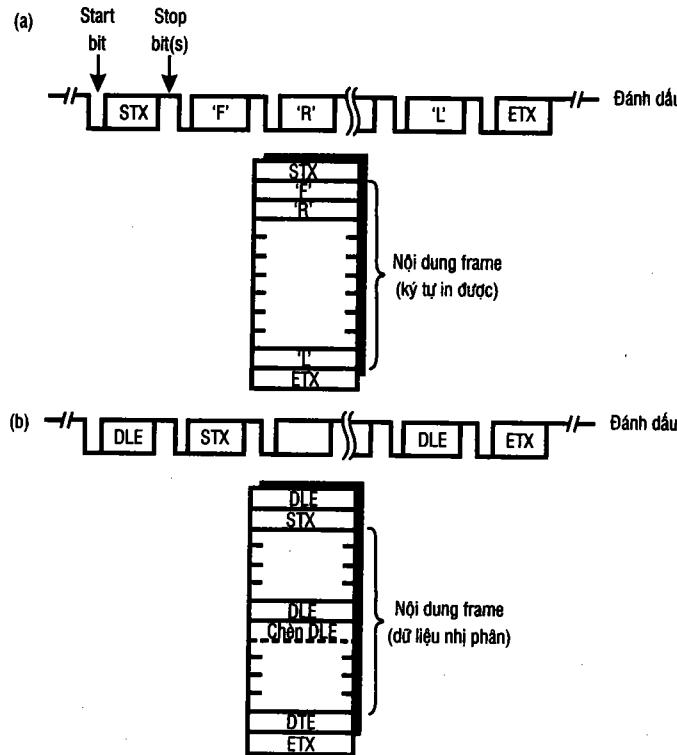
Rõ ràng trường hợp sau cùng là không thể chấp nhận được. Với đường dây chất lượng thấp, đặc biệt phải chịu trễ lớn thì ngay cả trường hợp thứ hai cũng không tin cậy. Điều này giải thích vì sao tỉ lệ x16 luôn được dùng bắt cứ khi nào có thể.

2.3. Nguyên tắc đồng bộ ký tự

Mạch điều khiển truyền nhận được lập trình để hoạt động với số bit bằng nhau trong một ký tự kể cả số stop bit, start bit và bit kiểm tra giữa thu và phát. Sau khi phát hiện và nhận start bit, việc đồng bộ ký tự đạt được tại đầu thu rất đơn giản, chỉ việc đếm đúng số bit đã được lập trình. Sau đó sẽ chuyển ký tự nhận được vào thanh ghi đệm thu nội bộ và phát tín hiệu thông báo với thiết bị điều khiển (CPU) rằng đã nhận được một ký tự mới. Và sẽ đợi cho đến khi phát hiện một start bit kế tiếp.

2.4. Nguyên tắc đồng bộ frame

Khi thông điệp gồm một khối các ký tự thường xem như một frame thông tin (information frame) được truyền, bên cạnh việc đồng bộ bit và đồng bộ ký tự, máy thu còn phải xác định được điểm bắt đầu và điểm kết thúc một frame. Điều này được gọi là sự đồng bộ frame.

**Hình 3.6 Đồng bộ frame:****(a) Các ký tự có thể in được (b) Dữ liệu nhị phân.**

Nguyên tắc đơn giản nhất để truyền một khối ký tự có thể in được là đóng gói chúng thành một khối hoàn chỉnh bằng hai ký tự điều khiển truyền đặc biệt là STX và ETX. Mặc dù kế hoạch này thỏa mãn cho đồng bộ frame nhưng có trở ngại là nếu trong dữ liệu lại có chuỗi bit giống STX hay ETX thì sao. Để khắc phục vấn đề này, khi truyền STX hay ETX chúng sẽ được kèm theo một DLE (Data Link Escape). Mặc khác để tránh nhầm lẫn giữa ký tự DLE đi kèm với STX hay ETX và byte giống DLE trong phần nội dung của frame, khi xuất hiện một byte giống DLE trong phần nội dung, nó sẽ được gấp đôi khi truyền đi. Xem hình 3.6.

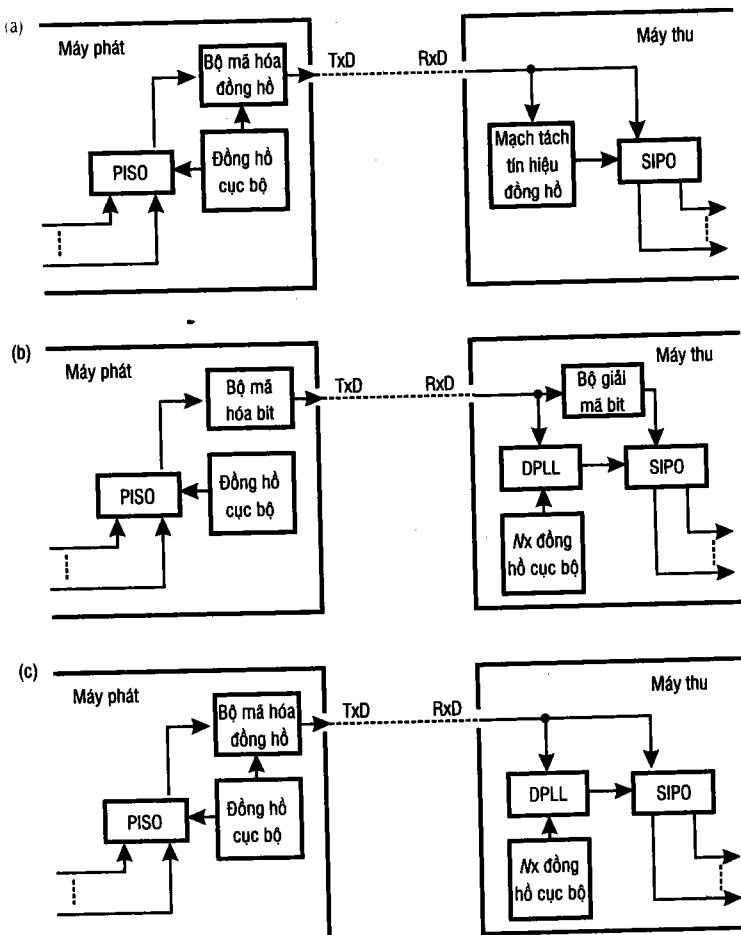
3. THÔNG TIN NỐI TIẾP ĐỒNG BỘ

3.1. Khái quát

Việc thêm các start bit và nhiều stop bit vào mỗi một ký tự hay byte trong thông tin nối tiếp bất đồng bộ làm cho hiệu suất truyền giảm xuống, đặc biệt là khi truyền một thông điệp gồm một khối ký tự. Một khía cạnh phương pháp đồng bộ bit được dùng ở đây trớn nên thiếu tin cậy khi gia tăng tốc độ truyền. Vì lý do này người ta đưa ra phương pháp mới gọi là truyền đồng bộ, truyền đồng bộ khắc phục được những nhược điểm nêu trên. Tuy nhiên, cũng giống như truyền bất đồng bộ chúng ta chỉ chấp nhận phương pháp nào cho phép máy thu đạt được sự đồng bộ bit, đồng bộ ký tự và đồng bộ frame. Trong thực tế có hai lược đồ truyền nối tiếp đồng bộ: truyền đồng bộ thiền hướng bit và truyền đồng bộ thiền hướng ký tự.

3.2. Nguyên tắc đồng bộ bit

Sự khác nhau cơ bản của truyền bất đồng bộ và đồng bộ là đối với truyền bất đồng bộ đồng hồ thu chạy một cách bất đồng bộ với tín hiệu đến, còn truyền đồng bộ thì đồng hồ thu chạy đồng bộ với tín hiệu đến. Các start bit và stop bit không được dùng, thay vì vậy mỗi frame được truyền như là dòng liên tục các ký số nhị phân. Máy thu đồng bộ bit trong hai cách. Hoặc là thông tin định thời được nhúng vào trong tín hiệu truyền và sau đó được tách ra bởi máy thu, hoặc máy thu có một đồng hồ cục bộ được giữ đồng bộ với tín hiệu thu nhờ một thiết bị gọi là DPLL (Digital Phase Lock-Loop). Như chúng ta sẽ thấy, DPLL lợi dụng sự chuyên trạng thái bit từ $1 \rightarrow 0$ hay từ $0 \rightarrow 1$ trong tín hiệu thu để duy trì sự đồng bộ qua một khoảng thời gian định kỳ nào đó. Lược đồ lai ghép là kết hợp cả hai cách. Nguyên lý hoạt động của các lược đồ này được trình bày trên hình 3.7.



Hình 3.7 Các phương pháp đồng bộ xung đồng hồ: (a) Mã hóa xung đồng hồ (b) Dùng DPLL (c) Phối hợp cả hai cách trên.

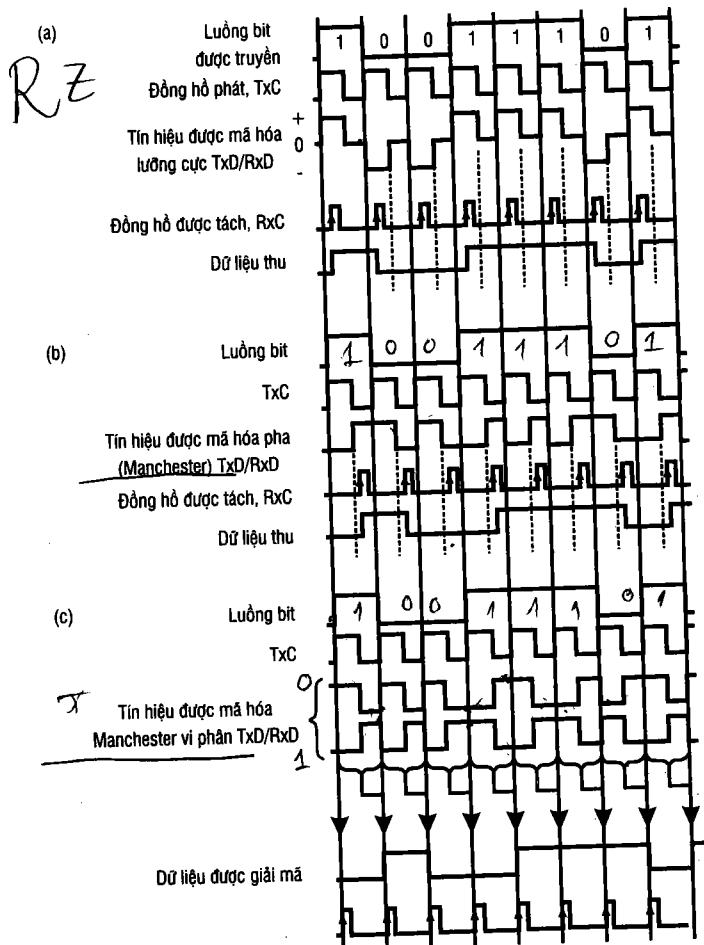
3.2.1. Mã hóa và giải mã tín hiệu đồng hồ

Ba phương pháp thường dùng để nhúng tín hiệu định thời vào dòng bit truyền được trình bày trên hình 3.8. Trong hình 3.8(a) luồng bit truyền được mã hóa sao cho một xung dương biểu diễn cho bit 1 và một xung âm biểu diễn cho bit 0. Phương pháp này được gọi là mã hóa lưỡng cực (bipolar encoding). Mỗi bit trong tín hiệu mã hóa lưỡng cực chứa thông tin đồng hồ. Một mạch điện tử đơn giản cho phép tín hiệu đồng hồ được giải mã từ tín hiệu thu được. Vì tín hiệu được mã hóa quay trở về mức 0 sau khi mỗi bit được mã hóa, nên tín hiệu này được xếp vào loại tín hiệu quay về 0_RZ (Return-to-Zero).

Mã hóa lưỡng cực dùng ba mức biên độ khác biệt (+, 0, -). Ngược lại ở hình 3.8(b) chỉ yêu cầu hai mức. Tín hiệu đầu ra của lược đồ này thuộc loại không quay về 0, NRZ (Non-Return-to-Zero) và được gọi là mã hóa nhị pha hay mã hóa Manchester. Với lược đồ này bit 1 được mã hóa thành sự chuyển tín hiệu từ thấp lên cao và bit 0 tương ứng với tín hiệu chuyển từ cao xuống thấp. Tuy nhiên, cũng có sự chuyển trạng thái ($1 \rightarrow 0$ hay $0 \rightarrow 1$) tại giữa thời bit (còn tùy thuộc vào Manchester loại gì ví dụ ở hình là Manchester theo mức (level), xem mục kế tiếp). Sự chuyển này được dùng bởi mạch giải mã xung đồng hồ để tạo ra một xung đồng hồ (clock) ngay tại điểm giữa phần thứ hai của thời bit. Tại thời điểm này, tín hiệu thu sẽ ở mức cao hoặc mức thấp (1 hay 0) và nhờ đó tín hiệu chính xác sẽ được dịch vào thanh ghi SIPO.

Lược đồ chỉ ra ở hình 3.8(c) được gọi là mã hóa Manchester vi phân. Nó khác với mã hóa Manchester thường ở chỗ, mặc dù vẫn còn chuyển trạng thái tại giữa mỗi thời bit, còn có chuyển trạng thái tại điểm bắt đầu của thời bit khi nào bit kế tiếp được mã hóa là 0. Hiệu quả của loại mã hóa này là tín hiệu được mã hóa ở ngõ ra sẽ nhận một trong hai dạng tùy thuộc vào mức khởi động giả sử ban đầu (cao hay thấp). Như chúng ta thấy, bit này chỉ đơn giản là phiên bản ngược của bit kia. Điều này rất hiệu quả với các liên kết hai dây, bởi vì nếu chúng ta dùng các bộ phát và thu vi phân thì sẽ không có vấn đề gì xảy ra tại đầu cuối của dây ở máy thu. Với lược đồ Manchester thường, ngõ ra sẽ bị đảo ngược dẫn đến hoạt động bị sai. Tín hiệu đồng hồ được phát ra tại điểm cuối của mỗi bit và sự chuyển trạng thái trong mỗi thời bit xác định được bit nhận là 1 hay 0.

Cả hai lược đồ Manchester ở trên được xem là dạng *mã cân bằng* vì không có thành phần một chiều liên kết với chúng. Điều này rất quan trọng vì có ảnh hưởng đến khả năng ghép AC với mạch điện tử của máy thu, ví dụ khi dùng phương pháp ghép biến áp. Các mạch điện tử này có thể hoạt động với nguồn cung cấp độc lập không liên quan gì đến nguồn cung cấp của máy phát. Mặt khác, vì không có thành phần một chiều trong tín hiệu nên khả năng kháng nhiễu rất tốt, đặc biệt là các loại nhiễu tác động vào biên độ.

**Hình 3.8 Các phương pháp mã hóa đồng hồ**

Một số mã hóa tín hiệu khác:

NRZ_L (Non-Return-to-Zero_Level)

NRZ_S (Non-Return-to-Zero_Space)

NRZ_M (Non-Return-to-Zero_Mark)

Manchester_L (Manchester_Level)

Manchester_M (Manchester_Mark)

Manchester_S (Manchester_Space)

Miller



3.2.2. Mạch vòng khóa pha số (Digital Phase Lock-Loop)

Một phương pháp mã hóa tín hiệu đồng hồ vào trong luồng bit truyền là dùng một nguồn xung đồng hồ ổn định tại máy thu để duy trì cơ cấu đồng bộ với luồng bit đến. Tuy nhiên, vì lược đồ truyền đồng bộ không có các start bit và stop bit, nên chúng ta phải mã hóa thông tin theo phương pháp sao cho có đủ sự chuyển trạng thái trong dạng sóng truyền, cho phép đồng hồ máy thu đồng bộ lại theo định kỳ. Một kỹ thuật nhằm đạt được điều này là chuyển dữ liệu truyền đi qua *bộ xáo trộn* (scramber), bộ phận này xáo trộn ngẫu nhiên luồng bit truyền để bỏ đi những chuỗi bit 1 hay bit 0 liên tục và dài. Có thể tùy chọn cách mã hóa tín hiệu sao cho sự chuyển trạng thái trong luồng bit truyền là phù hợp cho sự đồng bộ tại máy thu.

Trước khi truyền, mẫu bit được mã hóa như trên hình 3.9(a). Tín hiệu mã hóa này được xem là dạng sóng đảo của NRZ gọi là NRZI (non-return-to-zero-inverted). Với NRZI mức tín hiệu (1 hay 0) không thay đổi khi truyền một bit nhị phân 1, trong khi truyền bit nhị phân 0 thì thay đổi. Điều này có nghĩa là sẽ luôn có sự chuyển trạng thái của tín hiệu đến dưới dạng NRZI, luồng bit này vốn không có nhị phân 1 liên tục. Mới nhìn qua chúng ta thấy dạng này đường như không khác gì so với dạng NRZ thường, nhưng chúng ta sẽ thấy trong mục 3.4, nếu dùng một lược đồ thiên hướng bit có chèn các bit 0 thì đường dây sẽ luôn có một bit nhị phân 0 trong luồng bit truyền ít nhất là 5 thời bit. Dạng sóng kết quả đảm bảo chứa một số các chuyển đổi, vì chuỗi bit 0 dài sẽ tạo ra sự chuyển mức tại mỗi thời bit. Điều này cho phép máy thu chỉnh lại đồng hồ của nó sao cho đồng bộ với luồng bit đến.

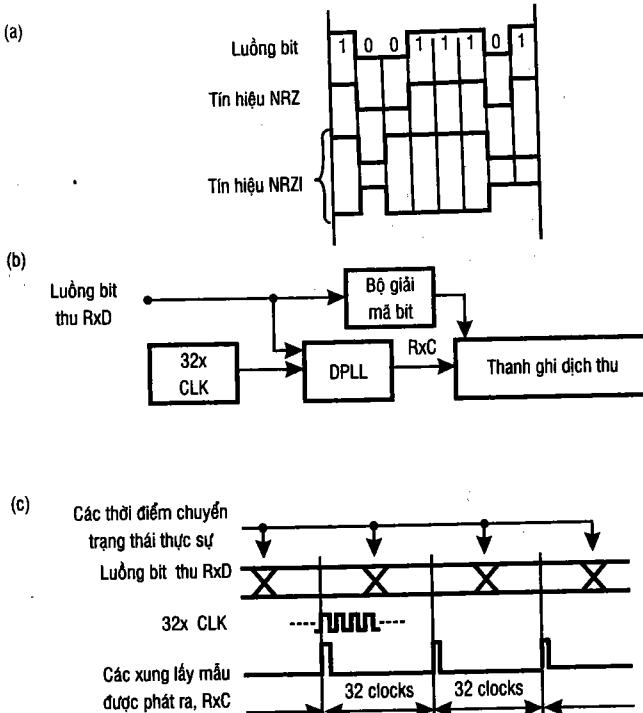
Mạch được dùng để duy trì cơ cấu đồng bộ bit được gọi là DPLL. Mạch gồm một bộ dao động thạch anh (đồng hồ nguồn) có thể điều khiển và có tần số ổn định chỉ cần điều chỉnh nhỏ theo định kỳ. Số độ nguyên lý được trình bày trên hình 3.9(b). Thông thường, tần số của đồng hồ gấp 32 lần tốc độ bit trên liên kết số liệu và được dùng bởi DPLL để suy ra khoảng định thời giữa các mẫu liên tiếp trong luồng bit thu được.

Giả sử luồng bit đến và đồng hồ cục bộ đang trong trạng thái đồng bộ, trạng thái của tín hiệu đến từ đường dây sẽ được lấy mẫu (và do đó được dịch chuyển vào trong thanh ghi SIPO) tại trung tâm của mỗi thời bit chính xác 32 xung giữa mỗi mẫu. Điều này được trình bày trên hình 3.9(c).

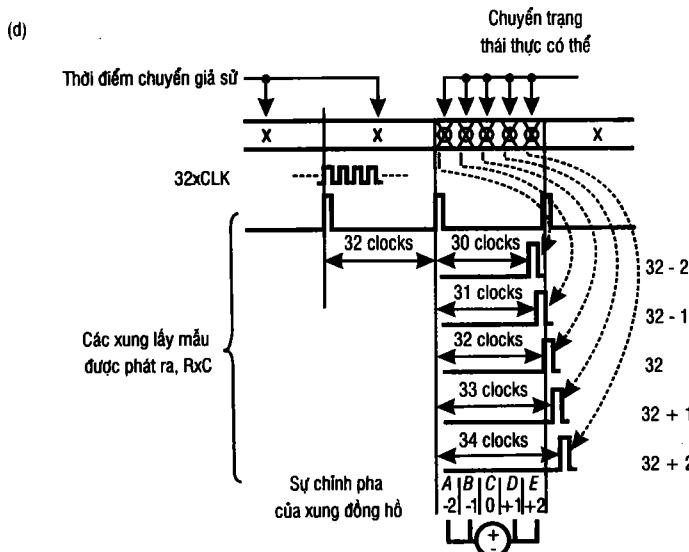
Bây giờ, giả sử luồng bit truyền và đồng hồ cục bộ bị mất đồng bộ do các thay đổi nhỏ phía sau. Thời khắc lấy mẫu được chỉnh tăng một cách rời rạc như trình bày trên hình 3.9(d). Nếu không có sự chuyển mức trên đường dây, DPLL chỉ đơn thuần phát ra một xung lấy mẫu sau 32 xung clock so với xung lấy mẫu trước đó. Tuy nhiên, bất cứ khi nào có một sự chuyển trạng thái thì khoảng thời gian giữa xung lấy mẫu được phát ra trước đó và xung lấy mẫu kế tiếp được xác định theo vị trí tương quan giữa điểm chuyển và thời điểm mà DPLL dự định sẽ lấy mẫu tiếp. Để đạt được điều này mỗi thời bit được chia thành 5 đoạn, tạm đặt là A, B, C, D, và E như trên hình 3.9(d). Ví dụ, nếu có một chuyển trạng thái xảy ra trong thời gian của đoạn A thì điều này chỉ ra xung lấy mẫu sau cùng gần với sự chuyển tiếp

theo và đó là trễ. Vì vậy khoảng thời gian phát xung kế tiếp được rút ngắn còn 30 xung clock. Tương tự, sự chuyển trạng thái xảy ra tại đoạn E chỉ ra xung lấy mẫu phía trước quá sớm so với vị trí chuyển. Do đó, khoảng thời gian đến xung kế sẽ được kéo dài thêm 2 xung clock nữa. Các chuyển trạng thái trong B và D rõ ràng gần với vị trí dự định của DPLL nên chỉ cần chỉnh ít hơn. Sau cùng, sự chuyển trạng thái trong C là dù gần với điểm dự định nên không cần phải chỉnh.

Bằng cách này, sự điều chỉnh diễn ra liên tục giữ cho các xung được tạo ra gần với trung tâm mỗi thời bit. Trong thực tế, bề rộng của mỗi đoạn là không bằng nhau. Hai đoạn A và E bên ngoài xa trung tâm hơn nên kéo dài hơn so với 3 đoạn trong. Trong mạch được trình bày, chia $A=E=10$, $B=D=4$ và $C=4$. Dễ dàng suy ra trong trường hợp xấu nhất DPPLL cần đến 10 chuyển đổi để qui về điểm trung tâm của bit trong dạng sóng: 5 thời bit chính thô (± 2) và 5 thời bit để chỉnh tinh (± 1). Do đó khi dùng DPPLL, trước khi truyền frame đầu tiên hay sau khoảng thời gian nhàn rỗi nào giữa các frame thường phải truyền một số byte/ký tự cung cấp tối thiểu 10 sự chuyển trạng thái. Ví dụ 2 byte/ký tự, mỗi byte gồm tất cả các bit 0 sẽ cho 16 lần chuyển trạng thái trong NRZI. Điều này đảm bảo rằng DPPLL phát ra các xung lấy mẫu tại trung tâm của mỗi thời bit ngay tại thời điểm nhận ký tự mở frame. Tuy nhiên chúng ta phải chú ý rằng một khi ở trong trạng thái đồng bộ chỉ có thể thực hiện các điều chỉnh nhỏ trong tiến trình nhận frame.



Hình 3.9 *Hoạt động của DPLL:*
 (a) Mã hóa bit (b) Lược đồ mạch (c) Đồng pha



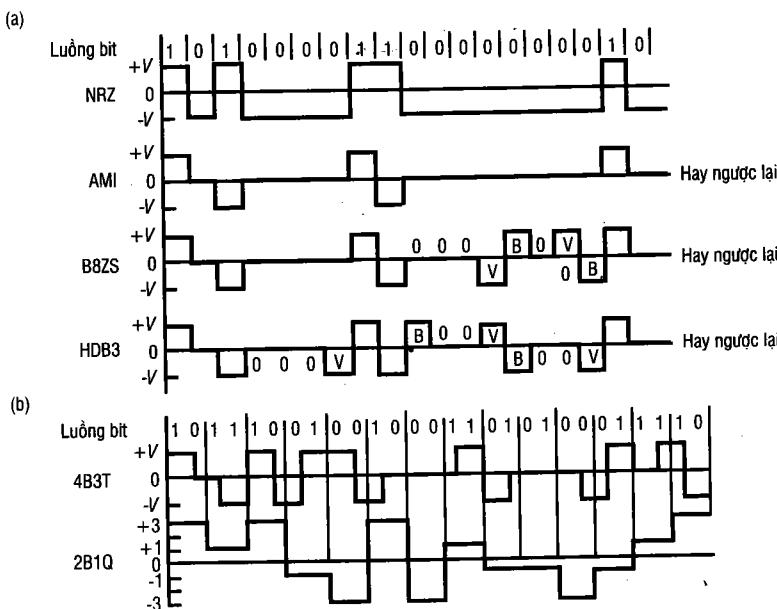
Hình 3.9 (tiếp theo) Hoạt động của DLL:
(d) các qui tắc điều chỉnh đồng hồ.

Có thể suy ra từ hình 3.9 rằng với mã hóa NRZI tốc độ tối đa mà tín hiệu mã hóa thay đổi cực tính chỉ bằng một nửa so với khi dùng mã hóa lưỡng cực và mã hóa Manchester. Nếu thời bit là T_b , với mã hóa NRZI thì tốc độ là $1/T_b$, trong khi đó với mã hóa lưỡng cực hay Manchester là $2/T_b$. Tốc độ tối đa này cũng được xem là tốc độ điều chế. Theo như chương trước thì thành phần tần số cao nhất của mỗi lược đồ tương ứng là $1/T_b$ và $2/T_b$. Điều này có nghĩa là để tốc độ bằng nhau, mã hóa lưỡng cực và mã hóa Manchester cần băng thông gấp hai lần so với tín hiệu mã hóa NRZI. Vì tốc độ điều chế càng lớn thì độ rộng băng thông càng lớn.

Chính vì các ảnh hưởng này mà mã hóa Manchester và Manchester vi phân thường được dùng rộng rãi trong các ứng dụng LAN trong khi lược đồ NRZI được dùng chủ yếu trong các mạng WAN. Các LAN hoạt động trong một văn phòng hay tòa nhà do đó chỉ dùng các dây cáp ngắn. Điều này có nghĩa là ngay cả khi hoạt động với tốc độ cao thì sự suy hao và băng thông của môi trường truyền cũng không là vấn đề đáng quan tâm đối với các LAN.

Ngược lại trong các mạng WAN thì cáp xoắn thường được sử dụng để truyền với tốc độ cao và qua khoảng cách vài km. Do đó các lược đồ mã hóa như NRZI thường được dùng với mỗi bit chiếm hết một xung. Một số ví dụ về lược đồ mã hóa thường dùng trong WAN được trình bày trên hình 3.10. Ba ví dụ đầu trong hình 3.10(a) đều là các tín hiệu được mã hóa vi phân và dùng nhiều mức trên một thời bit. Việc dùng mã hóa vi phân có ý nghĩa là các tín hiệu được biểu diễn có thể đảo lại tất cả nếu chọn một điểm bắt đầu có cực tính khác. Việc dùng đa mức với sự chuyển trạng thái luân phiên có nghĩa là bất kỳ một lối nào xuất hiện đều cưỡng bức lên tính luân phiên của sự chuyển trạng thái, nhờ đó dễ dàng phát hiện.

Ba dạng mã trong hình 3.10(a) dùng 3 mức (+V, 0, -V) để biểu diễn luồng bit. Với AMI (Alternate Mark Inversion), sự chuyển mức tín hiệu được kích hoạt bởi bit nhị phân 1 trong luồng bit nhập. Khuyết điểm của lược đồ cơ bản này là một chuỗi bit nhị phân 0 liên tiếp dài sẽ không có sự chuyển trạng thái nào. Hậu quả là DPLL sẽ bị mất đồng bộ bit bất cứ khi nào chuỗi bit 0 như thế xuất hiện.



Hình 3.10 Vài lược đồ mã hóa thường dùng trong WAN:
(a) Mã nhị phân (b) Mã đa mức.

Để khắc phục giới hạn này, một số dẫn suất của lược đồ cơ bản này đã được dùng. Ví dụ, một phiên bản sửa đổi được dùng trong đó các chuyển trạng thái được kích hoạt bởi các bit 0 trong luồng bit nhập thay vì bit 1. Như chúng ta sẽ thấy với phương pháp truyền theo hướng bit sẽ tự động chèn một bit 0 bất cứ khi nào có một chuỗi 5 bit 1 liên tiếp. Điều này có nghĩa là có ít nhất một sự chuyển trạng thái sau mỗi 5 bit, với tốc độ lên đến hàng trăm Kbps cũng đủ cho DPLL duy trì sự đồng bộ đồng hồ. Các lược đồ này thường dùng trong mạng ISDN.

Dẫn suất thứ hai của lược đồ mã hóa AMI được gọi là *luồng cực thay thế 8 bit 0 liên tiếp* B8ZS (*bipolar with eight zeros substitution*) cũng thường được dùng. B8ZS về cơ bản cũng giống như AMI ngoại trừ nếu có một chuỗi 8 bit 0 liên tiếp thì 8 bit này sẽ được mã hóa thành 000VB0VB, trong đó B biểu diễn sự chuyển trạng thái bình thường và V là sự chuyển trạng thái cưỡng bức. Số bit 0 tối đa có thể trong luồng tín hiệu là 7. Lược đồ này được dùng rộng rãi trong các mạng truyền dẫn số ở Bắc Mỹ.

Một lược đồ thứ 3 được dùng trong một số mạng truyền dẫn số không thuộc Bắc Mỹ là *lưỡng cực mật độ cao* HDB3 (High Density Bipolar 3). Hoạt động mã hoá này thay thế bất kỳ chuỗi 4 bit 0 bằng 3 bit 0 liên tiếp và kèm theo một cưỡng bức đảo cực, nghĩa là cùng cực tính với sự chuyển trạng thái lần kế trước. Do đó, ví dụ bốn bit 0 đầu tiên được thay thế là 000V. Tuy nhiên, với nguyên tắc cơ bản này, sự xuất hiện một chuỗi dài các bit 0 đồng nghĩa với sự thêm vào thành phần một chiều DC khi mỗi tập bốn bit 0 được mã hóa. Để tránh tình trạng này, với một chuỗi bit 0 dài sẽ mã hóa mỗi tổ hợp bốn bit 0 thành B00V, tạo ra sự đảo cực tính luân phiên. Từ đó có thể suy ra rằng với HDB3 thì số lượng các bit 0 liên tiếp tối đa là 3. Chúng ta thường xem HDB3 và hai lược đồ mã hóa trên như là *dạng điều chế*.

Hai lược đồ mã hóa được trình bày trong hình 3.10(b) được dùng trên các mạch truy xuất mạng ISDN. Chúng ta sẽ thấy, các dạng này hoạt động với tốc độ 160Kbps qua các đôi dây xoắn và trải rộng đến vài km. Cả hai dạng này là ví dụ cho dạng *mã hóa giảm tốc độ baud*, có nghĩa là có nhiều hơn một bit truyền ứng với một xung tín hiệu (time cell). Ưu điểm chính của dạng này là giảm được xuyên âm nhờ sự biến thiên về biên độ tín hiệu giữa các xung kề nhau nhỏ hơn.

Cả hai loại mã được xếp vào loại mBnL, có nghĩa là một tuần tự m bit nhập được biểu diễn bởi n xung, mỗi xung có thể lấy một trong L mức, trong đó $n < m$ và $L > 2$. Do đó với mã 4B3T cũng được gọi là trạng thái giám sát được điều chỉnh 43 hay MMS43 (modified monitoring state 43)_T cho biết có 3 mức được biểu diễn bởi ký hiệu +, 0, -. Do đó, 4 bit nhập được biểu diễn bởi 3 xung theo một trong 3 mức. Tốc độ baud vì thế là 3/4, giảm đi 1/4.

Ba ký hiệu mã được truyền cho mỗi tuần tự bốn bit nhập được chọn ra từ 1 trong bốn cột của bảng "các mẫu mã hóa 4B3T". Thông thường các biến áp được dùng trong các ứng dụng mạng điện rộng nhằm cài đặt phần phát và thu của đường dây với các phần khác. Điều này có nghĩa là không có đường dẫn nào cho DC. Do đó cần đảm bảo rằng mức DC của tín hiệu truyền là bằng 0, nếu khác 0 thì tín hiệu 0 tại máy thu sẽ thay đổi. Đây là hiện tượng được gọi là *DC wander* và tác động của nó làm cho bộ thu biên dịch không đúng tín hiệu nhận.

Bằng cách kiểm tra các mã trong các cột khác nhau, chúng ta có thể thấy rằng giá trị trung bình của mỗi từ mã là thay đổi. Ví dụ, trong cột 1, từ mã 0 - + có giá trị trung bình là 0 trong khi từ mã + + - có giá trị trung bình là +1. Rõ ràng, nếu một chuỗi từ mã có giá trị trung bình là +1 được truyền một cách trực tiếp, thì mức tín hiệu có nghĩa tại máy thu sẽ rời khỏi điểm 0 thực sự. Để khắc phục tình trạng này các mã được dùng trong mỗi tuần tự nhị phân thay đổi theo cột sao cho mức tín hiệu trung bình luôn hướng về 0.

Bảng 3.1 Các mẫu mã hóa 4B3T

Tuần tự nhị phân	1		2		3		4	
	Mã	Cột kế	Mã	Cột kế	Mã	Cột kế	Mã	Cột kế



0001	0 - +	1	0 - +	2	0 - +	3	0 - +	4
0111	- 0 +	1	- 0 +	2	- 0 +	3	- 0 +	4
0100	- + 0	1	- + 0	2	- + 0	3	- + 0	4
0010	+ - 0	1	+ - 0	2	+ - 0	3	+ - 0	4
1011	+ 0 -	1	+ 0 -	2	+ 0 -	3	+ 0 -	4
1110	0 + -	1	0 + -	2	0 + -	3	0 + -	4
1001	+ - +	2	+ - +	3	+ - +	4	- - -	1
0011	0 0 +	2	0 0 +	3	0 0 +	4	- - 0	2
1101	0 + 0	2	0 + 0	3	0 + 0	4	- 0 -	2
1000	+ 0 0	2	+ 0 0	3	+ 0 0	4	0 - -	2
0110	- + +	2	- + +	3	- - +	2	- - +	3
1010	+ + -	2	+ + -	3	+ - -	2	+ - -	3
1111	+ + 0	3	0 0 -	1	0 0 -	2	0 0 -	3
0000	+ 0 +	3	0 - 0	1	0 - 0	2	0 - 0	3
0101	0 + +	3	- 0 0	1	- 0 0	2	- 0 0	3
1100	+ + +	4	- + -	1	- + -	2	- + -	3

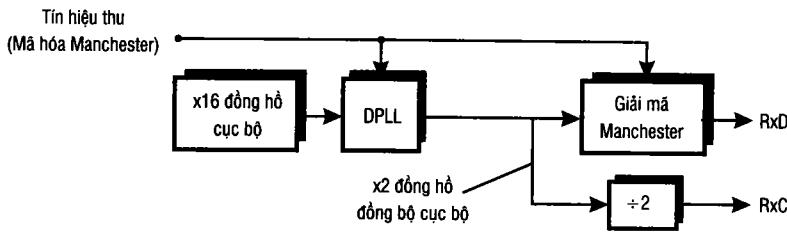
Liên kết chật chẽ với mỗi mã trong một cột là một số chỉ đến cột kế mà từ đây mã tiếp theo sẽ được chọn. Trong hình 3.10(b), tuần tự đầu tiên 1011 được chọn từ cột 1 (+ 0 -) và cột kế tiếp là 1. Tuần tự tiếp theo là 1001 (+ - +) và cột kế là 2, và cứ thế tiếp tục. Từ bảng có thể suy ra có 27 từ mã khác nhau. Vì chỉ có 16 tuần tự nhập có thể (4 bit) nên mã này hơi dư, có thể khai thác để phục vụ cho điều khiển lỗi. Nội dung của bảng cũng được chọn sao cho ứng với một tuần tự nhập ngẫu nhiên bằng thông trung bình được yêu cầu nhỏ hơn so với khi không dùng mã hóa.

Mã thứ hai trong hình 3.10(b) là 2B1Q, Q (quaternary) chỉ ra có 4 mức xung. Mỗi tuần tự nhịp 2 bit được truyền trong một xung ở một trong 4 mức. Như trong hình, bốn mức được biểu diễn bởi +3, +1, -1, -3 để chỉ ra tính đối xứng qua 0 và bằng khoảng cách giữa các trạng thái. Bit đầu tiên trong mỗi cặp ký số nhị phân xác định dấu (1 = +, 0 = -) và bit thứ hai chỉ độ lớn (1=1, 0=3). Không có dư thừa đối với mã này nhưng tốc độ baud là 1/2 so với 3/4 của 4B3T.

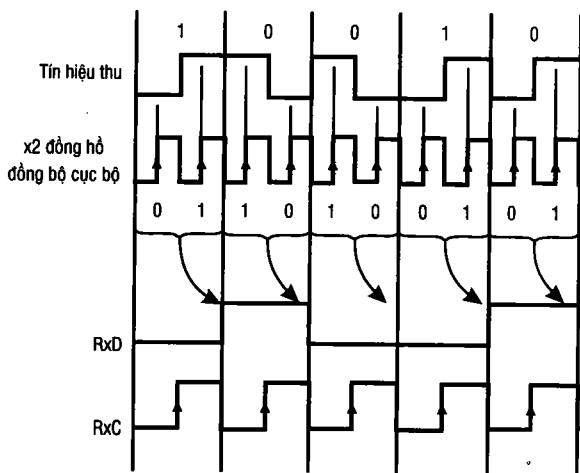
3.2.3. Các lược đồ lai

Khi tốc độ bit tăng lên, vì vậy rất khó đạt được và duy trì sự đồng bộ đồng hồ. Mặc dù các lược đồ Manchester và DPLL được dùng rộng rãi nhưng vẫn có một số các lược đồ lai. Một lược đồ tiêu biểu được trình bày trên hình 3.11(a). Nó kết hợp cả Manchester và DPLL.

(a)



(b)



Hình 3.11 Đồng bộ bit dùng mã hóa Manchester và DPLL:
(a) Sơ đồ mạch (b) Sóng dạng.

DPLL giữ tín hiệu đồng hồ đồng bộ với tín hiệu đến. Tuy nhiên, mã Manchester được dùng có nghĩa là có ít nhất một sự chuyển trạng thái tín hiệu trong một thời bit thay vì một chuyển trạng thái trong nhiều bit, ví dụ NRZI có một chuyển trạng thái trong 5 bit. Vì vậy tín hiệu đồng hồ cục bộ duy trì đồng bộ tin cậy hơn, từ dạng sóng ở hình 3.11(b) chúng ta có thể suy ra khả năng duy trì đồng bộ với tín hiệu đến của đồng hồ cục bộ, đem đến một phương tiện tin cậy để giải mã tín hiệu thu. Tuy nhiên, cái giá phải trả là nhu cầu băng thông gia tăng so với NRZI.

3.3. Truyền đồng bộ thiên hướng ký tự

Có hai kiểu điều khiển truyền đồng bộ: đồng bộ thiên hướng ký tự và đồng bộ thiên hướng bit. Cả hai đều dùng các nguyên tắc đồng bộ bit giống nhau. Khác nhau chủ yếu giữa hai lược đồ là phương pháp được dùng để đạt được sự đồng bộ ký tự và đồng bộ frame.

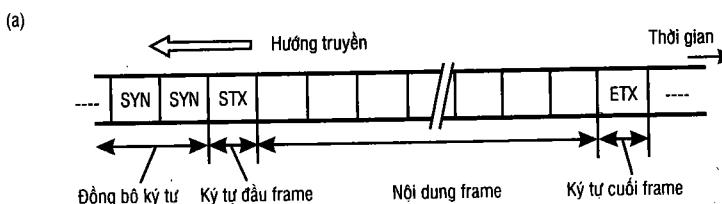
Truyền đồng bộ thiên hướng ký tự được dùng chủ yếu để truyền các khối ký tự, như là các tập tin dạng text. Vì không có start bit hay stop bit nên cần phải có cách thức để đồng bộ ký tự. Để thực hiện việc đồng bộ này, máy phát thêm vào các ký tự điều khiển truyền, gọi là các ký tự đồng bộ SYN, ngay trước các khối ký tự truyền. Các ký tự điều khiển này phải có hai chức năng: trước hết, chúng cho phép máy thu duy trì đồng bộ bit. Thứ hai, khi điều này đã được thực hiện, chúng cho phép máy thu bắt đầu biên dịch luồng bit thu theo các ranh giới ký tự chính xác _sự đồng bộ ký tự.

Phần 3.12(a) trình bày sự đồng bộ frame đạt được theo phương thức giống như truyền bất đồng bộ bằng cách đóng gói khối ký tự giữa cặp ký tự điều khiển truyền STX-ETX. Các ký tự điều khiển SYN thường được dùng bởi bộ thu để đồng bộ ký tự thì đứng trước ký tự STX (start of frame). Khi máy thu đã đạt được đồng bộ bit nó sẽ chuyển vào chế độ làm việc gọi là *chế độ bắt số liệu*. Điều này được trình bày trên hình 3.12(b).

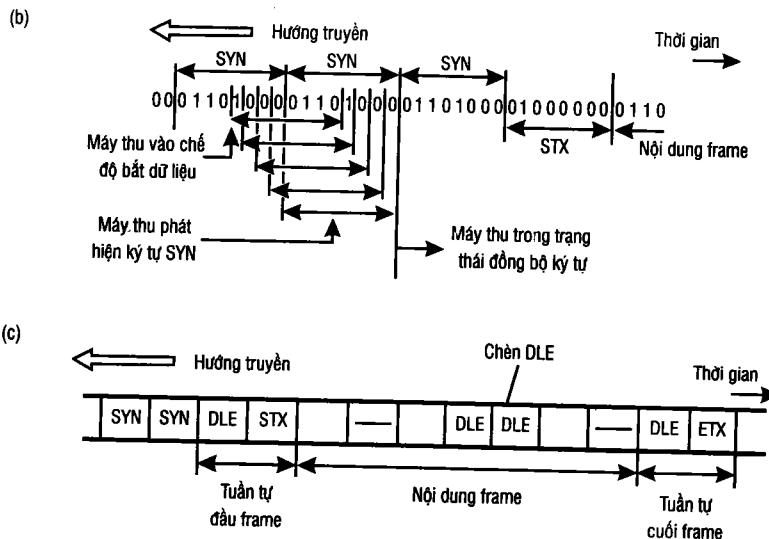
Khi bộ thu vào *chế độ bắt số liệu*, nó bắt đầu dịch dòng bit trong một cửa sổ 8 bit khi tiếp nhận một bit mới. Bằng cách này, khi nhận được mỗi bit, nó kiểm tra xem 8 bit sau cùng có đúng bằng ký tự đồng bộ hay không. Nếu không bằng, nó tiếp tục thu bit kế tiếp và lặp lại thao tác kiểm tra này. Nếu tìm thấy ký tự đồng bộ, các ký tự kế tiếp được đọc sau mỗi 8 bit thu được.

Khi ở trong trạng thái đồng bộ ký tự (và do đó đọc các ký tự theo đúng ranh giới bit), máy thu bắt đầu xử lý mỗi ký tự thu nối tiếp để dò ra ký tự STX đầu frame. Khi phát hiện một STX, máy thu xử lý nhận nội dung frame và chỉ kết thúc công việc này khi phát hiện ra ký tự ETX. Trên một liên kết điểm-nối-diểm, thông thường máy phát sẽ quay trở lại truyền các ký tự SYN để máy thu duy trì cơ cấu đồng bộ. Dĩ nhiên, toàn bộ thủ tục trên đều phải được lặp lại mỗi khi truyền một frame mới.

Khi dữ liệu nhị phân đang được truyền, sự trong suốt dữ liệu đạt được giống như phương pháp đã được mô tả trong mục 'nguyên tắc đồng bộ frame' trước đây, có nghĩa là dùng một ký tự DLE chèn vào trước STX và ETX, và chèn một DLE vào bất cứ vị trí nào trong nội dung có chứa một DLE. Trong trường hợp này, các ký tự SYN đứng trước ký tự DLE đầu tiên.



Hình 3.12 Truyền đồng bộ thiên hướng ký tự: (a) Khuôn dạng frame



**Hình 3.12 (tiếp theo) Truyền đồng bộ thiên hướng ký tự:
(b) Sự đồng bộ ký tự (c) Sự trong suốt dữ liệu.**

3.4. Truyền đồng bộ thiên hướng bit

Việc dùng một cặp ký tự bắt đầu và kết thúc một frame để đồng bộ frame, cùng với việc thêm vào các ký tự DLE không hiệu quả cho việc truyền số liệu nhị phân. Hơn nữa, dạng của các ký tự điều khiển truyền thay đổi theo các bộ mã ký tự khác nhau, vì vậy chỉ có thể sử dụng với một bộ ký tự. Để khắc phục các vấn đề này người ta dùng lược đồ truyền đồng bộ thiên hướng bit. Lược đồ này được xem như lược đồ điều khiển dùng cho việc truyền các frame dữ liệu gồm dữ liệu in được và dữ liệu nhị phân. Ba lược đồ thiên hướng bit chủ yếu được trình bày trên hình 3.13. Chúng khác nhau chủ yếu ở phương pháp bắt đầu và kết thúc mỗi frame.

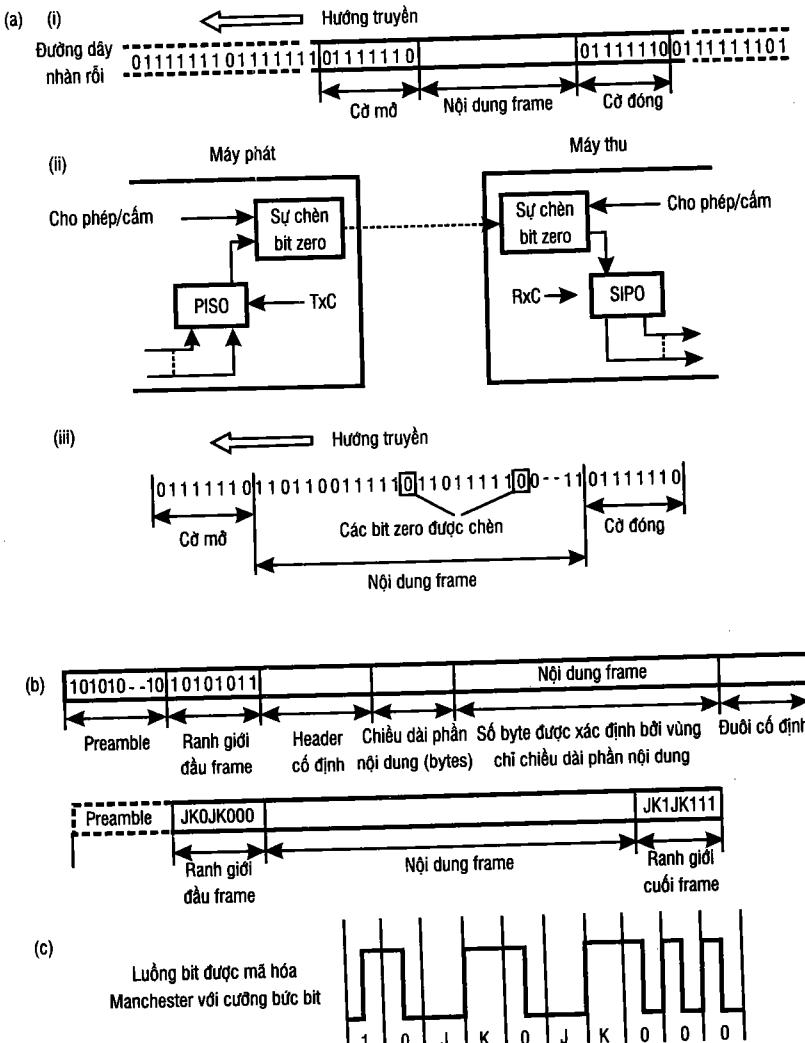
Lược đồ hình 3.13(a) được dùng nhiều cho các liên kết điểm-nối-điểm. Bắt đầu và kết thúc một frame bằng một 'cờ' 8 bit 01111110. Dùng thuật ngữ 'thiên hướng bit' vì luồng bit thu được dò theo từng bit. Do đó về nguyên lý nội dung của frame không nhất thiết phải là một bộ số của 8 bit.

Để cho phép máy thu tiếp cận và duy trì cơ cấu đồng bộ bit, máy phát phải gửi một chuỗi các byte idle (nhàn rỗi) 01111111 đứng trước cờ bắt đầu frame. Với NRZI mã hóa bit 0 trong byte idle cho phép DPLL tại máy thu tiếp cận và duy trì sự đồng bộ đồng hồ. Khi nhận được cờ khởi đầu frame, nội dung của frame được đọc và dịch theo các khoảng 8 bit cho đến khi gặp cờ kết thúc frame.

Để đạt được tính trong suốt dữ liệu, cần đảm bảo cờ không được nhận lầm trong phần nội dung. Vì lý do này người ta dùng kỹ thuật chèn bit 0 hay còn gọi là



kỹ thuật 'nhồi bit' (bit stuffing). Mạch thực hiện chức năng này đặt tại ngõ ra của thanh ghi PISO. Mạch này chỉ hoạt động trong quá trình truyền nội dung của frame. Khi có một tuần tự 5 bit 1 liên tục nó sẽ tự động chèn vào một bit 0. Bằng cách này sẽ không bao giờ có cờ trong phần nội dung truyền đi. Một mạch tương tự tại máy thu nắm ngay trước lối vào thanh ghi SIPO thực hiện chức năng gỡ bỏ bit 0 theo hướng ngược lại.



Hình 3.13 Các phương pháp đồng bộ frame thiền hướng bit: (a) Dùng cờ
(b) Chỉ định chiều dài và ranh giới bắt đầu frame (c) Cường bức mã hóa bit.



Lược đồ trình bày trong hình 3.13(b) được dùng trong một vài mạng LAN. Khi đó môi trường truyền là môi trường quảng bá và chia sẻ cho tất cả các DTE. Để cho phép tất cả các trạm khác nhau đạt được sự đồng bộ bit. Trạm truyền đặt vào trước nội dung frame một mẫu bit gọi là mẫu mở đầu *preamble* bao gồm mươi cặp 10. Một khi đã đồng bộ, máy thu dò dòng bit theo từng bit một cho đến khi tìm thấy byte khởi đầu khung 10101011. Một header cố định được xác định phía sau bao gồm địa chỉ, thông tin chiều dài phần nội dung. Do đó, với lược đồ này máy thu chỉ cần đếm số byte thích hợp để xác định sự kết thúc mỗi frame.

Lược đồ trình bày trên hình 3.13(c) cũng được dùng với LAN. Sự bắt đầu và kết thúc của mỗi frame được chỉ định bởi các mẫu mã hóa bit không chuẩn. Ví dụ mã Manchester, thay cho truyền một tín hiệu tại giữa thời bit, mức tín hiệu duy trì tại cùng mức như bit trước trong thời bit hoàn chỉnh (J) hay tại mức ngược (K). Một lần nữa, để phát hiện đầu và cuối frame, máy thu dò từng bit, trước hết phát hiện JK0JK000 và sau đó phát hiện mẫu kết thúc JK1JK111. Vì các ký hiệu J, K là các mã bit không chuẩn, nên trong phần nội dung của frame sẽ không chứa các ký hiệu này, như vậy đạt được sự trong suốt dữ liệu.

4. MẠCH ĐIỀU KHIỂN TRUYỀN SỐ LIỆU

4.1. Khái quát

Để thực hiện được các phương thức truyền một cách cụ thể, các nhà chế tạo đã cung cấp một loạt các IC chuyên dùng, các IC này chính là phần cứng thuộc lớp vật lý trong một hệ thống thông tin, chúng hoạt động theo nguyên tắc của kỹ thuật số và vì vậy chế độ truyền đồng bộ hay bất đồng bộ phụ thuộc vào việc sử dụng đồng hồ chung hay riêng khi truyền tín hiệu số đi xa.

Các IC đều là các vi mạch có thể lập trình. Đầu tiên lập trình chế độ hoạt động mong muốn bằng cách ghi một byte có nghĩa vào thanh ghi chế độ *mode register*. Sau đó ghi tiếp byte điều khiển vào thanh ghi lệnh *command register* để vi mạch theo đó mà hoạt động.

Vì các giao tiếp truyền nối tiếp được dùng khá rộng rãi trong các thiết bị điện tử hiện đại, các vi mạch ngoại vi LSI đặc biệt đã được phát triển cho phép thực hiện các loại giao tiếp này. Tên tổng quát của hầu hết các IC này là:

- UART (Universal Asynchronous Receiver Transmitter)
- USRT (Universal Synchronous Receiver Transmitter): mạch này đồng bộ thiền hướng ký tự
- USART có thể hoạt động theo UART hay USRT tùy chọn
- BOPs (Bit-Oriented Protocol circuits) mạch này đồng bộ thiền hướng bit
- UCCs (Universal Communication Control circuits) có thể lập trình cho cả 3 loại trên (UART,USRT hay BOPs).

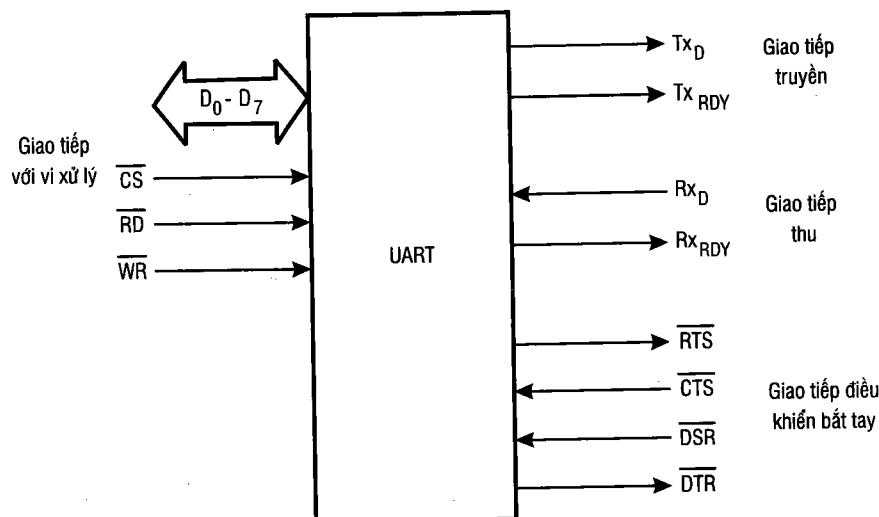
Cả UART và USART đều có khả năng thực hiện nhu cầu chuyển đổi song song sang nối tiếp để truyền số liệu đi xa và chuyển đổi nối tiếp sang song song khi tiếp nhận số liệu. Đối với số liệu được truyền theo chế độ bất đồng bộ chúng cũng có khả năng đóng khung cho ký tự một cách tự động với START bit, PARITY bit, và các STOP bit thích hợp.

Hơn nữa, để tiếp nhận dữ liệu, UART và USART đều có khả năng kiểm tra các ký tự một cách tự động để phát hiện lỗi parity, và hai loại lỗi khác là lỗi định dạng frame (framing error) và lỗi chồng chập ký tự nhận (overrun error). Lỗi định dạng frame có nghĩa là sau khi phát hiện đầu ký tự với một START bit, máy thu không phát hiện được số lượng STOP bit thích hợp. Điều này có nghĩa là ký tự truyền không được nhận một cách hoàn hảo và cần phải truyền lại. Lỗi chồng chập ký tự có nghĩa là ký tự đã được nhận nhưng không được bộ vi xử lý đọc ra khỏi thanh ghi dữ liệu thu của USART trước khi nhận tiếp một ký tự mới. Do đó, ký tự trước bị mất và sẽ phải truyền lại.

Một sơ đồ khái của UART được trình bày trên hình 3.14. Ở đây chúng ta thấy rằng nó có bốn giao tiếp tín hiệu chủ yếu: giao tiếp với bộ vi xử lý, giao tiếp truyền, giao tiếp thu, và giao tiếp điều khiển bắt tay (handshake control interface). Chúng ta sẽ tìm hiểu từng giao tiếp này một cách chi tiết hơn.

Các LSI UART và USART không thể đứng một mình trong hệ thống truyền tin. Hoạt động của chúng được điều khiển bởi một bộ xử lý có ứng dụng tổng quát ví dụ như các bộ vi xử lý thông thường. Giao tiếp với bộ vi xử lý là giao tiếp được dùng để kết nối UART vào đơn vị xử lý trung tâm CPU (Central Processing Unit). Xem hình 3.14, chúng ta thấy rằng giao tiếp này bao gồm một bus dữ liệu hai chiều 8-bit (D_0 đến D_7) và 3 đường điều khiển, CS, RD và WR.

Tất cả dữ liệu truyền giữa UART và CPU diễn ra qua bus dữ liệu 8 bit này. Hai hoạt động có sử dụng bus này là nạp dữ liệu từ phần thu của UART vào và xuất dữ liệu ra phần truyền của nó. Các loại thông tin khác cũng được chuyển qua giữa CPU và UART. Ví dụ như các chỉ thị điều khiển chế độ, các chỉ thị lệnh điều hành, và thông tin trạng thái.



Hình 3.14 Sơ đồ khái quát của UART.

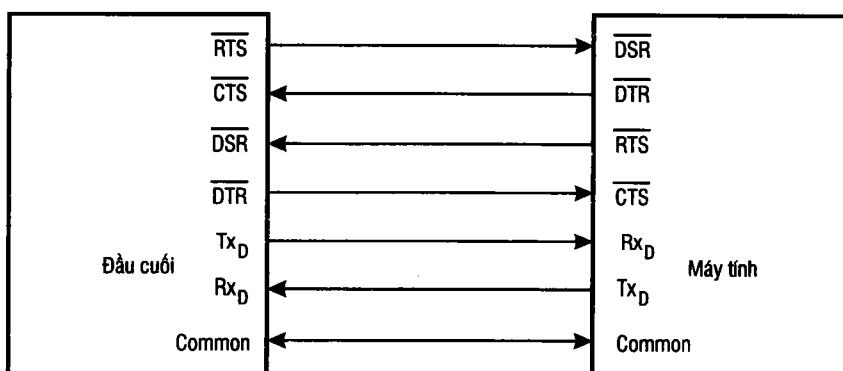
Các LSI UART có thể được cấu hình cho các chế độ hoạt động khác nhau thông qua phần mềm. Các chỉ thị điều khiển chế độ là những gì phải được gửi đến UART để khởi động các thanh ghi điều khiển của nó tạo chế độ hoạt động mong muốn. Ví dụ như khuôn dạng của frame được dùng để truyền hay nhận dữ liệu có thể được cấu hình thông qua phần mềm. Các tùy chọn tiêu biểu gồm chiều dài ký tự thay đổi từ 5 đến 8 bit; kiểm tra chẵn, kiểm tra lẻ hay không kiểm tra, và 1, 1,5 hay 2 STOP bit.

Phía trước chúng ta đã nói một UART không thể tự thực hiện chức năng truyền tin. Thật vậy, tuần tự của các sự kiện cần thiết để khởi động truyền và nhận được điều khiển bởi các lệnh của CPU gửi đến UART. Ví dụ CPU có thể bắt đầu yêu cầu truyền số liệu bằng cách ghi một lệnh vào UART khiến ngõ điều khiển hướng ra RTS được set ở mức tích cực (0). Mức tín hiệu tích cực 0 trên RTS báo cho hệ thống ở đầu bên kia của đường truyền (ví dụ DCE) chuẩn bị nhận dữ liệu. Tại đầu thu của đường truyền tin, CPU có thể chấp nhận sẵn sàng nhận dữ liệu bằng cách gửi một lệnh cho UART của nó, làm cho tín hiệu điều khiển DTR xuống mức thấp (0).

Hầu hết các UART đều có thanh ghi trạng thái (status register) chứa thông tin liên quan đến trạng thái hiện hành của nó. Ví dụ nó có thể chứa các bit cờ (flag bits) biểu thị trạng thái hiện hành của các đường tín hiệu như RTS và DTR. Điều này cho phép CPU kiểm tra các trạng thái luận lý của các đường dây này bằng phần mềm.

Ngoài các thông tin về mức luận lý (logic) của các đường điều khiển, thanh ghi trạng thái còn chứa các bit cờ biểu thị các điều kiện lỗi như lỗi parity, lỗi định dạng frame và lỗi chống chập ký tự. Sau khi nhận một ký tự, trước hết CPU đọc các bit này để chắc chắn rằng đã nhận được một ký tự hợp lệ, và nếu các bit này không ở mức tích cực (không lỗi) thì ký tự được đọc ra từ thanh ghi dữ liệu thu trong UART.

Phía bên phải của sơ đồ khối ở hình 3.14 chúng ta có thể thấy giao tiếp truyền và giao tiếp thu. Giao tiếp truyền có hai đường tín hiệu: transmit data (Tx_D) và transmitter ready (Tx_{RDY}). Tx_D là đường mà qua đó bộ phận truyền của UART xuất ký tự nối tiếp ra đường truyền. Như trình bày trên hình 3.15, đường ra này được nối đến ngõ nhập dữ liệu thu (Rx_D) của bộ phận thu trong hệ thống đầu xa của đường truyền.



Hình 3.15 Giao tiếp truyền bất đồng bộ đơn giản giữa một máy tính và một đầu cuối số liệu.

Thông thường bộ phận truyền của một LSI UART chỉ có thể giữ được một ký tự tại một thời điểm. Các ký tự này được giữ trong thanh ghi dữ liệu truyền (transmit data register) trong UART. Vì chỉ có một ký tự có thể được giữ trong UART, nên UART phải phát tín hiệu cho CPU mỗi khi nó hoàn thành truyền ký tự này. Đường Tx_{RDY} được cung cấp cho mục đích này. Ngay sau khi hoàn tất truyền ký tự trong thanh ghi dữ liệu truyền, bộ phận truyền chuyển Tx_{RDY} sang mức tích cực. Tín hiệu này sẽ gửi một ngắt (interrupt) vào CPU. Bằng cách này, sự xuất hiện của nó có thể khiến cho chương trình điều khiển chuyển qua chương trình phục vụ thích hợp và sẽ xuất ký tự khác ra thanh ghi dữ liệu truyền và sau đó hoạt động truyền được khởi động trở lại.

Bộ phận thu tương tự như bộ phận truyền mà chúng ta mới mô tả. Tuy nhiên, ở đây đường dữ liệu thu (RxD) là đường nhập, nó chấp nhận các chuỗi bit ký tự nối tiếp được truyền từ bộ phận truyền của hệ thống ở đầu xa của đường truyền. Lưu ý rằng trong hình 3.15 ngõ nhập dữ liệu nối đến ngõ truyền dữ liệu (TxD) của bộ phận truyền trong hệ thống tại đầu xa. Ở đây tín hiệu hướng ra RxD_{RDY} được dùng như một ngắt gửi đến CPU, thông báo cho CPU biết đã nhận được một ký tự. Chương trình con phục vụ ngắt này được khởi động, trước hết nó phải xác định ký tự này có hợp lệ không và nếu hợp lệ, nó phải đọc ký tự này ra khỏi thanh ghi dữ liệu thu của UART.

Dùng các tín hiệu điều khiển bắt tay RTS, DSR, DTR và CTS, các loại giao thức truyền bắt đồng bộ khác nhau có thể được hiện thực thông qua các giao tiếp này. Một giao tiếp truyền bắt đồng bộ dùng các đường điều khiển này được trình bày trên hình 3.15. Trong ví dụ, một giao thức có thể được thiết lập sao cho khi đầu cuối số liệu muốn gửi số liệu đến máy tính nó sẽ phát một yêu cầu qua ngõ ra RTS của nó. Để làm điều này, CPU của đầu cuối số liệu phải gửi một lệnh đến UART, lệnh này yêu cầu nó đưa đường tín hiệu RTS xuống mức tích cực (mức luận lý 0). Mức tích cực trên RTS của đầu cuối được áp vào ngõ nhập DSR của máy tính. Bằng cách này, nó báo cho máy tính biết rằng đầu cuối số liệu muốn truyền số liệu vào máy tính.

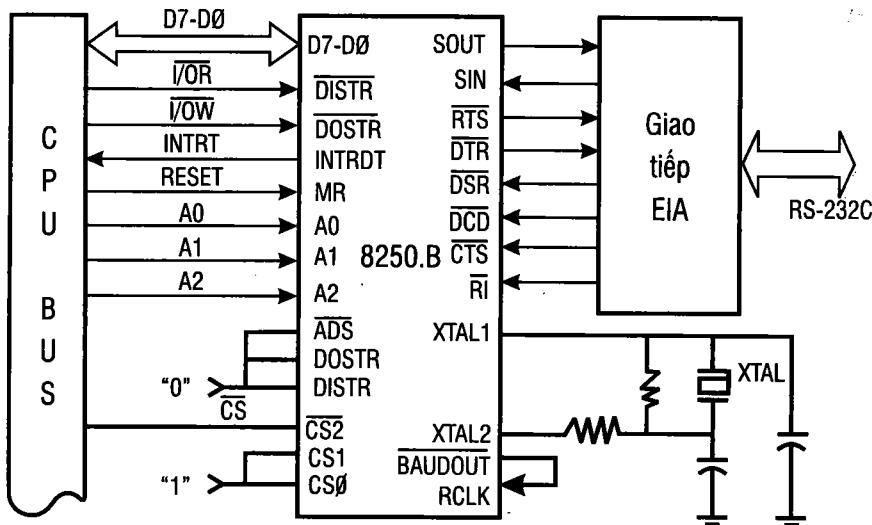
Khi máy tính sẵn sàng nhận số liệu, nó chấp nhận yêu cầu này bằng cách kích hoạt ngõ ra DTR trên UART của nó. CPU trong máy tính thực hiện điều này bằng cách gửi một lệnh cho UART bảo nó chuyển DTR xuống mức tích cực (mức 0). Tín hiệu này áp vào ngõ vào CTS của UART của đầu cuối số liệu và báo với UART trong đầu cuối số liệu biết có thể bắt đầu xuất số liệu lên TxD . Cùng lúc đó, bộ phận thu trong UART ở máy tính cũng bắt đầu đọc số liệu từ ngõ nhập RxD của nó.

4.2. Giao tiếp truyền có thể lập trình UART 8250 của Intel

Các hệ thống thông tin bắt đồng bộ thường dùng các chip LSI giá thành rẻ như: Motorola 6850 ACIA (Asynchronous Communication Interface Adapters), dùng với họ vi xử lý 6800/68000 của Motorola.

National 8250 UART dùng với họ vi xử lý 8088/80x86 của Intel.

Hình 3.16 mô tả cấu hình cơ bản của 8250 với ba phần giao tiếp chính là: giao tiếp với bus xuất/nhập (I/O) của hệ thống, mạch định thời và giao tiếp RS-232.



Hình 3.16 Cấu hình cơ bản của 8250

4.2.1. Giao tiếp bus

- Đệm dữ liệu hai chiều 3 trạng thái (D_0-D_7): là cửa ngõ trao đổi dữ liệu song song, các từ điều khiển, và từ trạng thái với CPU.
- DISTR,DISTR (Data Input Strobe): tín hiệu hướng vào chọn đọc thông tin từ 8250, sử dụng một trong hai đường.
- DOSTR,DOSTR (Data Output Strobe): tín hiệu hướng vào chọn ghi thông tin ra 8250, dùng một trong hai đường.
- A_0, A_1, A_2 : tín hiệu nhập, là địa chỉ dùng để chọn các thanh ghi bên trong 8250. Sự phân bố các địa chỉ tương ứng được trình bày trong bảng 3.2.
- CS_0, CS_1, CS_2 : cho phép 8250 hoạt động khi $CS_0 = 0$, $CS_1 = 1$, và $CS_2 = 0$.
- ADS (Adress Strobe): khi ở mức logic 0 cho phép thiết đặt địa chỉ A_0, A_1, A_2 và các CS, cho các tín hiệu này ổn định trước khi sử dụng.
- MR (Master Reset): khi ở mức logic 1 đưa 8250 về trạng thái ban đầu.

- INTRPT: là ngõ xuất yêu cầu ngắt quang về CPU. Tín hiệu này lên mức logic 1 khi xảy ra một trong bốn loại ngắt quang mà 8250 có thể phát ra: có cờ lỗi ở mức tích cực do số liệu nhận bị sai; nhận số liệu tốt; bộ đệm truyền không còn số liệu để truyền; có thay đổi trạng thái trên các đường tín hiệu điều khiển modem.
- CSOUT (Chip Select Output): ở mức logic 1 báo cho biết 8250 đã được chọn.
- DDIS (Driver Disable): ở mức logic 0 khi CPU đang đọc 8250.

Bảng 3.2 Phân bố địa chỉ chọn

DLAB	A ₂	A ₁	A ₀	Thanh ghi
0	0	0	0	Đệm nhận RBR/Đệm truyền THR
0	0	0	1	Cho phép ngắt IER
X	0	1	0	Nhận dạng ngắt IIR
X	0	1	1	Điều khiển đường dây LCR
X	1	0	0	Điều khiển modem MCA
X	1	0	1	Trạng thái đường dây LSR
X	1	1	0	Trạng thái modem MSR
X	1	1	1	Không dùng
1	0	0	0	Cài byte thấp cho số chia (Divisor Latch)
1	0	0	1	Cài byte cao cho số chia

4.2.2. Xung đồng hồ và sự định thời

Tần số xung đồng hồ của 8250 có thể được lấy từ một tín hiệu bên ngoài hoặc do một mạch dao động bên trong tạo ra nhờ nối với thạch anh. Các tần số này xác định ở chân XTAL1, sau đó qua một mạch chia tần (có thể lập trình được) để tạo ra một tần số tín hiệu đồng hồ chủ. Tần số này cao hơn tốc độ baud chọn 16 lần. Tín hiệu đồng hồ chủ được dùng để điều khiển mạch phát bên trong 8250 sao cho mạch phát và mạch thu có thể làm việc theo những tần số khác nhau.

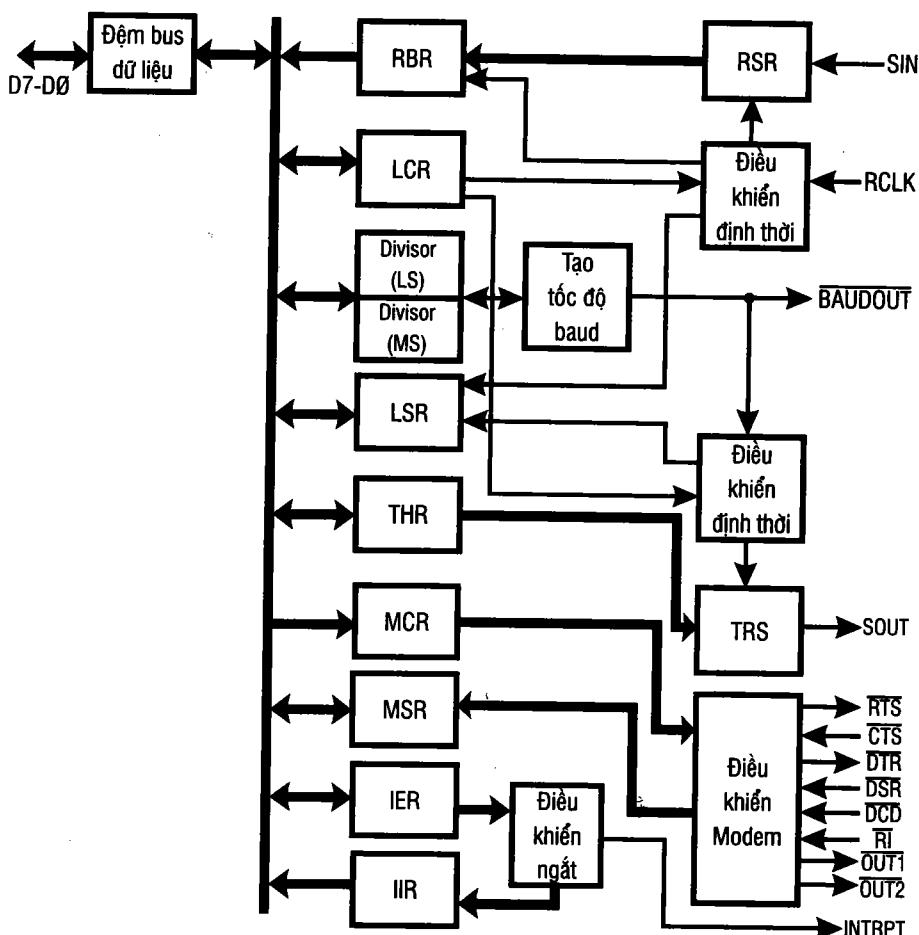
Tần số tín hiệu đồng hồ chủ này được đưa ra ở chân BAUDOUT, nếu chọn tần số đồng hồ khác cho mạch thu thì đưa vào chân RCLK, còn nếu dùng cùng tần số thì nối hai chân này lại với nhau. Cũng có thể xử lý lại tần số đồng hồ chủ để tạo ra tần số xung đồng hồ cho mạch thu.

4.2.3. Cấu trúc bên trong và hoạt động của 8250

Hoạt động của 8250 được điều khiển bởi các thành phần điều khiển và hổ trợ điều khiển gồm các thanh ghi: thanh ghi điều khiển đường truyền LCR (Line

Control Register), thanh ghi trạng thái đường truyền LSR (Line Status Register), thanh ghi nhận dạng ngắt quãng IIR (Interrupt Identification Register), thanh ghi cho phép ngắt quãng IER (Interrupt Enable Register), thanh ghi điều khiển modem MCR (Modem Control Register), thanh ghi trạng thái modem MSR (Modem Status Register), thanh ghi đệm truyền THR (Transmitter Holding Register), thanh ghi đệm nhận RBR (Receiver Buffer Register). Các thanh ghi này có thể được tác động bởi người lập trình, cấu trúc bên trong của 8250 được mô tả trên hình 3.17.

Trạng thái reset của các thanh ghi và các tín hiệu điều khiển của 8250 sau khi reset (mức logic 1 trên đường MR) được trình bày trong bảng 3.3.



Hình 3.17 Cấu trúc bên trong của 8250.



Bảng 3.3 Trạng thái reset của các thanh ghi và các tín hiệu xuất ra

Thanh ghi/tín hiệu	Tác nhân reset	Trạng thái reset
Thanh ghi IER	MR	Các bit đều là 0
Thanh ghi IIR	MR	$b_0=1, b_1=b_2=0$, bit 3-7 luôn 0
Thanh ghi LCR	MR	Các bit đều là 0
Thanh ghi MCR	MR	Các bit đều là 0
Thanh ghi LSR	MR	Bit 5,6 = 1, các bit còn lại =0
Thanh ghi MSR	MR	Bit 0-3 = 0, bit 4-7 là tín hiệu nhập
SOUT	Đọc LSR/MR	1
INTRPT (receiver line status)	Đọc RBR/MR	0
INTRPT (received data available)	Đọc RBR/MR	0
INTRPT (Transmit Holding register empty)	Đọc IIR/ghi THR/MR	0
INTRPT (modem status)	Đọc MSR/MR	0
OUT1	MR	1
OUT 2	MR	1
RTS	MR	1
DTR	MR	1

Thanh ghi điều khiển đường truyền LCR:

DLAB	SB	Stick	EP	PEN	Stop	DL1	DL0
------	----	-------	----	-----	------	-----	-----

- DLAB (Divisor Latch Access bit) : 1 = lập trình tốc độ truyền/ nhận; 0 = lập trình khuôn dạng dữ liệu.
- SB (Set Break): 1 = ngắt SOUT xuống mức thấp, các bit kế tiếp không còn ý nghĩa.
- Stick parity: chọn mức logic cho bit kiểm tra chẵn lẻ; 1 = chọn mức 0 cho parity chẵn.

- EP (Even parity): 1 = dùng kiểm tra chẵn; 0 = dùng kiểm tra lẻ.
- PEN (Parity Enable): 1 = dùng kiểm tra chẵn lẻ; 0 = không dùng kiểm tra chẵn lẻ.
- Stop : định số lượng bit STOP; 0 = 1 bit STOP; 1 = $1\frac{1}{2}$ bit STOP nếu sử dụng 5 bit dữ liệu; 2 bit STOP nếu sử dụng 6, 7 hay 8 bit dữ liệu.
- DL1 và DL0 (Data Length): chỉ định số bit dữ liệu sử dụng trong frame dữ liệu. 00 = 5 bit; 01 = 6 bit; 10 = 7 bit; 11 = 8 bit

Hoạt động của 8250 được đồng bộ bởi xung đồng hồ tần số XTAL1, thường là 1,8432 MHz hoặc 3,072MHz. Tần số này sẽ được chia cho một giá trị số chia 16 bit để tạo ra tần số đồng hồ chủ điều khiển các bộ phận truyền và thu. Tần số đồng hồ chủ này sau đó sẽ được chia tiếp cho 16 để tạo ra tốc độ baud:

$$\text{Tốc độ baud} = \text{Tần số đồng hồ} / (16 \times \text{Số chia})$$

Bảng 3.4 Một số giá trị số chia tương ứng với tốc độ baud

Tốc độ baud	Số chia (hexa)	
	1,8342MHz	3,072MHz
300	0180	0280
600	00C0	0140
1200	0060	00A0
2400	0030	0050
4800	0019	0028
9600	000C	0014
19200	0006	000A
38400	0003	0005

Giá trị số chia cho phép thay đổi tốc độ baud thông qua lập trình. Bảng 3.4 trình bày các số chia ứng với một số tốc độ baud thông dụng. Số chia được cắt trong thanh ghi cài bộ số chia 16 bit (RBR/THR và IER). Để cài đặt tốc độ baud thực hiện các bước sau:

- Cài bit DLAB trong LCR lên 1. Lúc này hai thanh ghi RBR/THR và IER sẽ được dùng để chứa byte thấp và byte cao của giá trị chia.
- Tùy theo tốc độ baud cần cài đặt, nạp giá trị số chia thích hợp vào 2 thanh ghi này.



- iii. Reset bit DLAB (về 0) để trả lại trạng thái hoạt động bình thường của hai thanh ghi trên.

Đoạn chương trình sau đây thực hiện việc chọn tốc độ 19200波特 với tần số thạch anh là 3,072MHz.

```

Serial_Port_Add = 03F8H ; địa chỉ COM1
Divisor        = 000A ; số chia
Mov      DX, Serial_Port_Add ; DX chứa địa chỉ COM1
Add      DX, 3           ;DX chỉ đến LCR
In AL,DX          ; đọc nội dung LCR
Or AL,80H         ; cài bit DLAB lên 1
Out     DX,AL          ;nạp LCR trở lại
Move DX, Serial_Port_Add
Move AX, divisor
Out     DX,AX          ;nạp giá trị divisor vào RBR/THR và IER
In AL, DX
And     AL,7FH
Out     DX,AL          ;trả DLAB về 0

```

Thanh ghi trạng thái đường truyền LSR:

0	TSRE	THRE	BI	FE	PE	OE	DR
---	------	------	----	----	----	----	----

- TSRE (Transmitter Shift Register Empty): khi có giá trị 1 cho biết dữ liệu đã được dịch hết ra đường dây. Bit này bị xóa về 0 khi có sự chuyển dữ liệu từ THR sang TSR.
- THRE (Transmitter Holding Register Empty): khi có giá trị 1 cho biết 8250 sẵn sàng nhận ký tự mới từ CPU. Bit này lên 1 do sự chuyển dữ liệu từ THR sang TSR và gây ra loại ngắt quãng báo đệm truyền rỗng về cho CPU. Bit THRE = 0 khi CPU xuất ký tự mới ra 8250.
- BI (Break Interrupt): bằng 1 khi ngõ SIN (Serial Input) bị kéo xuống mức 0 lâu hơn khoảng thời gian của một frame. Bit này lên 1 gây ra ngắt quãng loại báo lỗi nhận dữ liệu bị sai.
- FE (Framing Error): bằng 1 khi số bit STOP không đúng như đã lập trình.
- PE (Parity Error): bằng 1 khi có lỗi kiểm tra chẵn lẻ

- OE (Overrun Error): bằng 1 khi bị mất ký tự, nghĩa là ký tự mới đến xóa ký tự cũ chưa kịp nạp vào CPU.
- DR (Data Ready) : bằng 1 khi 8250 vừa nhận đủ một ký tự. Bit DR bị xóa về 0 khi CPU đọc dữ liệu hoặc do thao tác ghi 0 ngay tại bit này.

Thanh ghi nhận dạng ngắt quang IIR:

0	0	0	0	0	IT1	IT0	INTR
---	---	---	---	---	-----	-----	------

- INTR = 0 cho biết đang có ngắt quang chờ. Lúc đó có thể dùng nội dung IIR như một con trỏ chỉ đến chương trình phục vụ thích hợp. Bit này được sử dụng cho chế độ quét (polling)
- IT1 và IT0: chỉ loại ngắt quang
 - 11. = Việc thu có lỗi hay trạng thái đường dây
 - 10. = Nhận dữ liệu tốt
 - 01. = Bộ đệm truyền rỗng
 - 00. = Trạng thái modem

Thanh ghi cho phép ngắt quang IER:

0	0	0	0	MS	RLS	THRE	DA
---	---	---	---	----	-----	------	----

- MS(Modem Status): bằng 1 cho phép ngắt quang loại trạng thái modem.
- RLS(Receiver Line Status): bằng 1 cho phép ngắt quang loại lỗi nhận hay trạng thái đường dây.
- THRE (Transmitter Holding Register Empty): bằng 1 cho phép ngắt quang loại đệm truyền rỗng.
- DA (Data Available): bằng 1 cho phép ngắt quang loại nhận dữ liệu tốt.

Để làm việc với các thiết bị đầu nối vào cổng nối tiếp RS232, giống như các hệ thống khác UART 8250 cũng có thể làm việc theo cơ chế quét (polling) và ngắt quang (interrupt). Tuy nhiên 8250 thích hợp với cơ chế ngắt quang hơn bởi hai lý do chính:

- 1) 8250 chỉ có một vùng đệm duy nhất giữa thanh ghi RSR và bus dữ liệu, vậy nếu quét liên tục các ký tự xuất hiện trên bus vào thì sẽ bị chồng lấp lên ký tự cũ chưa đọc, để làm việc tốt ở cơ chế quét thì phải có tối thiểu hai vùng đệm tổ chức theo kiểu hàng đợi (FIFO).

- 2) 8250 có sẵn các thanh ghi IER và IIR ở trên để sử dụng cho cơ chế ngắt quãng. Để thực hiện cơ chế quét thì phải che các bit trong IER.

Do đó đa số các phần mềm truyền số liệu qua cổng RS-232 và UART 8250 đều chọn cơ chế ngắt quãng. UART 8250 sẽ tạo tín hiệu INTRPT đến CPU (thông qua các chip xử lý ngắt quãng ví dụ như PIC 8259A) để yêu cầu CPU phục vụ khi có một trong bốn sự kiện sau xảy ra:

- (1) Sự kiện về nhận sai và trạng thái đường dây RLS (Receive Line Status): xảy ra khi đường dây bị gián đoạn; nhận một ký tự trước khi đọc một ký tự trước đó; ký tự nhận bị sai; không tìm thấy bit STOP.
- (2) Sự kiện nhận được một ký tự đến RDA (Receive Data Available): xảy ra khi nhận một ký tự vào thanh ghi đệm nhận RBR.
- (3) Sự kiện bộ đệm truyền rỗng THRE (Transmit Holding Register Empty): xảy ra khi thanh ghi THR rỗng, cho biết sẵn sàng nhận tiếp ký tự mới từ CPU.
- (4) Sự kiện trên giao tiếp với modem MS (Modem Status): xảy ra khi thanh ghi modem có CTS, hoặc DSR, hoặc RI, hoặc RLSD được cài ở mức tích cực.

Khi có các sự kiện xảy ra đồng thời thì UART sẽ phải chọn phục vụ cho sự kiện nào trước, thứ tự ưu tiên của các ngắt này được phân bố giảm dần từ sự kiện (1) đến sự kiện (4) đã nêu ở trên.

Thanh ghi điều khiển modem MCR:

0	0	0	LOOP	OUT2	OUT1	RTS	DTR
---	---	---	------	------	------	-----	-----

- LOOP bằng 1 chọn chế độ kết nối vòng ngược để kiểm tra. Lúc đó SOUT = 1, SIN bị cách ly với bên ngoài. Đầu ra của thanh ghi dịch truyền nối với đầu vào của thanh ghi dịch nhận. Các ngõ xuất điều khiển modem nối vòng lại các ngõ nhập điều khiển theo các cặp tương ứng như sau: RTS→CTS; DTR→DSR; RLSD và OUT2→RI. Các hoạt động ngắt quãng vẫn diễn ra bình thường.
- OUT2 = 1 làm ngõ OUT2 xuống 0
- OUT1 = 1 làm ngõ OUT1 xuống 0
- RTS = 1 làm ngõ RTS xuống 0
- DTR = 1 làm ngõ DTR xuống 0

Thanh ghi trạng thái modem MSR:

RLSD	RI	DSR	CTS	DRLSD	DRI	DDSR	DCTS
------	----	-----	-----	-------	-----	------	------

- RLSD = 1 cho biết ngõ nhập RLSD đang ở mức 0

- RI = 1 cho biết ngõ nhập RI đang ở mức 0
- DSR = 1 cho biết ngõ nhập DSR đang ở mức 0
- CTS = 1 cho biết ngõ nhập CTS đang ở mức 0
- DRLSD (Delta RLSD) = 1 ngõ RLSD đã thay đổi trạng thái kể từ lần đọc trước
- DRI (Delta RI) = 1 ngõ RI đã thay đổi trạng thái kể từ lần đọc trước
- DDSR (Delta DSR) = 1 ngõ DSR đã thay đổi trạng thái kể từ lần đọc trước
- DCTS (Delta CTS) = 1 ngõ CTS đã thay đổi trạng thái kể từ lần đọc trước.

Thanh ghi đệm truyền/dệm nhận THR/RBR:

Hai thanh ghi này có cùng địa chỉ xuất nhập. Thanh ghi đệm truyền là thanh ghi chỉ xuất dữ liệu còn thanh ghi đệm nhận là thanh ghi chỉ nhập dữ liệu. Đây là hai thanh ghi được dùng để truyền nhận dữ liệu.

4.3. Giao tiếp truyền có thể lập trình USART 8251 của Intel

4.3.1. Giới thiệu

Vì mạch 8251 cho phép hiện thực một giao tiếp truyền số liệu nối tiếp. Ví dụ, có thể được dùng để hiện thực một cổng RS232. Đây là dạng giao tiếp kết nối modem, máy in vào máy tính như đã đề cập ở phần trước. 8251 là một USART được dùng rộng rãi nhất. Đúng như tên của nó, 8251 có khả năng thực hiện một giao tiếp truyền bất đồng bộ hay đồng bộ. Để minh họa chúng ta sẽ khảo sát hoạt động của nó ở chế độ truyền bất đồng bộ và đồng bộ. Khả năng lập trình được của 8251 đã cung cấp một giao tiếp truyền bất đồng bộ khá linh hoạt. Nó chứa một bộ truyền và một bộ thu song công hoàn toàn (full-duplex), có thể được cấu hình thông qua phần mềm để truyền dữ liệu dùng khuôn dạng gồm từ 5 đến 8 bit ký tự, một bit kiểm tra chẵn hoặc lẻ hay không cần kiểm tra, cùng với 1 hay 1.5 hoặc 2 STOP bit. Hơn nữa, nó còn có khả năng phát hiện các lỗi bit, lỗi định dạng frame và lỗi chồng chập dữ liệu một cách tự động.

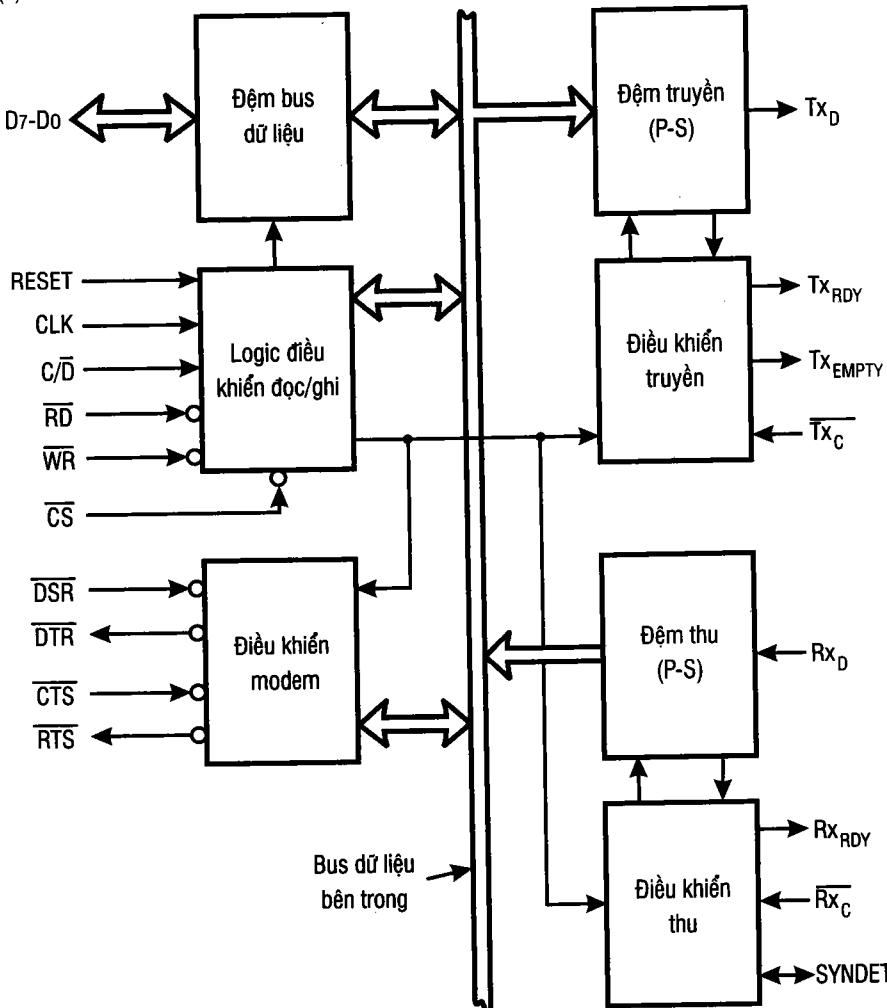
4.3.2. Kiến trúc và hoạt động của 8251 ở chế độ bất đồng bộ

Sơ đồ khối mô tả kiến trúc bên trong của 8251 được trình bày trên hình 3.18(a) và sơ đồ chân của chip được trình bày trên hình 3.18(b). Từ sơ đồ này chúng ta thấy rằng nó bao gồm bốn phần chính sau đây: phần giao tiếp bus, bao gồm bộ đệm bus dữ liệu và khối logic điều khiển đọc/ghi; phần truyền bao gồm bộ đệm truyền và khối điều khiển truyền; phần thu bao gồm bộ đệm thu và khối điều

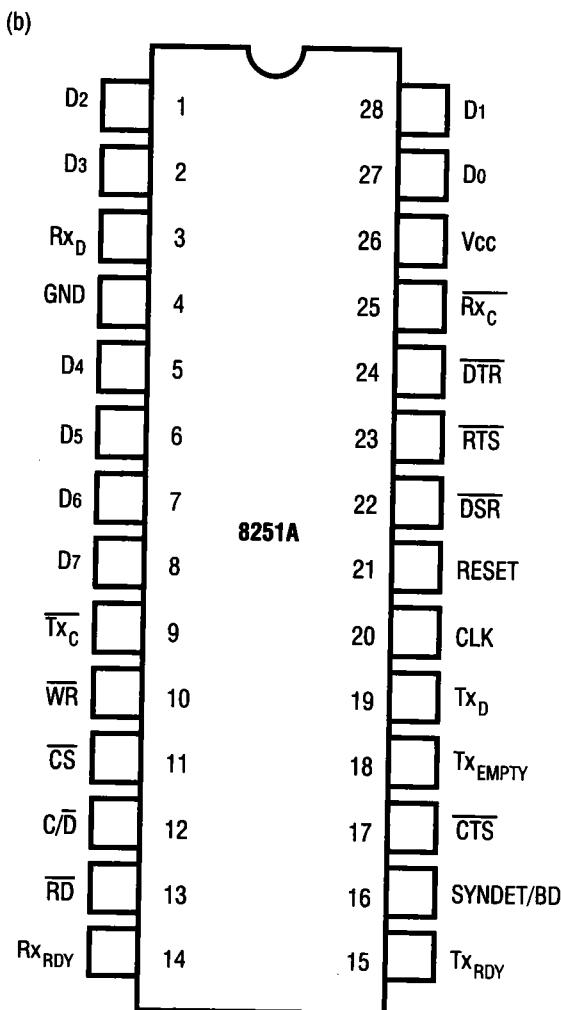
khiển thu; và cuối cùng là phần điều khiển modem. Chúng ta sẽ xem xét chi tiết hơn từng phần.

Phần giao tiếp bus được dùng để nối 8251 đến vi xử lý. Lưu ý rằng giao tiếp này bao gồm bus dữ liệu hai chiều 8-bit được ký hiệu từ D₀ đến D₇ được điều khiển bởi bộ đệm bus dữ liệu. Thông qua các đường dây này vi xử lý truyền các lệnh đến 8251, đọc thanh ghi trạng thái của nó, và xuất nhập ký tự.

(a)



Hình 3.18 Sơ đồ tóm tắt của 8251: (a) Sơ đồ khái niệm



Hình 3.18 (tiếp theo) Sơ đồ tổng quát của 8251: (b) Sơ đồ chân

Dữ liệu truyền qua bus được điều khiển bởi các tín hiệu C/D (control/data), RD (read), WR (write), và CS (chip select), chúng là tất cả các đường nhập vào khối điều khiển đọc ghi. Thông thường, 8251 được định vị tại một địa chỉ đặc biệt trong không gian địa chỉ xuất/nhập (I/O) hay không gian địa chỉ bộ nhớ của vi xử lý. Khi vi xử lý truy xuất các thanh ghi trong 8251, nó đặt địa chỉ này lên bus địa chỉ. Địa chỉ này được mã hóa bởi một mạch bên ngoài và phải tạo ra logic 0 tại ngõ vào CS. Tín hiệu nhập này phải ở mức logic 0 để một chu kỳ bus đọc hay ghi diễn ra đối với 8251.

Ba tín hiệu điều khiển khác, C/D, RD, và WR báo cho 8251 biết loại dữ liệu nào sẽ chuyển qua bus dữ liệu. Bảng 3.5 trình bày các hoạt động đọc ghi khác nhau có thể xảy ra. Ví dụ, trạng thái đầu tiên trong bảng C/D = 0, RD = 0, và WR = 1 tương ứng với một ký tự được truyền từ 8251 đến vi xử lý. Lưu ý RD = 0 báo cho biết vi xử lý đang đọc dữ liệu từ 8251, WR = 0 chỉ thị đang ghi dữ liệu ra 8251, và mức logic của C/D chỉ ra dữ liệu ký tự, thông tin điều khiển, hay thông tin trạng thái đang ở trên bus dữ liệu.

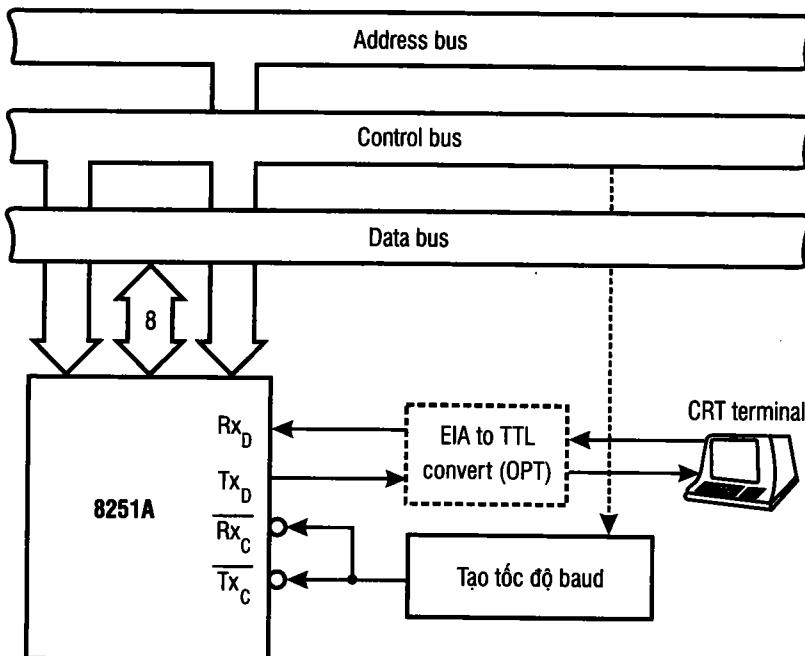
Bảng 3.5 Các hoạt động khác nhau trên bus

C/D	RD	WR	CS	Hoạt động
0	0	1	0	Dữ liệu từ 8251 đi vào bus dữ liệu
0	1	0	0	Dữ liệu từ bus dữ liệu đi ra 8251
1	0	1	0	Thông tin trạng thái đi vào bus dữ liệu
1	1	0	0	Thông tin điều khiển từ bus dữ liệu ra 8251
X	1	1	0	Bus dữ liệu ở trạng thái treo (trở kháng cao)
X	X	X	1	Bus dữ liệu ở trạng thái treo (trở kháng cao)

Phản thu chịu trách nhiệm đọc luồng bit dữ liệu nối tiếp từ ngõ vào Rx_D và chuyển đổi sang dạng song song. Khi phát hiện mức điện áp MARK trên đường dây này, bộ phản thu cho phép bộ đếm hoạt động. Khi bộ đếm gia tăng giá trị bằng với khoảng nửa thời bit, thì mức logic trên đường Rx_D lại được lấy mẫu một lần nữa. Nếu nó vẫn còn ở mức MARK, xem như đã phát hiện được một xung START hợp lệ. Sau đó Rx_D được kiểm tra mỗi khi bộ đếm tăng lên đúng một thời bit. Và cứ thế cho đến khi một ký tự hoàn chỉnh đã được tái hợp và đọc được STOP bit. Sau đó ký tự hoàn chỉnh được truyền vào thanh ghi dữ liệu thu.

Trong khi nhận một ký tự, bộ thu tự động kiểm tra để phát hiện lỗi. Nếu một trong các lỗi nào đó xảy ra, cờ tương ứng với lỗi sẽ được set (đặt) bằng cách xác lập bit tương ứng trong thanh ghi trạng thái. Sau đó ngõ ra Rx_{RDY} được chuyển lên mức logic 1. Tín hiệu này được gửi đến vi xử lý báo cho nó biết một ký tự đã có sẵn trong thanh ghi dữ liệu và nên đọc vào. Rx_{RDY} tự động trở về mức logic 0 khi CPU đọc nội dung của thanh ghi dữ liệu nhận.

8251 không có bộ phận phát sinh tốc độ baud bên trong. Vì lý do này, tín hiệu đồng hồ được dùng để xác lập tốc độ baud phải được tạo ra từ bên ngoài và được áp dụng vào đường tín hiệu Rx_C của máy thu. Thông qua phần mềm 8251 có thể thực hiện phép chia tần số tín hiệu đồng hồ bên trong đối với tín hiệu nhập Rx_C với thương số là 1, 16 hay 64.



Hình 3.19 Bộ phận truyền và thu được điều khiển bởi cùng tốc độ baud.

Bộ phận truyền làm việc ngược với phần thu. Nó nhận ký tự dữ liệu song song từ CPU qua bus dữ liệu. Sau đó định dạng ký tự này với START bit, parity bit thích hợp, một số STOP bit và đặt nó vào thanh ghi bộ đệm dữ liệu truyền. Sau cùng nó được dịch (shift) ra khỏi thanh ghi này để tạo luồng bit nối tiếp trên đường Tx_D. Khi thanh ghi bộ đệm dữ liệu truyền đã rỗng, ngõ ra Tx_{RDY} chuyển lên mức logic 1. Tín hiệu này được đưa đến CPU để báo cho nó biết có thể gửi tiếp một ký tự khác ra phần truyền. Khi CPU ghi ký tự dữ liệu ra thanh ghi bộ đệm truyền thì Tx_{RDY} tự động trở về mức logic 0.

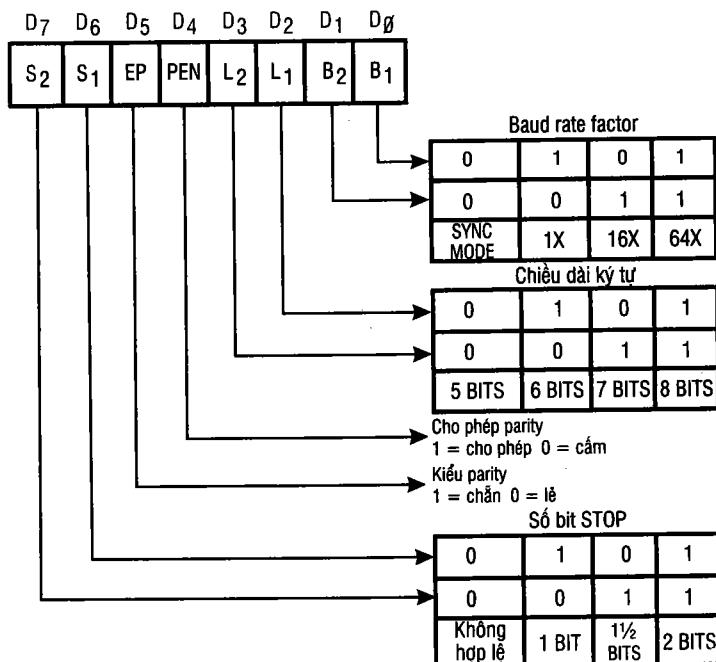
Dữ liệu được truyền lên dây với tốc độ được xác lập bởi tín hiệu đồng hồ truyền bên ngoài đặt vào ngõ vào Tx_C. Trong hầu hết các ứng dụng, hoạt động truyền và thu có cùng tốc độ baud. Do đó, cả Rx_C và Tx_C đều được cung cấp bởi cùng nguồn phát sinh tín hiệu đồng hồ. Mạch trên hình 3.19 trình bày cấu hình của loại hệ thống này.

Hoạt động của 8251 được điều khiển thông qua cài đặt các bit trong 3 thanh ghi điều khiển bên trong: thanh ghi điều khiển chế độ (mode control register), thanh ghi lệnh (command register), và thanh ghi trạng thái (status register). Ví dụ, phương thức mà bộ truyền và bộ thu của 8251 hoạt động được xác định bởi nội dung của thanh ghi điều khiển chế độ. Hình 3.20 trình bày tổ chức của thanh ghi điều khiển chế độ và chức năng của mỗi bit trong đó. Lưu ý hai bit có ý nghĩa thấp

nhất B₁ và B₂ xác định hoạt động theo đồng bộ hay bất đồng bộ và ngõ vào tốc độ baud từ bên ngoài được chia như thế nào ở bên trong 8251. Ví dụ, nếu hai bit này là 11, nó xác lập hoạt động bất đồng bộ với tốc độ baud vào được chia cho 64. Hai bit tiếp theo là L₁ và L₂ xác lập chiều dài của ký tự. Ví dụ khi thông tin đang được truyền nhận là các ký tự mã ASCII 7 bit thì các bit này sẽ được nạp là 10.

Hai bit kế tiếp, PEN và EP xác định có dùng kiểm tra parity hay không, nếu dùng thì là chẵn hay lẻ. Xem hình 3.20, chúng ta thấy rằng PEN cho phép hay cấm parity. Để cho phép parity, nó được đặt là 1. Hơn nữa, khi cho phép parity thì logic 0 trên EP chọn parity lẻ và logic 1 trên EP chọn parity chẵn. Để cấm parity, những gì cần làm là cho PEN = 0.

Chúng ta giả sử đang chọn chế độ truyền bất đồng bộ; do đó các bit S₁ và S₂ xác định số bit STOP được dùng. Lưu ý rằng nếu nạp vào hai bit này giá trị 11 thì ký tự được truyền có 2 bit STOP.

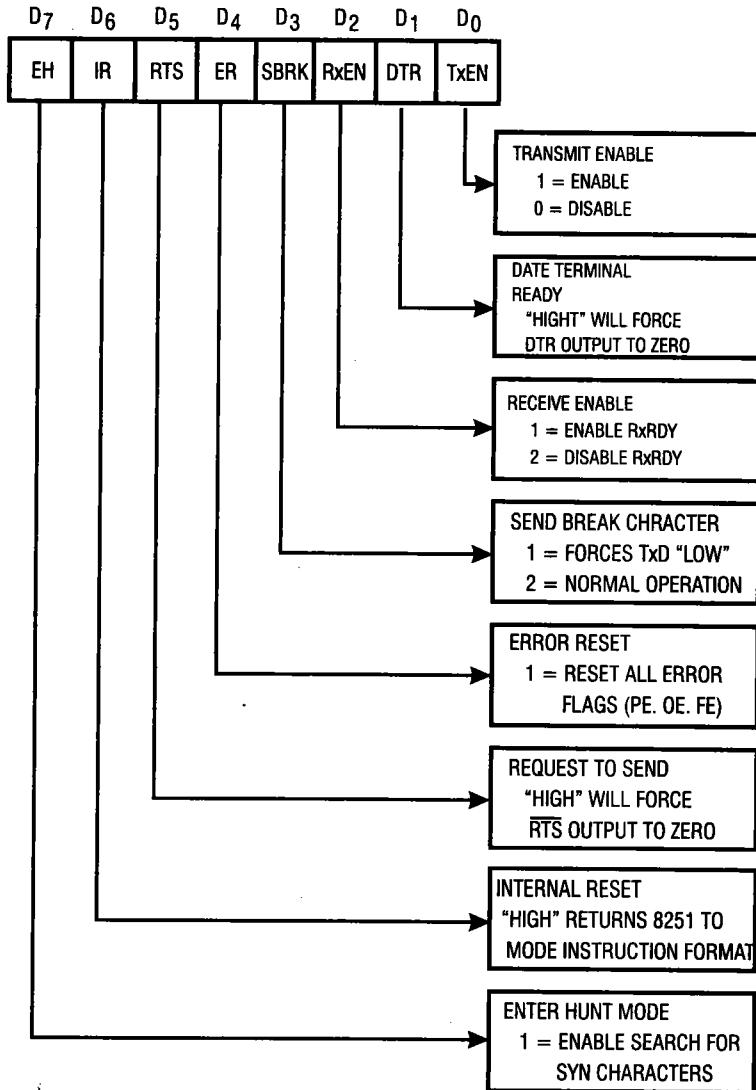


Hình 3.20 Thanh ghi điều khiển chế độ trong 8251

Một khi cấu hình truyền bất đồng bộ đã được thiết lập trong thanh ghi điều khiển chế độ, hoạt động của giao tiếp nối tiếp có thể được điều khiển bởi CPU bằng cách phát ra các lệnh vào thanh ghi lệnh trong 8251. Tổ chức của thanh ghi lệnh và chức năng của mỗi bit trong đó được trình bày trên hình 3.21.

T_{xEN} và R_{xEN} là các bit cho phép trong bộ truyền và bộ thu. Vì bộ truyền và bộ thu đều có thể hoạt động một cách đồng thời, nên hai bit này có thể cùng được set. R_{xEN} là tín hiệu cho phép thực sự đối với tín hiệu R_{xRDY}. Nó không đóng mở bộ thu. Bộ thu luôn hoạt động, nhưng nếu R_{xEN} được set là 0 thì 8251 không phát

tín hiệu cho CPU để báo cho biết đã nhận được một ký tự bằng cách đưa Rx_{RDY} lên mức logic 1. Tương tự, Tx_{EN} cho phép tín hiệu TxRDY.

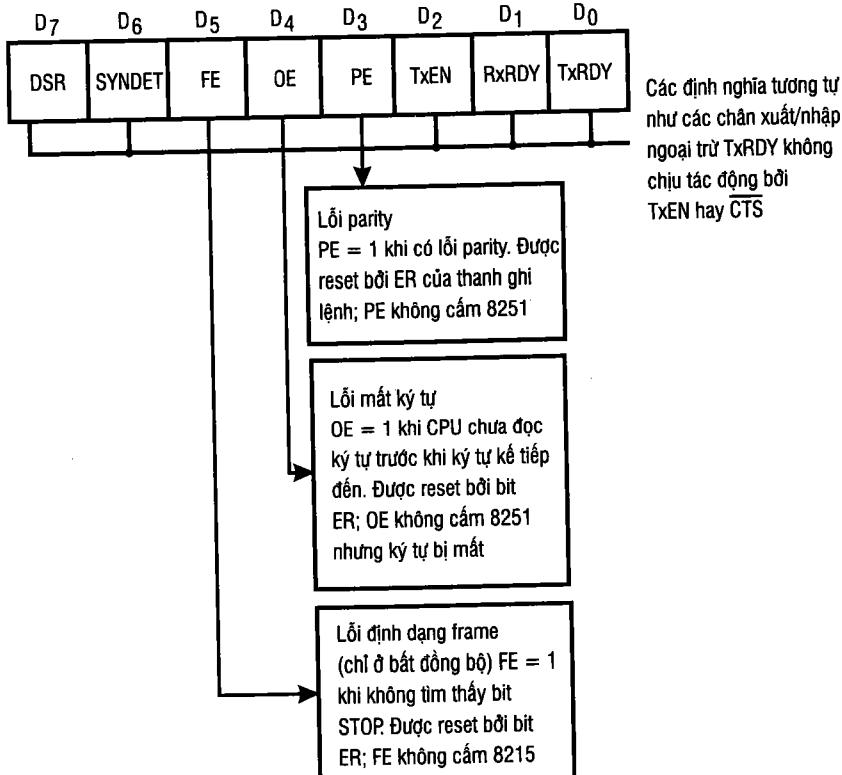


Hình 3.21 Thanh ghi lệnh trong 8251

FR bit của thanh ghi lệnh có thể được dùng để đưa các bit báo lỗi của thanh ghi trạng thái về giá trị mặc định (reset). Thanh ghi trạng thái của 8251 được trình bày trên hình 3.22. lưu ý rằng các bit PE, OE và FE là các cờ lỗi cho bộ thu. Nếu ký tự nhận bị phát hiện có một lỗi bit thì PE được set. Mặt khác, nếu lỗi định dạng



frame hay lỗi chồng chập ký tự xảy ra thì tương ứng là cờ FE và OE được set. CPU sẽ luôn luôn kiểm tra các bit lỗi này trước khi đọc một ký tự từ thanh ghi dữ liệu thu. Nếu lỗi xảy ra, một lệnh sẽ được gửi đến thanh ghi lệnh để ghi 1 vào bit ER. Điều này khiến tất cả 3 cờ lỗi trong thanh ghi trạng thái được reset. Sau đó một chương trình con có thể được khởi động để yêu cầu truyền lại ký tự này.



Hình 3.22 Thanh ghi trạng thái trong 8251

Chúng ta sẽ khảo sát một bit nữa của thanh ghi lệnh là bit IR, nó cho phép 8251 được khởi động lại dưới sự điều khiển của chương trình. Để khởi động thiết bị, CPU chỉ cần ghi giá trị vào bit IR này.

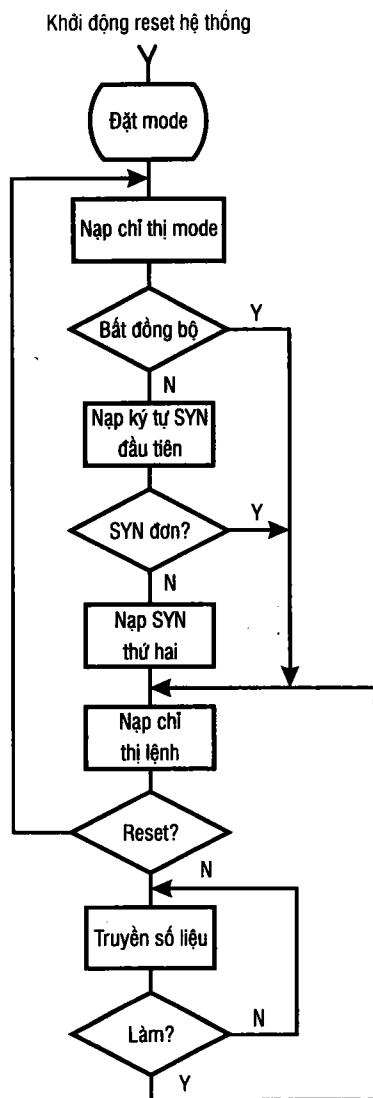
4.3.3. Hoạt động ở chế độ truyền đồng bộ của 8251A

Trong phương thức đồng bộ, mạch phát của 8251A tự động chèn ký tự điều khiển SYN vào trước mỗi khôi, không truyền các bit START và STOP. Ký tự SYN này được lập trình khi khởi tạo, thường dùng ký tự SYN của mã ASCII. Phần thu của USART tự động tìm ký tự SYN truyền đến và nếu thấy sẽ báo cho bộ vi xử lý biết.

Tần số xung đồng hồ (clock) dùng cho việc dịch các bit dữ liệu thu phải được cung cấp từ bên ngoài (ở chân Rx_C) và phải được đồng bộ ngoài với chuỗi bit dữ liệu đến. Tín hiệu Rx_C thường được cung cấp bởi modem đồng bộ. Còn giá trị chia (divisor) của tần số đồng hồ (clock) thu/ phát bên trong của 8251A sẽ cố định là 1.

Chân C/D ở mức cao sẽ cho phép chọn thanh ghi chế độ hoặc trạng thái, ở mức thấp sẽ chọn các thanh ghi đệm thu hoặc phát. Như vậy chân C/D bình thường được nối với đường địa chỉ A₀. Các thanh ghi chế độ và đệm phát chỉ cho phép ghi, còn các thanh ghi trạng thái và đệm thu chỉ cho phép đọc. Một số hoạt động chính của 8251A gồm có :

- **Khởi tạo 8251A**



Hình 3.23 Lưu đồ khởi động 8251A



Thủ tục khởi tạo cho 8251A ở phương thức đồng bộ chỉ khác phương thức bắt đồng bộ ở chỗ xác định mã của ký tự SYN, bao gồm các bước sau:

- Khởi động chip bằng chân RST (phần cứng) hoặc dùng bit 1 trong từ lệnh (phần mềm).
- Ghi từ điều khiển với các giá trị thích hợp vào thanh ghi chế độ.
- Ghi mã của ký tự SYN.
- Ghi từ lệnh với các giá trị thích hợp vào thanh ghi lệnh.

Trong thời gian ghi thì chân C/D phải ở mức cao. Byte ghi vào giữa từ điều khiển và từ lệnh sẽ được cài vào trong USART như là mã của ký tự SYN (USART tự nhận biết theo thứ tự).

Hoạt động khởi động tổng quát được trình bày qua lưu đồ ở hình 3.23.

- **Truyền ký tự**

Tương tự như ở phương thức bắt đồng bộ, để truyền 1 ký tự ở phương thức đồng bộ thì chân CTS phải ở mức thấp và bit 7 của thanh ghi lệnh (bit Tx_{EN}) phải có giá trị 1.

Chờ cho bit 0 của thanh ghi trạng thái (bit Tx_{RDY}) lên 1 hoặc chân Tx_{RDY} lên mức cao.

Ghi ký tự kế tiếp ra thanh ghi đệm phát.

Khi truyền xong ký tự cuối cùng, chân Tx_{EMPTY} lên mức cao và bit 2 của thanh ghi trạng thái (bit Tx_E) sẽ lên 1, USART tự động truyền các ký tự SYN trong khoảng thời gian rồi (nếu mạch phát được cho phép như vậy).

Các bit truyền được dịch ra ở các cạnh xuống của tín hiệu clock phát Tx_C (tác động ở cạnh xuống của xung đồng hồ).

- **Thu ký tự**

Để nhận một ký tự trong phương thức đồng bộ, lần lượt thực hiện các bước sau:

- Cài đặt bit B₀ ở thanh ghi lệnh lên 1 thông qua từ lệnh đầu tiên để cho phép tìm ký tự SYN.
- Chờ chân SYNDET lên mức cao hoặc bit SYNDET lên 1.
- Chờ chân Rx_{RDY} lên cao hoặc bit ở vị trí 1 của thanh ghi trạng thái (bit Rx_{RDY}) lên 1.
- Đọc một ký tự từ thanh ghi đệm thu.

- Đọc các bit kiểm tra sai từ thanh ghi trạng thái.

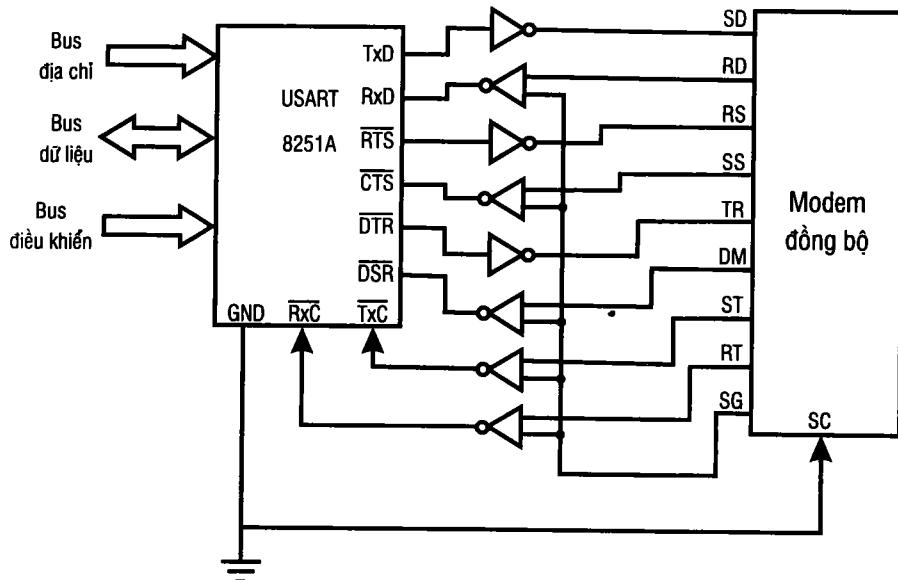
Các bit này sau đó phải được reset về 0 bằng cách đặt bit 3 của thanh ghi lệnh lên 1.

Các bit dữ liệu được dịch ở các sườn lên của xung clock thu Rx_C.

4.3.4. Giao tiếp 8251A

- Giao tiếp với modem

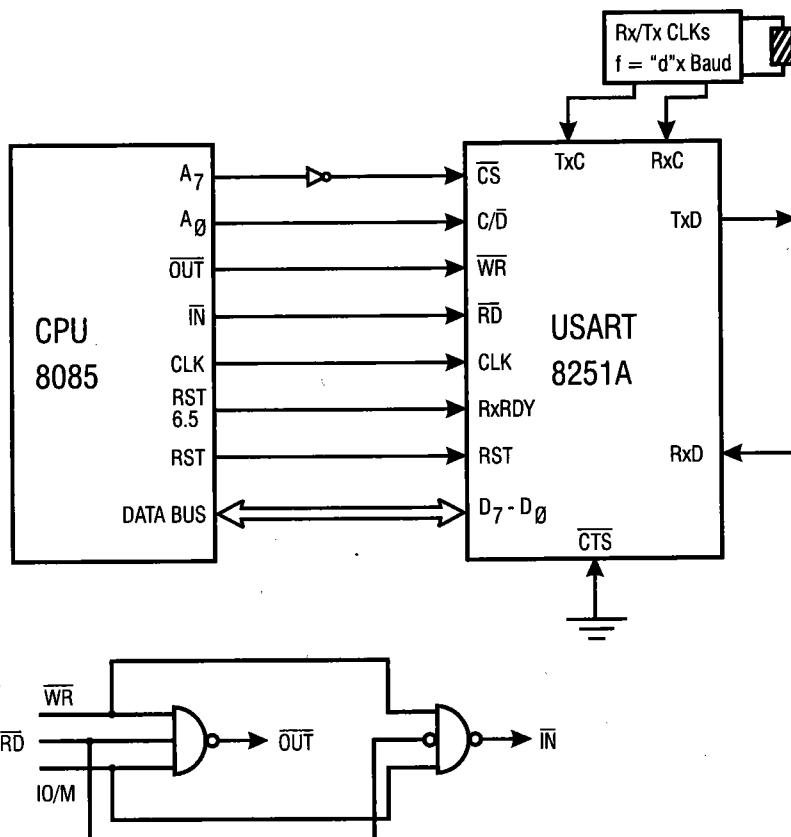
Hình 3.24 mô tả tổng quát phương thức giao tiếp USART 8251A với modem theo chuẩn giao tiếp RS-449. Các tín hiệu đồng hồ (clock) thu và phát sẽ do modem cung cấp.



Hình 3.24 Mạch giao tiếp với modem

- Giao tiếp với vi xử lý

Chip 8251A là loại NMOS được thiết kế phù hợp để giao tiếp với các họ vi xử lý 8086 của Intel. Hình 3.25 mô tả tổng quát giao tiếp giữa 8251A và CPU 8085 được điều khiển thông qua các ngõ IN, OUT. Trong mạch điện này CPU dùng cơ chế ngắt quãng (interrupt) trong mạch thu ký tự, khi chân Rx_{RDY} ở mức cao sẽ phát ra một ngắt vào CPU. Còn mạch phát CPU dùng cơ chế quét (poll) để kiểm tra bit Tx_{RDY} của thanh ghi trạng thái trước khi truyền ra USART một ký tự mới.



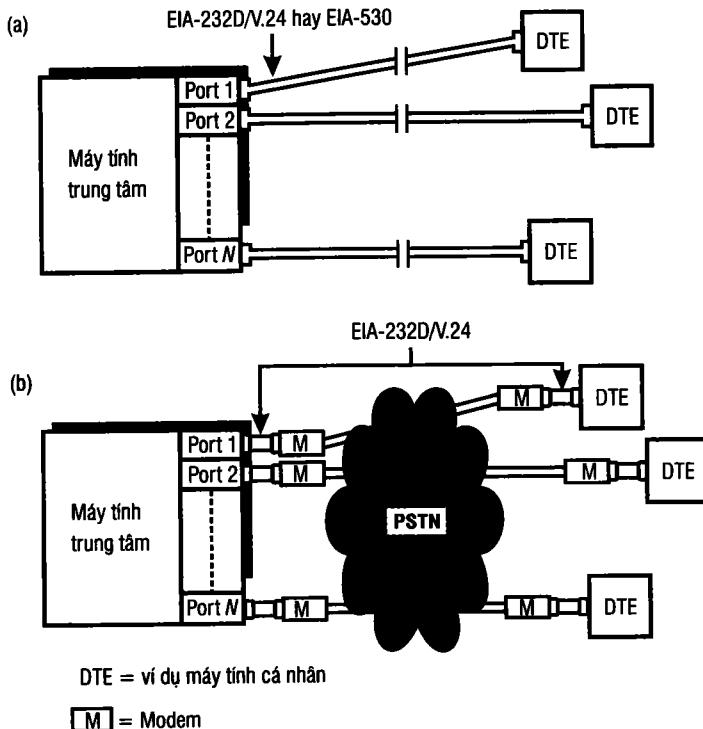
Hình 3.25 Mạch giao tiếp với vi xử lý

Đối với họ Motorola, chip đồng bộ thường dùng là NMOS 6852 với 24 chân, chip SSDA (Synchronous Serial Data Adapter) này có thể giao tiếp trực tiếp với các bus của CPU 6800. 6852 có một số tính năng mà 8251A không có. Ví dụ như mỗi vùng đệm thu phát gồm 3 byte tổ chức theo cơ chế hàng FIFO (first in first out) cho phép 6852 làm việc ở chế độ kép, trong đó CPU có thể ghi hoặc đọc liên tiếp hai ký tự mà không cần chờ. 6852 có đến 3 thanh ghi được dùng chỉ mục, cho phép truy xuất tất cả 7 thanh ghi của 6852 kết hợp với hai đường RS (register select) (nối với A₀) và R/W (read only/ write only).

5. Các thiết bị điều khiển truyền dữ liệu

5.1. Khái quát

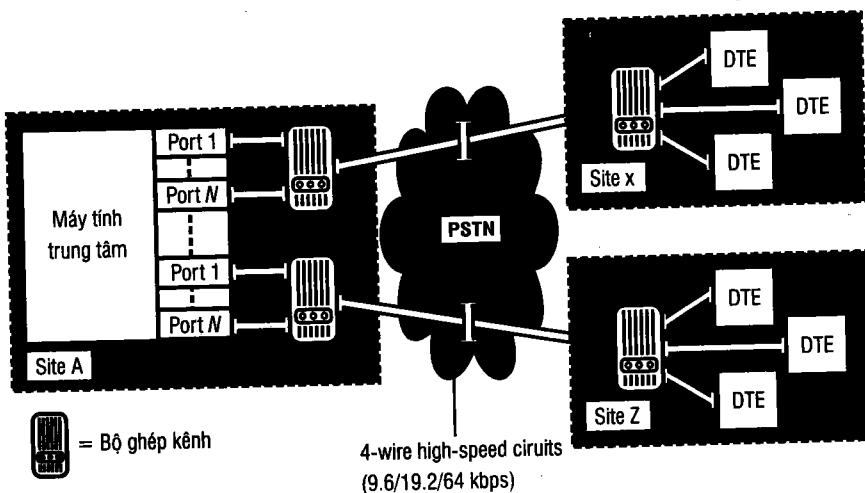
Trong nhiều ứng dụng truyền dữ liệu có một yêu cầu chung là phải đáp ứng sự phân tán các đầu cuối thông tin ví dụ như các máy tính cá nhân, tất cả các đầu cuối đều có nhu cầu truy xuất một dịch vụ tính toán trung tâm. Dịch vụ này có thể điều hành một dịch vụ thư điện tử trung tâm hay một cơ sở dữ liệu trung tâm.



Hình 3.26 Các mạng đầu cuối đơn giản: (a) Phân bố cục bộ (b) Phân bố xa

Nếu tất cả các đầu cuối đặt ở các vị trí khác nhau, chỉ có giải pháp cung cấp một đường thông tin riêng biệt cho mỗi đầu cuối, hình 3.26. Trong phần (a) giả sử rằng các đầu cuối được phân bố xung quanh nối trực tiếp vào một máy tính trung tâm, trong khi đó phần (b) giả sử rằng mỗi đầu cuối tọa lạc tại các vị trí khác nhau cách xa trung tâm. Trong trường hợp thứ hai, hầu hết đều phải cần một modem để thực hiện các cầu nối chuyển mạch hay leased line, tùy thuộc vào lượng dữ liệu cần truyền hay tần suất gọi. Trong trường hợp các cầu nối chuyển mạch, các đầu cuối thường phải có các dịch vụ quay số tự động liên quan đến giao tiếp thông tin.

Trong các ứng dụng có nhiều đầu cuối đặt gần nhau trong một phạm vi nào đó, có thể dùng một thiết bị gọi là bộ ghép kênh MUX (multiplexer) để tối thiểu số dây dẫn cần nối đến trung tâm. Các thiết bị này được dùng cùng với một đường truyền đơn có tốc độ cao hơn tốc độ truyền của các đầu cuối thành phần. Như trình bày trên hình 3.27 ở mỗi đầu của liên kết đều có một bộ ghép kênh như nhau. Trong phương pháp này, sự hiện diện các bộ ghép kênh là trong suốt đối với các đầu cuối và máy tính trung tâm.



Hình 3.27 Lược đồ mạng dựa vào bộ ghép kênh.

Có hai dạng thiết bị ghép kênh tổng quát : các bộ ghép kênh phân thời và các bộ ghép kênh thống kê. Một bộ ghép kênh phân thời phân phối cố định cho mỗi đầu cuối một phần khả năng truyền để cùng chia sẻ đường truyền tốc độ cao với các đầu cuối khác. Bộ ghép kênh thống kê chỉ phân phối khả năng truyền theo nhu cầu mang tính thống kê.

5.2. Bộ ghép kênh phân thời

Một ứng dụng tiêu biểu bộ ghép kênh phân thời được trình bày trên hình 3.28(a). Các đầu cuối tại một vùng cục bộ liên hệ với một ứng dụng thương mại nào đó, tất cả đều có nhu cầu truy xuất vào máy tính trung tâm. Ở đây giả sử rằng mỗi site có một số khá lớn các đầu cuối phát ra lượng tải vừa vẹn với tốc độ bit của các mạch đang được dùng để liên kết các site khác nhau đến site trung tâm. Thông thường là 64Kbps hay cao hơn, tùy vào số lượng đầu cuối.

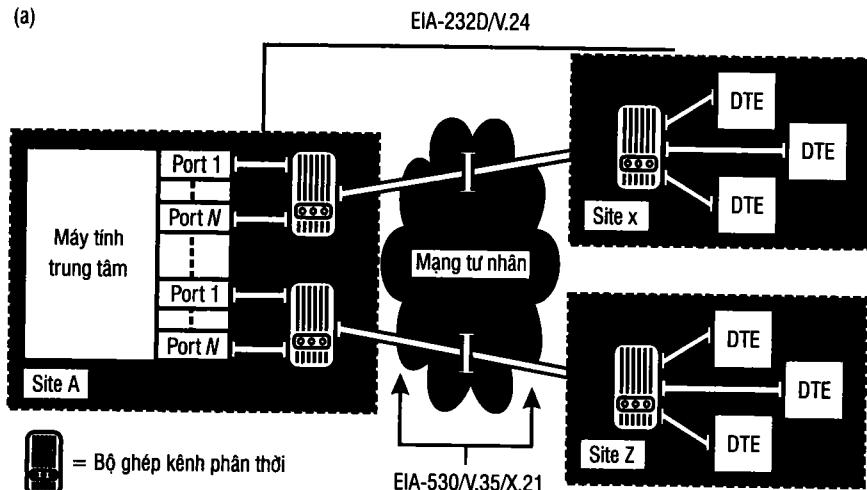
Hình 3.28(b) mô tả cấu trúc bên trong một MUX. Thường mỗi đầu cuối hoạt động theo chế độ truyền bắt đồng bộ và được kết nối đến UART. Các bộ xử lý bên trong MUX điều khiển truyền ký tự giữa các UART và mạch liên kết giao tiếp tốc độ cao. Mạch giao tiếp tốc độ cao hoạt động theo chế độ truyền đồng bộ hướng ký tự nên thường bao gồm một USART.

Để đảm bảo sự hiện diện của MUX là trong suốt đối với các đầu cuối và máy tính trung tâm, dung lượng truyền liên hệ với mạch tốc độ cao được phân chia theo phương thức sao cho các UART trong các cổng của đầu cuối và máy tính có thể hoạt động theo tốc độ đã được lập trình. Điều này có thể được thực hiện nhờ vào một kỹ thuật gọi là *thích nghi tốc độ*, kỹ thuật này liên quan đến việc phân dung lượng liên kết sẵn có thành một số các frame được trình bày trên hình 3.28(c).

Mỗi frame bao gồm N byte sao cho tốc độ bit liên hệ với một byte trong mỗi frame hình thành nên một tốc độ ghép kênh cơ bản thích hợp. Các tốc độ bit liên hệ

với mỗi đầu cuối được tạo ra nhờ dùng nhiều byte trong một frame. Tuy nhiên, không phải tất cả các bit trong mỗi frame đều được dùng cho dữ liệu hữu ích của user. Bit đầu tiên trong mỗi byte được dùng để đánh dấu frame; một mẫu bit cố định được gửi trong vị trí bit này của tất cả các byte trong một frame để máy thu có thể xác định bắt đầu và kết thúc của mỗi frame. Bit thứ tám cũng được dùng để truyền trạng thái của các bit điều khiển bắt tay ví dụ DSR/DTR và RTS/CTS liên hệ với các đường điều khiển giao tiếp V.24/EIA-232D của UART.

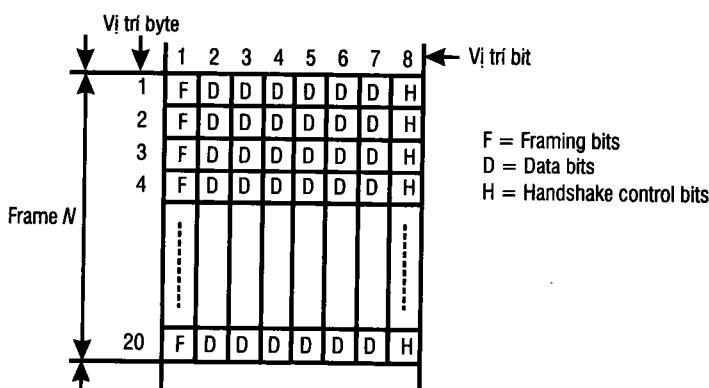
(a)



(b)



(c)



Hình 3.28 Nguyên lý bộ ghép kênh phân thời:
 (a) Ứng dụng (b) Lược đồ MUX (c) Sự thích nghi tốc độ.

Giả sử tốc độ bit của kênh tốc độ cao là 64kbps và dùng frame có chiều dài 20 byte. Tốc độ dữ liệu ứng với với 6 bit D trong mỗi byte là 2400bps. Do đó kênh tốc độ cao có thể được dùng để hỗ trợ một trong:

- 20 đầu cuối tốc độ 2400bps.
- 10 đầu cuối tốc độ 4800bps.
- 5 đầu cuối tốc độ 9600bps.
- 1 đầu cuối tốc độ 48kbps.

Điều đương nhiên là một hỗn hợp của những đầu cuối này đều có thể được hỗ trợ, ví dụ:

- 8 đầu cuối 2400bps.
- 4 đầu cuối 4800bps.
- 1 đầu cuối 9600bps.

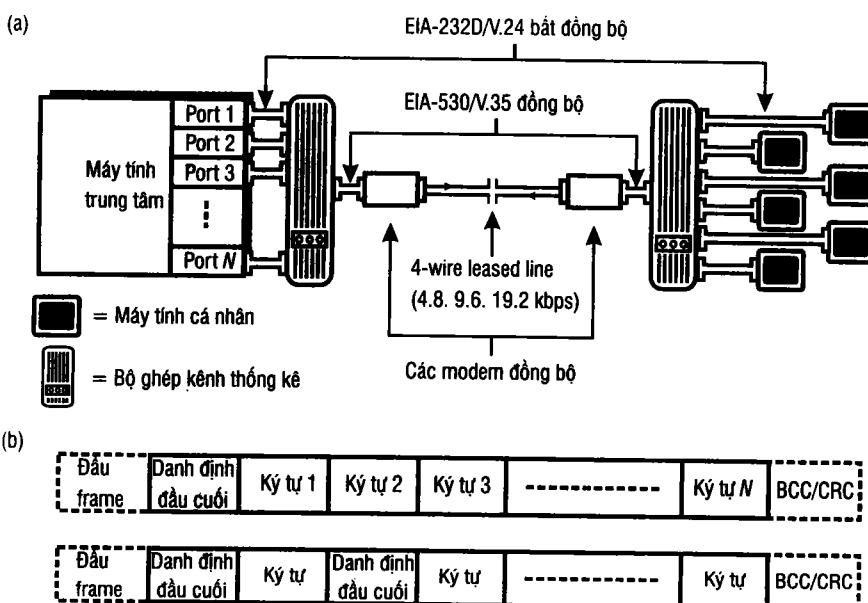
Để thực hiện hoạt động ghép kênh, vi xử lý dùng hai bộ đệm 2 byte cho mỗi UART_một cặp cho truyền và một cặp cho nhận. Để truyền, mỗi byte nhận từ UART được lưu trữ đơn giản theo phương pháp lặp trong bộ đệm 2 byte (vòng). Đồng thời, vi xử lý đọc nội dung hiện hành của mỗi bộ đệm 2 byte theo các đoạn 6-bit đồng bộ với tốc độ bit của liên kết tốc độ cao. Thủ tục theo chiều ngược lại cũng được xúc tiến để tiếp nhận từ liên kết tốc độ cao dùng bộ đệm kia. Các bit điều khiển bắt tay được cài theo phương pháp đã thống nhất trước nhằm phản ánh trạng thái tương ứng của đường dây liên quan đến giao tiếp tương ứng.

5.3. Bộ ghép kênh thống kê (statistical multiplexer)

Mỗi đầu cuối trong một bộ ghép kênh phân thời liên quan đến một khe ký tự cố định trong mỗi frame. Nếu đầu cuối hay máy tính không có dữ liệu truyền khi bộ xử lý quét qua UART tương ứng, vi xử lý phải chèn các ký tự NUL vào trong khe này, vì thế rất không hiệu quả trong việc sử dụng băng thông đường truyền sẵn có. Nếu liên kết dữ liệu thuộc sở hữu tư nhân thì điều này không cần bận tâm, nhưng nếu là đường truyền công cộng thì nó có ảnh hưởng lớn. Phương pháp ghép kênh hiệu quả hơn là ghép kênh thống kê (statistical multiplexing).

Các bộ ghép kênh thống kê hoạt động theo nguyên lý tốc độ trung bình dữ liệu của ký tự nhập tại một đầu cuối thường thấp hơn nhiều so với khả năng truyền dẫn của đường dây. Nếu dùng tốc độ dữ liệu của người dùng thay vì tốc độ của đường truyền thì tốc độ bit của liên kết dữ liệu chung thấp hơn nhiều và giá thành giảm đi đáng kể. Giả sử một vị trí ở xa có 8 đầu cuối cần nối đến một máy

tính trung tâm ở xa qua đường truyền dẫn công cộng, giả sử đường truyền có tốc độ tối đa là 4800bps. Dùng một MUX cơ bản và một đường dây đơn, tốc độ hoạt động danh định của mỗi đầu cuối phải nhỏ hơn 600bps, giả sử là 300bps. Ảnh hưởng của giới hạn này là thời gian đáp ứng của máy tính đối với mỗi ký tự được gõ vào tại đầu cuối thường thấp hoặc nếu một khôi ký tự đang được truyền đến đầu cuối thì thời gian trễ có thể nhận thấy được. Dĩ nhiên, nếu tốc độ dữ liệu trung bình của đầu cuối là 300bps thì với bộ ghép kênh thống kê, dữ liệu có thể được truyền bởi một đầu cuối với tốc độ tối đa có thể là 4800bps. Do đó, thời gian đáp ứng trung bình đối với mỗi ký tự gõ vào được cải thiện đáng kể.



Hình 3.29 Nguyên lý bộ ghép kênh thống kê:
(a) Lược đồ mạng (b) Các dạng frame.

Để hiện thực lược đồ này, bộ xử lý điều khiển trong bộ ghép kênh thống kê không chỉ thực hiện chức năng thăm dò (quét) thông thường đến các UART trong các đầu cuối liên quan, mà còn cung cấp và quản lý một số lượng phương tiện lưu trữ tạm thời để cho phép điều kiện quá tải trên liên kết số liệu chung khi có một số đầu cuối truy xuất đồng thời. Vì các ký tự được truyền trên liên kết số liệu chung căn cứ theo thống kê thay cho sự phân phối trước, nên mỗi ký tự hay nhóm ký tự được truyền cũng phải mang thêm thông tin nhận dạng.



Chức năng khác của bộ vi xử lý trong bộ ghép kênh có liên quan đến kiểm soát lỗi. Với một MUX dạng thường việc kiểm tra echo (dội lại) đơn giản thông thường là một chế độ làm việc hoàn toàn có thể chấp nhận được, vì mỗi cổng của đầu cuối hay máy tính được phân phối một khe ký tự cố định trong mỗi frame trên liên kết song công. Tuy nhiên, vì chế độ làm việc thống kê của bộ ghép kênh thống kê, nên chúng ta thường thực hiện kiểm soát lỗi đối với từng khối ký tự trên liên kết số liệu dùng chung. Một cấu hình tiêu biểu được trình bày trên hình 3.29(a).

Để giảm lượng overhead liên kết với mỗi ký tự truyền, chúng ta thường nhóm một số các ký tự lại với nhau để truyền lên liên kết dùng chung. Có thể thực hiện theo một số cách, trên hình 3.29(b) là hai ví dụ. Trong cách thứ nhất, bộ xử lý điều khiển đợi cho đến khi có được một số ký tự đến từ một đầu cuối ví dụ một chuỗi ký tự tạo nên một đường sau đó truyền chúng như một khối hoàn chỉnh kèm theo danh định đầu cuối đặt tại đầu của khối. Trong cách thứ hai, mỗi khối chứa một hỗn hợp các ký tự từ tất cả các đầu cuối đang hoạt động, mỗi ký tự đều có một danh định đầu cuối riêng biệt. Như chúng ta thấy trên hình, tập hợp các ký tự chiếm toàn bộ I-field của mỗi khối hay frame được truyền trên liên kết dùng chung. Giao thức truyền được dùng trên liên kết thường là một giao thức đồng bộ thiên hướng bit hay thiên hướng ký tự, chi tiết của các giao thức sẽ được đề cập trong chương kế tiếp.

5.4. Các thiết bị làm việc theo chế độ khối (block-mode)

Nhiều đầu cuối phức tạp hơn ví dụ như các đầu cuối nhập dữ liệu ngân hàng, các đầu cuối bán hàng thường hoạt động theo chế độ khối thay vì chế độ ký tự. Trong chế độ khối, khi mỗi ký tự được nhập vào sẽ được dội lại màn hình một cách trực tiếp bởi bộ xử lý cục bộ. Sau đó dữ liệu được chuyển đến trung tâm xử lý, tại đây một khối dữ liệu hoàn chỉnh được gói lại. Các đầu cuối như vậy thường hỗ trợ nhiều giao thức truyền khối phức tạp hơn để điều khiển truyền các bản tin. Cũng vì thời gian đáp ứng có thể chấp nhận đối với mỗi bản tin truyền chậm hơn nhiều so với những gì mong đợi trong một mạng tương tác theo chế độ ký tự, nên trang thiết bị chấp nhận lăng phí thời gian đáp ứng để giảm giá thành đường truyền thường được dùng trong các mạng đầu cuối theo chế độ khối.

5.4.1. Các đường dây đa điểm

Một phương pháp thông thường nhằm giảm giá thành đường dây thông tin trong mạng gồm các đầu cuối hoạt động theo chế độ khối là dùng các đường dây đa điểm (multidrop line). Hình 3.30(a) minh họa một mạng dùng các đường



dây đa điểm. Thay vì kết nối các đầu cuối một cách trực tiếp đến máy tính trung tâm bằng các dây riêng biệt, một số các đầu cuối phải cùng chia sẻ một đường dây. Bằng cách này sẽ giảm đi số lượng dây (và vì thế giảm số lượng modem hay các bộ điều khiển đường dây tùy vào vùng phủ của mạng). Tuy nhiên, chỉ một đường dây cho mỗi đầu cuối truyền tin thì chỉ có một khối thông điệp được gửi tại một thời điểm, bởi đầu cuối hoặc bởi máy tính trung tâm. Do đó, tất cả các hoạt động truyền trên mỗi đường dây đều được điều khiển bởi máy tính trung tâm dùng một thủ tục phân xử gọi là *poll-select*.

5.4.2. Poll-select

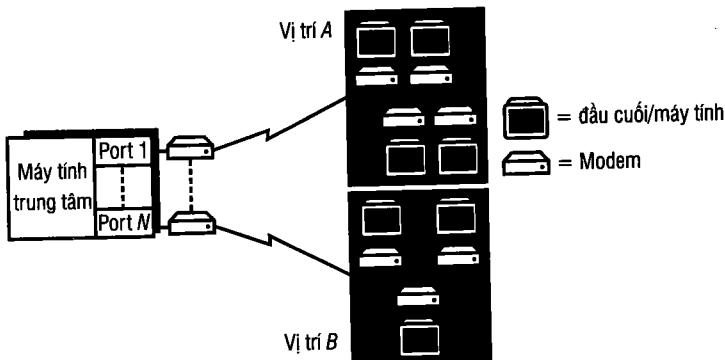
Để đảm bảo rằng chỉ một thông điệp được truyền tại bất kỳ thời điểm nào trên mỗi đường dây dùng chung, máy tính trung tâm hay đại diện cho nó sẽ quét (poll) hay chọn (select) mỗi đầu cuối được kết nối vào đường dây theo một tuần tự đặc biệt. Khi mỗi đầu cuối kết nối vào đường dây dùng chung sẽ được phân phối một danh định duy nhất, máy tính trung tâm sẽ thông tin với một đầu cuối bằng cách gửi thông điệp với danh định của đầu cuối được gắn ở đầu thông điệp (header). Thông điệp có thể ở hai dạng tổng quát là : điều khiển hay dữ liệu.

Theo định kỳ và lần lượt máy tính trung tâm sẽ gửi cho mỗi đầu cuối một thông điệp *poll* , điều này có ý nghĩa là hỏi đầu cuối có muốn truyền dữ liệu hay không. Nếu muốn, đầu cuối sẽ đáp ứng bằng một thông điệp dữ liệu, ngược lại nó sẽ đáp ứng bằng một thông điệp điều khiển có nghĩa là không có gì để truyền. Tương tự, bất cứ khi nào máy tính trung tâm muốn truyền dữ liệu đến đầu cuối, nó truyền một thông điệp điều khiển chọn hướng đến đầu cuối này. Giả sử đầu cuối được chọn có thể nhận một thông điệp, nó sẽ đáp ứng bằng cách gửi đáp bằng một thông điệp điều khiển báo sẵn sàng nhận. Sau đó, máy tính trung tâm sẽ bắt đầu truyền dữ liệu cho đầu cuối. Khi đã hoàn tất truyền máy tính trung tâm lại tiếp tục quét hay chọn các đầu cuối khác. Dạng quét này được gọi là *roll-call polling*, với cách làm này sẽ dẫn đến thời gian đáp ứng kéo dài trong các mạng lớn, vì mỗi đầu cuối trong mạng phải được quét hay được chọn trước khi có thể truyền hay nhận dữ liệu. Lượng thông tin overhead mà máy tính trung tâm gánh vác là rất nặng.

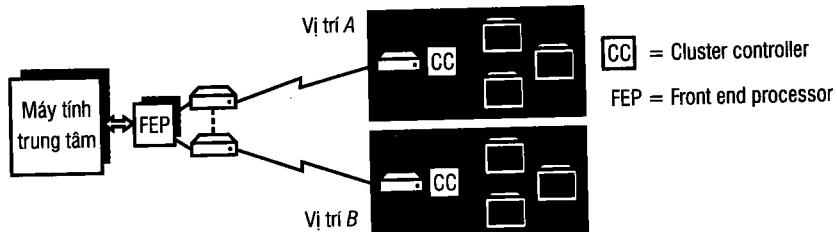
Để khắc phục các vấn đề này, một dạng phổ biến hơn trong đó các mạng đa điểm dùng bộ điều khiển theo nhóm gọi là *cluster controller* nhằm giảm thời gian đáp ứng của mạng và một bộ xử lý kết cuối gọi là *front-end processor* (FEP) để giảm lượng overhead trên máy tính trung tâm. Ví dụ của mạng này được minh họa trên hình 3.30(b). Mỗi bộ *cluster controller* đóng vai trò đại diện cho máy tính

trung tâm để quét và chọn các đầu cuối nối vào nó, do đó quản lý tất cả các thông điệp được truyền đến và đi từ các đầu cuối. Máy tính trung tâm hay FEP chỉ cần quét hay chọn mỗi bộ điều khiển.

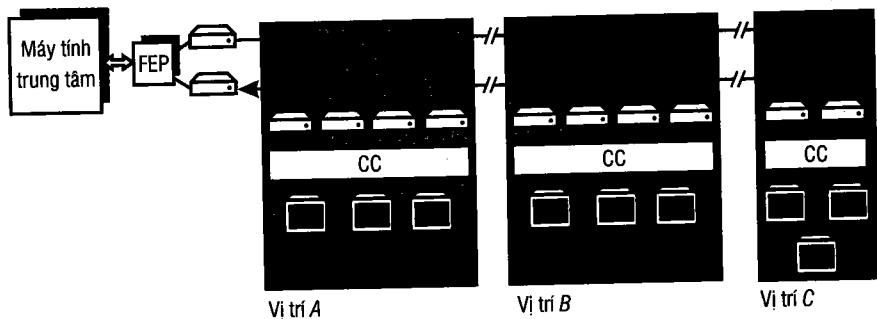
(a)



(b)



(c)

**Hình 3.30 Các mạng polling:****(a) Multidrop (b) Cluster controller (c) Hub polling.**

Một FEP thông thường là một máy tính nhỏ đi đôi với máy tính trung tâm. FEP được lập trình để kiểm soát tất cả các hoạt động quét và chọn, cho phép máy tính trung tâm dành hết thời gian của nó cho tác vụ xử lý ứng dụng chủ yếu. Ưu điểm của FEP là máy tính trung tâm chỉ cần can thiệp vào khi một thông điệp dữ

liệu được nhận hay được truyền. Hơn thế nữa, vì tất cả các overhead liên hệ đến mỗi thông điệp truyền đều được kiểm soát bởi FEP nên máy tính trung tâm chỉ cần khởi động truyền mỗi thông điệp đi và đến từ FEP.

Nếu các nhóm đầu cuối phân bố trên một vùng rộng, thì dùng một cơ cấu luân phiên gọi là **hub polling** sẽ rất kinh tế. Hub polling được trình bày trên hình 3.30(c). Trong cấu hình này, mỗi bộ điều khiển nhóm được kết nối đến láng giềng gần nhất của nó thay vì nối đến máy tính trung tâm. Như đề cập phía trước, máy tính trung tâm quản lý tất cả các hoạt động đến và đi từ các bộ điều khiển nhóm. Máy tính trung tâm chọn và gửi một thông điệp dữ liệu đến bất kỳ bộ điều khiển nào vào bất cứ lúc nào thông qua đường dây được vẽ phía trên cùng trong hình 3.30(c).

Để nhận thông điệp từ các bộ điều khiển, máy tính trung tâm gửi một thông điệp quét đến bộ điều khiển xa nhất, bộ điều khiển này sẽ đáp ứng bằng cách gửi một thông điệp dữ liệu hoặc gửi thông điệp điều khiển để báo rằng không có gì để gửi theo đường dây phía dưới (trên hình) đến bộ điều khiển láng giềng gần nhất của nó. Khi nhận thông điệp này, bộ điều khiển kế tiếp này biên dịch ra như là một thông điệp quét và nếu nó đang đợi truyền thì sẽ đáp ứng bằng cách thêm thông điệp dữ liệu của nó vào phía dưới của thông điệp vừa nhận từ láng giềng truyền lên (upstream) của nó. Sau đó thông điệp mới tạo lập được truyền đến láng giềng truyền xuống (downstream) của nó và vẫn dùng đường dây phía dưới (trong hình). Thủ tục này tiếp tục theo một chuỗi mắc xích, mỗi bộ điều khiển sẽ thêm thông điệp đáp ứng của nó vào thông điệp ghép trong quá trình chuyển thông điệp đến máy tính trung tâm. Sau cùng, khi nhận thông điệp ghép, FEP tách thông điệp ra và chuyển bất kỳ thành phần thông điệp hợp lệ nào chứa trong đó đến máy tính trung tâm để xử lý tiếp.

CHƯƠNG 4

XỬ LÝ SỐ LIỆU TRUYỀN

1. MÃ HÓA SỐ LIỆU MỨC VẬT LÝ

Số liệu cung cấp từ máy tính hoặc các thiết bị đầu cuối số liệu thường ở dạng nhị phân đơn cực (unipolar) với các bit 0 và 1 được biểu diễn với cùng mức điện áp âm hoặc dương. Tốc độ truyền của chúng được tính bằng số bit truyền trong một giây.

Các số liệu này khi truyền đi sẽ được biến đổi sang dạng tín hiệu số theo những kỹ thuật mã hóa khác nhau. Các tín hiệu này được đặc trưng bằng sự thay đổi mức điện, tốc độ truyền của chúng vì thế được xác định bằng tốc độ của sự thay đổi này còn được gọi là tốc độ điều chế và được tính bằng baud.

Các phương pháp mã hóa thông dụng bao gồm:

- Mã hóa NRZ
- Mã luồng cực
- Mã nhị pha
- Mã Miller
- Mã nhị phân đa mức

Để so sánh các loại mã này người ta căn cứ vào các yếu tố như phổ tín hiệu, khả năng đồng bộ tín hiệu, khả năng phát hiện sai, khả năng chống nhiễu và giao thoa tín hiệu, độ phức tạp và khả thi. Phổ tần của tín hiệu sau khi mã hóa sẽ quyết định đến một số khía cạnh của việc truyền số liệu như độ rộng băng tần cần thiết, khả năng ghép nối với đường truyền liên quan đến tín hiệu có thành phần một chiều (DC) hay không, nhiều hay ít. Nếu tín hiệu không có thành phần một chiều thì có thể ghép bằng biến áp, nhờ đó cách ly đường truyền bên ngoài với mạch thu phát bên trong, giảm sự giao thoa do ảnh hưởng của dòng một chiều. Khả năng đồng bộ tín hiệu liên quan đến đặc tính chuyển trạng thái của tín hiệu được mã hóa giúp xác định thời điểm bắt đầu và kết thúc của mỗi bit chính xác và thuận lợi hơn. Tùy theo phương pháp mã hóa có thể cung cấp khả năng phát hiện sai một cách đơn giản.

2. PHÁT HIỆN LỖI VÀ SỬA SAI

2.1.Tổng quan

Như đã đề cập ở các chương trước, khi dữ liệu được truyền giữa hai DTE, các tín hiệu điện đại diện luồng bit truyền rất dễ bị thay đổi do sự thâm nhập điện từ cảm ứng lên các đường dây từ các thiết bị điện gần đó, đặc biệt nếu các đường dây tồn tại trong môi trường xuyên nhiễu ví dụ như mạng điện thoại công cộng.

Điều này có nghĩa là các tín hiệu đại diện cho bit 1 bị máy thu dịch ra như bit nhị phân 0 và ngược lại. Để xác suất thông tin thu được bởi DTE đích giống như thông tin đã truyền đạt giá trị cao, cần phải có một vài biện pháp để nơi thu có khả năng biết được thông tin thu có chứa lỗi hay không. Hơn nữa, nếu phát hiện được lỗi, sẽ có một cơ cấu thích hợp để thu về bản copy chính xác của thông tin.

Có hai tiếp cận cho vấn đề này:

- Kiểm soát lỗi hướng tới (forward error control), trong đó mỗi ký tự hay frame dữ liệu được truyền sẽ chứa một vài thông tin bổ sung nhằm giúp máy thu không chỉ phát hiện lỗi mà còn xác định lỗi nằm ở đâu trong luồng bit truyền. Sau đó chỉ cần đảo ngược các bit lỗi để có được thông tin chính xác.
- Kiểm soát lỗi quay lui (feedback error control), trong đó mỗi ký tự hay frame dữ liệu được truyền chỉ chứa lượng thông tin đủ cho máy thu phát hiện lỗi, không thể xác định vị trí bit bị lỗi trong luồng dữ liệu thu được. Tuy nhiên sẽ có một lược đồ truyền lại để máy phát truyền bản copy khác của thông tin bị sai này.

Trong thực tế số lượng bit thêm vào để đạt được một độ tin cậy cần thiết trong điều khiển lỗi hướng tới sẽ gia tăng nhanh chóng khi số lượng bit thông tin tăng lên. Do vậy, điều khiển lỗi quay lui là phương pháp được dùng nhiều hơn trong các dạng truyền số liệu và các hệ thống mạng. Chúng ta sẽ trở lại với phương pháp điều khiển lỗi hướng tới qua một mô tả ngắn ở phần sau.

Điều khiển lỗi quay lui có thể được chia thành hai phần:

- (i) Các kỹ thuật được dùng để đạt được khả năng phát hiện lỗi tin cậy
- (ii) Các giải thuật điều khiển sẵn có được dùng để thực hiện các lược đồ điều khiển truyền lại.

Trong phần này chúng ta sẽ đề cập đến các kỹ thuật phát hiện lỗi thông dụng nhất đang được dùng. Một số giải thuật điều khiển truyền lại sẽ được trình bày trong chương sau.

Có hai yếu tố xác định dạng lược đồ phát hiện lỗi được dùng là tỉ lệ lỗi bit BER (Bit Error Rate) của đường dây hay mạch và loại lỗi. Loại lỗi chỉ ra rằng lỗi xảy ra là lỗi đơn bit ngẫu nhiên hay lỗi thuộc nhóm chuỗi bit liên tục. Các lỗi nhóm chuỗi bit liên tục được xem là các lỗi khối. BER chính là xác suất (P) một bit bị sai trong một khoảng thời gian xác định. Do đó nếu BER bằng 10^{-3} có nghĩa là trung bình cứ 1000 bit truyền sẽ có 1 bit lỗi trong một thời gian truyền này.

Nếu chúng ta truyền các ký tự đơn dùng phương pháp truyền bất đồng bộ (ví dụ 8 bit trong một ký tự cộng 1 start bit và một stop bit), xác suất một ký tự bị sai là $[1 - (1-P)^8]$, xấp xỉ 10^{-2} . Nếu truyền các khối ký tự dùng phương pháp truyền đồng bộ (ví dụ 125 ký tự trong một khối, mỗi ký tự có 8 bit), thì xác suất một khối bị lỗi xấp xỉ 1. Điều này có nghĩa là trung bình mỗi khối sẽ chứa một lỗi và phải được truyền. Rõ

ràng, chiều dài của frame là quá dài đối với loại đường dây này và chúng phải được giảm xuống để đạt được một lưu lượng có thể chấp nhận được.

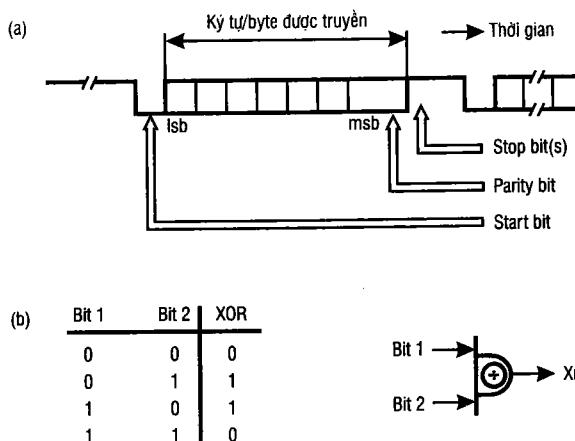
Dạng xuất hiện lỗi là yếu tố rất quan trọng, vì như chúng ta sẽ thấy, các kiểu lược đồ phát hiện lỗi khác nhau sẽ được dùng để phát hiện các lỗi khác nhau. Một số bit cũng được dùng trong một số lược đồ nhằm xác định chiều dài khồi được phát hiện. Có ba lược đồ được dùng rộng rãi nhất là lược đồ chẵn lẻ (parity), kiểm tra tổng BSC (Block Sum Check) và kiểm tra CRC (Cyclic Redundancy Check). Chúng ta sẽ xem xét từng lược đồ trong các mục sau đây.

2.2. Phương pháp kiểm tra bit chẵn lẻ (parity bit)

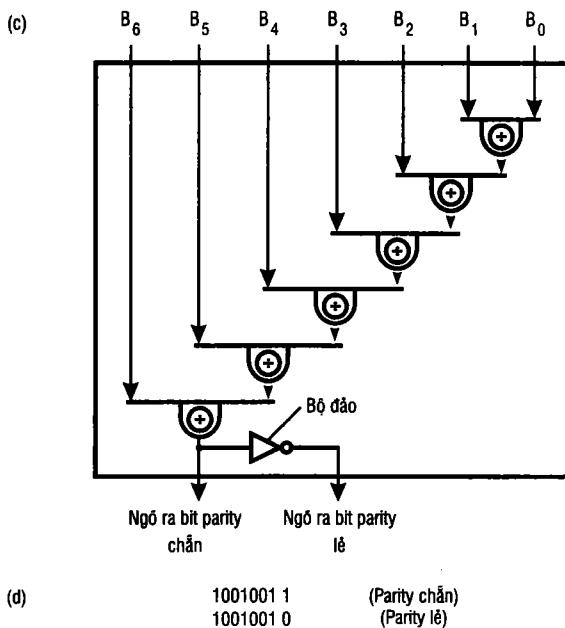
Phương pháp thông dụng nhất được dùng để phát hiện các lỗi của bit trong truyền dữ liệu bất đồng bộ và truyền dữ liệu đồng bộ hướng ký tự là phương pháp **parity bit**. Với lược đồ này máy phát sẽ thêm vào mỗi ký tự truyền một bit kiểm tra parity đã được tính toán trước khi truyền. Khi tiếp nhận thông tin, máy thu sẽ thực hiện các thao tác tính toán tương tự trên các ký tự thu được, và so sánh kết quả với bit parity thu được. Nếu chúng bằng nhau, được giả sử là không có lỗi xảy ra, ở đây dùng từ giả sử bởi vì với lược đồ này có thể không phát hiện được lỗi trong khi lỗi vẫn tồn tại trong dữ liệu. Nhưng nếu chúng khác nhau thì chắc chắn một lỗi xảy ra.

Để tính toán parity bit cho một ký tự, số các bit 1 trong mã ký tự được cộng với nhau (modulo 2) và parity bit được chọn sao cho tổng số các bit 1 (bao gồm cả parity bit này) là chẵn_even parity hoặc lẻ_odd parity. Các nguyên lý của lược đồ này được trình bày trên hình 4.1.

Hai ví dụ trên hình 4.1(d) cho thấy phương pháp parity bit chỉ phát hiện các lỗi đơn bit (hay là số lượng bit bị lỗi là một số lẻ) và không thể phát hiện các lỗi hai bit (hay số bit lỗi là một số chẵn).



Hình 4.1 Phương pháp parity bit:
(a) Vị trí trong ký tự (b) Bảng sự thật của cổng XOR và ký hiệu



Hình 4.1 (tiếp theo) Phương pháp parity bit:
(c) Mạch tạo parity bit (d) Hai ví dụ.

Mạch được dùng để tính toán parity bit cho mỗi ký tự gồm một tập các cổng XOR được nối với nhau như hình 4.1(c). Cổng XOR được xem như là một bộ cộng modulo-2, bảng sự thật (truth table) trong hình 4.1(b) giải thích cho điều này. Cặp bit có ý nghĩa thấp nhất được cộng modulo trước tiên và ngõ ra của cổng XOR này được cộng modulo với bit tiếp theo, và cứ thế. Ngõ ra của cổng XOR cuối cùng là bit parity cần, nó sẽ được nạp vào thanh ghi truyền PISO trước khi truyền ký tự này. Tương tự, khi nhận, bit kiểm tra này được tính toán lại và so với parity bit thu được. Nếu chúng khác nhau, điều này chỉ ra một lỗi truyền.

Thuật ngữ được dùng trong lý thuyết mã hóa mô tả đơn vị thông điệp kết hợp gồm các bit dữ liệu hữu dụng và các bit phát hiện lỗi thêm vào, được gọi là từ mã (codeword). Số vị trí bit tối thiểu mà hai từ mã hợp lệ khác nhau được gọi là khoảng cách Hamming của mã này. Ví dụ xem xét một lược đồ mã hóa có 7 bit dữ liệu và một bit parity trong một từ mã. Giả sử rằng dùng kiểm tra chẵn (even parity), các từ mã phối hợp trong lược đồ này sẽ là:

0000000 0

0000001 1

0000010 1

0000011 0

Chúng ta có thể suy ra từ danh sách này khoảng cách Hamming là 2, vì mỗi từ mã hợp lệ khác nhau ít nhất 2 vị trí bit. Điều này có nghĩa là lược đồ này không phát hiện được các lỗi 2 bit, vì mẫu bit kết quả sẽ khác nhưng từ thì hợp lệ. Tuy nhiên, nó sẽ phát hiện tất cả các lỗi đơn bit, vì nếu một bit đơn trong một từ mã bị sai thì từ mã sẽ không hợp lệ.

2.3. Kiểm tra tổng BSC (Block Sum Check)

Khi các khối ký tự đang được truyền, xác suất một ký tự chứa lỗi bit gia tăng. Xác suất một khối ký tự bị lỗi bit được gọi là tỉ lệ lỗi bit BER. Khi các khối ký tự (các frame) đang được truyền, chúng ta có thể mở rộng các khả năng phát hiện lỗi từ một parity bit trên một ký tự (byte) bằng cách dùng một tập parity bit được tính từ toàn bộ khối ký tự trong frame. Với phương pháp này, mỗi ký tự trong frame được phân phối một parity bit như trên (parity hàng). Ngoài ra một bit mở rộng được tính cho mỗi vị trí bit (parity cột) trong toàn bộ frame. Tập các parity bit cho mỗi cột được gọi là **ký tự kiểm tra khối, BCC (block check character)** vì mỗi bit tạo nên ký tự này là tổng modulo-2 của tất cả các bit trong cột tương ứng. Ví dụ trong hình 4.2 dùng parity lẻ cho các parity bit hàng và parity chẵn cho các parity bit cột, và giả sử rằng frame chứa các ký tự in được.

Điều chúng ta có thể suy ra từ ví dụ này là mặc dù các lỗi 2 bit trong một ký tự sẽ thoát khỏi kiểm tra parity theo hàng, nhưng chúng sẽ bị phát hiện bởi kiểm tra parity cột tương ứng. Dĩ nhiên, điều này là đúng chỉ khi không có lỗi 2 bit xảy ra trong cùng một cột tại cùng thời điểm. Rõ ràng xác suất xảy ra trường hợp này nhỏ hơn nhiều so với xác suất xảy ra lỗi 2 bit trong một ký tự. Việc dùng kiểm tra tổng khối cải thiện đáng kể các đặc trưng phát hiện lỗi của lược đồ kiểm tra chẵn lẻ.

(a)

P_R	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀
0	0	0	0	0	0	1	0
1	0	1	0	1	0	0	0
0	1	0	0	0	1	1	0
0	0	1	0	0	0	0	0
1	0	1	0	1	1	0	1
0	1	0	0	0	0	0	0
1	1	0	0	0	0	1	1
1	0	0	0	0	0	1	1
1	1	0	0	0	0	0	1

Các bit parity
duyệt theo hàng (lẻ)

= STX

} Nội dung frame

Hướng
truyền

Các bit parity
duyệt theo cột (chẵn)

= ETX

= BCC

█ = Ví dụ tổ hợp lỗi không phát hiện được

P_R = Bit parity hàng

Hình 4.2 Ví dụ kiểm tra BSC: (a) Các parity bit hàng và cột

(b)	Tại đầu phát:	Tại đầu thu:
	$ \begin{array}{r} 0\ 0\ 0\ 0\ 0\ 1\ 0 \\ 1\ 0\ 1\ 1\ 0\ 1\ 1 \\ 1\ 1\ 0\ 1\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\ \hline [1]\ 1\ 0\ 0\ 1\ 1\ 0\ 0 \end{array} $ <p style="text-align: center;">Nội dung</p> <p style="text-align: center;">$\xrightarrow{1}$</p> <p style="text-align: center;">$1\ 0\ 0\ 1\ 1\ 0\ 1 = \text{Tổng bù 1}$</p>	$ \begin{array}{r} 0\ 0\ 0\ 0\ 0\ 1\ 0 \\ 1\ 0\ 1\ 1\ 0\ 1\ 1 \\ 1\ 1\ 0\ 1\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\ 0\ 1\ 1\ 0\ 0\ 1\ 0 \\ \hline [1]\ 1\ 1\ 1\ 1\ 1\ 1\ 0 \end{array} $ <p style="text-align: center;">$\xrightarrow{1}$</p> <p style="text-align: center;">$1\ 1\ 1\ 1\ 1\ 1\ 1 = \text{Bằng 0 trong bù 1}$</p>
	$0\ 1\ 1\ 0\ 0\ 1\ 0$	

Hình 4.2 (tiếp theo) Ví dụ kiểm tra BSC: (b) Tổng bù 1.

Một dạng khác của lược đồ trên dùng tổng bù 1 làm cơ sở cho kiểm tra tổng khối, thay vì dùng tổng modulo-2. Nguyên lý của lược đồ này được minh họa trên hình 4.2(b).

Trong lược đồ này, các ký tự trong khối cần truyền được xem như các số nhị phân không dấu. Trước hết các số này được cộng với nhau dùng phép bù 1. Tất cả các bit trong kết quả được đảo ngược bit và kết quả đảo ngược được dùng như ký tự kiểm tra khối BCC. Tại máy thu, tổng bù 1 của tất cả các ký tự trong khối bao gồm cả ký tự kiểm tra, được tính và nếu không có lỗi xuất hiện thì kết quả sẽ bằng zero (0). Nên nhớ lại rằng, với phép toán bù 1, số nhớ cuối được dùng, là bất kỳ giá trị nào vượt ra ngoài vị trí bit có ý nghĩa lớn nhất được cộng vào tổng nhị phân hiện hữu. Zero trong phép bù 1 được biểu diễn bởi tất cả các bit nhị phân đều là 0 hoặc tất cả đều là 1.

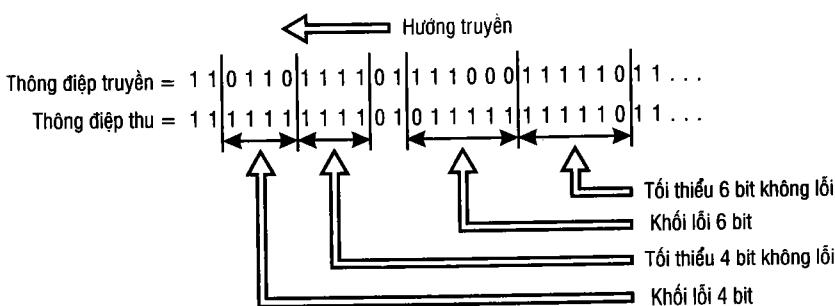
Từ hình 4.2(b) chúng ta có thể suy ra các đặc tính phát hiện lỗi của lược đồ này tốt hơn so với phương pháp tổng modulo-2. Như chúng ta sẽ thấy, vì tổng bù 1 được tính dễ dàng, nên được dùng như một phương pháp phát hiện lỗi trong một số ứng dụng yêu cầu hoạt động phát hiện lỗi chỉ được thực hiện bằng phần mềm.

2.4. Kiểm tra CRC (cyclic redundancy check)

Hai lược đồ trước đây khá thích hợp cho các ứng dụng trong đó xuất hiện các lỗi đơn bit ngẫu nhiên. Tuy nhiên khi các khối lỗi xuất hiện chúng ta phải dùng một phương pháp chắc chắn hơn. Một khối lỗi bắt đầu và kết thúc bằng một bit lỗi, mặc dù các bit trong khoảng giữa có thể hay không có thể bị lỗi. Do đó, một khối lỗi được định nghĩa là số bit giữa hai bit lỗi liên tiếp nhau bao gồm cả hai bit lỗi này. Hơn nữa, khi xác định chiều dài của một khối lỗi, bit lỗi cuối cùng trong một khối và bit lỗi đầu tiên trong khối sau đó phải được tách biệt bởi B hay nhiều hơn các bit không lỗi, trong đó B là chiều dài của khối lỗi này. Ví dụ hai khối lỗi có chiều dài khác nhau được trình bày trên hình 4.3. Chú ý rằng các bit lỗi thứ nhất và thứ 3 không thể dùng để định một khối lỗi 11-bit vì một lỗi xảy ra trong 11 bit kế tiếp.



Phương pháp parity hay dẫn xuất BSC không thể cung cấp một lược đồ phát hiện lỗi tin cậy để kháng lại các lỗi khôi. Trong các trường hợp này, chọn lựa phổ biến nhất là dựa trên các mã đa thức. Các mã đa thức được dùng với các lược đồ truyền frame. Một tập ký số kiểm tra được phát sinh cho mỗi frame được truyền, dựa vào nội dung của frame, và được máy phát gắn vào đuôi của frame. Sau đó máy thu sẽ thực hiện tính toán tương tự trên toàn bộ frame và kiểm tra các ký số. Nếu không có lỗi, kết quả thu được sẽ giống như mã kiểm tra thu được; nếu khác, nó chỉ ra một lỗi truyền.



Hình 4.3 Các ví dụ về khôi lỗi.

Số ký số kiểm tra trong một frame được chọn sao cho phù hợp với dạng lỗi truyền được dự báo trước, thường là 16 hay 32 bit. Các ký số kiểm tra có được từ tính toán được gọi là các ký số kiểm tra tuần tự frame FCS (Frame Check Sequence) hay CRC (Cyclic Redundancy Check).

Phản lý thuyết toán sau đây về các mã đa thức sẽ được trình bày kỹ hơn trong các sách về lý thuyết mã hóa. Về bản chất, phương pháp này khai thác đặc trưng của các số nhị phân khi dùng phép toán modulo-2.

Đặt

- $M(x)$ là một số k -bit (là thông điệp truyền).
- $G(x)$ là một số $(n+1)$ bit (phần tử chia hay phần tử sinh).
- $R(x)$ là một số n -bit sao cho $k > n$ (phần dư).

Nếu

- $M(x) \cdot x^n / G(x) = Q(x) + R(x)/G(x)$, trong đó $Q(x)$ là thương số,

- $[M(x) \cdot x^n + R(x)]/G(x) = Q(x)$, dùng phép modulo-2.

Chúng ta dễ dàng có kết quả này bằng cách thay biểu thức $M(x) \cdot x^n / G(x)$ vào phương trình thứ hai,

$$[M(x) \cdot x^n + R(x)]/G(x) = Q(x) + R(x)/G(x) + R(x)/G(x)$$

Vì với phép modulo-2 thì $R(x)/G(x) + R(x)/G(x) = 0$.

Để khai thác đặc điểm này, nội dung của toàn bộ frame $M(x)$ kết hợp với một tập các số 0 tạo thành số có số ký số bằng với số ký số của FCS, số này được chia modulo-2 cho số nhị phân $G(x)$, gọi là đa thức sinh, nhiều hơn FCS 1 ký số. Toán tử chia tương đương với thao tác XOR từng bit song song nhau khi xử lý từng bit trong frame. Số dư $R(x)$ sau đó là FCS được truyền tại đuôi của các ký số thông tin. Tương tự, khi tiếp nhận thông tin, luồng bit thu được bao gồm cả các bit FCS lần nữa được chia cho cùng đa thức sinh như phía phát, $[M(x) \cdot x^n + R(x)]/G(x)$, nếu không có lỗi thì phần dư sẽ chứa tất cả là số 0. Nếu có lỗi xuất hiện, thì phần dư khác không.

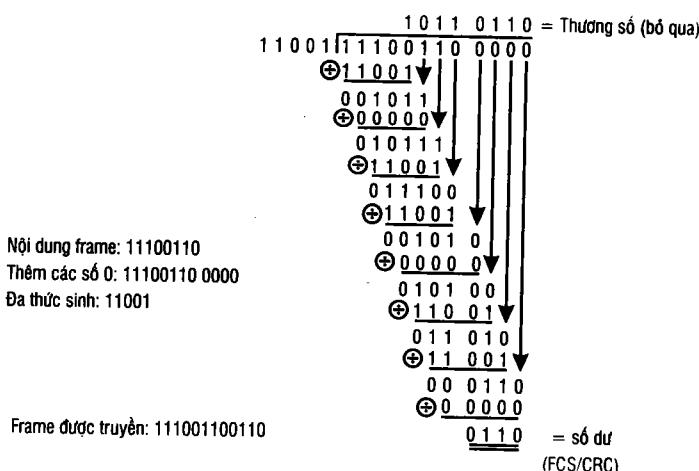
Ví dụ

Một luồng dữ liệu gồm các frame 8-bit được truyền qua một liên kết dùng CRC để phát hiện lỗi. Dùng đa thức sinh 11001. Hãy dùng ví dụ để mô tả:

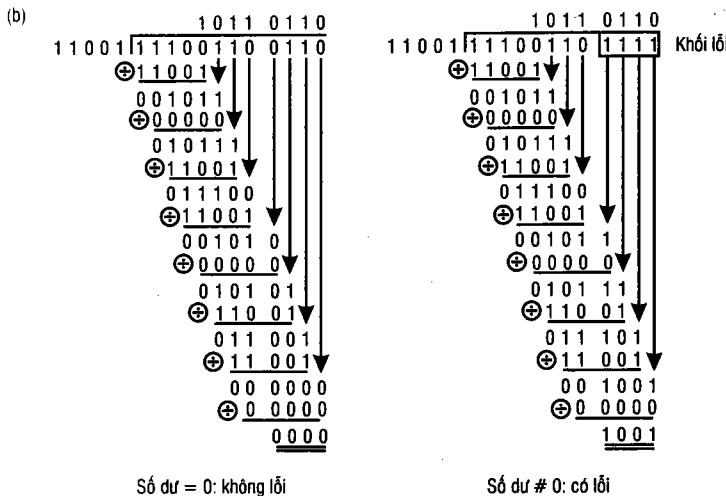
(a) Quá trình sinh FCS.

(b) Quá trình kiểm tra FCS.

(a)



Hình 4.4 Ví dụ CRC

**Hình 4.4 (tiếp theo) Ví dụ CRC**

Sự phát sinh FCS cho dòng dữ liệu 11100110 được trình bày trên hình 4.4. Trước hết, bốn bit zero được gắn vào dòng thông điệp, điều này tương đương với việc nhân thông điệp này với x^4 , vì FCS sẽ có bốn bit. Kết quả này sau đó được chia (modulo 2) cho đa thức sinh (số nhị phân). Thao tác chia modulo-2 tương đương với tiến hành XOR song song từng bit khi từng bit trong số bị chia được xử lý. Với phép toán modulo-2 chúng ta cũng có thể tiến hành chia từng phần dư, tạo hai số cùng chiều dài, như vậy cả hai bit có ý nghĩa lớn nhất đều là 1. Chúng ta không cần quan tâm đến độ lớn của cả hai số. Kết quả dư 4 bit 0110 chính là FCS, bốn bit này sẽ được gắn vào đuôi của thông điệp truyền khi chúng được truyền. Thương số không được dùng.

Tại máy thu, toàn bộ dòng bit đã nhận được chia cho cùng đa thức sinh với máy phát. Hai ví dụ được trình bày trong hình 4.4(b). Trong hình thứ nhất, được giả sử là không có lỗi, vì vậy phép chia có phần dư bằng 0, một lần nữa thương số không được dùng. Trong hình thứ hai có một khối lỗi 4 bit tại đuôi của luồng bit truyền. Kết quả, dư số của phép chia khác không, chỉ ra lỗi truyền này.

Việc chọn đa thức sinh là rất quan trọng vì nó quyết định loại lỗi sẽ được phát hiện. Giả sử frame truyền T(x) là:

110101100110

và mẫu lỗi E(x) là

00000000 1001

một bit 1 tại một vị trí chỉ ra một lỗi. Từ đó, với phép modulo-2 :

Frame nhận được = $T(x) + E(x)$

$$[T(x) + E(x)] / G(x) = T(x) / G(x) + E(x) / G(x)$$

nhưng $T(x)/G(x)$ không có dư số. Suy ra rằng một lỗi chỉ xuất hiện khi $E(x)/G(x)$ có dư số.

Ví dụ, tất cả $G(x)$ có ít nhất ba số hạng và $E(x)/G(x)$ sẽ cho một dư số cho tất cả các lỗi đơn bit và hai bit đối với phép modulo-2 và được phát hiện. Ngược lại, một khối lỗi có cùng chiều dài với $G(x)$ có thể là một bội số của $G(x)$ và do đó không có dư số và không được phát hiện.

Tóm lại, một đa thức sinh R bit sẽ phát hiện:

- Tất cả các lỗi đơn bit
- Tất cả các lỗi hai bit
- Tất cả lỗi có số bit lẻ
- Tất cả các khối lỗi $< R$
- Hầu hết các khối lỗi $\geq R$

Phương pháp chuẩn biểu diễn một đa thức sinh là ở những vị trí tồn tại bậc của x trong đa thức ghi bit 1. Ví dụ các đa thức sinh của CRC được dùng trong thực tế:

$$\text{CRC-16} = X^{16} + X^{15} + X^2 + 1$$

$$\text{CRC-CCITT} = X^{16} + X^{12} + X^5 + 1$$

$$\text{CRC-32} = X^{32} + X^{26} + X^{23} + X^{16} + X^{12} + X^{11} + 1 X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

Do đó CRC-16 tương đương dạng nhị phân:

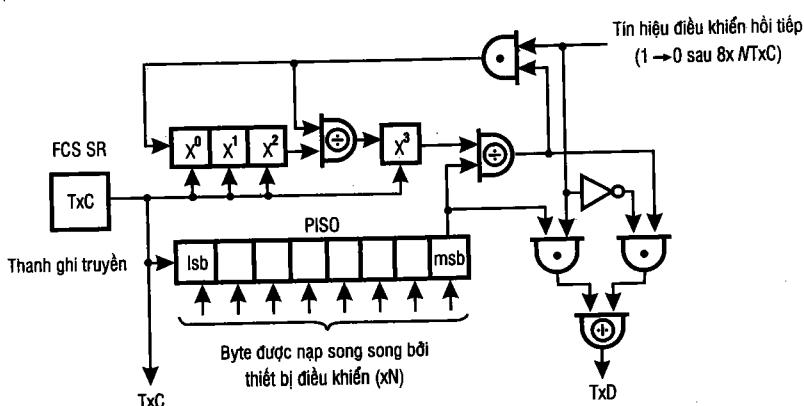
1 1000 0000 0000 0101

Với một đa thức sinh như vậy, 16 bit 0 sẽ được kết vào cuối thông điệp truyền trước khi sinh FCS. Thành phần này sau đó sẽ là dư số. CRC-16 sẽ phát hiện tất cả các lỗi khối ít hơn 16 bit và hầu hết các lỗi khối lớn hơn bằng 16 bit. CRC-16 và CRC-CCITT được dùng rộng rãi trong các mạng WAN, trong khi CRC-32 được dùng hầu hết trong các mạng LAN.

Mặc dù nhu cầu thực hiện có thể xuất hiện nhiều phép chia và trở nên phức tạp, nhưng trong thực tế điều này có thể thực hiện hoàn toàn bằng phần cứng hay phần mềm. Để minh họa cho điều này, trong hình 4.5(a) trình bày lược đồ phần cứng được dùng cho ví dụ ở hình 4.4.

Trong ví dụ này, vì cần tạo ra 4 bit FCS, nên chỉ cần một thanh ghi dịch 4-bit để đại diện cho x^3 , x^2 , x^1 và x^0 trong đa thức sinh. Thường xem chúng như các *bit tích cực* (*active bit*) của bộ sinh. Với đa thức sinh này x^3 và x^0 là nhị phân 1 trong khi x^2 và x^1 là nhị phân 0. Các trạng thái mới của các phần tử thanh ghi dịch x^1 và x^2 dựa trực tiếp vào trạng thái của x^0 và x^1 ; các trạng thái mới của các phần tử x^0 và x^3 được xác định bởi trạng thái của đường hồi tiếp được XOR với ký số ngay trước đó.

(a)

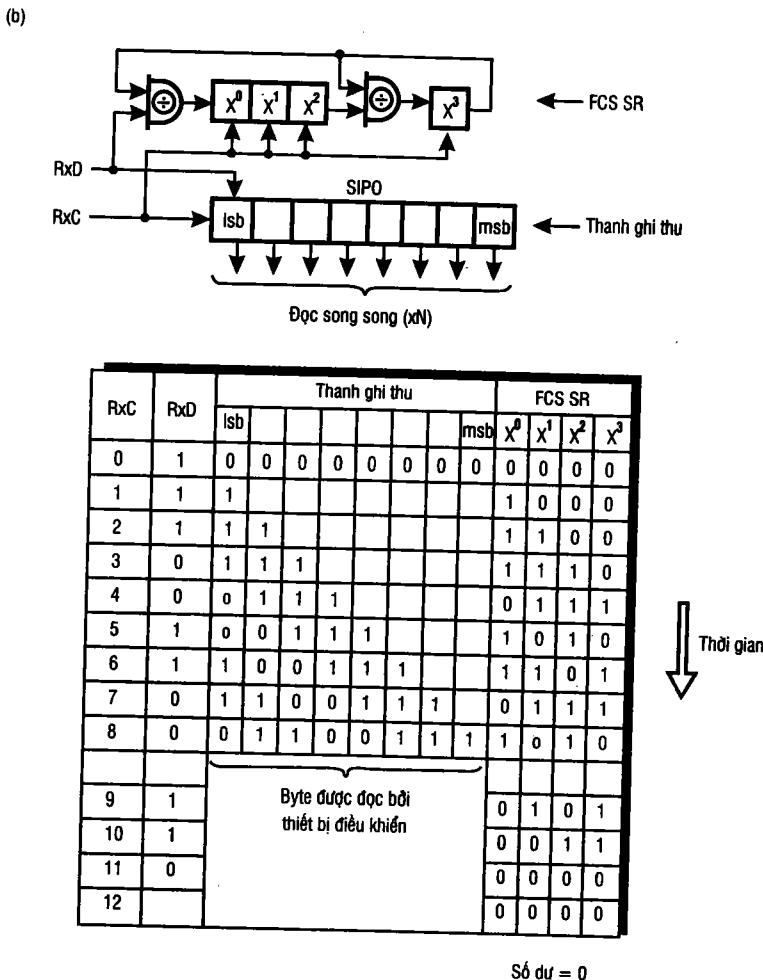


Xung Tx C	Thanh ghi truyền								FCS SR			
	lsb						msb	x^0	x^1	x^2	x^3	
0	0	1	1	0	0	1	1	1	0	0	0	0
1	0	0	1	1	0	0	1	1	1	0	0	1
2	0	0	0	1	1	0	0	1	0	1	0	0
3	0	0	0	0	1	1	0	0	1	0	1	1
4	0	0	0	0	0	1	1	0	1	1	0	0
5	0	0	0	0	0	0	1	1	0	1	1	0
6		0	0	0	0	0	0	1	1	0	1	0
7			0	0	0	0	0	0	1	1	0	0
8				0	0	0	0	0	0	1	1	0
9					0	0	0		0	1		1
10						0	0		0		0	1
11							0				0	1

Thời gian

Tx D

Hình 4.5 Lược đồ phần cứng của CRC: (a) Mạch sinh FCS



Hình 4.5 (tiếp theo) Lược đồ phần cứng của CRC: (b) Kiểm tra.

Mạch hoạt động như sau. Thanh ghi FCS được xoá và byte 8-bit đầu tiên của frame được nạp song song vào thanh ghi truyền PISO. Byte dữ liệu này sau đó được dịch ra đường truyền, bit có ý nghĩa lớn nhất được dịch trước, với tốc độ được xác định bởi đồng hồ truyền Tx_C . Trong cơ cấu đồng bộ thời gian như vậy, luồng bit tương tự được XOR với phần tử x^3 và được chuyển qua đường hồi tiếp đến ngõ vào được chọn của thanh ghi FCS. Khi mỗi byte 8 bit được nạp vào thanh ghi truyền và đã được truyền nối tiếp ra đường dây thì thủ tục được lặp lại. Sau cùng khi byte cuối cùng trong frame đã được truyền, thanh ghi dịch truyền nạp dây zero và tín hiệu điều khiển hồi tiếp thay đổi từ 1 xuống 0 để đưa nội dung hiện hành trong thanh ghi FCS (dư số) chạy nối tiếp với thông tin frame ra đường truyền.

Trong hình 4.5(a) nội dung của thanh ghi truyền cũng như thanh ghi FCS giả sử vừa vẹn một frame chỉ có 1 byte ($N=1$), do đó tương ứng với ví dụ trước đây trong hình 4.4. Trong hình 4.5 nội dung trong cả thanh ghi truyền và FCS được trình bày sau mỗi xung dịch truyền (transmit clock). Luồng bit truyền được chỉ ra trong phần khoanh chấm.

Phản cứng thu tương ứng giống như mạch phản cứng bên truyền, được trình bày trên hình 4.5(b). Dữ liệu thu sẽ được lấy mẫu (hay dịch) để đưa vào trong thanh ghi thu SIPO tại khoảng giữa của bit (hay cuối đối với mã Manchester). Cũng giống như mạch phía truyền, trong cơ cấu đồng bộ thời gian này luồng bit được XOR với phản tử x^3 và nạp vào thanh ghi FCS. Khi thu mỗi byte 8 bit, chúng được đọc bởi thiết bị điều khiển. Lần nữa, nội dung của một frame chỉ vừa một byte dữ liệu đơn.

Phản cứng trong hình 4.5 thường được kết hợp chặt chẽ với các mạch điều khiển truyền thiên hướng bit. Tuy nhiên, trong vài trường hợp một CRC được chuộng hơn so với kiểm tra BSC trong truyền thiên hướng ký tự. Trong các trường hợp như vậy, CRC phải được thực hiện trong phần mềm nhờ thiết bị điều khiển thay vì dùng phản cứng. Điều này có liên hệ với kiểm soát lỗi hướng tới khi xem xét mã giả sau:

(Giả sử một frame được định dạng trước, được truyền (bao gồm một byte zero tại đuôi của nó) hay một fram nhận, được lưu giữ trong một mảng $buff[1.....count]$. 8 bit tích cực của bộ chia 9 bit cũng được lưu trong 8 bit có ý nghĩa lớn nhất của CRCDIV 16 bit. Hàm sau đây sẽ tính toán và cho kết quả CRC 8-bit)

```

function    CRC: byte;
var          i,j:integer;
            data:integer;

begin
  data:= buff[1] shl 8;
  for j:=2 to count do
    begin
      data:=data+buff[j];
      for i:=1 to 8 do
        if ((data and S8000)=S8000) then
          begin
            data:= data shl 1;
            data:=data xor CRCDIV;
          end
        else    data:= data shl 1;
      end;
      CRC := data shr 8;
    End
  
```

Đoạn chương trình này giả sử một đa thức sinh 8 bit và frame được định dạng trước_STX, ETX,..._được lưu giữ trong một mảng . Mã tương tự có thể được dùng cho

sinh CRC và kiểm tra; để sinh CRC, mảng sẽ chứa một byte gồm tất cả các số zero tại phần cuối.

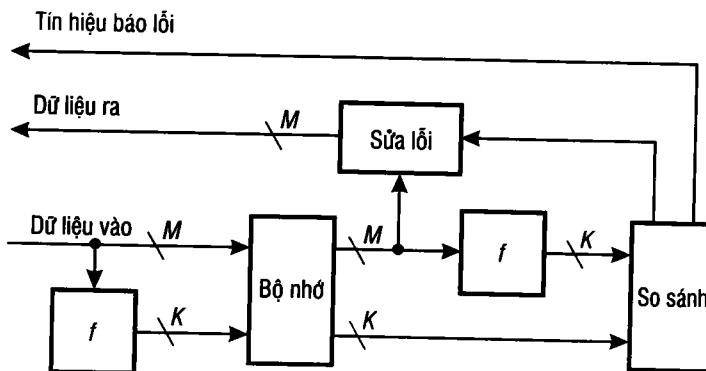
2.5. Phát hiện và sửa sai theo Hamming

Các phương pháp được trình bày ở trên chỉ phát hiện lỗi truyền, không thể sửa sai. Phương pháp được trình bày sau đây không chỉ phát hiện sai mà còn có thể sửa sai cho một số bit nhất định.

Hình 4.6 mô tả tổng quan quá trình xử lý phát hiện và sửa lỗi. Khi dữ liệu được đọc vào trong bộ nhớ, việc tính toán biểu diễn bởi hàm f được tiến hành trên dữ liệu này để tạo ra một mã sửa sai. Cả mã và dữ liệu được lưu giữ. Do đó, nếu một M bit từ (word) được lưu giữ, và mã có chiều dài là K bit thì kích thước thực sự được lưu giữ là $M+K$ bit.

Khi các từ được lưu giữ trước được đọc ra, mã được dùng để phát hiện và có thể sửa lỗi. Một tập mới của K bit mã được tạo ra từ M bit dữ liệu và được so sánh với các bit mã đã lưu giữ. Sự so sánh sẽ cho ra một trong 3 trường hợp:

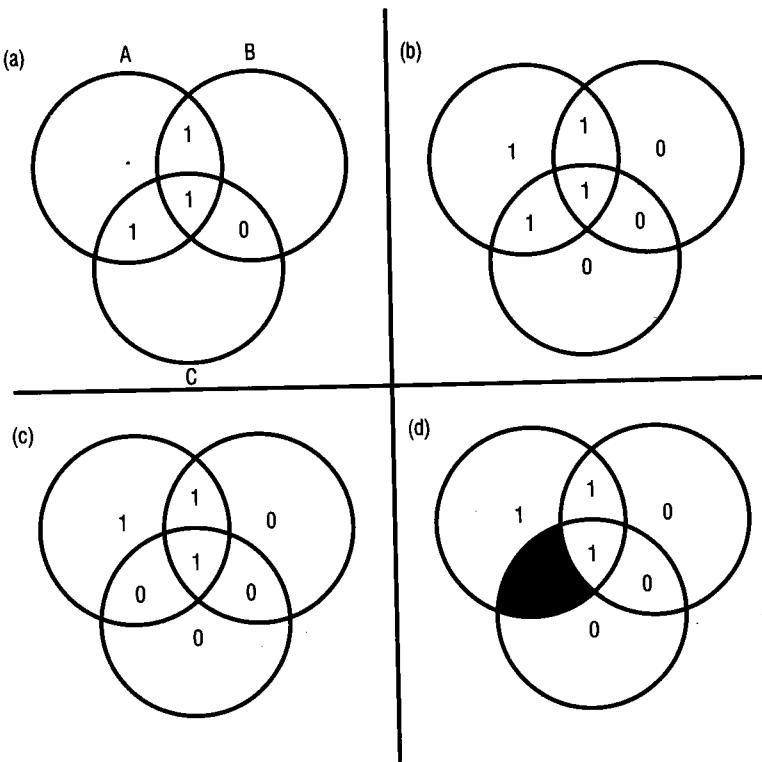
1. Không có lỗi. Các bit dữ liệu được gửi ra.
2. Có lỗi, và có khả năng sửa được. Các bit dữ liệu và các bit sửa sai được nạp vào một bộ sửa lỗi, nó tạo ra một tập M bit đã sửa được gửi ra.
3. Một lỗi nhưng không thể sửa được. Đây là điều kiện để gửi thông báo.



Hình 4.6 Chức năng sửa sai.

Một mã được đặc tính hóa bởi một số các bit lỗi trong một từ mà nó có thể phát hiện lỗi và sửa sai. Các mã sửa sai đơn giản nhất là mã Hamming được phát minh bởi Richard Hamming tại Bell Laboratories. Hình 4.7 dùng lược đồ Venn để mô tả việc dùng mã này trên các từ 4 bit ($M=4$). Với 3 vòng tròn giao nhau có 7 ngăn. Gán 4 bit dữ liệu vào các ngăn trong, hình 4.7(a). Các ngăn còn lại được làm đầy với parity bit (bit kiểm tra chẵn lẻ). Mỗi bit parity được chọn sao cho tổng số

bit 1 trong các vòng tròn là chẵn (hình 4.7(b)). Do đó, vì vòng A bao gồm ba bit dữ liệu 1, nên bit parity là 1. Bây giờ, nếu một lỗi xảy ra thay đổi một trong các bit này (hình 4.7(c)), sẽ dễ dàng nhận ra. Bằng cách kiểm tra các bit parity sẽ thấy sự trái ngược trong vòng tròn A và C. Chỉ một trong 7 ngăn là vừa thuộc A vừa thuộc C nhưng không thuộc B. Do đó có thể sửa lại bit trong ngăn này.



Hình 4.7 Cơ sở của mã sửa lỗi Hamming

Để sáng tỏ hơn khái niệm này, tiếp theo sẽ phát triển một mã có thể phát hiện và sửa sai các bit lỗi đơn trong các từ 8 bit. Khởi đầu cần xác định mã sẽ dài bao nhiêu. Xem hình 4.6, logic so sánh nhận hai giá trị K bit. Sự so sánh bit với bit được thực hiện bằng cách cộng modulo (xor) hai ngõ nhập. Kết quả được gọi là syndrome word. Do đó, mỗi bit của syndrom là 0 hay 1 tùy vào sự giống hay khác nhau của vị trí bit trong hai ngõ nhập.

Từ syndrom có chiều dài k bit và có giá trị trong dải từ 0 đến $(2^k - 1)$. Giá trị 0 cho biết không có lỗi. Vì một lỗi có thể xảy ra trên bất kỳ M bit dữ liệu hay K bit kiểm tra nào, nên phải có: $2^k - 1 \geq M + K$

Bất đẳng thức này cho biết số bit cần thiết để sửa sai một bit bị sai trong một từ có M bit. Bảng 4.1 liệt kê số bit kiểm tra cần đối với chiều dài từ thay đổi.



Từ bảng này, có thể thấy với từ 8 bit cần 4 bit kiểm tra. Để tiện lợi, dùng syndrom 4 bit với các đặc tính sau:

- Nếu syndrom chứa tất cả các bit 0, thì không có lỗi.
- Nếu syndrom chứa một và chỉ một bit đặt là 1 thì một lỗi xảy ra đối với một trong 4 bit kiểm tra. Không cần phải sửa.
- Nếu syndrom chứa nhiều hơn một bit 1 thì giá trị của syndrom chỉ vị trí của bit dữ liệu bị lỗi. Để sửa, bit dữ liệu này được đổi thành bit ngược lại.

Để đạt được đặc trưng này, các bit dữ liệu và kiểm tra được sắp xếp với nhau thành một từ 12 bit như hình 4.8. Các vị trí bit được đánh số từ 1 đến 12.

Bảng 4.1 Gia tăng chiều dài từ với sửa lỗi.

Số bit dữ liệu	Số bit kiểm tra	Sửa lỗi đơn	%Gia tăng	Số bit kiểm tra	Sửa lỗi đơn/Phát hiện lỗi kép
8	4		50	5	62,5
16	5		31,25	6	37,5
32	6		18,75	7	21,875
64	7		10,94	8	12,5
128	8		6,25	9	7,03
256	9		3,52	10	3,9

Các bit kiểm tra được tính toán như sau:

$$C1 = M1 \oplus M2 \oplus M4 \oplus M5 \oplus M7$$

$$C2 = M1 \oplus M3 \oplus M4 \oplus M6 \oplus M7$$

$$C4 = M2 \oplus M3 \oplus M4 \oplus M8$$

$$C8 = M5 \oplus M6 \oplus M7 \oplus M8$$

Mỗi bit kiểm tra tính toán trên mỗi vị trí bit dữ liệu có số 1 nằm trong cột tương ứng. Do đó các vị trí 3,5,7,9 và 11 chứa 2^0 , vị trí bit 3,6,7,10 và 11 chứa 2^1 , vị trí bit 5,6,7 và 12 chứa 2^2 và vị trí bit 9,10,11 và 12 chứa vị trí 2^3 . Tổng quát, vị trí bit n được kiểm tra bởi các bit C_i sao cho $\sum i = n$. Ví dụ, vị trí 7 được kiểm tra bởi các bit tại vị trí 4,2 và 1; và $7=4+2+1$.

Ví dụ có chuỗi bit dữ liệu là 00111001 và nếu vị trí bit 3 từ phải qua bị sai (thay vì 0 lại là 1) thì khi các bit kiểm tra mới được so sánh với các bit kiểm tra cũ, một từ syndrom được hình thành:

C8	C4	C2	C1
0	1	1	1
\oplus	0	0	1

0	1	1	0
---	---	---	---

Kết quả này chỉ rằng vị trí bit 6 từ trái qua bị sai.

Hình 4.8 mô tả quá trình tính toán ở trên. Vì kết quả chịu ảnh hưởng chỉ bởi các bit 1, chỉ những cột chứa bit 1 được khoanh tròn để nhận dạng. Các bit kiểm tra sau đó có thể được tính dọc theo hàng. Kết quả được trình bày cho các bit dữ liệu gốc và cho các bit dữ liệu có lỗi.

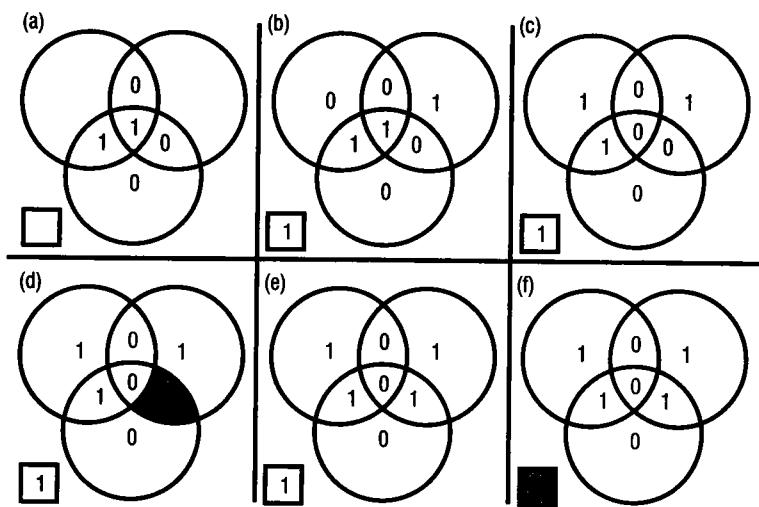
Vị trí bit	12	11	10	9	8	7	6	5	4	3	2	1	
Bit dữ liệu	M8	M7	M6	M5		M4	M3	M2		M1			
Bit kiểm tra					C8				C4		C2	C1	
	1	1	(1)	(1)		(0)	0	0		(0)			C80
	1	0	(0)	(0)		(1)	1	1		(0)			C41
	0	1	(1)	(0)		(1)	1	0		(1)			C21
	0	1	(0)	(1)		(1)	0	1		(1)			C11
Tử được lưu	0	0	(1)	(1)	0	(1)	0	0	1	(1)	1	1	
	0	0	(1)	(1)	0	(1)	(1)	0	1	(1)	1	1	
Tử thu được	0	1	(1)	(1)	0	(0)	(0)	0	1	(0)	0		C80
	1	1	(1)	(1)	0	(1)	(1)	1	1	(0)	0		C40
	1	0	(0)	(0)		(1)	(1)	1	1	(0)	0		C20
	0	1	(1)	(0)		(1)	(1)	0	1	(1)	1		C11
	0	1	(0)	(1)		(1)	(0)	1					

Hình 4.8 Sinh bit kiểm tra

Mã mới vừa trình bày ở trên được gọi là mã sửa sai đơn (SEC _single error correcting). Thông thường, bộ nhớ bán dẫn cần một mã sửa sai đơn kết hợp với phát hiện lỗi kép (double-error-detecting), gọi tắt là mã SEC-DED. Mã này cần thêm một bit nữa so với SEC.

Hình 4.9 diễn tả quá trình thực hiện của mã này, một lần nữa lại với một từ 4 bit. Tuần tự như sau : nếu hai lỗi xảy ra (4.9(c)), thủ tục kiểm tra chuyển đến (4.9(d)), và làm xấu hơn tình hình bằng cách tạo ra một lỗi thứ 3 (4.9(e)). Để khắc phục, bit thứ 8 được thêm vào và được đặt giá trị sao cho tổng số 1 là một số chẵn. Bit mở rộng này kiểm soát được lỗi (4.9(f)).

Mã sửa lỗi này cũng thường dùng để tăng cường độ tin cậy của bộ nhớ. Với tổ chức một bit trên một chip, một mã SEC-DED được xem xét một cách thích hợp. Ví dụ, IBM 30xx dùng một mã SEC-DED 8 bit cho mỗi 64 bit dữ liệu trong bộ nhớ chính. Do đó, kích thước của bộ nhớ chính thực sự lớn hơn khoảng 12%. Máy tính VAX dùng mã SEC-DED 7 bit cho mỗi 32 bit bộ nhớ, và overhead là 22%.



Hình 4.9 Mã SEC-DEC

3. NÉN SỐ LIỆU

3.1. Khái quát

Cho đến lúc này chúng ta đều giả sử rằng nội dung của các frame truyền bao gồm dữ liệu gốc dưới dạng chuỗi ký tự có chiều dài cố định. Cho dù đây là trường hợp của nhiều ứng dụng truyền số liệu, vẫn còn có những trường hợp khác, trong đó dữ liệu được nén trước khi được truyền đi. Nén dữ liệu là một việc làm thiết yếu trong các dịch vụ truyền dẫn công cộng, ví dụ như truyền qua mạng PSTN, vì trong các mạng như vậy việc tính cước dựa vào thời gian và cự ly truyền. Do đó, trong một cuộc gọi dữ liệu nếu thời gian truyền mỗi khối dữ liệu giảm xuống cũng đồng nghĩa với giảm giá thành của cuộc gọi này.

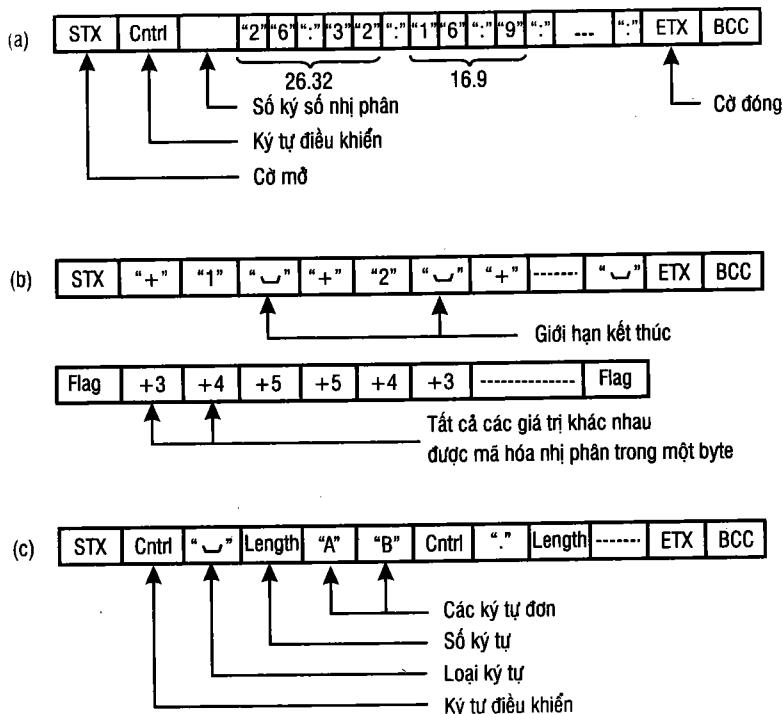
Ví dụ chúng ta truyền dữ liệu qua mạng PSTN dùng tốc độ 4800bps, và thời gian cần truyền hết dữ liệu là 20 phút. Rõ ràng nếu dùng nén dữ liệu chúng ta có thể giảm một nửa số lượng dữ liệu được truyền, và có thể tiết kiệm được 50% giá tiền. Điều này tương đương với dùng tốc độ truyền 9600bps nhưng không nén.

Trong thực tế chúng ta có thể dùng một loạt các giải thuật nén khác nhau, mỗi giải thuật sẽ phù hợp với một loại dữ liệu đặc biệt. Vài modem thông minh sẽ cung cấp đặc trưng nén thích nghi tự động thực hiện các giải thuật nén phù hợp cho loại dữ liệu đang được truyền. Chúng ta sẽ lần lượt đề cập đến một vài giải thuật nén thông dụng trong phần này.

3.2. Nén nhờ đơn giản mã cho các chữ số (Packed decimal)

Khi các frame chỉ bao gồm các ký tự số học đang được truyền, chúng ta có thể tiết kiệm đáng kể bằng cách giảm số bit trên mỗi ký tự từ 7 xuống 4 thông qua mã BCD (Binary Coded Decimal) thay cho mã ASCII. Chúng ta có thể thấy từ bảng mã ASCII rằng 10 ký số (0-9) đều có 011 tại ba vị trí bit thứ tự cao của chúng. Thông thường, ba bit (011) cần cho phân biệt giữa ký tự số và các ký tự khác trong tập mã. Tuy nhiên nếu dữ liệu chỉ nén các ký số thì 3 bit này là dư thừa và không cần truyền, (xem hình 4.10).

Như chúng ta có thể thấy, nếu sử dụng truyền đồng bộ thiên hướng ký tự, không dễ gì chèn các ký số nhị phân vào giữa các ký tự điều khiển STX và ETX vì cặp ký số 0, 3 sẽ được dịch ra là ETX. Vì vậy tại điểm bắt đầu của dữ liệu có một mã ký tự điều khiển được dùng để chỉ dữ liệu nén phía sau. Một ký tự thứ hai phía sau chỉ số lượng ký số. Hai ký tự trong cùng một cột, ví dụ : ; lần lượt được dùng như dấu chấm thập phân và khoảng trắng.



Hình 4.10 Các ví dụ về các giải thuật nén:
(a) *Packet decimal* (b) *Mã hóa quan hệ* (c) *Character suppression*.

3.3. Nén theo mã hóa quan hệ (Relative coding)

Một phương pháp khác được sử dụng khi truyền các dữ liệu số học kế tiếp chỉ khác nhau phần nhỏ về giá trị là chỉ gửi lượng khác nhau giữa các giá trị này cùng với một giá trị tham khảo. Điều này được gọi là *mã hóa quan hệ* và nó có thể đem lại hiệu quả đặc biệt trong các ứng dụng ghi nhận dữ liệu.

Ví dụ nếu giám sát từ xa mức nước của một con sông, thường đọc mức theo các khoảng thời gian thiết lập trước và lưu giữ trước khi truyền. Khi thu thập các giá trị đọc định trước, chúng ta truyền chúng đến trung tâm giám sát thông qua mạng điện thoại và modem quay số tự động. Để tối thiểu thời gian cần truyền thay vì truyền các giá trị chỉ mức nước tuyệt đối, chúng ta chỉ cần truyền các giá trị khác nhau. Ví dụ về dạng frame của lược đồ này được trình bày trên hình 4.10(b). Có thể thấy rằng hiệu quả tiết kiệm thu được phụ thuộc vào khuôn dạng được dùng để biểu diễn các giá trị khác nhau. Nhìn chung, các giá trị nhị phân với một giao thức hướng bit sẽ tiết kiệm được nhiều nhất.

3.4. Nén bằng cách bỏ bớt các ký tự giống nhau (Character suppression)

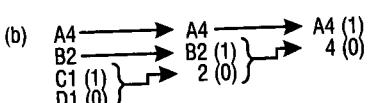
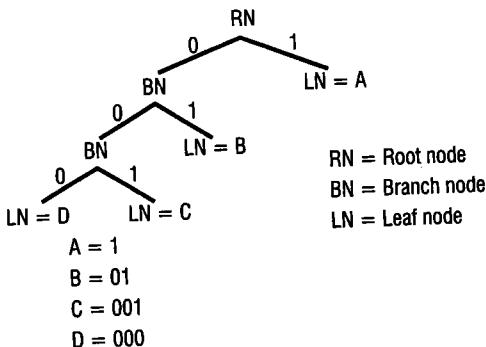
Một dẫn xuất của lược đồ mã hóa quan hệ có thể dùng như một phương pháp tổng quát hơn để nén các loại ký tự khác được trình bày trên hình 4.10(c). Thông thường, khi các frame gồm các ký tự có thể in đang được truyền, thường xuất hiện các chuỗi lặp lại các ký tự giống nhau, ví dụ như ký tự khoảng trắng. Thiết bị điều khiển tại máy phát sẽ quét nội dung của frame trước khi truyền và nếu có một chuỗi 3 hay nhiều hơn các ký tự liên tiếp được xác định thì chúng sẽ được thay thế bởi tuần tự 3 ký tự được trình bày trên hình.

Tuần tự thay thế bao gồm một ký tự điều khiển chỉ định một chuỗi nén phía sau, loại ký tự, và số lượng ký tự có trong chuỗi. Thông thường số đếm được biểu diễn dưới dạng nhị phân, vì trước nó có ký tự điều khiển nên bộ thu có thể phân biệt giữa giá trị đếm 3 và ký tự điều khiển ETX. Bất kỳ ký tự nào đều có thể được nén theo lược đồ này. Máy thu khi phát hiện ký tự điều khiển nén, chỉ cần đọc loại ký tự và giá trị bộ đếm rồi chèn số lượng ký tự thích hợp vào trong frame nhận được tại vị trí này. Lược đồ này là một ví dụ của một kỹ thuật giải mã tổng quát hơn được gọi là *run-length encoding*.

3.5. Nén theo mã hóa Huffman

Phương pháp nén theo mã Huffman khai thác một đặc tính là không phải tất cả các ký hiệu trong một frame truyền có cùng tần suất xuất hiện, ví dụ trong một frame bao gồm một chuỗi ký tự, vài ký tự nào đó xuất hiện nhiều hơn các ký tự khác. Thay vì dùng một số bit nhất định trên một ký tự, chúng ta dùng một lược đồ mã hóa khác trong đó các ký tự xuất hiện thường xuyên được mã hóa với số bit ít hơn các ký tự có tần suất xuất hiện thấp. Do đó lược đồ này là dạng *mã hóa thống kê*. Vì số bit trên mỗi ký tự thay đổi nên chúng ta phải dùng phương pháp truyền đồng bộ thiên hướng bit.

(a)

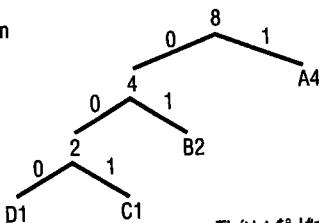


Tần số xuất hiện

$$\begin{aligned}
 A4 &= \{1\} & \rightarrow 1 \\
 B2 &= \{1\} \{0\} & \rightarrow 01 \\
 C1 &= \{1\} \{0\} \{0\} & \rightarrow 001 \\
 D1 &= \{0\} \{0\} \{0\} & \rightarrow 000
 \end{aligned}$$

Bắt đầu tại node lá

Bắt đầu tại node gốc



Thứ tự độ lớn = D1 C1 2 B2 4 A4 8

Hình 4.11 Cấu trúc cây mã Huffman: (a) Cây sau cùng với các mã (b) Sự dẫn xuất cây.

Trước hết, chuỗi ký tự truyền được phân tích và các loại ký tự và tần suất xuất hiện được xác định. Hoạt động mã hóa sau đó liên quan đến việc tạo ra một cây không cân bằng trong đó một số nhánh (là các từ mã trong thực tế) ngắn hơn một số nhánh khác. Mức độ thiếu cân bằng là một hàm của tần suất xuất hiện các ký tự, càng trái rộng ra độ mất cân bằng trong cây càng lớn. Kết quả cuối cùng sẽ được một cây được gọi là *cây Huffman*.

Một cây Huffman là một cây nhị phân với các nhánh được gán giá trị 0 hay 1. Xuất phát của cây thường là một đỉnh (hình học), trong thực tế được gọi là *node gốc* (root node), và một điểm tại đó một nhánh được rẽ ra coi như *node nhánh* (branch node). Điểm kết thúc của một nhánh được gọi là *node lá* (leaf node), là các vị trí được gán cho các ký hiệu đang được mã hóa. Một ví dụ về cây Huffman được trình bày trên hình 4.11(a). Cây này tương ứng với chuỗi ký tự AAAABBCD.

Khi mỗi nhánh chia ra, một giá trị nhị phân 0 hay 1 được gán cho mỗi nhánh mới: một giá trị nhị phân 0 cho nhánh trái và một giá trị nhị phân 1 cho nhánh phải. Mỗi từ mã được dùng cho mỗi ký tự (được ghi tại các node lá) được xác định bằng cách dò theo đường dẫn từ node gốc đến node lá tương ứng và hình thành một chuỗi các giá trị nhị phân nằm trên đường dẫn. Từ các mã này chúng ta có thể suy ra sẽ dùng

$$4x1 + 2x2 + 1x3 + 1x3 = 14 \text{ bit}$$

để truyền chuỗi hoàn chỉnh AAAABBCD.

Chúng ta cũng có thể suy ra lượng bit trung bình trên mỗi từ mã bằng cách lấy lượng bit trong mỗi từ mã nhân với xác suất xuất hiện rồi cộng toàn bộ lại. Cụ thể là

$$1x0,5 + 2x0,25 + 3x0,125 + 3x0,125 = 1,75 \text{ bit/t từ mã}$$

Lượng bit trung bình tối thiểu trên một từ mã theo lý thuyết để truyền chuỗi thông điệp được gọi là entropy, H, của thông điệp này. Có thể tính toán H bằng cách dùng công thức được suy ra từ luật Shannon:

$$H = -\sum_{i=1}^n P_i \log_2 P_i \text{ bit/codeword}$$

Trong đó n là số ký tự trong thông điệp và Pi là xác suất xuất hiện của ký tự i trong thông điệp. Do đó trong thông điệp AAAABBCD, lượng bit trung bình tối thiểu trên một từ mã là:

$$H = -(0,5 \log_2 0,5 + 0,25 \log_2 0,25 + 0,125 \log_2 0,125 + 0,125 \log_2 0,125) = 1,75 \text{ bit/t từ mã}$$

Trong trường hợp này chúng ta có cùng kết quả khi dùng mã hóa Huffman.

Dùng từ mã ASCII 7-bit sẽ cần $8 \times 7 = 56$ bit để truyền toàn bộ chuỗi thông điệp này, con số này nhiều hơn đáng kể so với 14 bit khi dùng mã hóa Huffman.

Trong thực tế, điều này không phải là một so sánh chính xác vì với mã hóa Huffman, máy thu phải biết được tập ký tự biến đổi đang được truyền và các từ mã tương ứng, hay nói cách khác là cây Huffman. Một so sánh tốt hơn là xem xét số bit yêu cầu với mã hóa nhị phân thường: bốn ký tự (A-D) cần 2 bit cho mỗi ký tự, vậy cần truyền 8 ký tự. Rõ ràng, lượng tiết kiệm là không đáng kể. Nhìn chung, mã hóa Huffman là hiệu quả nhất khi các ký tự đang truyền có phân bố tần số rộng và các chuỗi ký tự dài. Ngược lại, nó không phù hợp cho truyền dữ liệu mã nhị phân của máy tính vì các byte 8 bit thường xuất hiện với tần số như nhau.

Để diễn tả làm thế nào xác định cây Huffman trong hình 4.11(a), chúng ta phải dùng thêm thông tin liên quan đến tần số xuất hiện mỗi ký tự như liệt kê trong hình 4.11(b). Các ký tự được liệt kê trong một cột theo thứ tự giảm dần của tham số định lượng. Chúng ta có thể suy ra cây như sau:

Hai node lá đầu tiên phía dưới danh sách C1 và D1 lần lượt được gán là nhánh 1 và nhánh 0 của một node nhánh. Sau đó hai node lá sẽ được thay thế bởi một node nhánh có định lượng bằng tổng hai node lá_2. Một cột mới được hình thành chứa node mới, kết hợp với các node còn lại từ cột đầu tiên, một lần nữa việc sắp xếp lại phải theo thứ tự định lượng giảm dần. Thủ tục này được lặp lại cho đến khi chỉ còn lại hai node.

Để suy ra từ mã kết quả cho mỗi ký tự, chúng ta bắt đầu với ký tự trong cột đầu tiên và sau đó tiến hành liệt kê các số trong nhánh 0 hay 1 khi chúng gặp phải. Do đó cho ký tự A số nhánh đầu tiên là 1 trong cột cuối cùng trong khi cho ký tự C nhánh đầu tiên là 1 sau đó là 0 tại node nhánh 2 và cuối cùng là 0 tại node nhánh 4. Tuy nhiên, các từ mã bắt đầu tại gốc chứ không phải node lá vì các từ mã thực tế là thứ tự ngược của các số này. Sau đó cây Huffman được xây dựng dễ dàng từ tập từ mã.

Chúng ta kiểm tra xem đây có phải là cây tối ưu và do đó là tập từ mã tối ưu hay không bằng cách liệt kê kết quả của tất cả các node nhánh và node lá trong cây bắt đầu với định lượng nhỏ nhất và xử lý từ trái sang phải và từ dưới lên trên. Các từ mã là tối ưu nếu danh sách kết quả tăng theo định lượng.

Ví dụ:

Một chuỗi thông điệp được truyền giữa hai máy tính qua PSTN. Các thông điệp bao gồm chỉ các ký tự từ A đến H. Phân tích cho thấy tần số xuất hiện của mỗi ký tự như sau:

$$A \text{ và } B = 0,25 \quad C \text{ và } D = 0,14 \quad E, F, G \text{ and } H = 0,055$$

(a) Dùng công thức Shannon để suy ra lượng bit trung bình tối thiểu trên một ký tự.

(b) Dùng mã hóa Huffman để suy ra một tập từ mã và chứng minh tập này là tối ưu bằng cách xây dựng cây mã Huffman tương ứng.

(c) Suy ra số bit trung bình trên một ký tự cho tập mã này và so sánh với:

(i) Giá trị Shannon

(ii) Các từ mã nhị phân có chiều dài cố định

(iii) Các từ mã ASCII 7 bit

Tóm tắt lời giải:

(a) Theo công thức Shannon :

Entropy

$$H = -\sum_{i=1}^n P_i \log_2 P_i \text{ bit/codeword}$$

Do đó :

$$\begin{aligned} H &= -(2(0,25 \log_2 0,25) + 2(0,14 \log_2 0,14) + 4(0,055 \log_2 0,055)) \\ &= 1 + 0,794 + 0,921 = 2,715 \text{ bit/từ mã} \end{aligned}$$

(b) Việc suy ra bộ từ mã dùng mã hóa Huffman được trình bày trên hình 4.12(a). Trước hết các ký tự được liệt kê theo thứ tự trọng số và hai ký tự dưới cùng của danh sách được gán nhánh (1) và (0). Tuy nhiên trong trường hợp này, khi hai node kết hợp lại, trọng số của node nhánh (0,11) lớn hơn trọng số của hai ký tự E và F (0,055). Từ đó node nhánh này được

chèn vào danh sách thứ hai có thứ tự cao hơn cả hai ký tự E và F. Thủ tục tương tự được lặp lại cho đến khi danh sách chỉ còn hai node thì dừng.

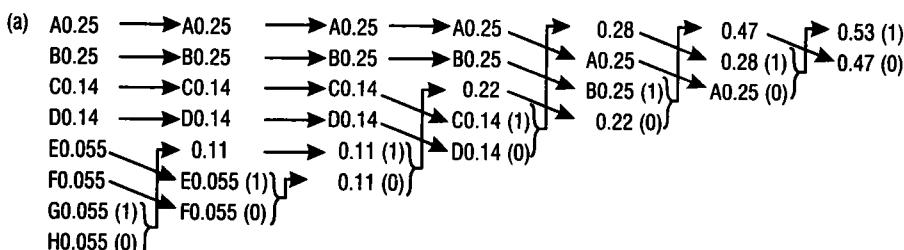
Cây Huffman tương ứng với tập từ mã được suy ra được biểu diễn trên hình 4.12(b), và đây là cây tối ưu vì tất cả các node nhánh và node lá đều tăng theo thứ tự số học.

(c) Số bit trung bình trong một từ mã dùng mã Huffman là :

$$2(2 \times 0,25) + 2(3 \times 0,14) + 4(4 \times 0,055) = 2,72 \text{ bit/từ mã, bằng } 99,8\% \text{ giá trị Shannon.}$$

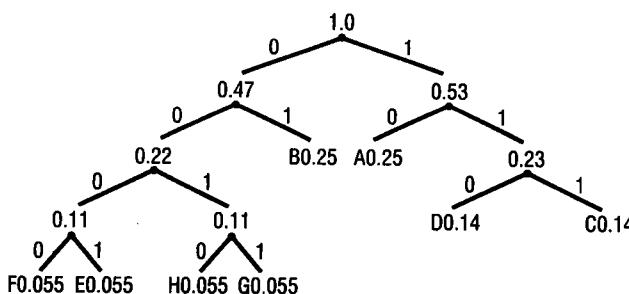
Dùng các từ mã nhị phân có chiều dài cố định :

Từ A đến H có 8 ký tự do đó có 3 bit trong mỗi từ mã, bằng 90,7% giá trị Huffman.



$A = (0) (1)$	$\rightarrow 10$
$B = (1) (0)$	$\rightarrow 01$
$C = (1) (1) (1)$	$\rightarrow 111$
$D = (0) (1) (1)$	$\rightarrow 110$
$E = (1) (0) (0) (0)$	$\rightarrow 0001$
$F = (0) (0) (0) (0)$	$\rightarrow 0000$
$G = (1) (1) (0) (0)$	$\rightarrow 0011$
$H = (0) (1) (0) (0)$	$\rightarrow 0010$

(b)



$$\text{Thứ tự độ lớn} = 0.055 \ 0.055 \ 0.055 \ 0.055 \ 0.11 \ 0.11 \ 0.14 \ 0.22 \ 0.25 \ 0.25 \ 0.28 \ 0.47 \ 0.53$$

Hình 4.12 Ví dụ mã hóa Huffman:

(a) Phát sinh từ mã (b) Cây mã Huffman.

Dùng các từ mã ASCII 7 bit :

7 bit trong một từ mã, bằng 38,86% giá trị Huffman.

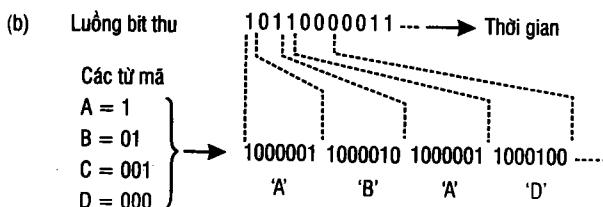
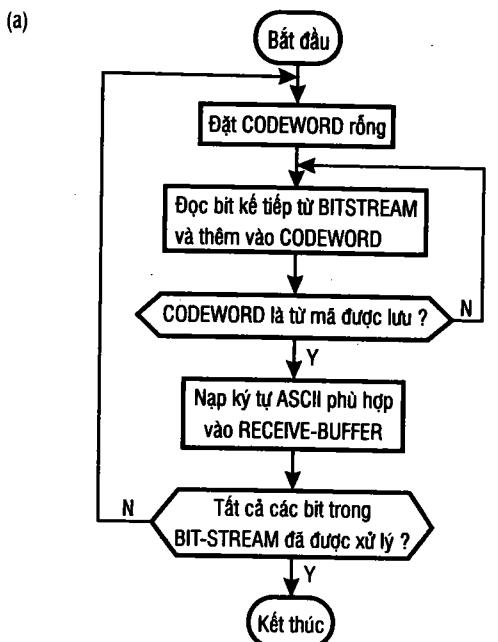
Vì mỗi ký tự trong dạng mã hóa này có số bit thay đổi, nên luồng bit nhận được phải được dịch ra theo từng bit một thay vì dịch theo ranh giới 8 bit cố định. Do thứ tự các bit được gán trong quá trình mã hóa mà các từ mã Huffman có một *đặc tính duy nhất* là từ mã ngắn hơn sẽ không bao giờ trùng với phần đầu của một từ mã dài hơn. Ví dụ nếu 011 là một từ mã hợp lệ thì không có một từ mã nào dài hơn chứa 011 kể từ bit đầu. Chúng ta có thể kiểm chứng điều này từ các mã được suy ra từ ví dụ trên.

Đặc tính này còn được gọi là *đặc tính tiền tố* (prefix), có nghĩa là luồng bit nhận được có thể được giải mã một cách đơn giản bằng cách dò từng bit một cho đến khi tìm thấy được từ mã hợp lệ. Một lưu đồ thuật toán giải mã thích hợp như hình 4.13. Giải thuật này giả sử ở máy thu đã có bảng mã và cũng giữ từ mã ASCII tương ứng. Luồng bit nhận được giữ trong biến BITSTREAM và biến CODEWORD được dùng để giữ mỗi từ mã trong quá trình nó được xây dựng. Từ giải thuật chúng ta có thể suy ra rằng khi một từ mã đã được xác nhận thì từ mã ASCII tương ứng được ghi trong biến RECEIVE_BUFFER. Thủ tục này được lặp lại cho tất cả các bit trong chuỗi bit nhận. Ví dụ một chuỗi giải mã tương ứng với tập từ mã được suy ra từ hình 4.11 được trình bày trong hình 4.13(b).

Khi cây mã Huffman (các từ mã được suy ra từ đây) thay đổi theo các tập ký tự đang được truyền, để máy thu thực hiện hoạt động giải mã nó phải biết tập từ mã liên hệ với dữ liệu đang được truyền. Điều này có thể được thực hiện theo hai cách. Tập từ mã liên hệ với tập dữ liệu kế được truyền trước khi dữ liệu này được truyền, hoặc máy thu được thông báo dùng tập từ mã nào.

Tiếp cận đầu tiên dẫn đến dạng nén thích nghi vì các từ mã có thể thay đổi để phù hợp với loại dữ liệu đang được truyền. Điều bất tiện ở đây là phần overhead chứa tập từ mã mới phải được truyền trước bất cứ khi nào một dạng dữ liệu mới được truyền. Cách thứ hai là máy thu phải biết một hay nhiều bộ từ mã khác nhau và máy phát chỉ cho máy thu (qua một thông điệp đàm phán) biết bộ từ mã dùng cho tập dữ liệu kế tiếp.

Ví dụ theo nhu cầu thông thường là truyền các tập tin dạng text được tạo ra từ các trình xử lý văn bản (và do đó chứa thông tin dạng text thông thường), các phân tích thống kê chi tiết được thực hiện theo tần suất xuất hiện các ký tự alphabet trong văn bản thông thường. Thông tin này được dùng để xây dựng cây mã Huffman cho alphabet. Nếu dạng dữ liệu alphabet được truyền, máy thu và máy phát sẽ tự động dùng tập từ mã này. Các tập dữ liệu chung khác được phân tích theo phương pháp tương tự.



Hình 4.13 Giải mã chuỗi bit nhận, giả sử các từ mã được lấy từ hình 4.11:
(a) Thuật toán giải mã (b) Ví dụ.

3.6. Mã hóa Huffman động

Phương pháp mã hóa Huffman cơ bản yêu cầu cả máy thu và máy phát biết được bảng mã liên hệ với dữ liệu đang được truyền. Một phương pháp khác cho phép máy phát và máy thu xây dựng cây Huffman một cách tự động khi các ký tự được truyền và nhận. Dạng mã hóa này được gọi là mã hóa Huffman động.

Với phương pháp này, nếu ký tự đang được truyền là hiện hữu trên cây thì từ mã của nó được xác định và được truyền đi theo phương pháp thông thường. Nếu ký tự không hiện hữu trên cây, đây là lần xuất hiện đầu tiên, thì ký tự được truyền dưới dạng không nén. Bộ mã hóa cập nhật cây Huffman của nó bằng cách gia tăng tần suất xuất hiện ký tự được truyền hoặc giới thiệu ký tự mới vào trong cây.

Mỗi từ mã truyền được mã hóa theo phương pháp này, bộ thu cũng có thể thực hiện các sửa chữa tương tự đối với bản copy của cây để nó có thể dịch từ mã nhận được kế tiếp tùy vào cấu trúc cây mới được cập nhật.

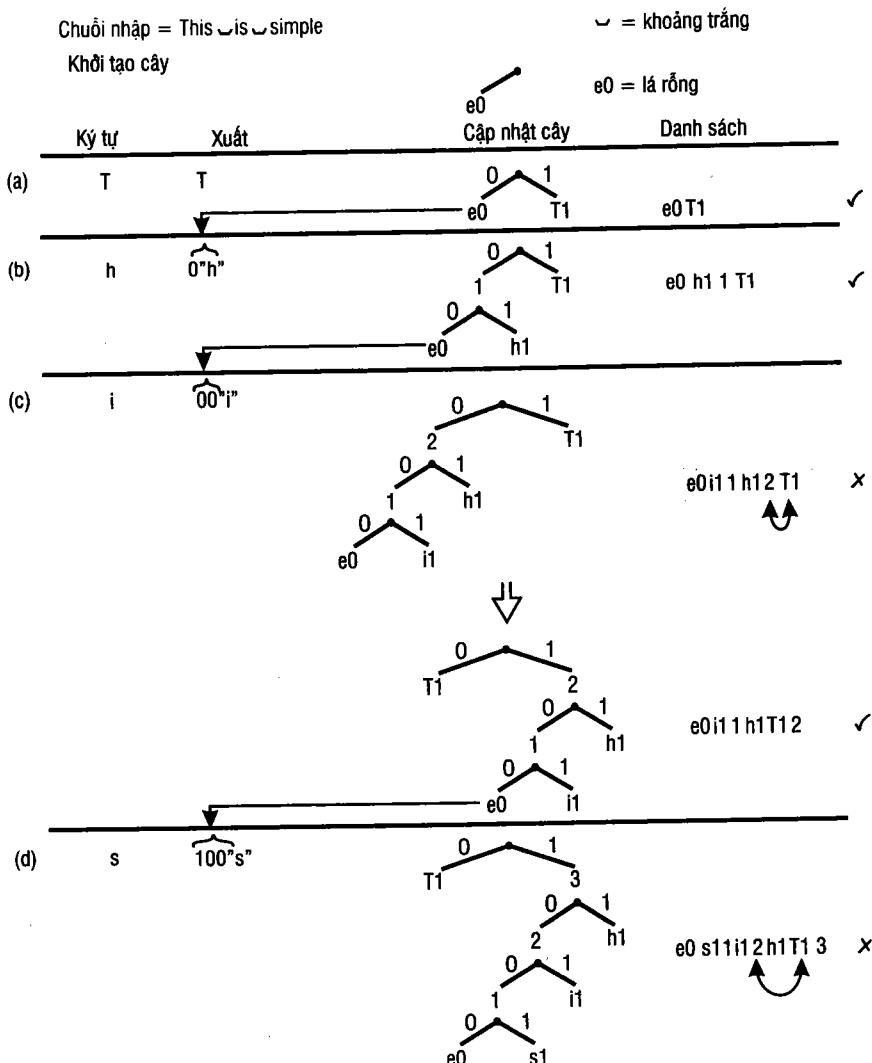


Để mô tả chi tiết phương pháp này, giả sử dữ liệu được truyền bắt đầu với chuỗi ký tự sau:

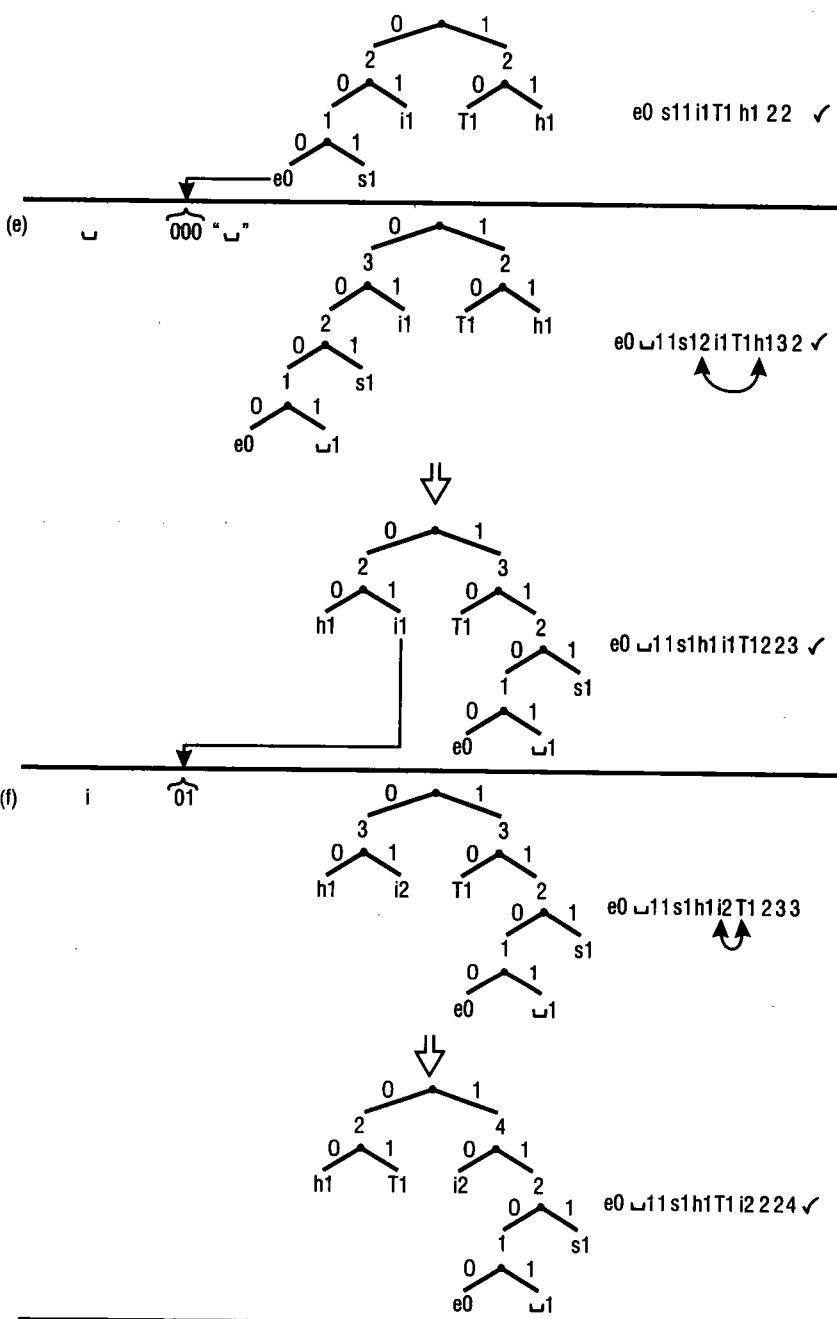
This is simple.....

Các bước thực hiện của máy phát được trình bày trên hình 4.14(a đến g).

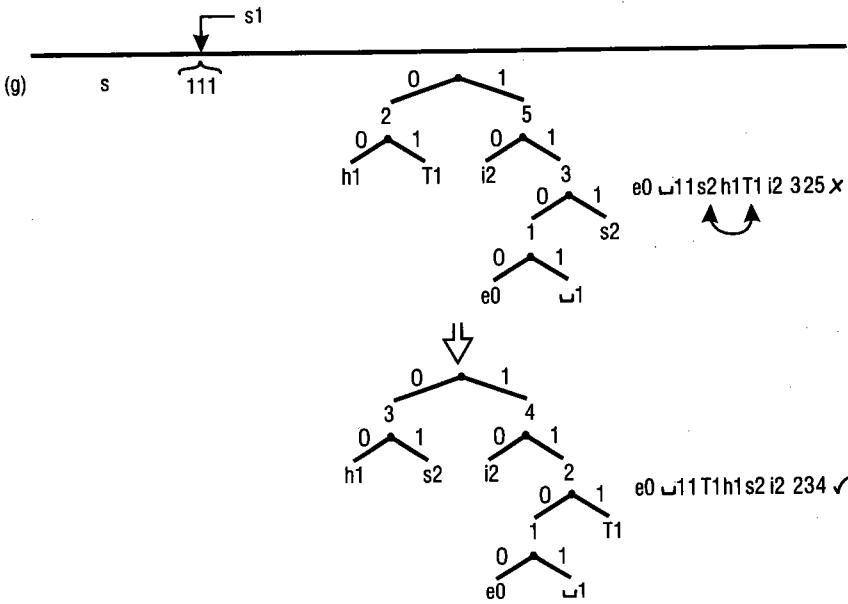
Cả máy phát và máy thu đều bắt đầu với cây bao gồm node gốc và một node lá rỗng_một node lá rỗng có tần suất xuất hiện 0 _được gán là nhánh 0. Luôn chỉ có một node như vậy trong cây và vị trí của nó thay đổi khi cây đang được xây dựng. Nó được trình bày trên hình 4.14 bằng ký hiệu e0.



Hình 4.14 Ví dụ mã hóa Huffman động.



Hình 4.14 (tiếp theo) Ví dụ mã hóa Huffman động.



Hình 4.14 (tiếp theo) Ví dụ mã hóa Huffman động

Sau đó bộ mã hóa bắt đầu đọc ký tự đầu T và gán nó vào nhánh 1 của gốc. Vì đây là lần xuất hiện đầu tiên của ký tự này, nó được ký hiệu là T1 và được truyền dưới dạng không nén, đó là mã ASCII. Vì cây giải mã cũng đang rỗng, nó dịch chuỗi bit nhận được như một ký tự không nén và tiến hành gán ký tự vào trong cây theo phương pháp tương tự.

Trong mỗi ký tự sau đó, trước hết bộ mã hóa kiểm tra ký tự đã có trong cây hay chưa. Nếu đã có, thì bộ mã hóa sẽ gửi từ mã của ký tự theo cách thông thường, từ mã hiện hành được xác định theo vị trí của ký tự trong cây. Nếu ký tự chưa có trong cây, bộ mã hóa gửi từ mã hiện hành cho lá rỗng cũng được xác định nhờ vào vị trí của nó trong cây kèm theo từ mã không nén cho ký tự này. Vì bộ giải mã có cùng cây mã với bộ mã hóa, nên có thể suy ra dễ dàng từ chuỗi bit nhận được từ mã của một ký tự (không nén) hay lá rỗng kèm theo ký tự ở dạng không nén của nó.

Bộ mã hóa và giải mã của nó tiến hành cập nhật bản copy cây mã dựa vào ký tự sau cùng được truyền hay nhận. Nếu nó là một ký tự mới, thì node lá rỗng tồn tại trong cây được thay thế bằng một node nhánh mới, node lá rỗng được gán là nhánh 0 và ký tự được gán là nhánh 1 (hình 4.14(b)).

Nếu ký tự có sẵn trong cây, thì tần suất xuất hiện node lá tăng một đơn vị. Trong quá trình thực hiện điều này, vị trí của node lá lúc này có thể không phải là vị trí tối ưu trong cây. Vì vậy cứ mỗi lần cây được cập nhật thêm vào một từ mới hay gia tăng tần suất xuất hiện một từ đã có, cả bộ mã hóa và giải mã đều kiểm tra và nếu cần thiết thì sửa đổi vị trí hiện hành của tất cả các ký tự trong cây.

Để đảm bảo cả bộ mã hóa và giải mã đều thực hiện điều này theo một phương pháp không đổi, trước hết chúng liệt kê trọng số của các node lá và node nhánh trong cây được cập nhật theo hướng từ trái qua phải và từ dưới lên trên bắt đầu là lá rỗng. Nếu tất cả chúng theo thứ tự trọng số thì tất cả các vị trí đều tốt và cây được giữ nguyên. Nếu có một node nằm ngoài trật tự, thì cấu trúc của cây được sửa đổi bằng cách trao đổi vị trí của node này với node khác trong cây_kết hợp với node lá và node nhánh của nó _để tạo ra danh sách theo đúng thứ tự. Sự xuất hiện lần đầu như trong hình 4.14(c) và các ví dụ khác ở trong (d) và (g).

Các bước tiếp theo khi một ký tự được truyền được mô tả trong hình 4.14(f). Ở đó ký tự được truyền là 'i' và khi bộ mã hóa duyệt cây, nó xác định rằng 'i' đã có và sẽ truyền từ mã của nó là 01. Bộ mã hóa sau đó tăng trọng số của ký tự -tần suất xuất hiện-lên một thành i2 và cập nhật vị trí của node được sửa đổi như trước. Một ví dụ khác được trình bày trên hình 4.14(g) khi ký tự 's' được truyền.

Từ ví dụ này có thể suy ra việc tiết kiệm băng thông truyền chỉ khi các ký tự bắt đầu lặp lại chính chúng. Trong thực tế, khả năng tiết kiệm đối với các tập tin dạng text là đáng kể, và ngày nay mã hóa Huffman được dùng nhiều trong các ứng dụng truyền số liệu, như giải thuật nén dữ liệu trong các modem dùng chuẩn V.32.

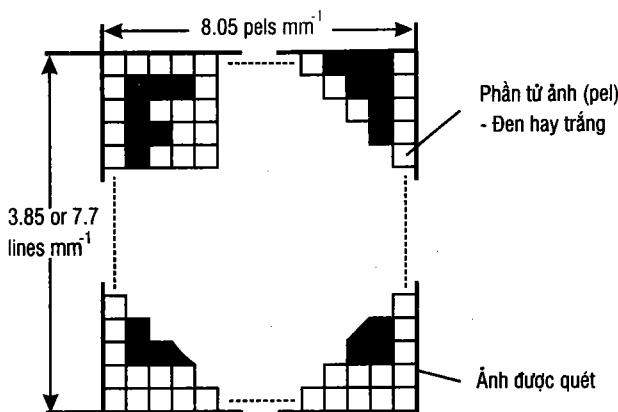
3.7. Nén fax (facsimile)

Mặc dù chúng ta có thể đạt được tỉ lệ nén 2:1 trong các tập tin dùng mã hóa Huffman nhưng tiết kiệm nhất vẫn là trong ứng dụng truyền các ảnh đã số hóa được tạo ra bởi các máy quét liên quan đến các máy fax. Như hình 4.15 trình bày, một trang được quét với phân giải doc 3,85 hay 7,7 dòng trên một mili mét_xấp xỉ 100 hay 200 dòng trên một inch. Mỗi dòng quét được số hóa với tốc độ 8,05 phần tử ảnh hay pel trên một mili mét_0 tương ứng cho một phần tử trắng và 1 cho một phần tử đen. Do đó một trang quét tiêu biểu có khoảng hai triệu ký số nhị phân. Để truyền trang này ở dạng không nén, giả sử dùng tốc độ 4800bps, sẽ cần tối thiểu 6 phút.

Thực tế, trong hầu hết các văn bản, nhiều dòng văn bản được quét chỉ bao gồm các chuỗi chứa các phần tử trắng trong khi các dòng khác gồm các chuỗi hỗn hợp các phần tử trắng và đen. Vì máy fax thường được dùng trong các mạng công cộng, nên ITU-T đã đưa ra các chuẩn liên quan đến chúng. Đó là chuẩn T2 (nhóm 1), T3 (nhóm 2), T4 (nhóm 3) và T6 (nhóm 4). Hai chuẩn đầu tiên nay ít dùng. Tuy nhiên, hai chuẩn sau đều ở dạng số hóa được dùng rộng rãi, nhóm 3 có điều chế được dùng trong mạng PSTN analog, và nhóm 4 là số hoàn toàn dùng cho các mạng số như ISDN. Cả hai đều dùng kỹ thuật nén dữ liệu, và tỉ lệ nén tối thiểu là 10:1 cho hầu hết các trang văn bản. Thời gian cần thiết truyền một trang giảm xuống dưới một phút đối với các máy fax dùng chuẩn nhóm 3, và nhờ vào ưu thế tốc độ truyền dẫn cao (64kbps) mà đối với các máy fax dùng tiêu chuẩn nhóm 4 thời gian cần để truyền một trang giảm xuống còn vài giây.

Trong quá trình chuẩn hóa, các phân tích các trang văn bản tiêu biểu được tiến hành. Các bảng mã được tạo ra dựa vào tần suất xuất hiện của số các phần tử trắng hay đen liên tục được tìm thấy trong dòng quét. Các từ mã kết quả là cố định và được nhóm vào trong hai bảng tách biệt: bảng terminateion-codes và bảng make-up codes. Các từ mã trong mỗi bảng được trình bày trên hình 4.16.

Các từ mã trong bảng termination-code là cho chiều dài của khoảng trắng hay đen từ 0 đến 63 tính theo các bước 1 phần tử; bảng make-up codes chứa các từ mã cho chiều dài trắng hay đen là bội số của 64. Dùng một kỹ thuật được gọi là overscanning, có nghĩa là tất cả các dòng được khởi đầu với tối thiểu một phần tử trắng. Trong phương pháp này, máy thu biết từ mã đầu tiên luôn liên hệ với các phần tử trắng và sau đó luân phiên giữa trắng và đen. Vì lược đồ này dùng hai tập mã nên chúng thường được gọi là **các mã Huffman sửa đổi**. Ví dụ một chiều dài gồm 12 phần tử trắng được mã hóa một cách trực tiếp thành 001000. Tương tự một chiều dài gồm 12 phần tử đen được mã hóa một cách trực tiếp thành 0000111. Tuy nhiên chiều dài của 140 phần tử đen được mã hóa là 000011001000 + 0000111; đó là 128 + 12 phần tử. Các chiều dài quá 2560 phần tử được mã hóa dùng nhiều hơn một mã make-up cộng với một mã termination.



Hình 4.15 Lược đồ ảnh được quét bằng máy fax.

Không có thủ tục sửa lỗi đối với nhóm 3. Từ danh sách từ mã, có thể suy ra rằng nếu một hay nhiều bit bị gián đoạn bởi các lỗi truyền, máy thu sẽ bắt đầu dịch các từ mã kế tiếp theo các ranh giới bit không chuẩn xác. Vì vậy máy thu đã mất đồng bộ và không thể giải mã chuỗi bit nhận được. Để máy thu đồng bộ trở lại, mỗi dòng quét được kết thúc bởi một mã kết thúc dòng EOL (End-Of-Line) biết trước. Bằng cách này khi máy thu thất bại trong việc giải mã một từ mã hợp lệ sau một số bit tối đa trong một từ mã đã được duyệt, nó bắt đầu vào công đoạn dò tìm mẫu EOL. Tương tự, nếu máy thu thất bại trong việc giải mã một EOL sau một số các dòng xác lập trước, nó sẽ hủy quá trình nhận và thông báo cho máy phát. Một chuỗi 6 mẫu EOL liên tục chỉ định kết thúc một trang.

Số phần tử trắng	Từ mã	Số phần tử đen	Từ mã
0	00110101	0	0000110111
1	000111	1	010
2	0111	2	11
3	1000	3	10
4	1011	4	011
5	1100	5	0011
6	1110	6	0010
7	1111	7	00011
8	10011	8	000101
9	10100	9	000100
10	00111	10	0000100
11	01000	11	0000101
12	0001000	12	0000111
13	000011	13	00000100
14	110100	14	00000111
15	110101	15	000011000
16	101010	16	0000010111
17	101011	17	0000011000
18	0100111	18	0000010000
19	0001100	19	00001100111
20	0001000	20	00001101000
21	0010111	21	00001101100
22	0000011	22	00000110111
23	0000100	23	00000101000
24	0101000	24	00000010111
25	0101011	25	00000011000
26	0010011	26	000011001010
27	0100100	27	000011001011
28	0011000	28	000011001100
29	00000010	29	000011001101
30	00000011	30	000001101000
31	00011010	31	000001101001
32	00011011	32	000001101010
33	0010010	33	000001101011
34	00010011	34	000011010010
35	00010100	35	000011010011
36	00010101	36	000001010100
37	00010110	37	000001010101
38	00010111	38	000001010110
39	00101000	39	000001010111
40	00101001	40	0000010101100
41	00101011	41	00000010101101
42	001010111	42	000000101011010
43	00101100	43	0000001010111
44	00101101	44	0000001010100
45	000000100	45	0000001010101
46	000000101	46	0000001010110
47	000001010	47	0000001010111
48	000001011	48	000000100100
49	010200010	49	0000001100101
50	01010011	50	0000001010010
51	01010100	51	0000001010011
52	01010101	52	000000100100
53	00100100	53	0000000110111
54	00100101	54	0000000111000
55	01011000	55	0000000100111

(a)

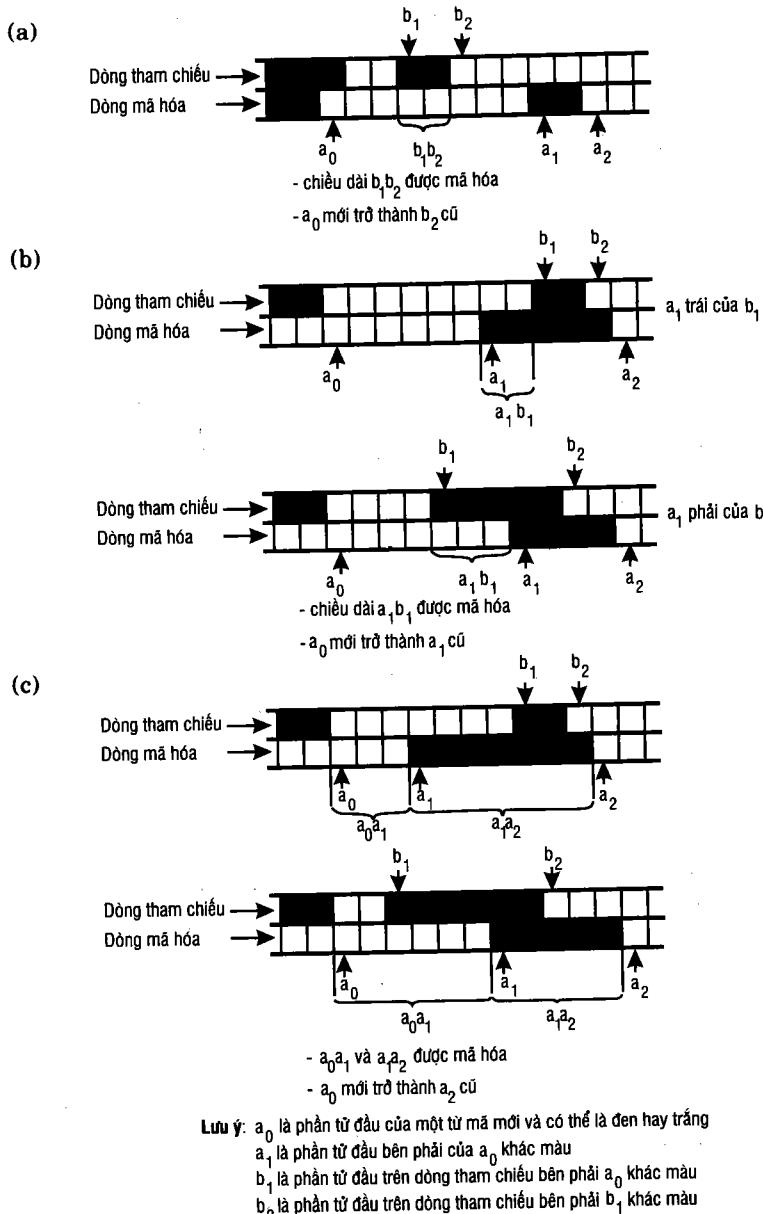
Số phần tử trắng	Từ mã	Số phần tử đen	Từ mã
56	01011001	56	000000101000
57	01011010	57	0000001011000
58	01011011	58	0000001011001
59	01001010	59	0000000101011
60	01001011	60	0000000101100
61	00110010	61	000001011010
62	00110011	62	000001100110
63	00110100	63	000001100111

(a)

Số phần tử trắng	Từ mã	Số phần tử đen	Từ mã
64	11011	64	0000001111
128	10010	128	000011001000
192	010111	192	000011001001
256	0110111	256	000001011011
320	0011010	320	000000110011
384	00110111	384	000000110100
448	01100100	448	000000110101
512	01100101	512	0000001101100
576	01101000	576	0000001101101
640	01100111	640	0000001001010
704	011001100	704	0000001001011
768	011001101	768	0000001001100
832	011010010	832	0000001001101
896	011010011	896	000000110010
960	011010100	960	000000110011
1024	011010101	1024	0000001110100
1088	011010110	1088	0000001110101
1152	011010111	1152	0000001110110
1216	011011000	1216	0000001110111
1280	011011001	1280	0000001010010
1344	011011010	1344	0000001010011
1408	011011011	1408	0000001010100
1472	010011000	1472	00000001010101
1536	010011001	1536	0000000101010
1600	010011010	1600	0000000101101
1664	011000	1664	00000001100100
1728	010011011	1728	00000001100101
1792	00000001000	1792	000000001000
1856	00000001100	1856	000000001100
1920	00000001101	1920	000000001101
1984	000000010010	1984	0000000010010
2048	000000010011	2048	0000000010011
2112	000000010100	2112	0000000010100
2176	000000010101	2176	0000000010101
2240	000000010110	2240	0000000010110
2304	000000010111	2304	0000000010111
2368	000000011100	2368	0000000011100
2432	000000011101	2432	0000000011101
2496	000000011110	2496	0000000011110
2560	000000011111	2560	0000000011111
EOL	000000000001	EOL	000000000001

(b)

Hình 4.16 Các mã fax nhóm 3 và nhóm 4 của ITU-T:
(a) Termination codes (b) Make-up codes.



Hình 4.17 Các ví dụ về khả năng:

(a) Pass mode (b) Vertical mode (c) Horizontal mode.



Vì mỗi dòng được quét được mã hóa một cách độc lập, nên lược đồ mã hóa T4 được gọi là lược đồ **one-dimensional coding**, và lược đồ này phù hợp với các ảnh có kích thước vùng trắng hay đen khá lớn, ví dụ các văn bản gồm các ký tự hay các ảnh out-line. Tuy nhiên các văn bản chứa các ảnh đồ họa thì không phù hợp vì độ bóng của đen hay trắng được biểu diễn qua mật độ các phần tử trắng hay đen. Như vậy có một số lớn các khoảng trắng hay đen rất ngắn mà với lược đồ T4 có thể dẫn đến một tỉ số nén âm, có nghĩa là số bit sau khi nén lại nhiều hơn số bit cần truyền ở dạng không nén.

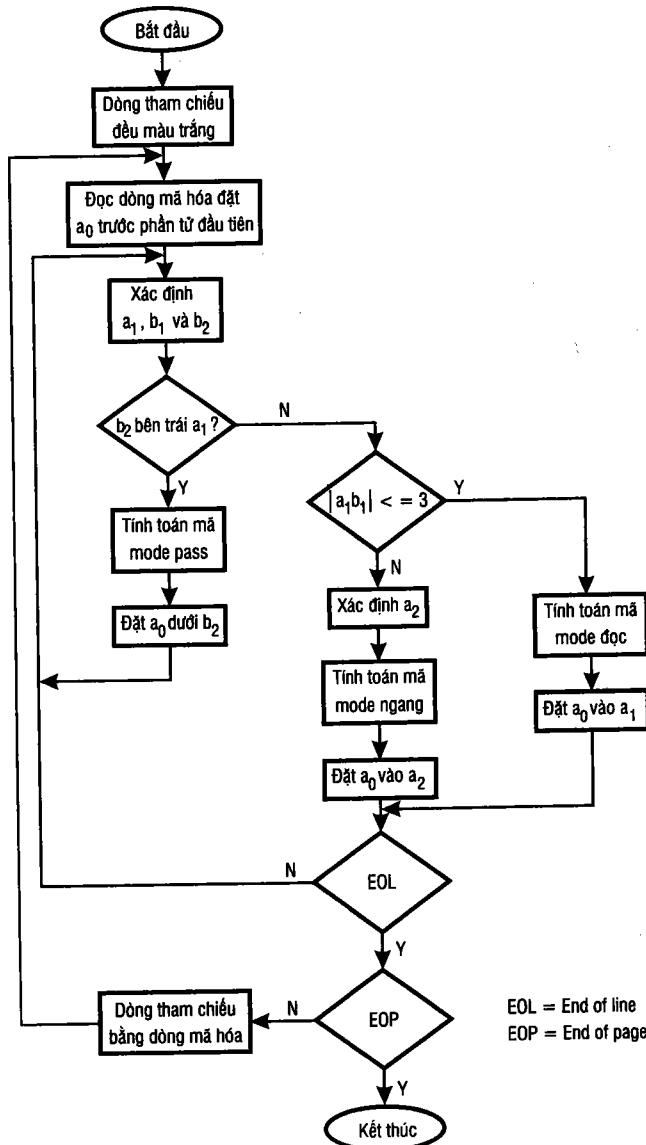
Vì lý do này mà lược đồ T6 đã được định nghĩa. Nó là một đặc tính tùy chọn đối với các máy fax dùng nhóm 3 nhưng lại là bắt buộc trong các máy fax nhóm 4. Khi hỗ trợ trong các máy nhóm 3, mã EOL tại cuối của mỗi dòng được nén có thêm vào một bit làm nhãn. Nếu đây là bit nhị phân 1 thì dòng kế tiếp được mã hóa theo lược đồ T4, nếu là bit nhị phân 0 thì dòng kế tiếp dùng lược đồ T6. Lược đồ T6 được gọi là mã hóa MMR (modified-modified READ). Cũng được gọi là **two-dimensional coding (2D coding)** vì nó nhận diện các khoảng trắng hay đen bằng cách so sánh các dòng quét kế cận. Từ READ là viết tắt của Relative Element Address Designate, từ ‘modified’ ngụ ý rằng nó là một phiên bản sửa đổi từ một lược đồ mã hóa có trước.

Mã hóa MMR khai thác một đặc tính: hầu hết các dòng quét chỉ khác nhau một ít phần tử ở đầu dòng. Ví dụ, nếu một dòng chứa một khoảng đen thì thông thường dòng kế sẽ chứa khoảng bằng chiều dài đó cộng hay trừ đi 3 phần tử. Với mã hóa MMR các chiều dài liên hệ với một dòng được nhận dạng bằng cách so sánh nội dung của dòng, dòng này được gọi là **coding line (CL)**, và liên hệ với dòng ngay trước đó, dòng đó gọi là **reference line (RL)**. Luôn giả sử dòng RL đầu tiên là dòng trắng hoàn toàn và dòng đầu tiên phù hợp được mã hóa theo quan hệ này. Kế đến dòng được mã hóa trở thành RL cho dòng theo sau và cứ thế. Để đảm bảo các trang được quét hoàn toàn, bộ quét luôn khởi đầu từ trái sang, và mỗi dòng luôn bắt đầu với một phần tử trắng.

Chúng ta nhận dạng các chiều dài liên kết với một dòng CL theo một trong ba khả năng hay ba chế độ quan hệ với dòng tham khảo (RL). Các ví dụ ứng với 3 chế độ quan hệ này được trình bày trên hình 4.17. Ba chế độ được nhận dạng bởi vị trí của chiều dài kế trong RL (b_1, b_2) liên hệ với điểm khởi đầu và kết thúc của cặp chiều dài trong CL (a_0, a_1 và a_1, a_2). Ghi chú rằng mỗi khoảng có thể là trắng hay đen. Ba khả năng là :

- 1) **Pass mode** : trường hợp này khi chiều dài trong RL (b_1, b_2) nằm bên trái của chiều dài kế trong CL (a_1, a_2), nghĩa là b_2 bên trái a_1 . Ví dụ trên hình 4.17(a); trong mode này chiều dài b_1, b_2 được mã hóa dùng các từ mã cho trong hình 4.16. Chú ý rằng nếu phần tử kế tiếp trên CL, a_1 , nằm dưới b_2 thì đây không phải là pass mode.
- 2) **Vertical mode** : đây là trường hợp khi chiều dài trong RL (b_1, b_2) chồng lấn lên chiều dài kế tiếp trong CL (a_1, a_2) một số tối đa cộng hay trừ 3 phần tử. Hai ví dụ được trình bày trên hình 4.17(b); trong mode này, chỉ chiều dài khác nhau a_1, b_1 được mã hóa. Hầu hết các từ mã đều thuộc loại này.
- 3) **Horizontal mode**: đây là trường hợp khi chiều dài trong RL (b_1, b_2) chồng lấn chiều dài (a_1, a_2) nhiều hơn cộng hay trừ 3 phần tử. Hai ví dụ được trình bày trên hình 4.17(c); trong chế độ này, hai chiều dài a_0, a_1 và a_1, a_2 được mã hóa dùng các từ mã trong hình 4.16.

Lưu đồ của giải thuật mã hóa được trình bày trên hình 4.18. Lưu ý rằng a_0 được gán là một phần tử trắng đứng trước phần tử đầu tiên của dòng và do đó chiều dài đầu tiên sẽ là $a_0a_1\ldots a_n$. Nếu trong quá trình mã hóa của một dòng a_1, a_2, b_1 , hay b_2 không được phát hiện, thì chúng được gán phần tử đặt ngay sau phần tử cuối cùng trên dòng.



Hình 4.18

Thủ tục mã hóa MMR.

Khi vị trí đầu tiên hay kế tiếp của a0 được xác định, các vị trí a1, a2, b1, và b2 trong từ mã kế tiếp được định vị. Mode được xác định sau đó bằng cách tính vị trí của b2 liên hệ với a1. Nếu nó ở bên trái, điều này chỉ ra pass mode. Nếu nó không phải ở bên trái, thì giá trị của a1b1 được dùng để xác định mode này là dọc (vertical) hay ngang (horizontal). Sau đó, từ mã trong mode đã xác định được tính toán và vị trí bắt đầu của từ mã kế tiếp a0, được cập nhật vị trí thích hợp. Đây là một phần tử được xác lập vị trí ngay sau phần tử sau cùng của dòng và giả sử có màu khác với phần tử sau cùng. Sau đó dòng mã hóa hiện hành trở thành RL mới và dòng được quét kế tiếp là CL mới.

Vì chiều dài được mã hóa liên quan đến một trong ba mode, các từ mã thêm vào được dùng để chỉ định mode nào liên hệ với các từ mã theo sau *_pass hay horizontal_* hay chỉ trực tiếp chiều dài của từ mã *_vertical*. Các từ mã thêm vào được cho trong bảng thứ ba được gọi là bảng **two-dimention code**, trong bảng 4.2 dòng cuối cùng trong bảng được gọi là **extension mode**, là một từ mã duy nhất kết thúc hoạt động mã hóa trước khi kết thúc trang. Điều này cho phép một phần của trang được truyền dưới dạng không nén hay có thể với lược đồ mã hóa khác.

Bảng 4.2 Nội dung của bảng mã two-dimensional.

Mode	Chiều dài được mã hóa	Ký hiệu	Từ mã
Pass	b1b2	P	0001 + b1b2
Horizontal	A0a1, a1a2	H	001+a0a1+a1a2
Vertical	A1b1=0	V(0)	1
	A1b1=-1	V _R (1)	011
	A1b1=-2	V _R (2)	000011
	A1b1=-3	V _R (3)	0000011
	A1b1=+1	V _L (1)	010
	A1b1=+2	V _L (2)	000010
	A1b1=+3	V _L (3)	0000010
Extension			0000001000

4. MẬT MÃ HÓA SỐ LIỆU

4.1. Khái quát

Đường truyền số liệu trong một số ứng dụng cần phải bảo mật, ví dụ như quốc phòng, ngân hàng .v.v. Như vậy, ngoài những biện pháp xử lý số liệu cần thiết để truyền thành công và hiệu quả, số liệu còn được mã hóa bằng phương pháp nào đó, theo một khóa nào đó mà chỉ máy phát và máy thu mới biết được.

Quá trình mã hóa và giải mã thường được thực hiện ở mức liên kết số liệu (data link) trở lên. Tuy nhiên cũng có những vi mạch cỡ lớn chuyên thực hiện mã hóa và giải mã số liệu. Các chip này cho phép người sử dụng thay đổi các giải thuật mã phức tạp với rất nhiều khóa khác nhau để lựa chọn. Ngày nay mã hóa mức vật lý cũng được quan tâm nhiều, đặc biệt là mã theo hướng công nghệ, một số phương pháp lợi dụng công nghệ cao để tiến hành mã hóa, thám mã muốn biết bản gốc phải đạt được trình độ công nghệ tương đương mới có thể thực hiện được.

4.2. Mật mã hóa cổ điển

Bản gốc sẽ được mã hóa bằng một khóa được xác định trước để tạo ra một bản mã. Bản mã chính là bản được truyền lên kênh. Khi thâm nhập vào kênh, đối phương có thể thu trộm được bản mã nhưng vì không biết khóa nên khó tìm ra được bản gốc.

Về mặt toán học có thể mô phỏng hoạt động mật mã cổ điển như sau:

Một hệ thống mật mã là một tập có 5 thành phần (P, C, K, E, D) trong đó:

- P là tập hữu hạn các bản gốc có thể
- C là tập hữu hạn các bản mã có thể
- K là tập khóa có thể
- Đối với mỗi $k \in K$ có một luật mật mã $e_k: P \rightarrow C$, $e_k \in E$ và một luật giải mã tương ứng $d_k: C \rightarrow P$, $d_k \in D$. Mỗi e_k và d_k là những ánh xạ sao cho: $d_k(e_k(x)) = x \quad \forall x \in P$

Giả sử thông tin gốc cần gửi đi là một chuỗi $x = x_1, x_2, x_3, x_4, \dots, x_n$, với n là một số nguyên lớn hơn hay bằng 1. Mỗi một ký hiệu x_i ($1 \leq i \leq n$) được mã hóa bằng luật e_k với khóa K đã thống nhất với bên thu. Nơi phát sẽ xác định các y_i theo $y_i = e_k(x_i)$ và bản mã sẽ phát lên kênh là: $y = y_1, y_2, y_3, \dots, y_n$. Ở phía thu hợp lệ, sẽ tìm ra bản gốc bằng cách dùng ánh xạ $d_k(y_i) = x_i$. Cũng cần nói thêm rằng e_k và d_k phải là các ánh xạ 1-1, nghĩa là với $x_1 \neq x_2$ thì $y_1 \neq y_2$.

Có nhiều phương pháp mật mã cổ điển đã được dùng trong quá khứ, ví dụ như mã dịch vòng, mã thay thế, mã Affine, mã Vigenére . . . Sau đây sẽ trình bày phương pháp mật mã dịch vòng.

Phương pháp mã dịch vòng có cơ sở là phép toán modulo. Để minh họa cho phương pháp này ta xem xét công việc mật mã trên bộ chữ cái tiếng Anh gồm 26 chữ cái. Ở đây sẽ dùng phép modulo 26 như sau:

$$e_K(x) = x + K \bmod 26$$

$$d_K(x) = y - K \bmod 26$$

Sự tương ứng của các chữ cái và các số dư theo modulo 26 như sau:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Ví dụ:

Giả sử dùng khóa cho mã dịch vòng này là 9 và bản gốc là :

gonewiththewind

Tiến hành mật mã như sau:

Trước hết chúng ta sẽ biến đổi bản gốc thành chuỗi các số nguyên theo phép lấy tương ứng bảng đã cho ở trên. Ta được:

6	14	13	4	22	8	19	7
19	7	4	22	8	13	3	

sau đó cộng thêm 9 vào mỗi giá trị rồi modulo 26 ta được:

15	23	22	13	5	17	2	16
2	16	13	5	17	22	12	

từ chuỗi các giá trị trên đây lấy các ký tự tương ứng trong bảng trên ta được bảng mã sẽ truyền lên kênh là:

pxwnfrcqcqnfrwm

khi thu được bản mã này, máy thu sẽ tiến hành biến đổi thành dãy các giá trị tương ứng trong bảng. Lấy từng giá trị trừ bớt cho 9 rồi modulo 26, đổi giá trị của kết quả thành ký tự, cuối cùng sẽ được bản gốc.

4.3. Mật mã khóa công khai

Một trong những phương pháp mật mã hóa hiện đại là mật mã khóa công khai. Phương pháp mật mã này ứng dụng tính chất đặc biệt của các *hàm bẩy sập một chiều* để tăng độ khó và cản trở hoạt động thám mã. Hệ mật mã khóa công khai dựa trên logarit rời rạc được dùng khá phổ biến và được gọi là hệ mật mã Elgamal. Để tiếp cận phương pháp mật mã khóa công khai, chúng ta sẽ tìm hiểu về hệ mật Elgamal và logarit rời rạc.

Bài toán logarit rời rạc trong trường hữu hạn Z_p là đối tượng của nhiều công trình nghiên cứu và được xem là bài toán khó nếu chọn p một cách cẩn thận. Nghĩa là không có một thuật toán khả thi nào cho bài toán logarit rời rạc. Elgamal đã phát triển một hệ mật mã khóa công khai dựa trên bài toán logarit rời rạc như sau:

$I = (p, \alpha, \beta)$ trong đó p là số nguyên tố, $\alpha \in Z_p^*$ (nhóm nhân

Z_p^* là nhóm tuần hoàn) là phần tử nguyên thủy, $\beta \in Z_p^*$.

Mục đích: Tìm một số nguyên duy nhất a , $0 \leq a \leq p-2$ sao cho:

$\alpha^a \equiv \beta \pmod{p}$, nghĩa là α^a cùng dư với β theo modulo p .

Sẽ xác định số nguyên a bằng $\log_{\alpha}\beta$.

Từ đó chọn số nguyên tố p sao cho bài toán logarit rời rạc ở trên là khó giải. Ta định nghĩa:

$$K = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}$$

Các giá trị p, α, β được công khai, còn a giữ bí mật.

Với $K = (p, \alpha, a, \beta)$ và một số ngẫu nhiên bí mật $k \in Z_{p-1}$, ta xác định:

$$e_K(x, k) = (y_1, y_2)$$

trong đó

$$y_1 = \alpha^k \pmod{p}$$

và

$$y_2 = x\beta^k \pmod{p}$$

với $y_1, y_2 \in Z_p^*$ ta xác định:

$$d_k(y_1, y_2) = y_2(y_1^{-1}) \pmod{p}$$

Để minh họa hoạt động của hệ mật Elgamal, sau đây sẽ trình bày một cách hình thức các bước. Trước hết bản gốc x sẽ được dấu bằng cách nhân với β^k để tạo ra y_2 . Giá trị α^k cũng được gửi đi như một phần của bản mã. Nơi thu hợp lệ biết được a sẽ suy ra được β^k từ α^k . Sau đó sẽ chia y_2 cho β^k để thu được bản gốc x .

Ví dụ : Cho $p = 2579$, $\alpha = 2$, $a = 765$. Khi đó $\beta = 2^{765} \text{ mod } 2579 = 949$.

Giả sử muốn gửi bản tin $x = 1299$. Chọn số ngẫu nhiên $k = 853$.

$$y_1 = 2^{853} \text{ mod } 2579 = 435$$

và $y_2 = 1299 \cdot 949^{853} \text{ mod } 2579 = 2396$.

Ở đâu thu khi nhận được bản mã $y = (435, 2396)$ sẽ tiến hành tính ra bản gốc
 $x = 2396 \cdot (435^{765})^{-1} \text{ mod } 2579 = 1299$.

CHƯƠNG 5

CƠ SỞ CỦA GIAO THỨC

TỔNG QUAN

Chương 3 đã mô tả các mạch điện và các kỹ thuật có thể dùng để truyền một frame thông tin giữa hai thiết bị đầu cuối qua một liên kết số liệu điểm-nối-điểm. Trong chương 4 đã trình bày các lược đồ phát hiện lỗi khác nhau, cho phép máy thu xác định được bất kỳ lỗi nào xảy ra trong luồng bit truyền. Nếu có thêm cơ cấu khắc phục lỗi thì máy thu hoàn toàn có thể suy ra thông tin chi tiết từ luồng bit nhận, ngay cả khi có lỗi truyền.

Tuy nhiên, nhìn chung các kỹ thuật đã được mô tả chỉ cung cấp cơ cấu căn bản để truyền thông tin và để máy thu phát hiện bất kỳ lỗi nào khi nó xuất hiện. Khi một lỗi đã bị phát hiện, ngay cả khi chỉ có một bit lỗi thì khối số liệu hoàn chỉnh cũng phải bị loại bỏ. Loại lược đồ như vậy được gọi là *truyền tổng lực* (*best-trty transmission*), cũng còn gọi là *truyền theo chế độ không tạo cầu nối* (*connectionless mode transmission*), bởi một số lý do mà chúng ta sẽ đề cập đến trong mục 4.

Bên cạnh chế độ hoạt động này còn có một chế độ hoạt động khác được xem như là *truyền tin cậy* (*reliable transmission*), cũng thường được gọi là *truyền theo chế độ có tạo cầu nối* (*connection-oriented transmission*) được dùng trong nhiều ứng dụng. Với chế độ hoạt động này ngoài phát hiện lỗi, còn định nghĩa một tập các luật hay thủ tục điều khiển mà cả hai chủ thể truyền thông đều phải tuân theo để đảm bảo tính tin cậy của hoạt động truyền (xác suất xảy ra lỗi thấp). Để thực hiện điều này, thiết bị điều khiển tại đích phải thông báo với máy khi phát hiện một lỗi truyền và yêu cầu máy nguồn gửi một bản copy khác của frame bị hỏng do lỗi này. Chu kỳ phối hợp phát hiện và sửa lỗi được gọi là kiểm soát lỗi (*error control*). Ngoài ra, cũng có các cơ cấu điều khiển khác phải được tuân thủ bởi các chủ thể truyền tin. Tập hợp các cơ cấu này tạo thành giao thức (liên kết số liệu) của liên kết. Trong chương này chúng ta sẽ xem xét một vài thành phần cơ bản của một giao thức liên kết số liệu.

KIỂM SOÁT LỖI

Nhập dữ liệu vào máy tính từ bàn phím, chúng ta ấn vào phím thì một từ mã được truyền vào trong máy tính theo từng bit nối tiếp thông qua một UART và truyền dẫn vào trong theo chế độ truyền bất đồng bộ. Chương trình trong máy tính sẽ điều khiển quá trình nhập, sau đó đọc, lưu ký tự này và khởi động quá trình xuất ra màn hình. Nếu những gì hiển thị lên màn hình khác với ký tự đã gõ, thì chúng ta chỉ cần gõ vào một ký tự điều khiển xóa thích hợp, ví dụ backspace. Khi nhận được ký tự điều khiển, chương trình điều khiển loại bỏ ký tự được nhập vào trước đó và xóa nó khỏi màn hình. Bằng cách này chúng ta đang thực hiện một dạng kiểm soát lỗi bằng tay.

Thủ tục tương tự cũng được dùng khi một đầu cuối kết nối đến một máy tính ở xa thông qua PSTN và một modem. Thay vì mỗi ký tự được hiển thị một cách trực tiếp trên màn hình của đầu cuối, trước tiên nó lại được truyền đến máy tính đầu xa. Sau đó ký tự được máy tính đầu xa đọc, lưu giữ và truyền trở lại đầu cuối để hiển thị lên màn hình. Nếu những gì hiển thị lên màn hình khác với ký tự đã gõ, người dùng có thể khởi động truyền một ký tự xóa thích hợp. Chế độ kiểm soát lỗi này được gọi là *kiểm tra dội lại (echo checking)*.

Ngược lại, khi máy tính đang truyền các khối ký tự qua một liên kết số liệu nối tiếp đến máy tính khác, chương trình trong máy tính đích chịu trách nhiệm điều khiển quá trình tiếp nhận phải thực hiện thủ tục kiểm soát lỗi một cách tự động mà không có sự can thiệp nào từ người dùng (user). Thông thường, máy tính thu kiểm tra để phát hiện các lỗi truyền có thể và sau đó gửi ngược lại nguồn tin một thông điệp điều khiển ngắn để báo nhận thành công hoặc yêu cầu nguồn tin gửi một bản copy khác của frame vừa đến (do lỗi). Thủ tục kiểm soát lỗi này được gọi là ARQ (Automatic Repeat Request).

Có hai loại ARQ cơ bản: **idle RQ**, là kiểm soát lỗi được dùng với các lược đồ truyền số liệu thiên hướng ký tự (character-oriented) và **continuous RQ**, được dùng trong chiến lược truyền lại có chọn lựa (*selective repeat retransmission*) hoặc truyền lại một nhóm (*go-back-N retransmission*). Continuous RQ được dùng chủ yếu trong các lược đồ truyền thiên hướng bit (bit-oriented). Mặc dù idle RQ đang bị continuous RQ thay thế trong nhiều ứng dụng để việc truyền đạt hiệu quả cao hơn, nhưng vẫn còn nhiều giao thức liên kết số liệu đang được dùng có cơ sở là idle RQ. Tuy vậy, điều quan trọng hơn vẫn là do idle RQ là lược đồ kiểm soát lỗi đơn giản nhất, nó tạo nên một phương tiện lý tưởng để lý giải nhiều vấn đề tổng quát hơn liên quan đến các giao thức liên kết số liệu.

2. Idle RQ

Lược đồ điều khiển idle RQ được định nghĩa để cho phép truyền các khối ký tự bao gồm ký tự điều khiển định dạng và ký tự in được qua một liên kết số liệu nối tiếp giữa DTE nguồn và DTE đích một cách tin cậy. Để phân biệt giữa các frame dữ liệu (thường đề cập đến như là I-frame) của máy phát và của máy thu thuật ngữ sơ cấp *primary* (*P*) và thứ cấp *secondary* (*S*) lần lượt được dùng. Do đó lược đồ điều khiển idle RQ liên hệ đến tính tin cậy của hoạt động truyền các I-frame giữa một sơ cấp và một thứ cấp qua một liên kết số liệu nối tiếp.

Giao thức idle RQ hoạt động theo chế độ bán song công vì sau khi sơ cấp gửi một I-frame, nó phải đợi cho đến khi nhận một thông báo của thứ cấp cho biết frame được nhận thành công hay không. Sau đó, sơ cấp gửi frame kế tiếp nếu frame trước đó được nhận tốt hoặc truyền lại một bản copy của nó nếu không được nhận thành công.

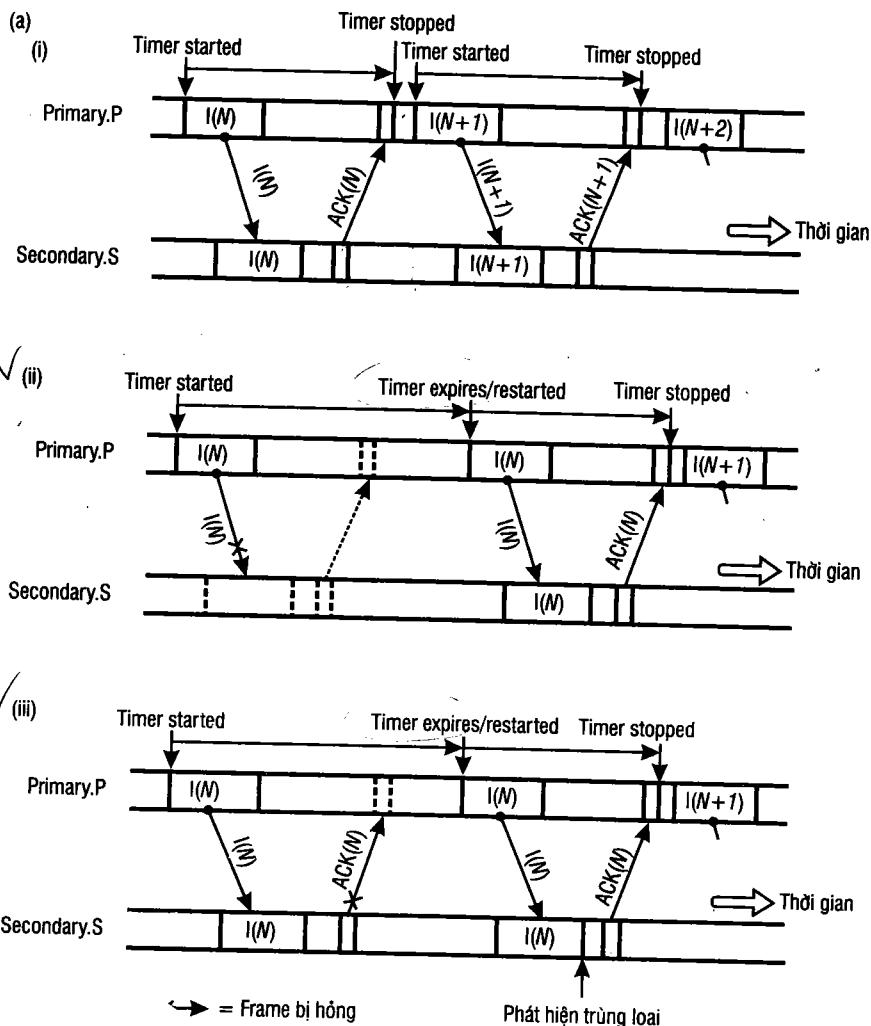
Có hai phương pháp để thực hiện lược đồ này. Trong phương pháp *truyền lại ngầm định* (implicit retransmission), thứ cấp chỉ báo nhận ACK (acknowledge) đối với những frame được nhận tốt và P hiểu rằng không có báo nhận có nghĩa là frame đã bị hỏng và tự động truyền lại. Còn đối với phương pháp thứ hai, khi thứ cấp phát hiện một frame bị hỏng, nó gửi lại một thông báo từ chối NAK (negative acknowledge) để yêu cầu truyền lại, do đó phương pháp thứ hai được gọi là *yêu cầu rõ* (explicit request).

Một số ví dụ về tuần tự frame trong lược đồ điều khiển implicit retransmission được trình bày trên hình 5.1(a). Sau đây là các điểm nên lưu ý khi giải thích các tuần tự frame:

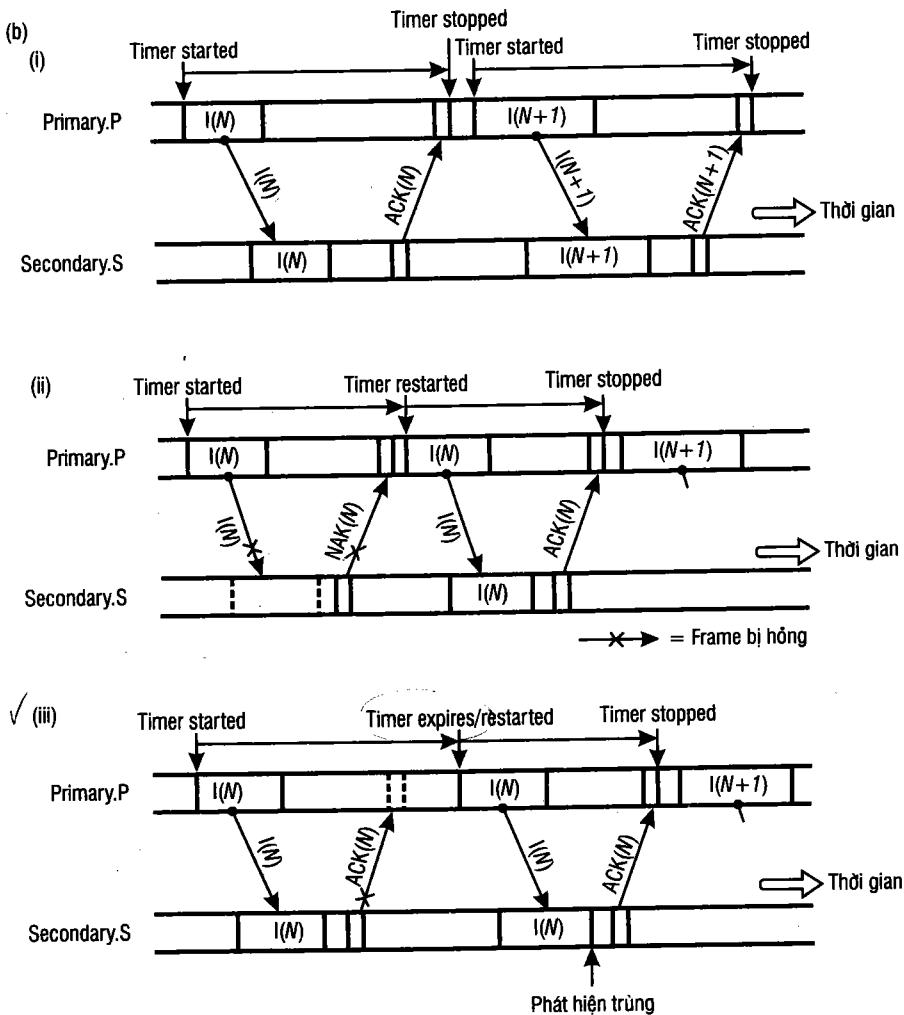
- P chỉ có thể có một I-frame đang chờ (chờ ACK) tại một thời điểm.
- Khi nhận một I-frame không bị lỗi, S phúc đáp một ACK-frame cho P.
- Khi một I-frame không lỗi đã được nhận, P có thể truyền một I-frame khác phần (i).
- Khi P khởi động truyền một I-frame, nó kích hoạt một bộ định thời (timer).



- Nếu S nhận một I-frame hay P nhận một ACK-frame có chứa lỗi thì frame sẽ bị loại bỏ.
- Nếu P không nhận được ACK-frame trong khoảng thời gian xác định trước thì P sẽ truyền lại I-frame _phần (ii).
- Nếu một ACK-frame bị hỏng, thì S lại nhận một bản copy khác của I-frame vừa nhận và S sẽ loại bỏ bản copy này _phần (iii).



Hoạt động của idle RQ: (a) Truyền lại ngầm định



Hình 5.1 (tiếp theo) Hoạt động của idle RQ: (b) Yêu cầu rõ.

Như chúng ta thấy trong phần (i), sau khi khởi động truyền một I-frame, P phải đợi trong một khoảng thời gian tối thiểu trước khi truyền I-frame kế tiếp. Thời gian đợi bằng với thời gian hành trình của I-frame đến đích và được S xử lý cộng với thời gian hành trình của ACK-frame đến P và được xử lý bởi P. Trong trường hợp xấu nhất, P phải đợi trong khoảng thời gian bằng với khoảng thời gian bất khả dụng (timeout), thời gian này phải vượt quá thời gian tối thiểu một lượng

thích hợp, nhằm tránh trường hợp nhận một ACK-frame ngay sau khi truyền lại một bản copy khác của I-frame trước đó.

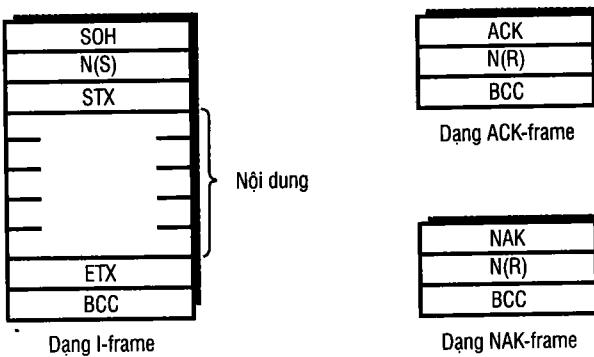
Mức độ quan hệ của mỗi thành phần làm cho khoảng thời gian tối thiểu thay đổi tùy vào các loại liên kết số liệu khác nhau. Nó được xác định bởi các yếu tố như khoảng cách vật lý giữa hai hệ thống truyền tin (P và S) và tốc độ truyền số liệu trong liên kết. Tuy nhiên, có thể cải thiện đáng kể việc sử dụng dung lượng có sẵn của liên kết nếu S thông báo cho P ngay tức thời bất cứ khi nào nhận được một I-frame bị hỏng bằng cách gửi một NAK-frame. Vài ví dụ về tuần tự frame theo lược đồ này được trình bày trên hình 5.1(b).

Sau đây là các điểm lưu ý khi giải thích các tuần tự:

- Giống như lược đồ báo nhận ngầm định (implicit acknowledgment), khi nhận một frame không lỗi, S phúc đáp một ACK-frame cho P.
- Khi nhận một ACK-frame không lỗi, P dừng bộ định thời và có thể khởi động truyền một frame khác_phần (i).
- Nếu S nhận một I-frame bị lỗi, frame này sẽ bị loại và S phúc đáp bằng một NAK-frame_phần (ii).
- Nếu P không nhận được ACK-frame (hay NAK-frame) trong khoảng thời gian của bộ định thời, khi hết hạn chờ P sẽ truyền lại I-frame_phần (iii).

Với idle RQ, sơ cấp phải đợi báo nhận sau khi gửi một frame, cho nên lược đồ này còn được gọi là truyền-và-chờ (send-and-wait) hay dừng-và-chờ (stop-and-wait). Như chúng ta thấy trong hình 5.1, lược đồ đảm bảo rằng S nhận ít nhất một bản copy của mỗi I-frame được P truyền. Tuy nhiên trong cả hai lược đồ, rất có thể S nhận được hai bản copy của một I-frame đặc biệt nào đó. Để S phân biệt giữa I-frame hợp lệ kế tiếp và I-frame trùng lặp, mỗi I-frame đều chứa một danh định duy nhất được gọi là chỉ số tuần tự frame ($N, N+1, \dots$) như trình bày trên hình 5.1. Do đó, S phải giữ lại một mục ghi số tuần tự chứa trong I-frame sau cùng đã được nhận tốt. Nếu S nhận một bản copy khác của frame này, thì bản copy này phải bị loại bỏ. Để P đồng bộ lại, S gửi một ACK-frame cho mỗi I-frame nhận tốt cùng với danh định của I-frame này.

Chúng ta có thể nhận thấy sự cải tiến trong việc sử dụng liên kết ở lược đồ yêu cầu rõ so với lược đồ truyền lại ngầm định bằng cách xem xét tuần tự frame (phần ii) của mỗi lược đồ. Với truyền lại ngầm định, thời gian trước khi truyền một I-frame kế tiếp chính là thời gian timeout, trong khi thời gian này có thể bớt đi rất nhiều nếu dùng một NAK-frame. Sự cải thiện về hiệu suất sử dụng liên kết này sau đó sẽ còn tùy vào tốc độ lỗi bit BER (Bit Error Rate) của liên kết và do đó tùy vào số lượng frame bị hỏng cần truyền lại. Tuy nhiên, trong hầu hết các ứng dụng truyền số liệu phù hợp với giao thức idle RQ thì lược đồ explicit request (yêu cầu rõ) với các NAK-frame thường được dùng.



Hình 5.2 Dạng frame của idle RQ (PDU).

Chỉ số tuần tự được mang trong mỗi I-frame được gọi là *chỉ số tuần tự gửi* hay N(S), và chỉ số tuần tự trong mỗi ACK-frame hay NAK-frame được gọi là *chỉ số tuần tự nhận* hay N(R). Như trong chương trước đã mô tả, các tập ký tự ASCII (hay EBCDIC) chứa một số các ký tự điều khiển, một số trong đó đã được dùng để điều khiển truyền. Có các ký tự điều khiển truyền khác được yêu cầu để thực hiện thủ tục điều khiển idle RQ cơ bản: SOH (start-of-header), NAK, và ACK. Việc sử dụng chúng được trình bày trên hình 5.2.

Mỗi I-frame phải chứa một chỉ số tuần tự tại đầu của frame, và như chúng ta có thể thấy, chỉ số này đứng trước ký tự STX. Ký tự SOH được chèn vào đầu của một khối hoàn chỉnh để báo hiệu bắt đầu của một frame mới.

Các ký tự điều khiển ACK và NAK được dùng cho mục đích báo nhận, theo sau là chỉ số tuần tự thu. Để tăng cường khả năng phát hiện lỗi, một ACK-frame hay NAK-frame hoàn chỉnh chứa một ký tự kiểm tra tổng khối. Cả 3 loại frame_I-frame, ACK-frame, NAK-frame đều được gọi là đơn vị dữ liệu giao thức PDU (Protocol Data Unit) của giao thức idle RQ, P và S lần lượt là thực thể giao thức sơ cấp và thực thể giao thức thứ cấp.

2.1. Kiến trúc phân lớp

Lực đồ tuần tự frame được trình bày trên hình 5.1 mô tả các đặc trưng chủ yếu của hoạt động giao thức idle RQ. Tuy nhiên, trước khi đi vào các chi tiết liên quan đến giao thức, chúng ta sẽ xem xét cụ thể hơn khái niệm phân lớp. Khái niệm này có liên quan đến sự tách riêng một ứng dụng phối hợp và các tác vụ truyền tin thành hai tác vụ con rõ ràng hơn, tác vụ con được gọi là lớp, cùng với một hình thức giao tiếp giữa chúng.

Để mô tả khái niệm, hãy xem xét một quá trình ứng dụng AP (Application Process) chạy trên một máy tính truyền một tập tin đến một quá trình ứng dụng tương tự chạy trên một máy tính thứ hai qua một liên kết số liệu nối tiếp dùng một giao thức điều khiển lỗi idle RQ. Như đã mô tả, giao thức idle RQ sẽ cố gắng gởi các khối thông tin nối tiếp qua một liên kết số liệu theo một phương thức tin

cậy. Cũng tùy vào BER trên liên kết, một kích thước khối tối đa sẽ được chỉ định để đảm bảo một tỉ lệ cao của các I-frame không bị lỗi.

Lớp giao thức idle RQ trong máy tính nguồn vì thế phải cung cấp một dịch vụ xác định cho lớp AP của người dùng bên trên nó, dịch vụ này là truyền một chuỗi nối tiếp các khối thông tin, mỗi khối có một chiều dài tối đa xác định, đến quá trình ứng dụng tương tự (ngang hàng) trên máy tính đích. Hai thực thể giao thức ngang hàng idle RQ có liên quan đến các vấn đề phát hiện lỗi được thảo luận trước đây: tạo ra và phản hồi các frame báo nhận, timeout, phân phối các khối thông tin theo tuần tự giống như khi chúng được truyền.

Ngược lại, hai quá trình ứng dụng ngang hàng chỉ liên quan đến thủ tục truyền tập tin dùng dịch vụ được cung cấp bởi lớp truyền tin (idle RQ), như tên tập tin, chiều dài của nó, sự phân đoạn nội dung tập tin thành các khối nhỏ hơn trước khi truyền đến lớp truyền tin, và tái thiết chúng thành một tập tin hoàn chỉnh khi nhận. Do đó, trong mỗi ứng dụng, tuần tự của các khối thông điệp là rất quan trọng, cú pháp và cấu trúc của thông điệp phải được xác định. Điều này ngụ ý rằng một giao thức AP-to-AP phải có tập PDU riêng của nó. Hai thực thể giao thức AP dùng các dịch vụ của lớp truyền tin bên dưới để truyền các PDU của chúng. Dĩ nhiên, đối với lớp truyền tin, tất cả các PDU này chỉ đơn giản là các khối thông tin sẽ được truyền theo cùng một cách.

Thông thường, dịch vụ được cung cấp bởi một lớp truyền tin được biểu diễn dưới dạng một hàm thực thể dịch vụ (service primitive) với dữ liệu truyền đóng vai trò là tham số. Vì dịch vụ liên hệ với lớp liên kết (L) và thực hiện truyền số liệu (các khối thông tin) nên hàm dịch vụ của người dùng (user) tại giao tiếp truyền được ký hiệu là L_DATA.request và hàm dịch vụ tại giao tiếp nhận được ký hiệu là L_DATA.indication. Điều này được trình bày trên hình 5.3(a).

Trong nhiều trường hợp, vì các user của một lớp liên quan đến dịch vụ được cung cấp chứ không liên quan đến cách thực hiện dịch vụ, nên khi định nghĩa các dịch vụ liên quan đến một lớp giao thức, chúng ta thường trình bày dưới dạng tương tự hình 5.3(b). Dạng trình bày này được gọi là *lược đồ tuần tự theo thời gian*. Để tách biệt rõ ràng giữa hai lớp, dùng một hàng đợi giữa chúng, như trình bày trên hình 5.3(c). Đây là dạng cấu trúc dữ liệu đơn giản thực hiện theo nguyên lý hàng đợi first-in-first-out (FIFO). Các phần tử được đưa vào đuôi của hàng đợi và được lấy ra từ đầu hàng đợi.

Thông thường, các hàm dịch vụ liên hệ đến một lớp được chuyển qua giữa các lớp bằng một cấu trúc dữ liệu gọi là *khối điều khiển sự kiện_ECB* (Event Control Block). Tổng quát, nó là một record hay cấu trúc chứa kiểu hàm tại filed đầu tiên và một chuỗi ký tự hay byte chứa dữ liệu của user trong field thứ hai. Bất cứ khi nào nguồn AP hay lớp giao thức cao hơn muốn truyền một khối thông điệp, trước hết nó lấy một ECB tự do, ghi đoạn dữ liệu của user vào chuỗi ký tự hay byte, thiết lập filed chỉ loại hàm thành L_DATA.request và chèn ECB vào đuôi của hàng đợi nhập của lớp liên kết (LS_user) để chuẩn bị sẵn cho thực thể giao thức sơ cấp idle RQ đọc.

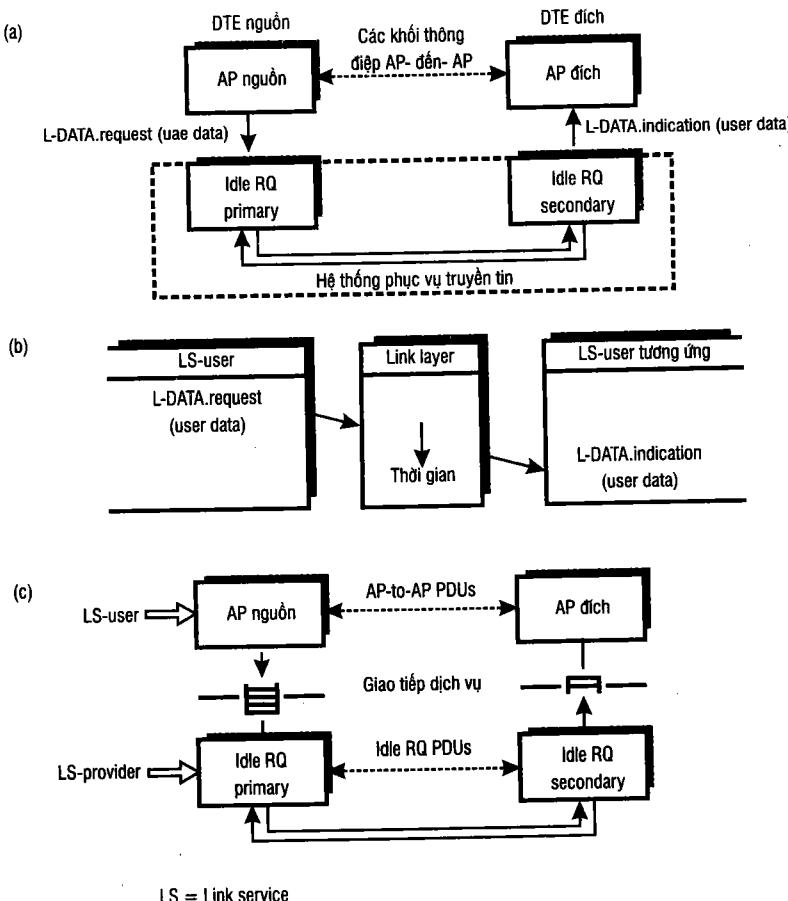
Khi phần mềm thực thể giao thức idle RQ kế tiếp thực thi, nó phát hiện ra sự hiện diện của một ECB tại hàng đợi nhập của lớp liên kết, nó đọc ECB từ đầu của hàng đợi, và xử lý để tạo ra một I-frame hoàn chỉnh_gồm nội dung thông điệp, header và trailer thích hợp. Sau đó nó khởi động truyền frame đến thực thể giao thức thứ cấp. Giả sử frame nhận được không bị lỗi, thực thể thứ cấp sẽ loại bỏ header và trailer, và chuyển nội dung frame đến AP đích theo một ECB bằng cách



dùng hàng đợi ngõ ra của lớp liên kết (LS_provider) kèm theo hàm dịch vụ được đổi thành L_DATA.indication. Sau đó, tạo và phản hồi một ACK-frame đến P.

Khi đến lượt AP đích chạy, nó phát hiện và đọc ECB từ hàng đợi LS_provider và xử lý nội dung của khối thông tin phụ thuộc vào giao thức AP-to-AP đã được định nghĩa. Thông thường, nếu đây là khối thông diệp đầu tiên chứa tên tập tin, nó sẽ liên quan đến việc tạo ra một tập tin (file) có tên theo tên này và mở sẵn cho các hoạt động ghi tập tin ngay sau đó.

Ở phía truyền, giả sử nhận được ACK-frame không lỗi, P xóa bộ đệm giữ I-frame vừa được báo nhận và kiểm tra hàng đợi nhập LS_user để tiếp nhận một ECB đang đợi khác. Nếu có một ECB khác, thủ tục sẽ được lặp lại cho đến khi tất cả các đoạn của tập tin đã được truyền. Thông thường AP nguồn gửi một khối thông diệp kết thúc truyền (end-of-transfer) để thông báo cho AP đích biết toàn bộ nội dung của tập tin đã được truyền.



Hình 5.3

Kiến trúc phân lớp: (a) Các hàm dịch vụ (b) Lược đồ tuần tự theo thời gian (c) Các giao tiếp dịch vụ.



Có thể kết luận rằng sự tuân theo kiến trúc phân lớp có nghĩa là mỗi lớp thực hiện một chức năng quan trọng nào đó trong ngữ cảnh của tác vụ truyền tin toàn cục (ví dụ truyền tập tin). Lớp bên dưới cung cấp một dịch vụ xác định cho lớp phía trên của nó và hoạt động tùy theo giao thức đã được định nghĩa. Một kiến trúc phân lớp có nghĩa là các hệ thống thông tin con phức tạp có thể được thực hiện một cách đơn giản bằng cách thêm các lớp vào giữa lớp ứng dụng và lớp liên kết bên dưới, mỗi lớp mới liên quan đến một chức năng hoàn chỉnh.

2.2. Đặc tả giao thức

Mặc dù 3 lược đồ tuần tự theo thời gian được trình bày theo mỗi phần của hình 5.1, kèm theo một đoạn miêu tả, có lẽ đủ để mô tả hoạt động của một giao thức idle RQ, nhưng với các giao thức phức tạp hơn thì việc mô tả đầy đủ hoạt động của giao thức theo phương pháp này là không thể. Thật vậy, định nghĩa hoạt động của một giao thức trong khi cho phép tất cả các sự kiện và các trường hợp kiểm soát lỗi có thể làm cho vấn đề trở nên vô cùng phức tạp. Vì vậy, sẽ mô tả giao thức bằng cách dùng một trong số các phương pháp rõ ràng và hình thức hơn. Chúng ta dùng sơ đồ tuần tự frame đơn giản như dạng trình bày trên hình 5.1 để mô tả các khía cạnh nào đó của giao thức thay vì xem nó là phương tiện để đặc tả giao thức.

Ba phương pháp phổ biến nhất để đặc tả một giao thức truyền tin là dùng các sơ đồ chuyển trạng thái (state transition diagram), các bảng sự kiện-trạng thái mở rộng (extended event-state table), và chương trình có cấu trúc mức cao (high-level structured program). Trong nhiều trường hợp, chúng ta định nghĩa một giao thức như là một tổ hợp của các sơ đồ này đi đôi với lược đồ tuần tự thời gian để mô tả các hàm dịch vụ user liên hệ với giao thức.

Bất chấp phương pháp đặc tả nào được dùng, chúng ta đều mô phỏng một giao thức như một thiết bị máy móc tự động. Điều này có nghĩa là giao thức, hay chính xác hơn là thực thể giao thức, chỉ có thể ở một trong một số trạng thái xác định tại bất kỳ thời điểm nào. Ví dụ, nó có thể ở trạng thái nhàn rỗi đợi truyền một thông điệp, hay đợi nhận một báo nhận. Sự chuyển trạng thái xảy ra khi có kết quả của một sự kiện đến, ví dụ khi nhận một thông điệp, một sự kiện đi tương ứng được tạo ra, ví dụ khi nhận một thông điệp, gửi tiếp một I-frame lên liên kết, hay khi nhận một NAK-frame thì truyền lại I-frame.

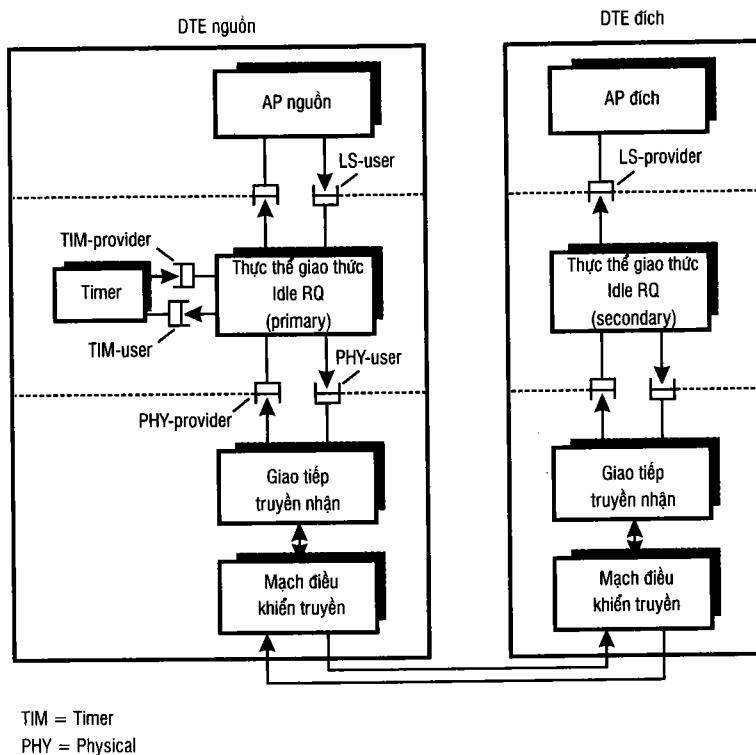
Có thể thấy trong một số sự kiện đến có thể tạo ra một số các sự kiện đi có thể. Sự kiện đi đặc biệt được chọn là sự kiện được xác định bởi trạng thái được tính toán trên cơ sở một hay nhiều thuộc tính (các biến luận lý). Ví dụ thuộc tính P1 có thể là *true* nếu N(R) trong ACK-frame trùng với N(S) trong I-frame đang đợi báo nhận. Do đó, nếu P1 là *true* thì có thể làm rỗng bộ đệm đang lưu giữ I-frame này; nếu là *false* thì khởi động truyền lại frame.

Sau cùng, một sự kiện đến, ngoài việc tạo ra một sự kiện đi (và có thể thay đổi trạng thái), cũng có thể có một hay nhiều hành động cục bộ đặc trưng liên quan được tiến hành. Ví dụ khởi động một bộ định thời và gia tăng biến tuần tự gởi.

Chúng ta sẽ mở rộng tất cả các khía cạnh đặc tả một giao thức bằng cách xem xét đặc tả thủ tục kiểm soát lỗi liên quan đến giao thức idle RQ.

2.3. Đặc tả idle RQ

Tất cả các máy có trạng thái xác định hay các thực thể giao thức đều hoạt động theo một phương pháp vi mô. Điều này có nghĩa là một khi có sự kiện đến là bắt đầu quá trình xử lý, tất cả các chức năng xử lý liên quan đến sự kiện này, bao gồm tạo ra bất cứ sự kiện đi nào, bất kỳ một hành động cục bộ nào cũng như sự thay đổi trạng thái là những gì được thực hiện theo phương thức riêng của chúng trước khi sự kiện đến được chấp nhận.



Hình 5.4 Kiến trúc hệ thống truyền tin và các giao tiếp thực thể giao thức.

Để chắc chắn điều này xảy ra, các giao tiếp sự kiện đến khác nhau được tách ra khỏi thực thể giao thức nhờ các hàng đợi, như trình bày trên hình 5.4. Có một cặp hàng đợi thêm vào giữa thực thể giao thức và thủ tục truyền/nhận điều khiển mạch truyền/nhận đặc biệt đang được dùng. Tương tự, cũng có một cặp hàng đợi giữa thực thể giao thức và thủ tục định thời. Thông thường thủ tục định thời chạy

theo khoảng thời gian đều đặn bởi tác động của *ngắt* (interrupt), và nếu bộ định thời đang chạy thì giá trị hiện hành của nó bị giảm từng khoảng một. Nếu giá trị đến 0, bộ định thời hết hạn, một thông điệp được gửi cho thực thể giao thức qua hàng đợi thích hợp.

Vai trò của thủ tục truyền/nhận chỉ đơn giản là truyền một frame được định dạng trước đi qua nó và nhận một frame từ liên kết và xếp vào hàng đợi chờ thực thể giao thức xử lý. Thủ tục này cũng có thể được kích hoạt bởi một ngắt, nhưng lần này là từ mạch điều khiển truyền. Mặc dù, về mặt nguyên lý chỉ cần một hàng đợi nhập và xuất để giao tiếp sơ cấp và thứ cấp với các quá trình ứng dụng tương ứng của nó, trong thực tế cần một cặp hàng đợi tại mỗi giao tiếp để kiểm soát dòng các hàm dịch vụ trong chế độ song công.

Để đơn giản thủ tục đặc tả, chúng ta cung cấp cho mỗi sự kiện đến, sự kiện đi, yếu tố xác lập (thuộc tính luận lý), hành động đặc biệt và trạng thái liên hệ với mỗi thực thể giao thức một tên ngắn gọn (viết tắt). Trước khi đặc tả một giao thức, các tên thường được liệt kê; tất cả các tham khảo tiếp theo đây đều dùng các tên này. Trong thành phần kiểm soát lỗi của giao thức idle RQ, danh sách các tên ghi tắt của sơ cấp và thứ cấp lần lượt được trình bày trên hình 5.5(a) và 5.5(b).

Vì mỗi thực thể giao thức là một hệ thống tuần tự liên tục, nên phải giữ lại thông tin nào có thể bị thay đổi khi có một sự kiện khác đến. Thông tin được giữ trong một số *biến trạng thái*. Ví dụ đối với sơ cấp thì biến tuần tự gửi là V(S)_Vs trong đặc tả_ nó giữ chỉ số tuần tự được gán cho I-frame được truyền kế tiếp. Biến PresentState giữ trạng thái hiện hành của thực thể giao thức. RetxCount được dùng để giới hạn số lần truyền lại frame, và ErrorCount chứa số lượng frame lỗi đã nhận. Thông thường, nếu RetxCount hoặc ErrorCount đạt đến giới hạn tối đa thì frame sẽ bị loại bỏ, và một thông báo lỗi được gửi lên cho lớp AP ở trên và thực thể giao thức được khởi động lại.

(a) Các sự kiện đến

Tên	Giao tiếp	Ý nghĩa
LDATAreq	LS_user	Hàm dịch vụ L_DATA.request nhận
ACKRCVD	PHY_provider	Nhận ACK-frame từ S
TEXP	TIM_provider	Đợi bộ định thời ACK hết hạn
NAKRCVD	PHY_provider	Nhận NAK-frame từ S

Các trạng thái

Tên	Ý nghĩa
IDLE	Nhàn rỗi, không có thông điệp truyền
WTACK	Đợi một báo nhận

Hình 5.5 Các tên viết tắt được dùng trong đặc tả idle RQ:

**Các sự kiện di**

Tên	Giao tiếp	Ý nghĩa
<i>TxFrame</i>	<i>PHY_user</i>	<i>Định dạng và truyền một I-frame</i>
<i>RetxFrame</i>	<i>PHY_user</i>	<i>Truyền lại một I-frame đang đợi báo nhận</i>
<i>LERROnD</i>	<i>LS_provider</i>	<i>Thông điệp bị lỗi: frame bị loại vì lý do được chỉ định</i>

Các thuộc tính xác lập

Tên	Ý nghĩa
<i>P0</i>	<i>N(S) trong I-frame đang đợi =N(R) trong ACK-frame</i>
<i>P1</i>	<i>Kiểm tra tổng khối (BSC) trong ACK/NAK-frame là đúng</i>

Các hành động đặc biệt

Các hành động đặc biệt		Các biến trạng thái
[1]= <i>Start_timer</i> dùng hàng đợi <i>TIM_user</i>		<i>Vs</i> = biến tuần tự gởi
[2]= <i>Gia tăng Vs</i>		<i>PresentState</i> = trạng thái hiện hành của thực thể giao thức
[3]= <i>Stop_timer</i> dùng hàng đợi <i>TIM_user</i>		<i>ErrorCount</i> = số frame lỗi đã nhận
[4]= <i>Gia tăng RetxCount</i>		
[5]= <i>Gia tăng ErrorCount</i>		<i>RetxCount</i> = số lần truyền lại frame này
[6]= <i>Gán lại RetxCount bằng 0</i>		

(b) Các sự kiện đến

Tên	Giao tiếp	Ý nghĩa
<i>IRCVD</i>	<i>PHY_provider</i>	<i>Nhận I-frame từ P</i>

Các trạng thái

Tên	Ý nghĩa
<i>WTIFM</i>	<i>Đợi một I-frame mới từ P</i>

Các sự kiện di

Tên	Giao tiếp	Ý nghĩa
<i>LDATAind</i>	<i>LS_provider</i>	<i>Chuyển nội dung của I-frame nhận được đến AP với một hàm dịch vụ L_DATA.indication</i>

Hình 5.5 (tiếp theo) Các tên viết tắt được dùng trong đặc tả idle RQ:



<i>TxACK(X)</i>	<i>PHY_user</i>	<i>Định dạng và truyền một ACK-frame với N(R)=X</i>
<i>TxNAK(X)</i>	<i>PHY_user</i>	<i>Định dạng và truyền một NAK-frame với N(R)=X</i>
<i>LERRORind</i>	<i>LS_provider</i>	<i>Phát ra một thông điệp lỗi vì lý do được chỉ định</i>

Các thuộc tính xác lập

<i>Tên</i>	<i>Ý nghĩa</i>
<i>P0</i>	<i>N(S) trong I-frame = Vr</i>
<i>P1</i>	<i>Kiểm tra tổng khối (BSC) trong I-frame là đúng</i>
<i>P2</i>	<i>N(S) trong I-frame = Vr - 1</i>
Các hành động đặc biệt	Các biến trạng thái
<i>[1] = gia tăng Vr</i>	<i>Vr = biến tuần tự nhận</i>
<i>[2] = gia tăng ErrorCount</i>	<i>ErrorCount = số frame lỗi đã nhận</i>

Hình 5.5 (tiếp theo) Các tên viết tắt được dùng trong đặc tả idle RQ:
(a) Cho sơ cấp P (b) Cho thứ cấp S.

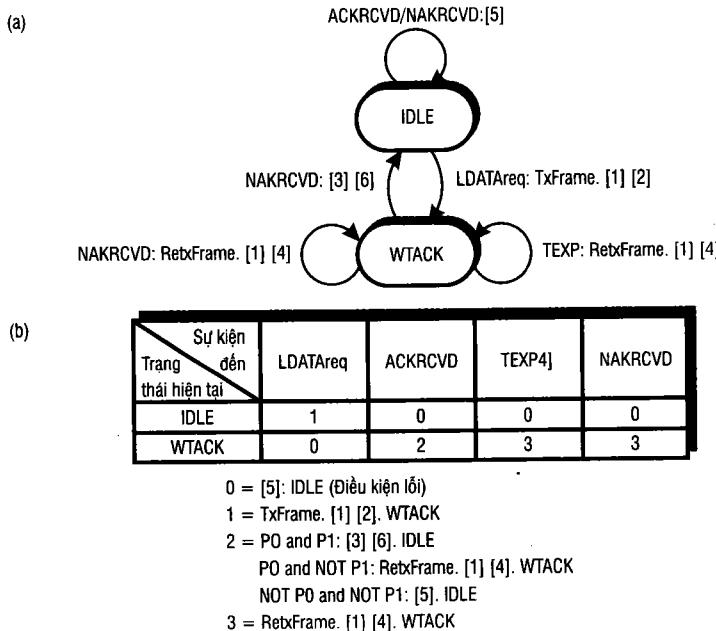
Chỉ có hai biến trạng thái là cần cho thứ cấp: biến tuần tự thu V(R) _Vr trong đặc tả _nó giữ số tuần tự của I-frame nhận tốt sau cùng, và ErrorCount giữ số lượng frame lỗi đã nhận. Một lần nữa, nếu ErrorCount đạt đến một giới hạn tối đa xác định thì một thông báo lỗi sẽ được gửi đến lớp AP ở trên.

Đặc tả chính xác của cả hai thực thể P và S lần lượt được trình bày trên hình 5.6 và 5.7 dưới dạng sơ đồ chuyển trạng thái và dạng bảng sự kiện_trạng thái.

Sử dụng phương pháp sơ đồ chuyển trạng thái, các trạng thái có thể của thực thể giao thức được trình bày trong các hình oval, tên trạng thái được viết bên trong đó. Các mũi tên có hướng chỉ ra chiều dịch chuyển có thể giữa các trạng thái, sự kiện đến gây ra sự dịch chuyển và bất kỳ sự kiện đi và hành động đặc biệt nào đều được viết trên biên ngoài. Ví dụ nếu nhận được một L_DATA.request (LDATAReq) từ giao tiếp LS_user thì frame được định dạng và đưa đến giao tiếp PHY_user (TxFrame), một timer được khởi động cho frame này [1], biến tuần tự gửi tăng lên [2], và vào trạng thái WTACK. Tương tự, nếu nhận một ACK-frame với một N(R) bằng với N(S) trong frame đang đợi và kiểm tra BSC là đúng thì timer bị dừng [3] và bộ đếm truyền lại được trả về zero [6], và vào trạng thái IDLE. Sự chuyển trạng thái khác có thể được diễn giải theo cách thức tương tự.

Mặc dù lược đồ chuyển trạng thái là hữu dụng để trình bày hoạt động của một giao thức, nhưng sự giới hạn về không gian luôn là trở ngại thực tế trong việc trình bày tất cả các khả năng của sự kiện đến bao gồm cả các điều kiện lỗi. Do đó

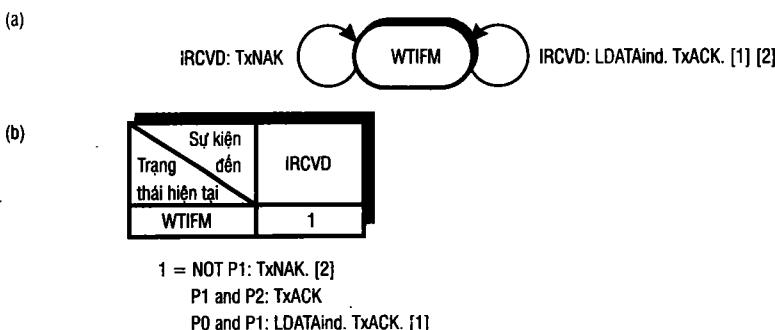
hầu hết các sơ đồ chuyển trạng thái là các đặc tả không hoàn chỉnh. Hơn thế nữa, ngoại trừ các giao thức đơn giản nhất còn hầu hết chúng ta đều cần dùng nhiều lược đồ để định nghĩa, ngay cả một hoạt động đúng của giao thức. Vì các lý do này mà chúng ta sẽ dùng bảng sự kiện-trạng thái và phương pháp mã lệnh (chương trình) có cấu trúc.



Hình 5.6 Đặc tả sơ cấp của idle RQ:

(a) Sơ đồ chuyển trạng thái (b) Bảng sự kiện-trạng thái mở rộng.

Dùng phương pháp bảng sự kiện-trạng thái mở rộng như ở hình 5.6 và 5.7 chúng ta có thể trình bày tất cả các sự kiện đến có thể và các trạng thái của giao thức dưới dạng một bảng. Đối với mỗi trạng thái, mục chỉ định trong bảng (table entry) xác định một sự kiện đi, bất kỳ hành động đặc biệt nào, và trạng thái mới tương ứng với tất cả các sự kiện đến có thể, cũng như các yếu tố xác lập liên quan và tập các hành động tùy chọn. Rõ ràng, bảng sự kiện-trạng thái mở rộng là một phương pháp trình bày tốt hơn vì nó cho phép trình bày tất cả các tổ hợp trạng thái hiện hành và sự kiện đến có thể. Một bảng sự kiện-trạng thái cơ bản chỉ có một hoạt động có thể và một trạng thái kế cho mỗi tổ hợp sự kiện đến và trạng thái hiện hành. Sự xuất hiện các yếu tố xác lập và do đó là các trạng thái kế tiếp hay hành động tùy theo có thể kéo theo việc sử dụng thuật ngữ "mở rộng" trong tên của phương pháp này (bảng sự kiện-trạng thái mở rộng).



Hình 5.7 Đặc tả thứ cấp của idle RQ:

(a) Sơ đồ chuyển trạng thái (b) Bảng sự kiện-trạng thái mở rộng.

Khi chúng ta giải thích các hành động theo sau các yếu tố xác lập (thuộc tính luận lý), cần lưu ý rằng các hành động này được trình bày theo thứ tự. Do đó, một hành động theo sau khi sơ cấp ở trong trạng thái WTACK và nhận một ACK-frame là trước hết xác định P0 và P1 đều là *true*. Nếu cả hai đúng như thế, thì thực hiện hành động [3] và [6] và vào trạng thái IDLE. Nếu ngược lại thì xác định xem {P0 & NOTP1} có là *true* hay không và cứ thế. Nếu không có điều kiện nào là *true* thì nghi ngờ có một lỗi và các hành động được trình bày như hình.

Một đặc trưng của bảng sự kiện-trạng thái mở rộng là làm cho công việc hiện thực theo mã chương trình (program code) dễ dàng hơn so với một sơ đồ chuyển trạng thái. Có thể thấy rõ điều này bằng cách xem xét một ví dụ: hiện thực sơ cấp idle RQ và thứ cấp idle RQ. Cấu trúc của chương trình được trình bày lần lượt trên hình 5.8 (a) và (b). Chúng được viết theo mã giả mức cao (high-level pseudocode) để dễ hiểu và được trình bày như là một chương trình riêng biệt. Trong thực tế, chúng có thể là những thủ tục nhưng điều này không ảnh hưởng đến hoạt động cơ bản của chúng.

Khi mỗi chương trình bắt đầu chạy, thủ tục khởi động (Initialize) được yêu cầu. Thủ tục này thực hiện các chức năng như là khởi động tất cả các biến trạng thái bằng cách gán các giá trị ban đầu cho chúng và nội dung của mảng EventStateTable (EventStateTable array) theo các giá trị trong bảng sự kiện-trạng thái mở rộng. Sau đó chương trình đi vào vòng lặp vô hạn đợi một sự kiện đến tại một trong các hàng đợi nhập của nó.

Sự kiện đến trước hết sẽ làm cho chương trình được gán loại sự kiện vào biến EventType. Sau đó nội dung hiện hành của biến PresentState và EventType được dùng như là chỉ số trỏ đến mảng EventStateTable (EventStateTable array) để xác định số nguyên 0, 1, 2 hay 3, những số nguyên này chỉ định các hành động xử lý liên quan đến sự kiện này. Ví dụ, nếu truy xuất được số nguyên là 2, điều này dẫn đến nhu cầu tham khảo các hàm (functions) xác lập P0 và P1, và tùy vào trạng

thái được tính toán của chúng (*true* hay *false*), việc gọi thủ tục thực hiện sự kiện đi thích hợp, cùng với bất kỳ thủ tục thực hiện hành động đặc biệt nào được định nghĩa trong đặc tả giao thức, ví dụ như khởi động hay reset bộ định thời, và cập nhật biến PresentState.

Ở đây chúng ta dùng mã giả đơn giản vì mục đích làm sáng tỏ cấu trúc của chương trình và từ đó làm sáng tỏ phương pháp luận của sự hiện thực. Không có mã nào trình bày cho các thủ tục thực hiện sự kiện đi khác nhau, cũng không có mã nào trình bày cho các hàm xác lập (predicate functions). Trong thực tế, chúng được hiện thực theo một cách thức rõ ràng dùng các bước cần thiết đã được chỉ định trong đặc tả.

Mặc dù nhiều giao thức được đặc tả dưới dạng một bảng sự kiện-trạng thái mở rộng, nhưng cũng có một số giao thức được đặc tả dưới dạng một đoạn mã chương trình có cấu trúc. Loại đặc tả này rất giống với các hiện thực bằng mã giả được trình bày trên hình 5.8. Tuy nhiên, mã chương trình có cấu trúc được dùng ở đây chuẩn tắc hơn vì tất cả các biến yêu cầu, các sự kiện đi, các thủ tục thực thi hành động đặc biệt và các hàm xác lập đều được trình bày chi tiết. Do đó sự hiện thực một đặc tả giao thức được mô tả theo mã cấu trúc thì đơn giản hơn nhiều và ít có khả năng xảy ra lỗi.

```

(a) program           IdleRQ_Primary;
const                 MaxErrCount;
                     MaxRetxCount;
type                 Events = (LDATAreq,ACKRCVD,TEXP,NAKRCVD);
                     States = (IDLE,WTACK);
var                  EventStateTable = array[Events,States] of 0..3;
                     PresenState: States;
                     Vs,ErrorCount,RetxCount: integer;
                     EventType: Events;
procedure             Initialize; {khởi động các biến trạng thái và nội dung của
EventStateTable}
                     procedure TxFrame;
                     procedure RetxFrame;
                     procedure LERRORind;
                     procedure Start_timer;
                     procedure Stop_timer;
function P0:boolean;

```

Hình 5.8 Đặc tả Idle RQ

```
function P1:boolean;
begin
    Initialize;
repeat  Đợi nhận một sự kiện đến
    EventType:= loại sự kiện;
    Case EventStateTable[EventType,PresentState] of
        0: begin ErrorCount := ErrorCount +1; PresentState:=IDLE;
            if (ErrorCount = MaxErrCount) then ERRORind end;
        1: begin TxFrame; Start_timer; Vs:=Vs +1; PresenState:=WTACK
    end;
        2: begin if(P0 and P1) then begin Stop_timer;RetxCount:=0;
    PresentState:=IDLE end
            else if(P0 and NOTP1) then begin
                RetxFrame; Satrt_timer;
                RetxCount:=RetxCount + 1;
                PresentState:=WTACK end
            else if(NOTP0 and NOTP1) then begin
                PresentState:=IDLE; ErrorCount:=ErrorCount + 1;
                end;
            If (ErrorCount =MaxErrorCount) then begin LERRORind;Initialize; end;
            end;
        3:begin
            RetxFrame;
            Start_timer;
            RetxCount:=RetxCount + 1;
            PresentState:=WTACK;
            if (RetxCount = MaxRetxCount) then begin LERRORind; Initialize;end; end;
            untill Mái mái;
    end.
```

(b) program

IdleRQ_Secondary;

Hình 5.8 (tiếp theo) Đặc tả Idle RQ



```

const MaxErrorCount;
type Events=IRCVD;
States=WTIFM;
var EventStateTable = array[Events,States] of 1;
PresentState: States;
Vr,X,ErrorCount: integer;
EventType: Events;
procedure Initialize;
procedure LDATAind(X);
procedure TxACK(X);
procedure TxNAK(X);
procedure LERRORind;
function P0:boolean;
function P1:boolean;
function P2:boolean;
begin Initialize;
repeat Đợi nhận một sự kiện đến;
    EventType:= loại sự kiện;
    Case EventStateTable[EventType,PresentState] of
        1: X:=N(S) từ I-frame;
            if (NOTP1) then TxNAK(X)
            else if(P1 and P2) then TxACK(X)
        else if(P0 and P1) then begin LDATAind; TcACK(X); Vr:=Vr+1; end
        else begin ErrorCount:=ErrorCount +1; if(ErrorCount = MaxErrorCount) then
            begin LERRORind; Initialize; end;
        end;
        until Mãi mãi;
    end.

```

(c)-Định nghĩa hằng số và loại

Hình 5.8 (tiếp theo) Đặc tả Idle RQ

- Các định nghĩa giao tiếp (các tên của hàng đợi giao tiếp và các định nghĩa cấu trúc record ECB)

- Định nghĩa các trạng thái máy xác định.

Module

- Các trạng thái
- Các sự kiện đến
- Các thủ tục ứng với sự kiện đi
- Các thủ tục ứng với hành động đặc biệt
- Các hàm xác nhận

trans(các định nghĩa chuyển trạng thái*)*

từ IDLE đến IDLE

khi PHY_provider.ACKRCVD

begin

end;

từ IDLE đến IDLE

khi TIM_provider.TEXP

begin

end;

:

:

từ WTACK đến IDLE

được cung cấp (P0 and P1)

khi PHY_provider.ACKRCVD

begin

end;

:

:

end.

Hình 5.8 (tiếp theo) Đặc tả Idle RQ: (a) Mã giả cho sơ cấp (b) Mã giả cho thứ cấp (c) Nét cấu trúc chính của sơ cấp theo Estelle.

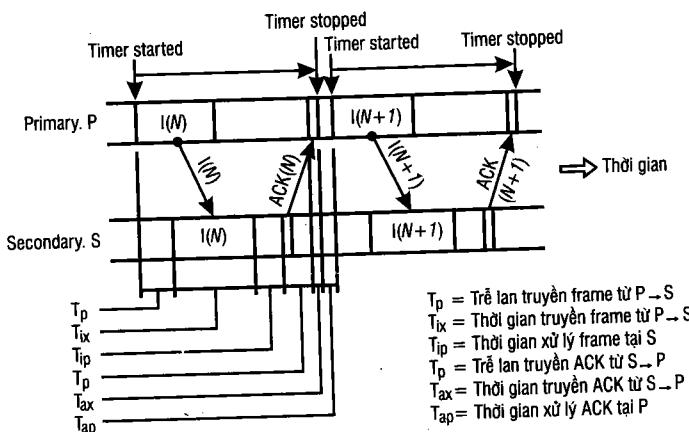
Một cấu trúc của thủ tục kiểm soát lỗi của sơ cấp idle RQ dùng phương pháp mô tả theo mã cấu trúc được trình bày trên hình 5.8(c), việc trình bày cấu trúc này chỉ nhằm các mục đích so sánh. Tiếp cận này được gọi là mô hình chuyển trạng thái mở rộng hay Estelle (extended state transitional model). Như chúng ta sẽ thấy, điều này tương tự như cấu trúc của các đặc tả trong phần (a) và phần (b) ngoại trừ thực thể giao thức được định nghĩa dưới dạng một module và tất cả các hướng chuyển có thể được định nghĩa một cách riêng biệt theo một phương pháp chuẩn hơn. Trong thực tế chỉ một số nhỏ các đặc tả là thuộc dạng này, hầu hết là dưới dạng bảng sự kiện-trạng thái mở rộng. Bằng cách áp dụng phương pháp luận hiện thực đã được trình bày chúng ta có thể tạo ra mã chương trình có cấu trúc tương tự.

2.4. Hiệu suất sử dụng liên kết

Trước khi xem xét các thủ tục kiểm soát lỗi liên hệ với hai loại giao thức RQ liên tục (continuous RQ protocol), chúng ta sẽ đánh giá hiệu quả sử dụng khả năng có sẵn của liên kết đối với giao thức idle RQ. Hiệu suất sử dụng U là tỷ số của hai đại lượng thời gian, mỗi đại lượng được đo từ thời điểm mạch phát bắt đầu gửi một frame. Nó được định nghĩa như sau:

$$U = T_{ix} / T_t$$

Trong đó : T_{ix} là thời gian để mạch phát truyền một frame và T_t bằng T_{ix} cộng với bất kỳ một lượng thời gian nào mà máy phát đợi một báo nhận.



Hình 5.9. Lược đồ sử dụng liên kết Idle RQ.

Để đánh giá hiệu suất sử dụng liên kết đối với Idle RQ, một sơ đồ tuần tự frame cùng với các thành phần thời gian được xác định trong hình 5.9. Trong thực tế hầu hết các trường hợp phù hợp với giao thức Idle RQ, thời gian xử lý một I-frame, T_{ip} và thời gian đợi báo nhận ACK-frame liên quan, T_{ap} đều nhỏ so với các thời gian truyền T_{ix} và T_{ax} . Cũng vì ACK-frame ngắn hơn nhiều so với một I-

Frame nén T_{ax} có thể bỏ qua so với T_{ix} . Do đó tổng thời gian tối thiểu trước khi I-frame kế tiếp có thể được truyền xấp xỉ bằng $T_{ix} + 2T_p$. Biểu thức của U là :

$$U = T_{ix} / (T_{ix} + 2T_p)$$

Hay :

$$U = 1 / (1 + 2T_p / T_{ix})$$

Như đã mô tả trong chương 2, tỷ số T_p / T_{ix} thường được ký hiệu là a và do đó:

$$U = 1 / (1 + 2a)$$

Ví dụ 1

Một chuỗi các frame, mỗi frame 1.000 bit, sử dụng giao thức Idle RQ để truyền. Hãy xác định hiệu suất sử dụng liên kết cho các loại liên kết sau đây, giả sử tốc độ truyền dữ liệu của (i) 1Kbps và (ii) 1Mbps. Tốc độ lan truyền của liên kết là 2×10^8 m/s và bỏ qua BER :

- (a). Cáp xoắn đôi có chiều dài 1km.
- (b). Đường leased-line chiều dài 200km.
- (c). Một liên kết vệ tinh chiều dài 50.000km.

Giải

Thời gian cần truyền một frame T_{ix} là :

$T_{ix} = \text{số bit trong frame} / \text{tốc độ bit (bps)}$.

Tốc độ bit R theo bps

Tại 1Kbps :

$$T_{ix} = 1.000 / 10^3 = 1s$$

Tại 1 Mbps :

$$T_{ix} = 1.000 / 10^6 = 10^{-3}s$$

$$T_p = S/V \text{ và } U = 1 / (1 + 2T_p / T_{ix}) = 1 / (1 + 2a)$$

(a). $T_p = 10^3 / 2 \times 10^8 = 5 \times 10^{-6}s$

(i). $a = 5 \times 10^{-6} / 1 = 5 \times 10^{-6}$

và do đó : $1+2a \approx 1$ và $U=1$

(ii). $a = 5 \times 10^{-6} / 10^{-3} = 5 \times 10^{-3}$

và do đó : $1+2a \approx 1$ và $U=1$

(b). $T_p = (200 \times 10^3) / 2 \times 10^8 = 1 \times 10^{-3}s$

(i). $a = 1 \times 10^{-3} / 1 = 1 \times 10^{-3}$

và do đó : $1+2a \approx 1$ và $U=1$



$$(ii). a = 1 \times 10^{-3} / 10^{-3} = 1$$

và do đó : $1+2a > 1$ và $U=1/(1+2) = 0,33$

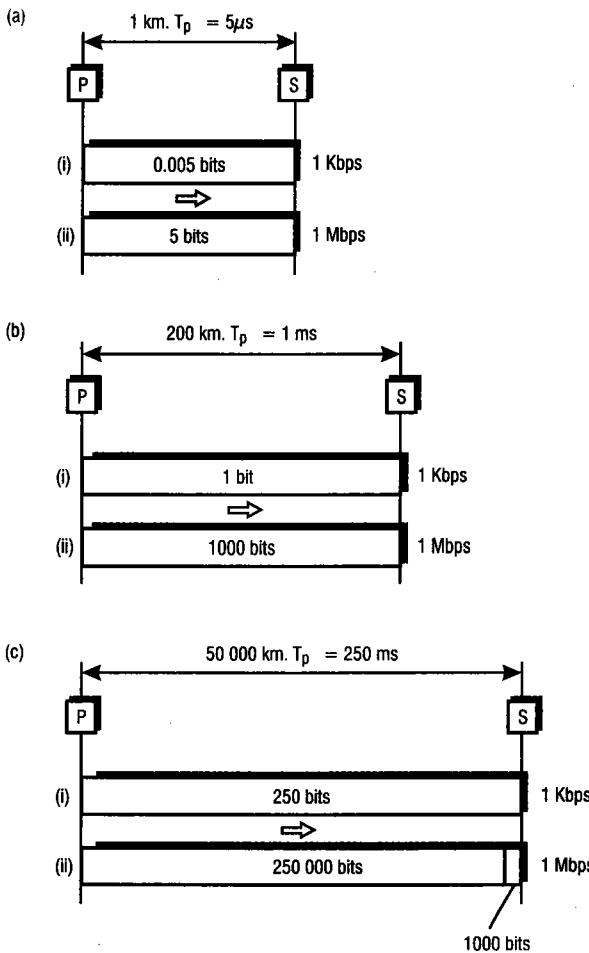
$$(c). T_p = (50 \times 10^6) / 2 \times 10^8 = 0,25\text{s}$$

$$(i). a = 0,25 / 1 = 0,25$$

và do đó : $1+2a > 1$ và $U=1 / (1+0,5) = 0,67$

$$(ii). a = 0,25 / 10^{-3} = 250$$

và do đó : $1+2a > 1$ và $U=1/(1+500) = 0,002$



Hình 5.10 *Ảnh hưởng của trễ do lan truyền là một hàm của tốc độ truyền số liệu: các phần tương ứng với ví dụ 1*

Kết quả được tóm tắt trong hình 5.10, từ đó chúng ta có thể rút ra một số điểm lưu ý. Trước hết liên quan đến các liên kết ngắn trong đó $a < 1$, hiệu suất sử

dụng liên kết là 100% và độc lập với tốc độ dữ liệu. Điều này có nghĩa là giao thức Idle RQ hoàn toàn thích hợp cho các liên kết ngắn và có tốc độ vừa phải. Ví dụ các mạng sử dụng các modem và PSTN analog. Thứ hai, các liên kết mặt đất dài hơn thì hiệu suất sử dụng liên kết sẽ cao ứng với tốc độ dữ liệu thấp và do đó giá trị của a thấp, nhưng hiệu suất sẽ giảm đáng kể khi tốc độ dữ liệu gia tăng. Thứ ba, hiệu suất sử dụng liên kết rất thấp đối với các liên kết vệ tinh ngay cả khi tốc độ dữ liệu thấp. Chúng ta có thể kết luận rằng một giao thức Idle RQ không phù hợp cho các ứng dụng như vậy và cũng không phù hợp cho các liên kết mặt đất tốc độ cao ví dụ như LAN và các mạng tải công cộng WAN.

Trong tính toán hiệu suất liên kết ở ví dụ 1 chúng ta đã giả sử không có lỗi truyền, nhưng trong thực tế các liên kết đều có $BER \neq 0$. Do đó để truyền một frame thành công, trung bình N_r , nỗ lực truyền sẽ được yêu cầu. Chúng ta có thể hiệu chỉnh lại biểu thức hiệu suất liên kết như sau:

$$U = T_{ix} / (N_r \cdot T_{ix} + 2N_r \cdot T_p) = 1 / N_r(1 + 2T_p/T_{ix})$$

Chúng ta có thể suy ra giá trị của N_r từ những giá trị đã biết là BER và P của liên kết. Nếu P là xác suất mà 1 bit bị hỏng và nếu giả sử các lỗi là ngẫu nhiên thì xác suất 1 frame có chiều dài là N_i bit được nhận mà không bị lỗi là :

$$P_{fe} = (1 - P)^{N_i}$$

Do đó xác suất một frame được thu mà có lỗi là:

$$P_f = 1 - (1 - P)^{N_i}$$

$$\approx N_r \cdot P \quad \text{nếu } N_r \cdot P \ll 1$$

Ví dụ : $BER = 10^{-4}$, có nghĩa là trung bình 1 trong 10^4 bit sẽ bị lỗi và do đó các frame 1.000 bit sẽ có :

$$P_f = 1 - (1 - 10^{-4})^{1000} = 0,095$$

Hay :

$$P_f = 10^3 \times 10^{-4} = 0,1$$

Nếu P_f là xác suất một frame bị hỏng thì $1 - P_f$ là xác suất một frame không bị hỏng sẽ được nhận và do đó:

$$N_r = 1 / (1 - P_f)$$

Ví dụ, nếu $P_f = 0,5$, $N_r = 2$, nghĩa là trung bình 50% frame bị hỏng, mỗi frame sẽ phải được truyền 2 lần, bởi vì 50% số frame truyền lại cũng sẽ bị hỏng, giả sử ACK-Frame không bị hỏng. Do các I-Frame là ngắn nên những điều giả sử ở trên là hợp lý. Trong thực tế tất cả các giá trị về hiệu suất đều phải được chia cho N_r :

$$U = (1 - P_f) / (1 + 2a)$$

Idle RQ với $BER \neq 0$

Ưu điểm chính của lược đồ idle RQ là nó yêu cầu lượng bộ đệm tối thiểu để hiện thực, vì cả P và S chỉ cần chứa đủ một frame. Ngoài ra S chỉ phải duy trì một



record danh định của frame nhận thành công sau cùng để phát hiện sự trùng lặp. Nhìn chung, các lược đồ truyền lại khác nhau cần dùng các bộ đệm cũng vì hiệu suất truyền. Bởi nhu cầu lưu trữ thấp nên idle RQ được dùng rộng rãi trong các ứng dụng có sự tham gia của các thiết bị đơn giản (ví dụ như đầu cuối hay máy tính) tại một đầu của liên kết.

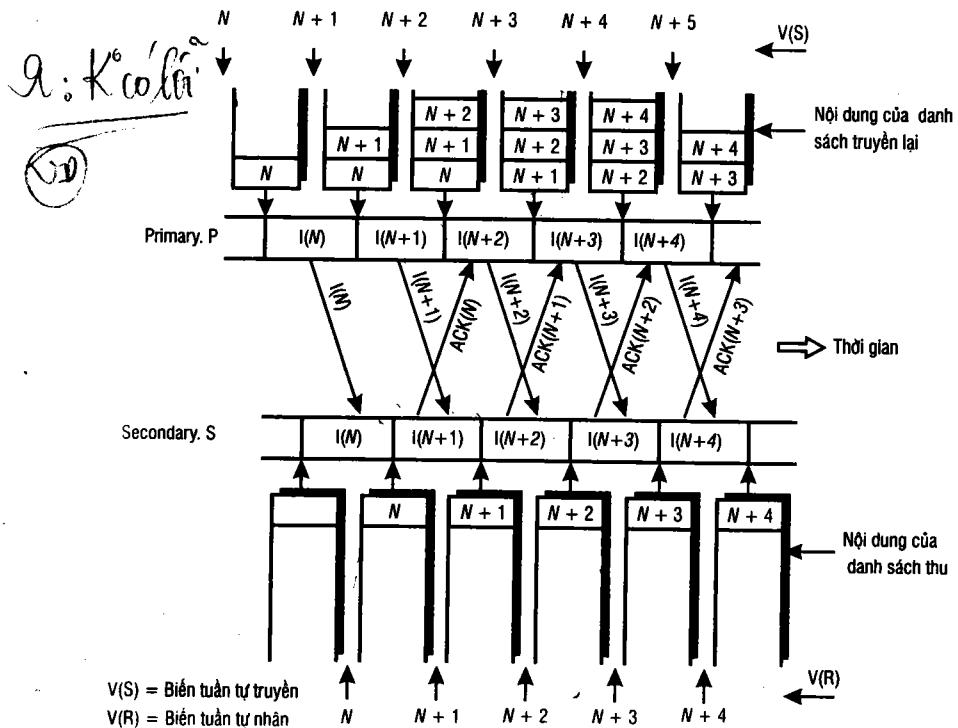
3. RQ LIÊN TỤC (CONTINUOUS RQ)

Đối với lược đồ kiểm soát lỗi RQ liên tục, hiệu suất sử dụng liên kết được cải thiện rất nhiều nhờ chấp nhận chi phí để mở rộng bộ đệm (bộ nhớ). Như chúng ta sẽ thấy, một liên kết song công được yêu cầu để hiện thực RQ liên tục. Ví dụ trên hình 5.11 mô tả hoạt động truyền các I-frame và các ACK-frame phúc đáp cho chúng theo tuần tự.

Lưu ý các điểm sau đây khi giải thích hoạt động của lược đồ:

- P gửi I-frame một cách liên tục mà không đợi một ACK-frame trả về.
- Vì có nhiều I-frame đang đợi báo nhận nên P phải giữ bản copy của mỗi I-frame truyền trong một *danh sách truyền lại* (*retransmission list*), danh sách này hoạt động theo nguyên tắc hàng đợi FIFO (First In-First Out).
- S phúc đáp một ACK-frame cho mỗi I-frame được nhận tốt.
- Mỗi I-frame chứa một danh định duy nhất, danh định này cũng được gửi về trong ACK-frame tương ứng.
- Khi nhận được một ACK-frame, P sẽ loại I-frame tương ứng ra khỏi danh sách truyền.
- Các I-frame đã nhận mà không có lỗi được đặt vào trong danh sách liên kết nhận (link receive list) để chờ xử lý.
- Khi nhận I-frame kế đang chờ theo tuần tự, S chuyển nội dung thông tin trong I-frame cho lớp trên (LS_user) ngay khi nó đã xử lý frame này.
- Trong trường hợp (sự kiện) nhận các I-frame không theo tuần tự, S giữ chúng trong danh sách liên kết nhận cho đến khi nhận được I-frame kế tiếp đúng theo tuần tự.

Như đã mô tả trong mục 2.1, giao tiếp giữa phần mềm mức cao và phần mềm truyền tin thường tồn tại ở dạng hai hàng đợi FIFO. Tuy nhiên, chúng ta phải ghi nhớ rằng không có mối quan hệ nào giữa chúng với danh sách liên kết truyền lại (tại P) và danh sách liên kết thu (tại S). Hai danh sách này được dùng bên trong bởi lớp truyền tin để bảo đảm độ tin cậy của hoạt động truyền khỏi dữ liệu giữa hai thực thể lớp cao hơn.



Hình 5.11 Tuần tự frame của Continuous RQ.

Để thực hiện lược đồ này, P phải giữ một biến tuần tự gửi $V(S)$, nó chỉ ra chỉ số tuần tự $N(S)$ được phân phối vào I-frame kế tiếp sẽ được truyền. Tương tự S cũng phải duy trì một biến tuần tự thu $V(R)$, nó chỉ ra I-frame kế tiếp đang mong đợi.

Từ hình 5.11, chúng ta có thể kết luận rằng, khi không có lỗi, hiệu suất sử dụng liên kết của một lược đồ RQ liên tục (sự xấp xỉ hợp lý) luôn là 100% với điều kiện hoạt động gửi I-frame của P không bị giới hạn. Tuy nhiên, như chúng ta sẽ thấy trong mục 3.3, trường hợp này là không thể, vì thường có giới hạn số I-frame mà P có thể gửi trước khi nhận một ACK-frame tương ứng. Do đó chúng ta tạm gác lại việc thảo luận hiệu suất sử dụng liên kết của RQ liên tục ở đây cho đến mục 3.6.

Hình 5.11 giả sử rằng không có lỗi truyền nào xảy ra. Khi một lỗi truyền xảy ra thì một trong hai chiến lược truyền lại sau đây có thể sử dụng:

- S chỉ phát hiện và yêu cầu truyền lại đối những frame trong tuần tự bị hỏng, chiến lược này được gọi là *truyền lại có chọn lựa - selective repeat*.
- S phát hiện sự tiếp nhận một I-frame không theo tuần tự và yêu cầu P truyền lại tất cả các I-frame chưa được báo nhận kể từ thời điểm nhận thành công một I-frame sau cùng (được báo nhận), chiến lược này được gọi là *truyền lại một nhóm - go-back-N*.

Lưu ý rằng trong tất cả các lược đồ RQ liên tục, các frame bị hỏng sẽ bị loại và yêu cầu truyền lại chỉ được kích hoạt sau khi nhận được I-frame không bị lỗi kế tiếp.

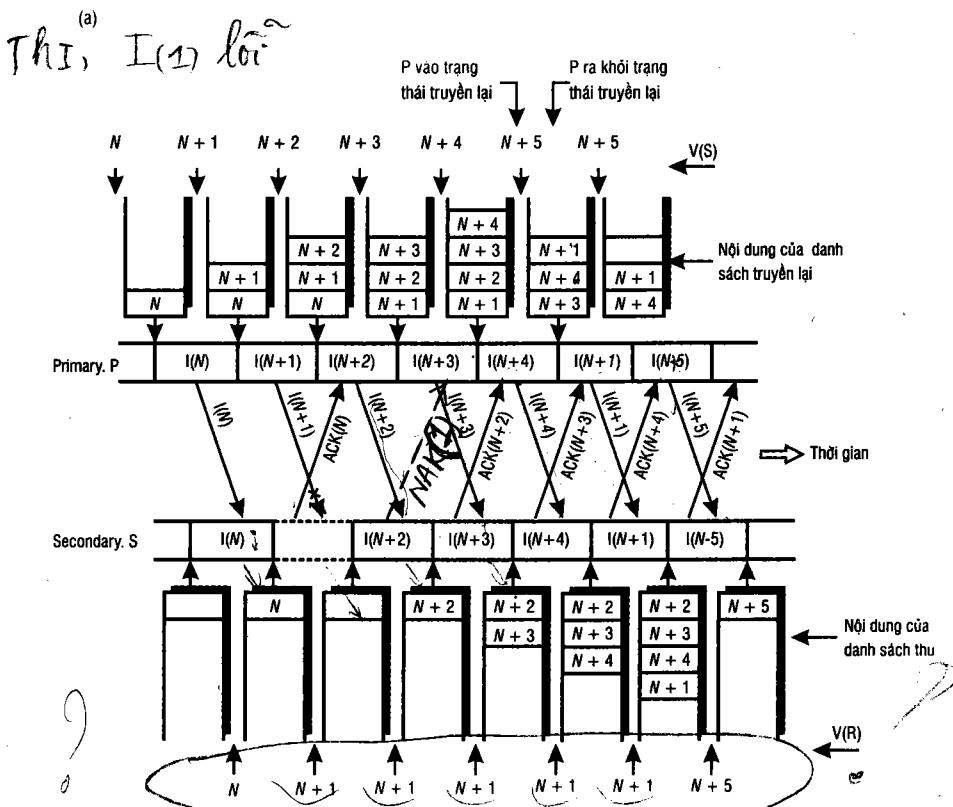
3.1. Truyền lại có chọn lựa (selective repeat)

Giống như lược đồ kiểm soát lỗi idle RQ, lược đồ truyền lại có chọn lựa (selective repeat) có thể được thực hiện theo một trong hai cách; S biết các frame

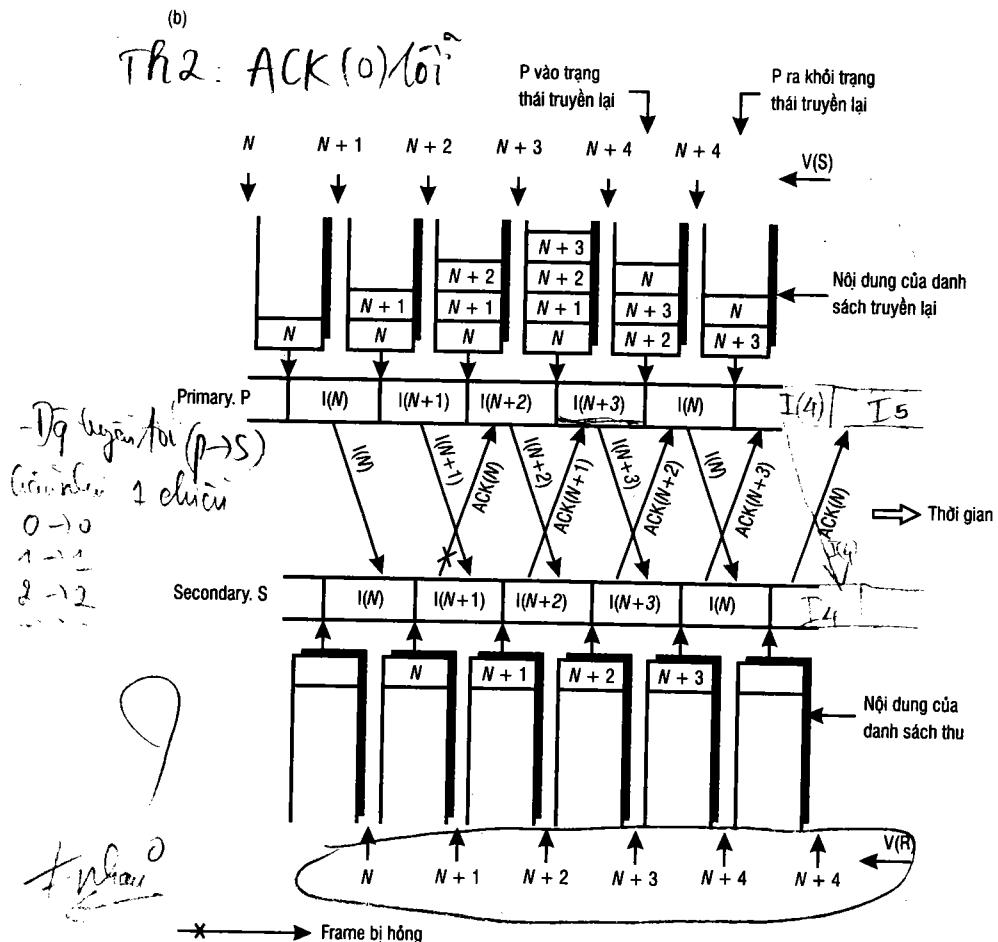
dược nhận tốt và P xác định một frame đã bị mất dựa vào tuần tự của các ACK-frame đã nhận_truyền lại ngầm định (implicit retransmission)_hoặc S trả về một thông báo từ chối nhận NAK-frame (negative acknowledgment) đối với một frame bị lệch tuần tự_yêu cầu rõ (explicit request). Trong cả hai lược đồ này, đối với sự kiện các frame nhận không đúng theo tuần tự, S giữ chúng trong một danh sách (liên kết) nhận cho đến khi nhận được frame kế tiếp đúng theo tuần tự.

Hai sơ đồ tuần tự frame mô tả các khía cạnh của lược đồ thứ nhất được trình bày trên hình 5.12. Phần (a) giả sử tất cả các ACK-frame đều được nhận tốt, trong khi phần (b) trình bày ánh hưởng khi một ACK-frame bị hỏng. Để theo dõi tuần tự trong hình 5.12(a), lưu ý các điểm sau đây:

- Giả sử I-frame N+1 bị hỏng
- S phục dáp một ACK-frame cho mỗi I-frame được nhận tốt ngay trước đó.
- S phục dáp một ACK-frame cho các I-frame N, N+2, N+3...
- ✓ • Khi nhận ACK-frame cho I-frame N+2, P phát hiện rằng frame N+1 đã không được báo nhận.
- ✓ • Để chấp nhận khả năng có nhiều hơn một I-frame đang bị hỏng, khi phát hiện một frame không có báo nhận P chuyển vào trạng thái truyền lại.
- Khi ở trong trạng thái này, việc truyền một frame mới bị hoãn lại cho đến khi tất cả các frame không được báo nhận đều đã được truyền lại.



Hình 5.12 Truyền lại có chọn lựa_ngầm định (selective repeat_implicit retransmission): (a) I-frame bị hỏng



Hình 5.12 (tiếp theo) Truyền lại có chọn lựa_ ngầm định (selective repeat_implicit retransmission): (b) ACK-frame bị hỏng.

- P xóa I-frame N+2 khỏi danh sách truyền lại và truyền lại I-frame N+1 trước khi truyền frame N+5.
- Khi nhận I-frame N+1, nội dung của các frame được xếp hàng trong danh sách liên kết nhận được S phân phối đến LS_user theo đúng tuần tự.

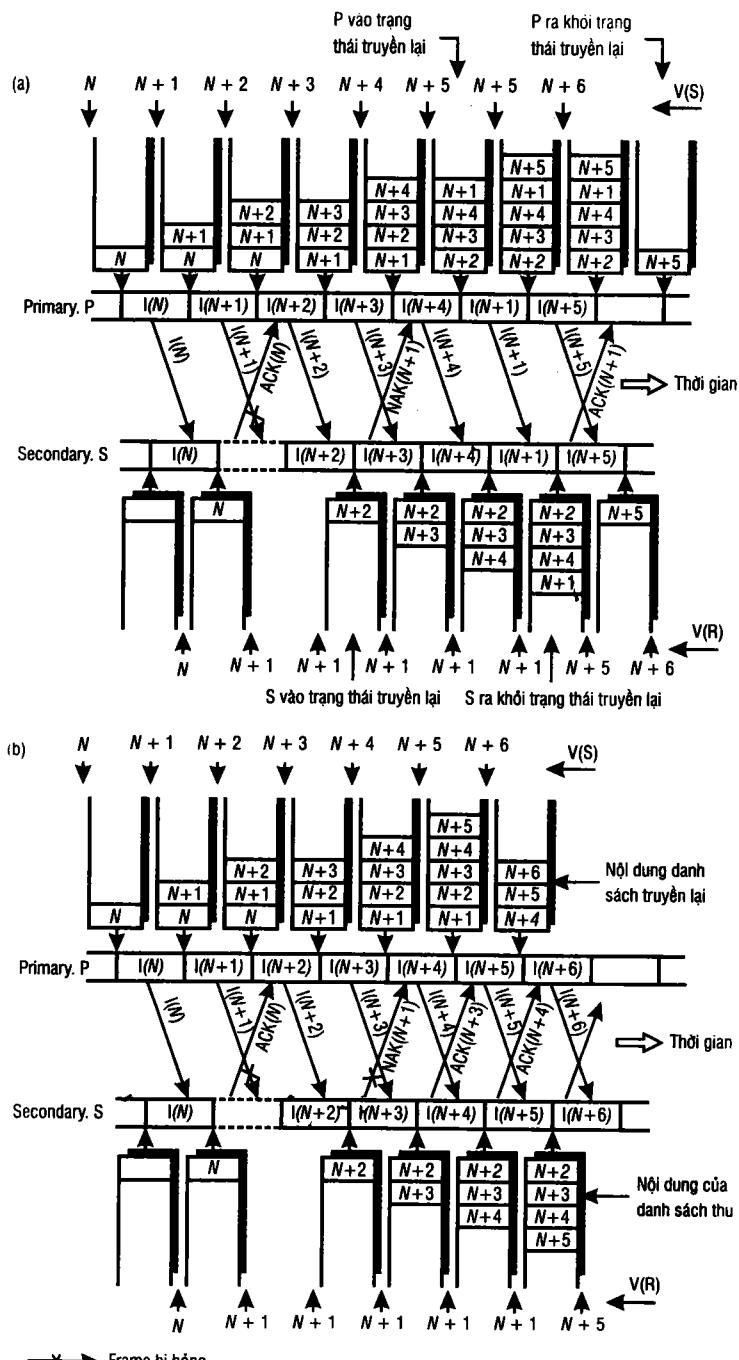
Tuần tự frame được trình bày trên hình 5.12(b) liên quan đến tình huống trong đó tất cả các I-frame đều được nhận tốt nhưng ACK-frame N bị hỏng. Lưu ý rằng:

- Khi nhận ACK-frame N+1, P phát hiện ra rằng I-frame N vẫn còn chờ báo nhận và do đó truyền lại nó.
- Khi nhận I-frame truyền lại N, nhờ vào biến tuần tự nhận mà S xác định frame này đã được nhận tốt và do đó I-frame này là bản trùng lặp.

- S loại bỏ frame trùng lặp này nhưng vẫn gửi trả một ACK-frame cho nó để đảm bảo P xóa frame này khỏi danh sách truyền lại.

Hoạt động của lược đồ trên căn cứ vào việc nhận các ACK-frame liên hệ đến các I-frame tiếp theo để sớm khởi động truyền lại một frame bị hỏng. Một tiếp cận khác là dùng một frame từ chối rõ ràng để yêu cầu truyền lại một I-frame xác định. Sự từ chối (negative acknowledgment) được xem như loại bỏ có chọn lọc *selective reject*. Hình 5.13 trình bày hai sơ đồ ví dụ về tuần tự frame minh họa hoạt động của lược đồ này. Tuần tự được trình bày trên phần (a) giả sử không có báo nhận nào bị hỏng, trong khi phần (b) trình bày ảnh hưởng của một báo nhận bị hỏng. Lưu ý rằng:

- Một ACK-frame báo nhận cho tất cả các I-frame trong danh sách truyền lại tính từ I-frame có số tuần tự đang chứa trong ACK-frame trở về trước (bao gồm I-frame có số tuần tự này).
- Giả sử I-frame N+1 bị hỏng.
- S trả một ACK-frame cho I-frame N
- Khi S nhận I-frame N+2, nó phát hiện I-frame N+1 bị thất lạc và do đó đáp lại bằng một NAK-frame chứa danh định của I-frame thất lạc N+1.
- Khi nhận NAK-frame N+1, P xem như S vẫn còn đang đợi I-frame N+1, do đó sẽ truyền lại I-frame này.
- Khi S gửi một NAK-frame nó sẽ vào trạng thái truyền lại.
- Một khi đã ở trong trạng thái truyền lại thì việc gửi ACK-frame tạm thời bị treo.
- Khi nhận I-frame N+1, S giải phóng trạng thái truyền lại và khôi phục việc gửi ACK-frame.
- ACK-frame N+4 báo nhận cho tất cả các frame có chỉ số đến N+4, bao gồm cả N+4.
- Một timer được dùng cho mỗi NAK-frame để đảm bảo rằng nếu nó bị hỏng (và vì vậy không nhận được I-frame N+1) thì được truyền lại.



Hình 5.13 Truyền lại có chọn lựa yêu cầu rõ (selective repeat_explicit request): (a) Hoạt động tốt (b) Sai sót do không có trạng thái truyền lại.

Để hiểu tại sao chỉ có một NAK-frame có thể tồn tại tại một thời điểm, hãy xem xét sơ đồ tuần tự frame ở hình 5.13 (b). Lưu ý rằng:

- Giả sử I-frame N+1 lại bị hỏng một lần nữa.
- S gửi đáp một NAK-frame N+1 như trước đây nhưng lần này nó lại bị hỏng
- Khi nhận ACK-frame N+3, điều này báo nhận tốt cho tất cả các frame từ N+3 trở xuống, có nghĩa là bao gồm cả I-frame N+1, vì vậy I-frame này không được truyền lại.
- I-frame N+1 sẽ bị mất.

Từ các tuần tự frame đã trình bày chúng ta có thể suy ra rằng mặc dù S nhận một bản copy tốt của mỗi I-frame do P gửi, nhưng thứ tự nhận có thể không được duy trì. Ví dụ, S nhận I-frame N+2, N+3 và N+4 trước N+1. Vì vậy selective repeat được dùng chủ yếu cho trường hợp các I-frame đang được truyền là các thực thể độc lập (self-contained entities), nghĩa là thứ tự nhận không quan trọng, hoặc được dùng khi tất cả các I-frame liên quan đến cùng một bản tin (hay frame lớn hơn) và các bản tin (frame) được tái thiết trở lại hình dáng ban đầu bởi thứ cấp (S). Trường hợp sau chính là khi chúng ta dùng một liên kết có BER cao (như các liên kết radio) và kích thước I-frame dùng trên liên kết là nhỏ. Tiếp cận thông thường cho ứng dụng này là dùng selective repeat cho tất cả các phân mảnh frame nhỏ của mỗi frame lớn hơn và tái thiết lập các frame nhỏ hơn thành một frame lớn ban đầu trước khi phân phối frame. Một NAK được dùng để phát ra yêu cầu truyền lại cho bất cứ mảnh nào bị mất.

Trong nhiều ứng dụng, frame phải được phân phối theo tuần tự giống như khi được tạo ra. Do đó các frame nào được nhận không theo đúng thứ tự phải được đệm (giữ) bởi S cho đến khi nhận được các frame thất lạc. Thông thường các frame này có kích thước lớn và số lượng frame cần đệm cũng có thể lớn, điều này đòi hỏi hệ thống phần cứng phải có dung lượng lưu trữ quá lớn đến mức không thể chấp nhận được. Vì lý do này mà hầu hết các ứng dụng đã được giới thiệu, cũng như trong hầu hết các mạng đều dùng lược đồ điều khiển truyền lại một nhóm _go-back-N.

3.2. Truyền lại một nhóm (Go-back_N)

Đối với go-back-N thì như tên gọi của nó, khi thứ cấp phát hiện một frame không đúng tuần tự, nó báo cho sơ cấp để bắt đầu truyền lại các frame bắt đầu từ chỉ số frame đã chỉ định. Nó thực hiện điều này bằng cách gửi một frame phủ nhận đặc biệt gọi là **reject**. Hai tuần tự frame mô tả hoạt động của go-back-N được trình bày trên hình 5.14. Khi giải thích tuần tự trong hình này cần lưu ý:

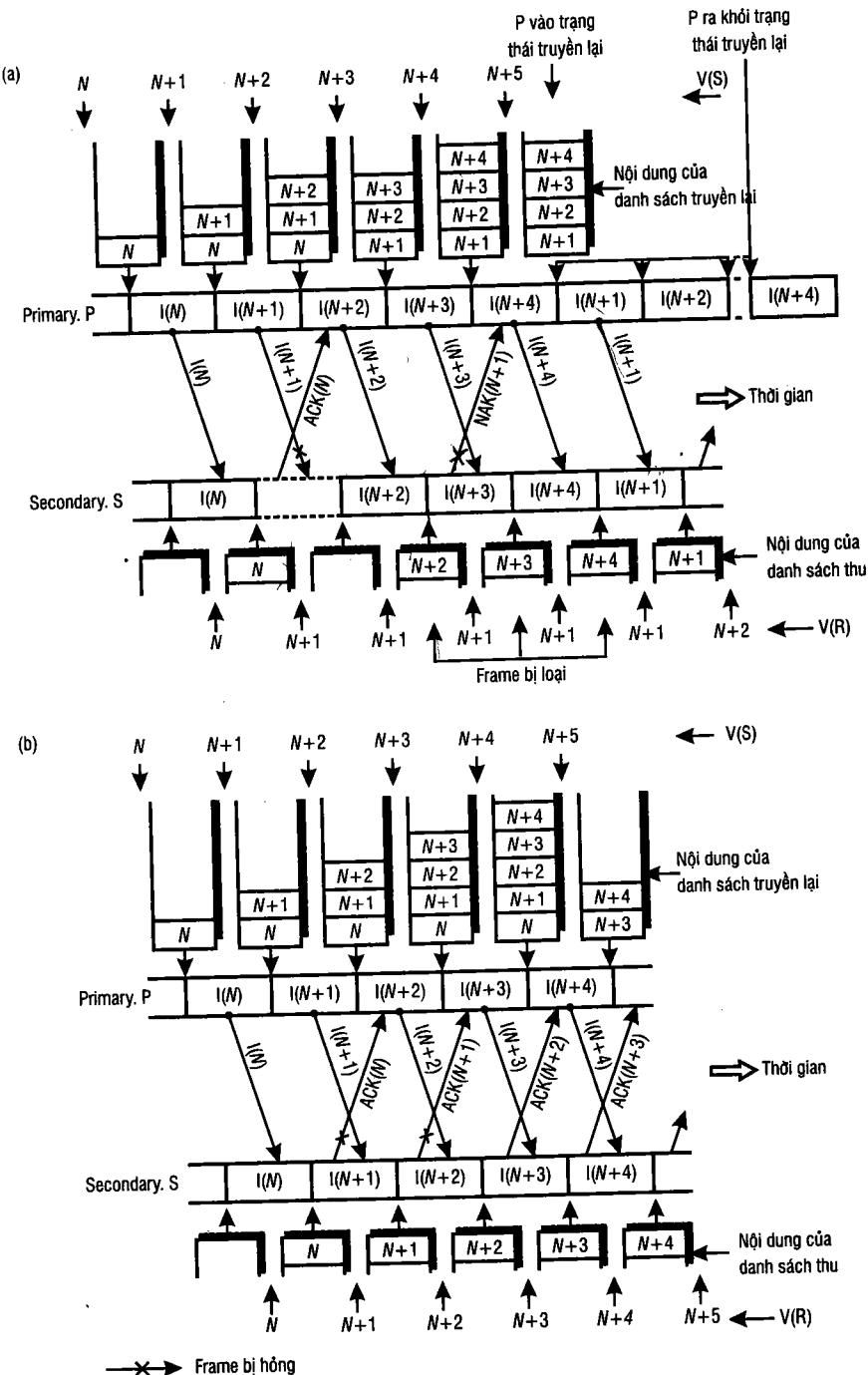


- Giả sử I-frame N+1 bị hỏng.
- S nhận I-frame N+2 không đúng tuần tự.
- Khi nhận I-frame N+2, S gửi NAK-frame N+1 báo cho P quay trở lại và bắt đầu truyền lại từ I-frame N+1.
- Khi nhận NAK-frame N+1, P vào trạng thái truyền lại.
- Khi ở trong trạng thái này, nó tạm thời dừng truyền frame mới và bắt đầu truyền lại các frame đang đợi báo nhận trong danh sách truyền lại.
- S loại bỏ frame cho đến khi nhận được I-frame N+1
- Khi nhận I-frame N+1, S trả lại trạng thái sẵn sàng nhận frame mới và gửi báo nhận cho P
- Một khoảng thời gian **timeout** được S áp dụng cho các NAK-frame và một NAK-frame thứ hai được gửi nếu không nhận được một I-frame theo tuần tự trong khoảng thời gian của **timeout**

Lần nữa, chúng ta lại giả sử rằng trong hình 5.14(a) một I-frame bị hỏng và các frame báo nhận đều được nhận tốt. Ảnh hưởng do báo nhận bị hỏng trên tuần tự truyền frame được trình bày trên hình 5.14(b). Lưu ý rằng:

- S nhận mỗi I-frame được truyền thành công.
- Giả sử ACK-frame N và N+1 đều bị hỏng.
- Khi nhận ACK-frame N+2, P phát hiện có hai I-frame bị mất tuần tự báo nhận trong danh sách truyền lại (N và N+1);
- Vì nó là ACK-frame chứ không phải là một NAK-frame, nên P giả sử rằng hai ACK-frame ứng với I-frame N và N+1 đã bị hỏng và do đó chấp nhận ACK-frame N+2 cũng như là báo nhận cho các frame còn đang đợi này.

Hình 5.14(a) cho thấy với chiến lược go-back-N, tuần tự frame chính xác được duy trì, do đó tối thiểu được nhu cầu dùng bộ đệm (bộ nhớ) trong hoạt động. Tuy nhiên, vì một vài frame đã được nhận tốt mà vẫn cứ phải truyền lại, do đó hiệu suất sử dụng băng thông truyền có sẵn thấp hơn so với khi dùng lược đồ truyền lại có chọn lựa (selective repeat). Do đó cần một sự cân đối nhất định giữa nhu cầu bộ nhớ và hiệu quả sử dụng liên kết.

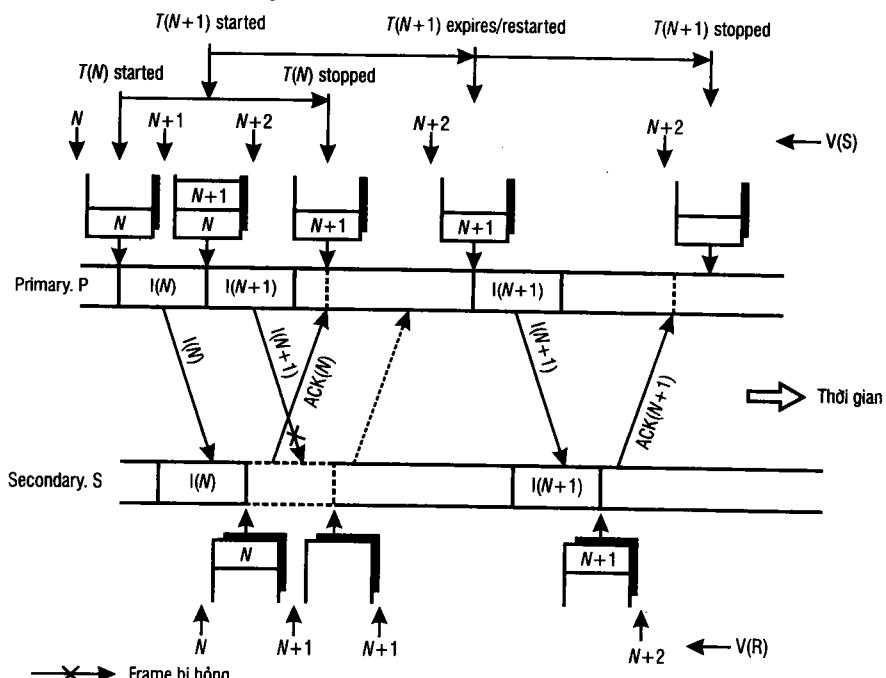


Hình 5.14 Chiến lược truyền lại go-back-N: (a) *I-frame bị hỏng* (b) *ACK-frame bị hỏng*.

Sau cùng, chúng ta giả sử rằng trong các hình 5.11 đến 5.13 liên hệ với Continuous RQ, việc bắt I-frame chỉ được phát hiện sau khi I-frame kế tiếp đã được nhận thành công. Điều này yêu cầu một luồng I-frame liên tục luôn sẵn sàng gửi, nếu không, phải chịu một khoảng thời gian trễ khá dài không thể chấp nhận được khi mà S đang đợi P truyền I-frame kế tiếp.

Để thỏa mãn điều này, P thường dùng một cơ cấu timeout tương tự như cơ cấu đã được mô tả trong lược đồ điều khiển idle RQ. Có một số lược đồ có thể sử dụng, nhưng một lược đồ đã được chọn trong các định nghĩa giao thức trước đây là dùng một bộ định thời riêng, được khởi động mỗi khi P truyền một I-frame và sẽ dừng khi một báo nhận đến báo là đã nhận tốt. Nếu không nhận được báo nhận cho I-frame trước khi thời gian timeout hết hạn, thì I-frame được truyền lại. Điều này được trình bày trên hình 5.15.

Khoảng thời gian timeout được chọn phải lớn hơn thời gian trễ lan truyền xấu nhất giữa thời điểm truyền một I-frame và thời điểm nhận được báo nhận tương ứng. Với cơ cấu timeout này cũng tạo ra khả năng S nhận trùng frame ngay cả khi frame đã được nhận tốt, bởi đó là kết quả do báo nhận bị hỏng trên con đường đến P. Không có báo nhận, P coi như I-frame đã bị hỏng và truyền lại một bản copy khác. Đối với lược đồ điều khiển go-back-N, điều này không thành vấn đề vì $N(S)$ trong các frame trùng sẽ không bằng với $V(R)$ hiện hành đang được S giữ và sẽ bị loại bỏ một cách tự động. Tuy nhiên, với lược đồ truyền lại có chọn lựa (selective repeat) cho phép có một hay nhiều frame trùng được truyền bởi P nếu như S duy trì một danh sách thứ tự có N I-frame sau cùng được nhận tốt. Bằng cách này, S có thể kiểm tra một I-frame vừa nhận có trùng hay không. Như chúng ta sẽ thấy trong mục 4 của phần này, số lượng frame N được duy trì bị chi phối bởi giải thuật điều khiển luồng (flow control).



Hình 5.15 Cơ cấu timeout.



Để rõ ràng hơn chúng ta giả sử rằng các I-frame chạy theo một hướng và đường quay về chỉ đơn giản là cho báo nhận. Tuy nhiên, thông thường hầu hết các liên kết truyền tin dùng continuous RQ là các liên kết song công hoàn toàn (full-duplex) và mang các frame thông tin theo cả hai hướng. Để phù hợp với điều này, mỗi phía của liên kết đều chứa cả sơ cấp và thứ cấp: sơ cấp điều khiển tuần tự truyền các frame và thứ cấp điều khiển tuần tự nhận các frame. Do đó, mỗi phía của liên kết đều chứa cả V(S) được điều khiển bởi P và V(R) được điều khiển bởi S. Mặc dù các ACK-frame và NAK-frame được dùng riêng biệt, nhưng vì có khả năng một I-frame đang đợi truyền theo hướng ngược lại trong khi một ACK-frame hay một NAK-frame cũng đang được gửi lại (cùng hướng với I-frame truyền ngược lại), nên một vài giao thức dùng các I-frame theo hướng quay về này để mang thông tin báo nhận cho hoạt động truyền các I-frame theo hướng kia, cách này được xem như một cách để cải thiện hiệu suất sử dụng liên kết. Vì vậy mỗi I-frame được truyền chứa cả N(S) chỉ ra số tuần tự gửi và N(R) chứa thông tin báo nhận cho hướng ngược lại. Lược đồ này được gọi là *báo nhận ký sinh* (piggyback acknowledgment). Một trong những giao thức dùng kỹ thuật này là giao thức điều khiển liên kết số liệu mức cao HDLC (High-level Data Link Control).

Tóm lại, mặc dù tất cả các ACK-frame và NAK-frame mà chúng ta đã nói đến đều mang một N(R) bằng với N(S) mà chúng liên quan, nhưng trong thực tế V(R) tại thứ cấp gia tăng ngay khi nhận tốt một I-frame và trước khi báo nhận được phát ra. Điều này có nghĩa là với hầu hết các hiện thực giao thức thực tế thì ACK-frame N báo nhận cho I-frame N-1. HDLC cũng dùng kỹ thuật này.

3.3. Điều khiển luồng (flow control)

Kiểm soát lỗi chỉ là một thành phần của một giao thức liên kết số liệu. Một thành phần quan trọng khác là điều khiển luồng. Như tên gọi, thành phần này liên quan đến điều khiển tốc độ truyền ký tự dữ liệu (hay frame) trên liên kết sao cho nơi thu luôn luôn có đủ tài nguyên bộ nhớ để tiếp nhận chúng trước khi xử lý. Ví dụ, với một liên kết đầu cuối nối máy tính theo hướng ký tự, nếu máy tính đầu xa phục vụ cho nhiều đầu cuối, nó có thể bị quá tải tạm thời và không thể nào xử lý tất cả các ký tự gửi đến với tốc độ truyền hiện có. Tương tự, với lược đồ truyền lại có chọn lựa theo hướng frame, máy thu có thể hết dung lượng bộ nhớ nếu như nó đang cố gắng đệm một số frame không xác định. Chúng ta sẽ lần lượt xem xét hai lược đồ điều khiển luồng phổ biến nhất.

3.3.1. X-ON/X-OFF

Từ mục 1 chúng ta có thể suy ra rằng đối với kiểm tra dội lại (echo checking) thì ngoài thực hiện kiểm soát lỗi (bằng tay) thao tác này còn bao hàm cả chức năng điều khiển luồng vì nếu máy tính ở xa hết bộ nhớ, nó sẽ dừng dội



(echoing) các ký tự trở lại màn hình. Do đó người dùng sẽ tự động không gõ thêm ký tự nữa. Thông thường sự thiếu vắng các ký tự dội lại là do máy tính bị quá tải tạm thời. Trong trường hợp này, nếu người dùng không ngừng truyền ký tự mới thì máy tính sẽ chịu các lãng phí xử lý không cần thiết bởi chỉ đọc mỗi ký tự rồi lại loại bỏ nó.

Do đó thường yêu cầu một phương tiện điều khiển luồng tự động để đảm bảo đầu cuối không gửi thêm ký tự cho đến khi điều kiện quá tải đã được khắc phục. Để làm điều này, máy tính gửi lại ký tự điều khiển đặc biệt X-OFF đến thiết bị điều khiển bên trong đầu cuối, chỉ thị cho nó ngừng truyền ký tự mới. Khi nhận ký tự X-OFF, đầu cuối bỏ qua tất cả các ký tự được nhập vào từ bàn phím hoặc đệm chúng trong bộ nhớ nội bộ cho đến khi điều kiện quá tải đã kết thúc. Bằng cách này máy tính tránh được bất kỳ một lãng phí xử lý nào. Sau đó, khi điều kiện quá tải đã được khắc phục và máy tính đầu xa có thể chấp nhận ký tự mới, nó gửi ký tự điều khiển bắt tay X-ON để thông báo cho thiết bị điều khiển đầu cuối truyền rằng có thể khởi động truyền ký tự trở lại. Cơ cấu này cũng được dùng khi máy tính gửi các ký tự đến máy in hay đầu cuối khác không thể giữ tốc độ xuất dữ liệu bằng máy tính. Trong trường hợp như vậy, thiết bị điều khiển trong máy in (hay đầu cuối) sẽ điều khiển luồng ký tự.

Các đường điều khiển bắt tay (handshake lines) đã được thảo luận trong phần giao tiếp vật lý EIA-232D/V.24 có thể được dùng để thực hiện chức năng tương tự. Để phân biệt giữa hai phương pháp, người ta gọi phương pháp sau là điều khiển luồng ngoài băng (out-of-band) trong khi X-ON/X-OFF là một trường hợp của điều khiển luồng trong băng (in-band).

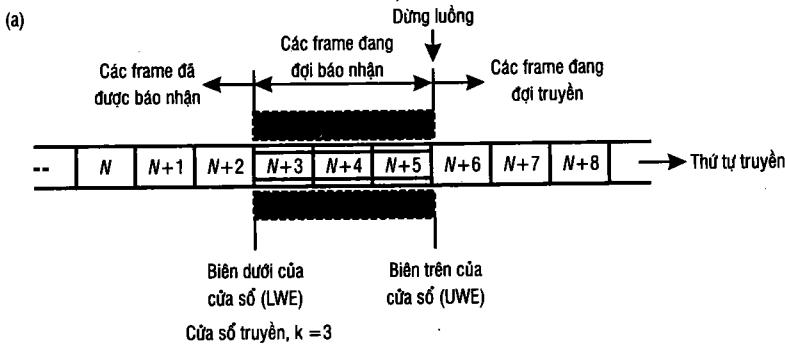
3.3.2. Cửa sổ trượt (sliding window)

Để điều khiển luồng frame đi qua liên kết số liệu, có một cơ cấu khác được gọi là **cửa sổ trượt** được dùng. Nhớ lại rằng trong lược đồ kiểm soát lỗi idle RQ, mặc dù việc sử dụng băng thông của nó là không hiệu quả, nhưng nó yêu cầu dung lượng bộ nhớ tối thiểu, vì sau khi một frame đã được truyền bởi P, nó phải đợi cho đến khi nhận một báo nhận từ S trước khi truyền một frame mới. Do đó luồng I-frame đi qua liên kết được điều khiển một cách chặt chẽ và tự động.

Tuy nhiên, đối với lược đồ kiểm soát lỗi continuous RQ, P có thể gửi các I-frame một cách liên tục trước khi nhận bất kỳ một báo nhận nào. Với dạng lược này máy thu có thể hết bộ nhớ, ví dụ nếu nó không thể chuyển các frame đi theo tốc độ nhận frame. Thường phải đưa thêm vào các lược đồ này một thao tác điều chỉnh. Tiếp cận này giống với lược đồ điều khiển idle RQ, trong đó chủ yếu là đặt một giới hạn đối với số I-frame đang đợi hiện hành được giữ trong danh sách truyền lại. Nếu đầu thu của liên kết không thể chuyển các I-frame gửi đến nó, S



ngừng gửi các frame báo nhận, danh sách truyền lại tại P được xây dần lên và đến lúc nào đó có thể đến giới hạn, được xem là dấu hiệu để P ngừng gửi I-frame cho đến khi các báo nhận khởi động luồng trở lại.



(b)

Giao thức	Cửa sổ truyền	Cửa sổ thu
Idle RQ	1	1
Selective repeat	K	K
go-back-N	K	1

Hình 5.16 Nguyên lý điều khiển luồng:

(a) Ví dụ về cửa sổ trượt (b) Các giới hạn cửa sổ truyền và nhận.

Để hiện thực lược đồ này, số lượng I-frame có thể đợi các báo nhận bị giới hạn bởi một con số tối đa. Giới hạn này được gọi là cửa sổ truyền (send window) của liên kết, thường được ký hiệu là K. Nếu K bằng 1 thì lược đồ truyền này quay trở lại idle RQ với hiệu suất truyền thấp. Giới hạn này thường được chọn sao cho cửa sổ truyền không làm suy thoái luồng I-frame xuyên qua liên kết và cung cấp cho đầu thu khả năng chuyển qua hay hấp thụ hết tất cả các I-frame gửi đến. Các yếu tố như kích thước tối đa của frame, dung lượng bộ nhớ có sẵn, thời gian trễ lan truyền trên liên kết và tốc độ truyền bit đều phải được xem xét khi chọn cửa sổ truyền.

Hoạt động của lược đồ này được trình bày trên hình 5.16. Khi truyền mỗi I-frame, biên trên của cửa sổ _UWE (upper window edge)_ tăng một đơn vị. Tương tự, khi mỗi I-frame được báo nhận, biên dưới của cửa sổ _LWE (lower window edge)_ cũng tăng một đơn vị. Nếu sai biệt giữa UWE và LWE bằng với cửa sổ truyền K thì ngưng chấp nhận I-frame mới. Giả sử việc truyền không bị lỗi, thì cửa sổ K là cố định và trượt (di chuyển) qua một khối frame hoàn chỉnh đang được truyền. Vì thế kỹ thuật này được gọi là "cửa sổ trượt".

Kích thước lớn nhất của bộ đệm frame tại S được gọi là cửa sổ thu (receive window). Từ các sơ đồ tuần tự frame trước đây chúng ta có thể suy ra đối với các lược đồ idle RQ và go-back-N chỉ yêu cầu đệm một frame (1 bộ đệm). Tuy nhiên,

đối với truyền lại có chọn lựa (selective repeat), yêu cầu đệm K frame để đảm bảo các frame được phân phối đến LS_user theo đúng tuần tự.

3.4. Các chỉ số tuần tự

Cho đến bây giờ, chúng ta vẫn giả sử rằng số tuần tự được P chèn vào mỗi frame chỉ đơn giản bằng số tuần tự trước đó cộng thêm 1 và dãy số có sẵn là không xác định. Định nghĩa giới hạn tối đa đối với số lượng các frame đang được truyền qua liên kết không chỉ giới hạn kích thước của danh sách (liên kết) truyền lại và danh sách (liên kết) thu mà còn giới hạn dãy số tuần tự được yêu cầu để nhận dạng duy nhất mỗi frame được truyền. Số lượng chỉ số danh định tùy thuộc vào lược đồ điều khiển truyền lại, kích thước cửa sổ truyền và kích thước cửa sổ thu.

Ví dụ, với lược đồ điều khiển idle RQ, cả hai cửa sổ truyền và nhận đều là 1 và do đó cần hai danh định cho phép S xác định I-frame vừa nhận là mới hay trùng. Thông thường, hai danh định là 0 và 1; biến tuần tự gửi được P tăng theo modulo-2.

Với lược đồ điều khiển go-back-N có một cửa sổ truyền là K thì số lượng danh định tối thiểu phải là K+1. Chúng ta có thể thấy điều này bằng cách xem xét ví dụ được trình bày trong hình 5.17(a). Lưu ý rằng:

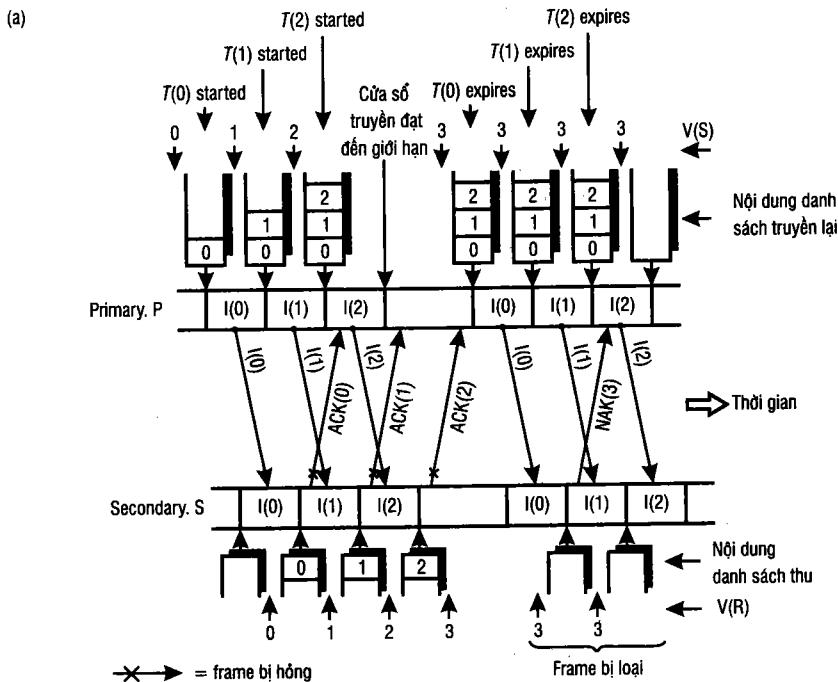
- Cửa sổ truyền được dùng trong ví dụ là 3.
- P truyền nguyên một cửa sổ: 3 I-frame.
- 3 I-frame được S nhận tốt.
- Cá 3 ACK-frame được S phục đáp đều bị hỏng.
- P xác định timeout cho mỗi I-frame và truyền lại chúng.
- S loại bỏ mỗi I-frame trùng và gửi một NAK-frame chỉ I(3) là frame kế đang được đợi.
- Vì số tuần tự trong NAK-frame bằng với V(S), nên P xem nó như một báo nhận cho 3 frame đang đợi, từ đó mở lại cửa sổ.

Nếu chỉ dùng 3 danh định_nghĩa là bằng với kích thước cửa sổ truyền_S sẽ không thể xác định I(0) là một I-frame mới hay I-frame trùng, vì 0 sẽ là danh định kế tiếp theo tuần tự. Với bốn danh định (K+1), S biết rằng I-frame kế tiếp trong tuần tự có một trong 3 danh định, ngược lại I-frame truyền lại có danh định là 0. Do đó nó loại bỏ chính xác I-frame sau cùng.

Với lược đồ truyền lại có chọn lựa và cửa sổ truyền và nhận là K, có thể suy ra số lượng danh định không thể nhỏ hơn 2K. Một lần nữa, chúng ta có thể suy ra điều này bằng cách xem xét trường hợp khi P gửi đầy đủ một cửa sổ (K I-frame) và tất cả các báo nhận sau đó đều bị hỏng. S phải có thể xác định được bất kỳ K I-frame kế tiếp nào có phải là I-frame mới không. Chỉ có một cách để S xác định được điều này là gán một tập K danh định mới hoàn toàn cho cửa sổ I-frame được truyền kế tiếp, nó yêu cầu ít nhất là 2 K danh định. Các giới hạn cho mỗi lược đồ được trình bày trên hình 5.17(b).

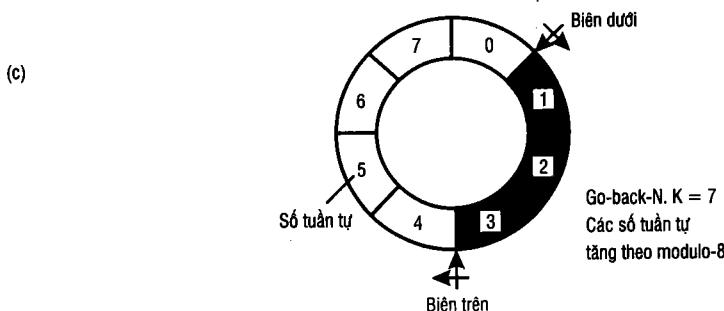
Trên thực tế, vì danh định của một I-frame là dưới dạng số nhị phân, nên một tập ký số nhị phân phải được dành riêng để dùng cho nó. Ví dụ, với cửa sổ

truyền là 7 và lược đồ là go-back-N thì cần dùng 3 ký số nhị phân cho các số tuần tự truyền và tuần tự nhận tạo ra 8 danh định có thể: 0 đến 7. Các biến tuần tự truyền và nhận lần lượt được P và S tăng theo modulo-8. Điều này được trình bày trên hình 5.17(c).



(b)

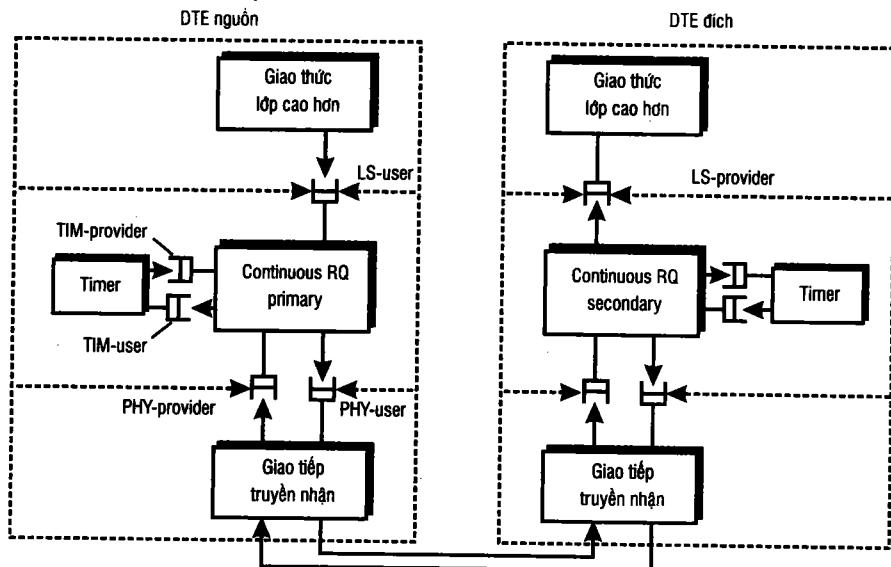
Giao thức	Số danh định tối đa
Idle RQ	2
Selective repeat	2^k
Go-back-N	$k+1$



Hình 5.17 Các số tuần tự: (a) Ví dụ dài tối đa (b) Số lược tối đa cho mỗi giao thức (c) Giả sử có 8 số tuần tự.

3.5. Đặc tả giao thức

Các sơ đồ tuần tự frame trong các hình từ 5.11 đến 5.18 cùng với các chú thích kèm theo đã cung cấp một mô tả về các thành phần điều khiển luồng và kiểm soát lỗi của giao thức continuous RQ. Bây giờ chúng ta có thể có được một đặc tả hình thức hơn dựa vào phương pháp luận đã được trình bày trong mục 2. Để đơn giản việc mô tả, chúng ta sẽ chỉ xem xét luồng I-frame theo một hướng từ DTE nguồn đến DTE đích. Tuy nhiên trong hầu hết các ứng dụng đều yêu cầu luồng hai hướng và cả hai phía đều có P và S.



Hình 5.18 Các giao tiếp lớp giao thức continuous RQ.

Trước hết chúng ta có thể nhận dạng các giao tiếp khác nhau liên quan đến mỗi thực thể giao thức trong hình 5.18. Chiến lược kiểm soát lỗi go-back-N, cùng với một cơ cấu điều khiển luồng theo phương pháp cửa sổ trượt, được dùng như là cơ sở của hai đặc tả được trình bày trên các hình 5.19 và 5.20. Sơ cấp được đặc tả trong hình 5.19 và thứ cấp được đặc tả trong hình 5.20.

Trong cả hai đặc tả, các tên viết tắt dùng ở đây được định nghĩa trong phần (a) của hình. Các đặc tả của mỗi giao thức sơ cấp và thứ cấp dùng 3 phương pháp được trình bày trong phần (b), (c) và (d). Từ mỗi đặc tả chúng ta có thể thấy, mặc dù các sơ đồ chuyển trạng thái đã cung cấp một tổng quan khá tốt về hoạt động cơ bản của mỗi giao thức, nhưng các khía cạnh có thể trong hoạt động của chúng không thể được miêu tả tất cả bằng phương pháp này.

Các bảng sự kiện-trạng thái mở rộng cung cấp nhiều chi tiết hơn bao gồm các hàm luận lý có thể dùng trong đặc tả. Đặc tả của mỗi giao thức theo mã có cấu trúc có thể được thực hiện theo một phương pháp có hệ thống. Kết quả đặc tả theo mã giả được trình bày trong phần (d) của hình 5.19 và 5.20. Trong các chương tiếp theo chúng ta sẽ có các bảng sự kiện-trạng thái mở rộng cho các giao thức khác nhau đã được đề cập đến, hay các đặc tính thiết yếu của chúng. Bạn đọc có thể ứng dụng phương pháp này để suy ra mã cấu trúc tương ứng.



(a) Các sự kiện đến

Tên	Giao tiếp	Ý nghĩa
LDTAreq	LS-user	Nhận L-DATA.request
ACKRCVD	PHY-provider	Nhận ACK-frame từ S
TEXP	TIM-provider	Bộ định thời chờ ACK cho frame hết hạn
NAKRCVD	PHY-provider	Nhận NAK-frame từ S

Các trạng thái

Tên	Giao tiếp
DATA	Data transfer

Các sự kiện đi

Tên	Giao tiếp	Ý nghĩa
TxFrame(X)	PHY-user	Định dạng một frame với N(S) = X và truyền đến đuôi của hàng đợi PHY-user
RetxFrame(X)	PHY-user	Chuyển frame X đến đuôi của hàng đợi PHY-user

Các thuộc tính quyết định

Tên	Ý nghĩa
PO	Mô cửa sổ truyền

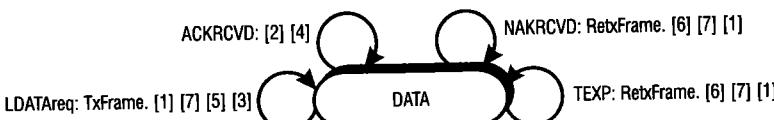
Các hành động

- [1] = Start-timer (X) dùng TIM-user queue
- [2] = Stop-timer (X) dùng TIM-user queue
- [3] = Tăng Va
- [4] = Giảm Va
- [5] = Tăng Vs
- [6] = Lấy frame từ đầu của danh sách truyền lại
- [7] = Đặt frame vào đuôi của danh sách truyền lại

Các hành động

- Vx = Biến tuần tự truyền
- PresentState = Trạng thái hiện tại của thực thể giao thức
- Va = Số lượng frame chưa báo nhận trong danh sách truyền lại

(b)



(c)

Sự kiện đến	LDTAreq	ACKRCVD	TEXP	NAKRCVD
Trạng thái hiện tại	1	2	3	4
DATA				

1 = PO: TxFrame. [1] [7] [5] [3]

2 = [2] [4]

3 = RetxFrame. [6] [7] [1]

4 = RetxFrame. [6] [7] [1]

Hình 5.19 Đặc tả continuous RQ (go-back-N): (a) Các tên viết tắt
(b) Sơ đồ chuyển trạng thái (c) Bảng sự kiện-trạng thái mở rộng

(d) *program Continuous RQ_Primary;*

const K; (giới hạn cửa sổ truyền)

type Events=(LDAReq, ACKRCVD, TEXP, NAKRCVD);

States=DATA;

var EventStateTable=array[Events, States] of 1..4;

EventType:Events;

PresentState:States;

Vs=0..K;

Va=0..K-1;

RextList; (hàng đợi FIFO giữ các con trỏ của bộ đệm I-frame đang đợi báo nhận)

procedure Initialize; (khởi động nội dung cho biến EventStateTable và các biến trạng thái)

procedure TxFrame(X:integer);

procedure RetxFrame(X:integer);

procedure Start_timer(X:integer);

procedure Stop_timer(X:integer);

procedure Get_frame;

procedure Put_frame;

function P0:boolean;

begin Initialize;

repeat đợi một sự kiện đến;

EventType:= loại sự kiện đến;

Case EventStateTable[EventType, PresentState] of

1: begin if P0 then begin TxFrame(Vs); Start_timer(Vs); Put_frame;
Vs:=Vs+1; Va:=Va+1;
if Va=K then P0:=false end;
end;

2: repeat X:=N(R) từ frame; Stop_timer(X);

Va:=Va-1; P0:=true;

Until X=N(R) trong ACK-frame;

3: begin Get_frame; X:=N(S) từ frame; RetxFrame(X);

Start_timer(X); Put_frame; end;

4: repeat Get_frame; X:=N(S) từ frame; RetxFrame(X);

Start_timer(X); Put_frame;

Until X=N(R) trong NAK-frame;

Until mãi mãi

End.

Hình 5.19 (tiếp theo) Đặc tả continuous RQ (go-back-N):

(d) Mã giả có cấu trúc.

(a) Các sự kiện đến

Tên	Giao tiếp	Ý nghĩa
IRCVD	PHY-provider	Nhận I-frame từ P

Các trạng thái

Tên	Giao tiếp
DATA	Đợi I-frame kề với N(S) = Vr
NAK-SENT	Đợi I-frame mất với N(S) = Vr

Các sự kiện đi

Tên	Giao tiếp	Ý nghĩa
LDADataInd(X)	LS-provider	Chuyển nội dung của I-frame có N(S) = X đến lớp trên
TxACK(X)	PHY-user	Định dạng và truyền một ACK-frame với N(R) = X
TxNAK(X)	PHY-user	Định dạng và truyền một NAK-frame với N(R) = X
LERRORInd	LS-provider	Phát ra thông báo lỗi

Các thuộc tính quyết định

Tên	Ý nghĩa
PO	N(S) trong I-frame bằng Vr

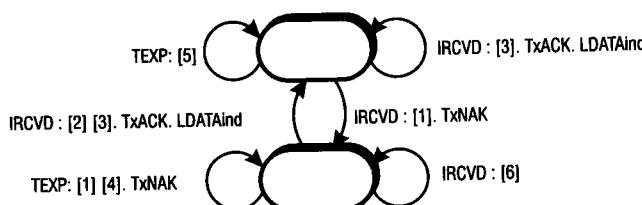
Các hành động

- [1] = Start-timer
- [2] = Stop-timer
- [3] = Tăng Vr
- [4] = Tăng RetxBCount
- [5] = Tăng ErrorCount
- [6] = Loại bỏ frame

Các hành động

- Vr = Biến tuần tự nhận
- RetxBCount = Số NAK-frames được truyền lại
- ErrorCount = Số điều kiện lỗi
- PresentState = Trạng thái hiện tại của thực thể giao thức

(b)



(c)

Sự kiện Trang thái hiện tại	IRCVD	TEXP
DATA	1	0
NAK-SENT	2	3

0 = [5]

1 = PO:TxACK. LDADataInd. [3]

NOT PO: TxNAK. [1]. NAK-SENT

2 = PO:TxACK. LDADataInd. [2] [3]. DATA

NOT PO: [6]

3 = TxNAK. [1] [4]

Hình 5.20 Đặc tả continuous RQ (go-back-N)-thứ cấp: (a) Các tên viết tắt
 (b) Sơ đồ chuyển trạng thái (c) Bảng sự kiện-trạng thái mở rộng

(d) *program Continuous RQ_Secondary;*

const MaxRetxCount;

MaxErrorCount;

type Events=(IRCVD,TEXP);

States=(DATA,NAK_SENT);

var EventStateTable=array[Events,States] of 0..3;

EventType:Events;

PresentState:States;

Vr,RetxCount,ErrorCount: integer;

procedure Initialize; /khởi động nội dung cho biến EventStateTable và các biến trạng thái/

procedure LDATAind(X);

procedure TxACK(X);

procedure TxNAK(X);

procedure LERRORind;;

procedure Start_timer;

procedure Stop_timer;

procedure DiscardFrame;;

function P0:boolean;

begin Initialize;

repeat đợi một sự kiện đến;

EventType:=loại sự kiện;

Case EventStateTable[EventType,PresentState] of

0: begin ErrorCount:=ErrorCount-1;

if (ErrorCount=MaxErrorCount) then

begin LERRORind; Initialize;end;

end;

1: X:=N(S) từ I-frame nhận được;

begin if P0 then begin TxACK(X); LDATAind(X); Vr:=Vr+1; end else begin

TxNAK(X); Start_timer; PresentState:=NAK_SENT;end;end;

Hình 5.20 (tiếp theo) Đặc tả continuous RQ (go-back-N)-thứ cấp:

(d) Mô giả có cấu trúc.



2: $X:=N(S)$ từ I-frame nhận được;
 begin if $P0$ then begin $TxACK(X)$; $LDATAind(X)$; $Vr:=Vr+1$; $Stop_timer$; end
 else $DiscardFrame$; end;

3: begin $TxNAK(Vr)$; $Start_timer$;
 $RetxCount:=RetxCount + 1$;
 If ($RetxCount = MaxRetxCount$) then begin $LERRORind$; $Initialize$; end; end;
 Until māi māi;
 End.

Hình 5.20 (tiếp theo) Đặc tả continuous RQ (go-back-N)-thứ cấp:
(d) Mă giả có cấu trúc.

3.6. Hiệu suất sử dụng liên kết

Trong khi thảo luận giao thức idle RQ, chúng ta đã thấy hiệu suất sử dụng liên kết U là một hàm theo thời gian truyền một frame T_{ix} và thời gian trễ lan truyền T_p của liên kết. Tuy nhiên, trong các liên kết với T_p lớn hơn T_{ix} thì hiệu suất sử dụng liên kết cũng bị chi phối bởi cửa sổ truyền K .

Trở lại ví dụ trong hình 5.10 trình bày một liên kết vệ tinh tiêu biểu, T_p thường lớn hơn T_{ix} . Trong ví dụ này T_{ix} là 1 ms và T_p khoảng 250ms đối với một kênh 1Mbps và frame chứa 1000bit. Do đó theo lý thuyết, tại bất cứ một thời điểm nào đều có thể truyền 250 frame giữa P và S. Để đạt được hiệu suất liên kết là 100%, chúng ta cần một cửa sổ truyền lớn hơn 500, vì sẽ không nhận được một ACK-frame cho I-frame đầu tiên theo tuần tự mãi cho đến 500ms ($2T_p$) sau.

Tổng quát, hiệu suất liên kết U với một cửa sổ truyền K được cho bởi:

$$U=1$$

Nếu K lớn hơn hay bằng $1+2a$ và:

$$U=KT_{ix}/(T_{ix}+2T_p)=K/(1+2T_p/T_{ix})=K/(1+2a)$$

Nếu K nhỏ hơn $1+2a$.

Chúng ta có thể kiểm tra lại điều này khi $T_p=T_{ix}$. Trong trường hợp này, bit cuối cùng của một frame được P gửi phải sau thời gian $2T_p$ (và do đó $2T_{ix}$) thì S mới có thể nhận được. ACK-frame liên quan phải mất thêm T_p nữa (và do đó T_{ix}) để P nhận. Nếu $K=1$ (idle RQ), thì $U=1/3$. Để tăng lên 100% (K lớn hơn $1+2a$), K phải lớn hơn 3, nghĩa là 3 frame hay nhiều hơn phải được P gửi trước khi nhận một ACK.

Ví dụ 2

Một chuỗi frame 1000 bit được truyền dùng một giao thức continuous RQ. Xác định hiệu suất liên kết cho các loại liên kết số liệu sau nếu tốc độ lan truyền là 2.10^8 ms^{-1} và tốc độ lối bit của các liên kết thấp không đáng kể:

- (a) Một liên kết dài 1km có tốc độ 1Mbps và cửa sổ truyền K=2
 (b) Một liên kết dài 10km có tốc độ 200Mbps và cửa sổ truyền K=7
 (c) Một liên kết vệ tinh 50 000km có tốc độ 2Mbps và cửa sổ truyền K=127

$$T_p = S/V; T_{ix} = N_i/R; \quad a = T_b/T_{ix}, N_i = \text{số bit thông tin}$$

Tóm tắt lời giải:

$$(a) T_p = 10^3 / 2 \cdot 10^8 = 5 \cdot 10^{-6} \text{ s}$$

$$T_{ix} = 1000 / 1 \cdot 10^6 = 10^{-3} \text{ s}$$

Do đó:

$$a = 5 \cdot 10^{-6} / 10^{-3} = 5 \cdot 10^{-3}$$

K=2 lớn hơn 1+2a và do đó U=1

$$(b) T_p = 10 \cdot 10^3 / 2 \cdot 10^8 = 5 \cdot 10^{-5} \text{ s}$$

$$T_{ix} = 1000 / 200 \cdot 10^6 = 5 \cdot 10^{-6} \text{ s}$$

Do đó:

$$a = 5 \cdot 10^{-5} / 5 \cdot 10^{-6} = 10$$

K=7 nhỏ hơn 1+2a và do đó

$$U = K / (1 + 2a) = 7 / (1 + 20) = 0,33$$

$$(c) T_p = 50 \cdot 10^6 / 2 \cdot 10^8 = 0,25 \text{ s}$$

$$T_{ix} = 1000 / 2 \cdot 10^6 = 5 \cdot 10^{-4} \text{ s}$$

Do đó:

$$a = 0,25 / 5 \cdot 10^{-4} = 500$$

K=127 nhỏ hơn 1+2a và do đó

$$U = K / (1 + 2a) = 127 / (1 + 1000) = 0,127$$

Từ kết quả của bài toán chúng ta có thể suy ra rằng việc chọn K có ảnh hưởng nhiều đến hiệu quả sử dụng liên kết trong các trường hợp nào đó. Như chúng ta có thể thấy, ngay cả với K=127, hiệu suất sử dụng liên kết của một liên kết vệ tinh vẫn còn rất thấp. Vì lý do này nên thường dùng K lớn. Điều này cũng được áp dụng đối với các liên kết mặt đất hoạt động với tốc độ cao.

Kết quả trong ví dụ trên được tính toán với giả sử không có lỗi truyền. Bất kỳ lỗi truyền nào xảy ra đều làm giảm hiệu suất liên kết vì vài frame phải được truyền lại. Tuy nhiên, ảnh hưởng này khác nhau rất ít giữa hai lược đồ truyền lại. Ví dụ, với lược đồ truyền lại có chọn lọc (selective repeat), U bị giảm chỉ do số lần truyền mỗi frame N_r , vì chỉ truyền lại những frame bị hỏng. Nếu gọi P_f là tốc độ lỗi frame của liên kết, giả sử các lỗi là ngẫu nhiên:

K^o có lỗi.

$$N_r = 1/(1-P_f)$$

Do đó giá trị của U được chỉnh lại:

$$U = K/N_r(1+2a) = K(1-P_f)/(1+2a)$$

Selective repeat.

với K nhỏ hơn $1+2a$. Với các giá trị của K lớn hơn hay bằng $1+2a$, chúng ta tính U chỉ đơn giản thay $K=1+2a$ vào biểu thức, vì $1+2a$ là số I-frame tối đa có thể truyền trước khi nhận một báo nhận trong khoảng thời gian $T_{ix} + 2T_p$. Do đó:

$$U = (1+2a)(1-P_f)/(1+2a) = 1-P_f$$

Với lược đồ go-back-N, hiệu suất liên kết bị giảm nhiều hơn vì nếu một I-frame bị hỏng, nhiều hơn một I-frame phải được truyền lại. Số lượng I-frame được truyền thêm được xác định bởi độ lớn của K liên quan đến $1+2a$.

Đối với K nhỏ hơn $1+2a$, số lần phải truyền lại ($K-1$) frame đơn giản là $P_f(K-1)$. Ứng với mỗi lần như vậy, phải trễ thêm $1+2a$. Do đó, biểu thức được điều chỉnh lại như sau:

$$U = K(1-P_f)/[(1+2a)+(1+2a)P_f(K-1)] = K(1-P_f)/(1+2a)[1+P_f(K-1)]$$

với K nhỏ hơn $1+2a$, và :

$$U = (1+2a)(1-P_f)/(1+2a)[1+P_f(K-1)] = (1-P_f)/[1+P_f(K-1)]$$

với K lớn hơn hay bằng $1+2a$.

Các công thức này chỉ là xấp xỉ và sẽ cho kết quả có ý nghĩa chỉ khi đảm bảo các giá trị xấp xỉ trước. Tuy nhiên, chúng đã cung cấp một mô tả tốt về mức phẩm chất mong đợi và phẩm chất liên quan đến mỗi phương pháp.

Ví dụ 3

Một chuỗi các frame nối tiếp, mỗi frame có 1000 bit được truyền qua một liên kết số liệu có chiều dài 100km, tốc độ liên kết là 20Mbps. Nếu liên kết có tốc độ lan truyền là $2 \cdot 10^8 \text{ ms}^{-1}$ và hệ số BER là $4 \cdot 10^{-5}$, hãy xác định hiệu suất liên kết khi dùng các giao thức liên kết sau:

(a) Idle RQ

(b) Truyền lại có chọn lựa (selective repeat) với cửa sổ truyền là 10.

(c) Truyền lại một nhó (go-back-N) với cửa sổ truyền là 10.

Tóm tắt lời giải:

$$T_p = S/V = 100 \cdot 10^3 / 2 \cdot 10^8 = 5 \cdot 10^{-4} \text{ s}$$

$$T_{ix} = N_i/R = 1000/20 \cdot 10^6 = 5 \cdot 10^{-5} \text{ s}$$

$$a = T_p/T_{ix} = 5 \cdot 10^{-4} / 5 \cdot 10^{-5} = 10$$

Do đó:

$$1 + 2a = 21$$

Với $P_f = N_i P = 1000 \cdot 4 \cdot 10^{-5} = 4 \cdot 10^{-2}$

Do đó:

$$\frac{1}{1 - P_f} = 96 \cdot 10^{-2}$$

$$(a) U = (1 - P_f) / (1 + 2a) = 96 \cdot 10^{-2} / 21 = 0,046$$

(b) Đối với K nhỏ hơn 1+2a

$$U = K(1 - P_f) / (1 + 2a) = 10 \cdot 96 \cdot 10^{-2} / 21 = 0,46$$

(c) Đối với K nhỏ hơn 1+2a

$$U = K(1 - P_f) / (1 + 2a)(1 + P_f(K - 1)) = 0,336$$

4. QUẢN LÝ LIÊN KẾT

Kiểm soát lỗi và điều khiển luồng cả hai đều liên quan đến truyền I-frame chính xác qua một đường thông tin không hoàn hảo. Trong các lược đồ đã được mô tả, chúng ta giả sử rằng cả hai đối tác thông tin đều được khởi động sao cho chúng sẵn sàng trao đổi thông tin. Ví dụ cả hai phía của liên kết phải khởi động với biến tuần tự truyền và biến tuần tự nhận như nhau trước khi bắt kỳ frame thông tin nào được truyền. Tổng quát, hoạt động này được gọi là sự khởi động hay pha thiết lập liên kết (link set-up phase). Thông thường sau khi tất cả các frame đã được trao đổi qua một liên kết, lại có pha xóa liên kết (link disconnection phase). Vì cả pha thiết lập liên kết và pha xóa liên kết không thực sự truyền dữ liệu của người dùng, nên chúng được gọi chung là *quản lý liên kết* (link management).

Trong một liên kết giữa một đầu cuối và một máy tính có cự ly tương đối ngắn, ví dụ như 20m, các chức năng quản lý liên kết có thể đạt được bằng cách trao đổi các tín hiệu trên các đường bắt tay thêm vào liên quan đến giao tiếp vật lý. Điều này được gọi là thủ tục bắt tay (handshake procedure). Một user muốn xúc tiến một cuộc hội thoại với máy tính, trước hết phải mở đầu cuối lên. Điều này khiến cho một trong các đường điều khiển được set ở mức tích cực (active) và chỉ cho máy tính biết rằng đầu cuối đang sẵn sàng truyền ký tự _đầu cuối số liệu sẵn sàng. Sau đó đầu cuối phải đợi cho đến khi máy tính đáp ứng, máy tính đáp ứng bằng cách thiết lập mức tích cực trên đường dây điều khiển tương ứng để chỉ cho đầu cuối biết rằng nó cũng đang sẵn sàng tiếp nhận ký tự. Hoạt động trao đổi ký tự có thể bắt đầu sau đó.

Khi cả hai thiết bị đều là máy tính và chúng đang cùng truyền frame qua một liên kết số liệu, liên kết được thiết lập khi giao thức mức liên kết trong mỗi máy tính trao đổi một tập các frame điều khiển và quản lý để thống nhất hành động. Trong ví dụ liên kết đầu cuối đến máy tính ở trên, liên kết được thiết lập khi user bật công tắc trên đầu cuối. Tuy nhiên, trong trường hợp liên kết máy tính đến máy tính, sự thiết lập liên kết thường được khởi động khi phần mềm lớp cao hơn (ví dụ một trình ứng dụng) trong một máy tính phát tín hiệu đến phần mềm (chương trình) truyền tin để chỉ ra yêu cầu muốn mở một cuộc hội thoại với một máy tính đầu xa. Thông thường điều này có lẽ là sự thực thi một câu lệnh hay



hàm thực thể (primitive) yêu cầu truyền trong chương trình ứng dụng, đến lượt mình hàm thực thể lại thỉnh cầu phần mềm truyền số liệu. Trong thực tế, phần mềm truyền số liệu được xây dựng từ một số các lớp giao thức riêng biệt, mỗi lớp chịu trách nhiệm thực hiện một chức năng đặc biệt trong toàn bộ công việc truyền tin. Trước khi truyền bất kỳ dữ liệu nào, mỗi lớp đều được khởi động. Ví dụ khởi động các hàm thực thể liên quan đến lớp liên kết số liệu được trình bày trên hình 5.21(a).

Như chúng ta có thể thấy, trước khi truyền bất kỳ dữ liệu nào, dùng dịch vụ L_DATA.request, LS_user gửi một hàm thực thể dịch vụ L_CONNECT.request đến lớp liên kết. Không giống như dịch vụ L_DATA, dịch vụ này được xem như là dịch vụ xác nhận (confirmed service) vì khi thực thể giao thức liên kết nguồn đã thiết lập một liên kết (cầu nối luận lý) với thực thể giao thức đích, nó gửi lại một hàm thực thể L_CONNECT.confirm đến LS_user nguồn. Lưu ý rằng điều này xác nhận một liên kết luận lý đã được thiết lập chỉ với thực thể giao thức đích chứ chưa phải với LS_user đích. Liên kết thực sự với LS_user đích chỉ xảy ra đối với trường hợp của các lớp giao thức cao hơn.

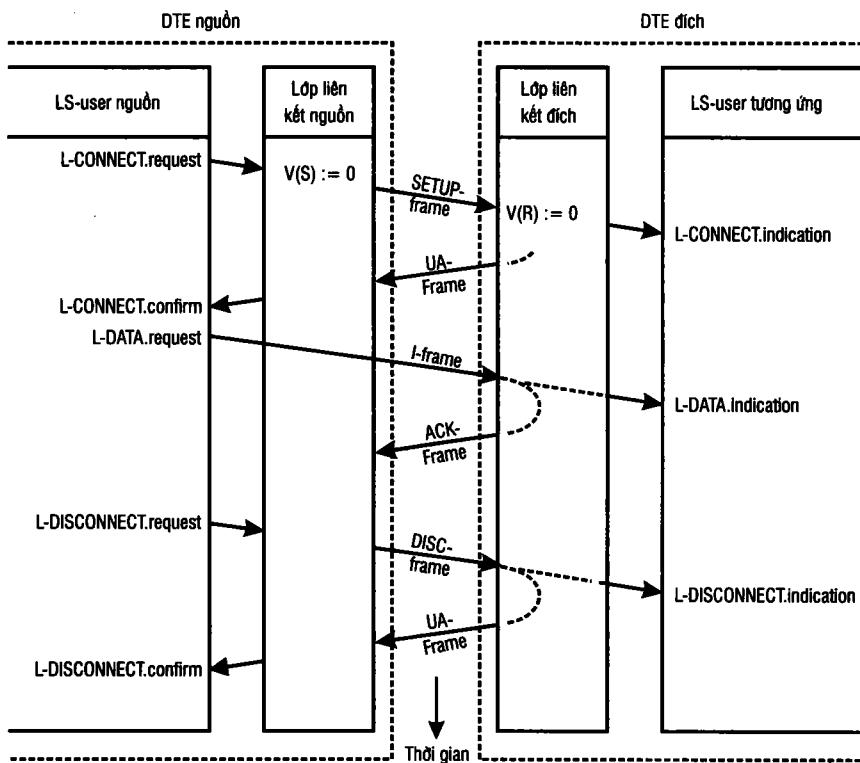
Khi nhận L_CONNECT.request, bằng cách dùng một ECB (xem mục 2), thực thể giao thức liên kết tại nguồn khởi động tất cả các biến trạng thái và sau đó tạo ra một frame thiết lập liên kết – link SETUP frame (PDU). Frame này được gửi đến thực thể giao thức liên kết tương ứng (ngang hàng) tại DTE đích theo chế độ truyền đã chọn. Khi nhận frame SETUP này, DTE đích khởi động các biến trạng thái của nó và xử lý bằng cách gửi một L_CONNECT.indication đến LS_user tương ứng và một frame báo nhận được gửi về nguồn.

Vì báo nhận này không liên hệ đến I-frame nào, nên nó không chứa số tuần tự. Do đó, nó được gọi là báo nhận không được đánh số (Unnumbered Acknowledgment) hay UA-frame. Khi nhận UA-frame này, thực thể giao thức nguồn phát ra L_CONNECT.confirm đến LS_user và lúc này liên kết đã sẵn sàng cho truyền số liệu thông qua dịch vụ L_DATA. Sau cùng, khi tất cả số liệu đều đã được truyền, liên kết được giải phóng bằng dịch vụ L_DISCONNECT, nó cũng là một dịch vụ được xác nhận. Frame tương ứng được gọi là DISC-frame, chúng được báo nhận bằng một UA-frame. Chế độ hoạt động này được gọi là chế độ tạo cầu nối (connection-oriented mode).

Rõ ràng, việc thêm chức năng quản lý liên kết sẽ ảnh hưởng đến các đặc tả giao thức idle RQ và Continuous RQ đã bàn đến trước đây. Để minh họa điều này, có một số điều chỉnh trên sơ đồ chuyển trạng thái của sơ cấp được trình bày trên hình 5.21(b).

Như chúng ta sẽ thấy, cần thêm ba trạng thái mới nữa; các sự kiện đến gây ra các chuyển trạng thái giữa các trạng thái mới này và trạng thái truyền số liệu được trình bày trong hình. Đối với sơ đồ chuyển trạng thái của thứ cấp có thể được mô tả theo cách tương tự. Hơn thế nữa, cấu trúc của bảng sự kiện-trạng thái rộng và cả mã giả tương ứng cũng được thêm các trạng thái mới này mà không bắt kỳ thay đổi lớn nào về cấu trúc của chúng.

(a)

**Hình 5.21 Quản lý liên kết:****(a) Sơ đồ tuần tự theo thời gian (b) Sơ đồ chuyển trạng thái.**

CHƯƠNG 6

CÁC GIAO THỨC ĐIỀU KHIỂN LIÊN KẾT SỐ LIỆU

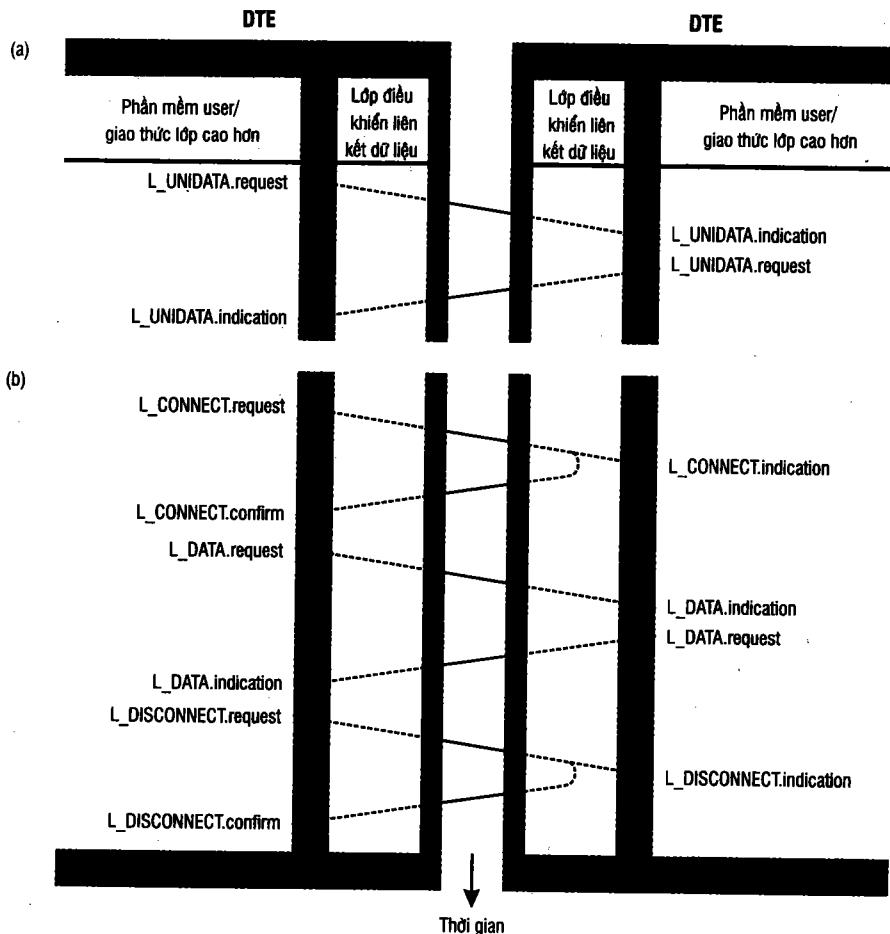
TỔNG QUAN

Lớp điều khiển liên kết số liệu liên quan đến việc chuyển thông tin số liệu qua một liên kết số liệu nối tiếp. Liên kết số liệu có thể là một kênh vật lý điểm-nối-diểm (dùng cáp xoắn, cáp đồng trực hay cáp quang) hoặc một kênh vô tuyến như liên kết vệ tinh hoặc một liên kết vật lý hay luận lý qua các mạng chuyển mạch. Chế độ truyền có thể là đồng bộ hay bất đồng bộ và dựa trên giao thức điều khiển truyền thiên hướng bit hay thiên hướng ký tự. Do đó lớp điều khiển liên kết số liệu là nền tảng hoạt động của tất cả các ứng dụng truyền số liệu và thường gọi tắt là lớp liên kết số liệu.

Trong các ứng dụng điểm-nối-diểm đơn giản, lớp liên kết số liệu đóng vai trò là lớp ứng dụng trực tiếp. Trong các ứng dụng phức tạp hơn, chẳng hạn như các ứng dụng thông qua các mạng chuyển mạch, lớp liên kết số liệu cung cấp một dịch vụ xác định cho tập các lớp giao thức mức cao hơn. Tùy thuộc vào ứng dụng, dịch vụ user được cung cấp bởi lớp liên kết số liệu có thể là dịch vụ không tạo cầu nối (connectionless) hay dịch vụ có tạo cầu nối (connection-oriented). Hai loại dịch vụ được trình bày trên sơ đồ tuần tự theo thời gian ở hình 6.1.

Dịch vụ không tạo cầu nối có nghĩa là cho dù có các bit kiểm tra để phát hiện lỗi, nhưng nếu phát hiện bất kỳ frame nào bị lỗi thì thực thể giao thức lớp liên kết chỉ làm một thao tác đơn giản là loại bỏ frame này. Dịch vụ này cũng được xem là dịch vụ không báo nhận và hoạt động truyền lại trở thành một chức năng hiển nhiên của một lớp giao thức cao hơn. Ví dụ, được thực hiện trong các ứng dụng dựa trên các mạng chuyển mạch trong đó tham số BER của các đường truyền dẫn rất thấp, do đó xác suất truyền lại nhỏ, chẳng hạn như trong các mạng LAN và ISDN.

Các hàm thực thể dịch vụ (service primitives) liên quan đến chế độ hoạt động có tạo cầu nối giống như các hàm thực thể đã được bàn đến trong chương 5. Nhớ lại rằng với loại dịch vụ này, giao thức liên kết số liệu dùng các thủ tục kiểm soát lỗi và điều khiển luồng để tạo ra dịch vụ tin cậy. Do đó xác suất số liệu không lỗi, không trùng khá cao và các thông điệp sẽ được phân phối theo thứ tự giống như khi được nạp vào để truyền đi. Để đạt được điều này, trước khi truyền bắt đầu một frame thông tin nào, một cầu nối luận lý giữa hai thực thể giao thức được thiết lập thông qua dịch vụ L_CONNECT. Tất cả số liệu được chuyển giao nhờ vào giao thức điều khiển luồng và truyền lại thích hợp. Khi tất cả số liệu đã được trao đổi, cầu nối luận lý bị xóa bằng dịch vụ L_DISCONNECT.

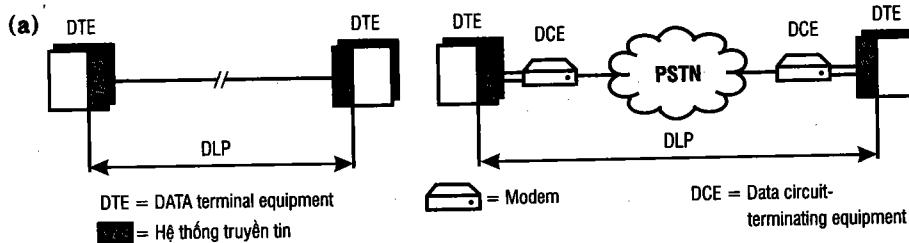


Hình 6.1 Các hàm thực thể dịch vụ lớp điều khiển liên kết dữ liệu:
(a) Không tạo cầu nối (b) Có tạo cầu nối.

Bởi dải ứng dụng của lớp liên kết số liệu khá rộng nên trước hết chúng ta sẽ xem xét vài môi trường ứng dụng khác nhau liên quan đến nó. Chúng ta sẽ xem xét hoạt động chi tiết của các giao thức khác nhau trong các mục tiếp theo.

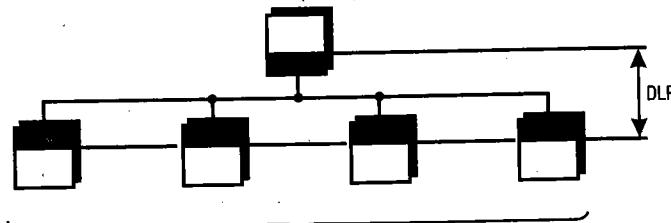
2. CÁC MÔI TRƯỜNG ỨNG DỤNG

Một vài môi trường ứng dụng được trình bày trên hình 6.2. Chúng ta có thể thấy rằng, trong vài trường hợp các giao thức liên kết số liệu tọa lạc ngay trong hai đầu cuối thông tin (DTE)_ví dụ như máy tính và tầm hoạt động của giao thức được xem như từ đầu cuối đến đầu cuối. Trong các trường hợp khác, giao thức hoạt động thông qua liên kết cục bộ, ví dụ liên kết nối DTE vào mạng. Trường hợp như vậy, ta nói giao thức chỉ có ý nghĩa cục bộ.



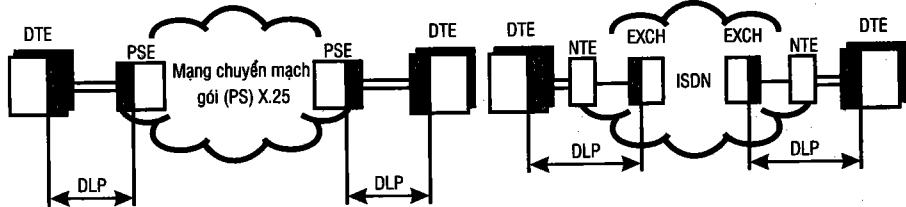
(b)

DTE chủ (master)



(c)

Các DTE từ (slave)



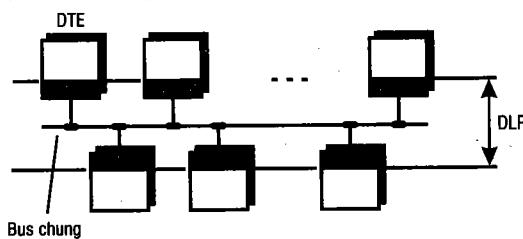
PS = Packet switching

PSE = Packet switching exchange

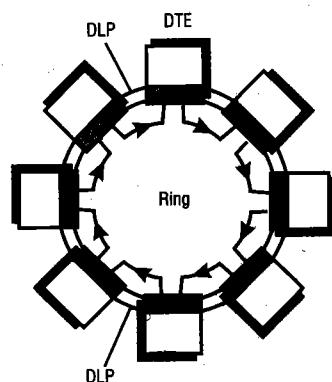
EXCH = Switching exchange

NTE = Network termination equipment

(d)



DLP = Data link protocol



Hình 6.2 Các môi trường ứng dụng giao thức liên kết dữ liệu:
(a) Điểm-nối-điểm (b) Đa điểm (c) Các WAN (d) Các LAN.



Trong hình 6.2(a), liên kết số liệu là một kênh điểm-nối-diểm, nó có thể là một kết nối vật lý trực tiếp (dùng cáp xoắn đôi, cáp đồng trục hay cáp quang), một kênh được thiết lập qua mạng điện thoại công cộng dùng modem, một kênh thông qua mạng ghép kênh tư nhân, hoặc một liên kết vô tuyến như liên kết vi ba mặt đất hay liên kết vệ tinh. Liên kết số liệu hoạt động trên cơ sở đầu cuối đến đầu cuối và trong nhiều ứng dụng như vậy, nó phục vụ cho ứng dụng một cách trực tiếp. Do đó, thường dùng dịch vụ theo hướng kết nối tin cậy.

Loại giao thức liên kết số liệu được dùng tùy vào khoảng cách hai đầu cuối thông tin và tốc độ bit của liên kết. Đối với các liên kết tốc độ thấp như liên kết dùng modem, thì giao thức hướng ký tự idle RQ được dùng. Các giao thức loại này ví dụ như Kermit và X-modem. Cả hai là các giao thức truyền tập tin đơn giản được dùng để mở rộng truyền tin giữa các máy tính cá nhân. Chúng rất giống với giao thức idle RQ được mô tả trong chương 5.

Đối với các liên kết tốc độ cao hơn và đặc biệt là các liên kết có cự ly xa như các liên kết vệ tinh hay các kênh xuyên qua các mạng ghép kênh tư nhân, một giao thức thuộc loại continuous RQ được gọi là HDLC (High-Level Data Link Control) được dùng. Đây là giao thức thiên hướng bit phù hợp với nhiều chế độ hoạt động khác nhau.

Kiến trúc ứng dụng được trình bày trên hình 6.2(b) được gọi là topo đa điểm. Như chúng ta thấy, có một đường dây truyền được gọi là bus được dùng để kết nối tất cả các máy tính lại với nhau. Do đó chúng ta phải đảm bảo rằng tất cả các hoạt động truyền đều được thực hiện theo một phương pháp có kiểm soát và không bao giờ có hai hoạt động truyền xảy ra đồng thời. Các kiến trúc như vậy thường được dùng trong các ứng dụng liên quan đến mô hình thông tin máy tính chủ/tớ (master/slave), trong đó có một máy tính chủ (master) kết nối với một nhóm phân tán các máy tính tớ (slave). Ví dụ như một máy tính chủ điều khiển một số các đầu cuối đặt phân tán tại các điểm bán hàng của một siêu thị hay máy tính quản lý trong một qui trình điều khiển một nhóm các trang thiết bị thông minh (dùng công nghệ máy tính) đặt phân tán trong một nhà máy. Tất cả các hoạt động truyền đều diễn ra giữa máy tính chủ và máy tính tớ đã chọn, vì vậy máy tính chủ điều khiển thứ tự của tất cả các hoạt động truyền.

Để điều khiển truy nhập vào môi trường truyền chia sẻ một cách bình đẳng, thường dùng một giao thức liên kết dữ liệu có tạo cầu nối. Các giao thức trước đây dùng cho các kiến trúc như vậy chủ yếu dựa vào sự phát triển một giao thức idle RQ thiên hướng ký tự được gọi là BSC (Binary Synchronous Control) hay bisync. Các hiện thực gần đây dựa vào một trong các chế độ hoạt động của giao thức HDLC thiên hướng bit được gọi là chế độ đáp ứng thông thường NRM (Normal Response Mode). Cả bisync và NRM đều hoạt động theo chế độ quét-chọn (poll-select); khi máy chủ muốn nhận dữ liệu từ một máy tớ, nó gửi cho máy tớ một thông điệp quét (poll message), và nếu nó muốn gửi dữ liệu đến cho máy tớ thì sẽ gửi cho máy tớ một thông điệp chọn (select message).

Hai kiến trúc được trình bày trên hình 6.2(c) đều liên hệ đến các ứng dụng có liên quan đến các mạng chuyển mạch điện rộng (WAN). Trong ví dụ đầu tiên, giao



thức liên kết chỉ có ý nghĩa cục bộ và chỉ hoạt động giữa DTE và DCE, như trong trường hợp một mạng chuyển mạch gói X.25. Tập giao thức X.25 được dùng trong các mạng như vậy chỉ áp dụng cho liên kết cục bộ giữa DTE và DCE. Giao thức liên kết số liệu dùng với X.25 cũng dẫn xuất từ HDLC, được gọi là LAP-B (Link Access Procedure_Balanced).

Kiến trúc thứ hai được dùng với các mạng số liệu chuyển mạch mạch (circuit-switched data networks) ví dụ như ISDN. Khi một mạch đã được thiết lập thông qua mạng, nó cung cấp một liên kết điểm-nối-diểm được xem như mạch ảo_cho cung đoạn chuyển dữ liệu. Giao thức này có thể có tạo cầu nối hay không tạo cầu nối, được gọi tương ứng là *chuyển frame* (frame switching) và *tiếp frame* (frame relay). Ngoài ra, thủ tục thiết lập cuộc gọi liên hệ với ISDN được thực hiện nhờ dùng liên kết riêng gọi là kênh báo hiệu hay kênh D. Thủ tục này dùng một giao thức liên kết là một dẫn xuất của HDLC được gọi là LAPD (Link Access Procedure D-channel).

Sau cùng, hai cấu hình được trình bày trên hình 6.2(d) liên quan đến các ứng dụng trên mạng cục bộ (LAN). Một đặc trưng của các mạng này là dùng các liên kết có tần số lõi bit thấp, cự ly ngắn và hoạt động với tốc độ bit cao (xấp xỉ 10Mbps). Kết quả là lõi hiếm khi xảy ra và thời gian chuyển frame giữa đầu cuối với đầu cuối diễn ra rất nhanh. Các mạng như vậy thường hoạt động theo chế độ không tạo cầu nối. Trong chế độ này tất cả các hoạt động truyền lại và các chức năng điều khiển luồng được giao cho một lớp giao thức cao hơn trong hai hệ thống đầu cuối. Giao thức liên kết được dùng với các LAN là một lớp con của HDLC được gọi là LLC (Logical Link Control).

Tóm lại, có một dải các giao thức liên kết số liệu, mỗi giao thức được thiết kế để dùng cho một môi trường ứng dụng đặc biệt.

3. CÁC GIAO THỨC THIỀN HƯỚNG KÝ TỰ

Các giao thức thiên hướng ký tự được dùng trong các ứng dụng điểm-nối-diểm và cả đa điểm. Đặc trưng của các giao thức này là dùng các ký tự điều khiển truyền để thực hiện các chức năng điều khiển truyền liên quan đến quản lý liên kết, đánh dấu đầu và cuối frame, kiểm soát lỗi và 'trong suốt' dữ liệu. Trong suốt dữ liệu là chức năng đặc biệt nhằm ngăn chặn sự nhầm lẫn giữa dữ liệu và thông tin điều khiển.

Trong khi đề cập đến các giao thức thiên hướng ký tự ở chương 5, chúng ta đã xem xét một liên kết số liệu điểm-nối-diểm và một luồng frame đơn công (một chiều) để trình bày các khía cạnh khác nhau của các giao thức liên kết. Tuy nhiên, trong hầu hết các ứng dụng thực tế chúng ta phải mở rộng các khái niệm đã được giới thiệu để chấp nhận số liệu được trao đổi theo cả hai hướng. Tương tự, nếu như có nhiều hơn hai chủ thể truyền tham gia trong cấu hình đa điểm, chúng ta phải cần đến một phương pháp điều khiển truy nhập vào môi trường truyền chia sẻ. Chúng ta sẽ bàn đến các chủ điểm này khi khảo sát các giao thức khác nhau.

3.1. Các giao thức đơn công (simplex protocols)

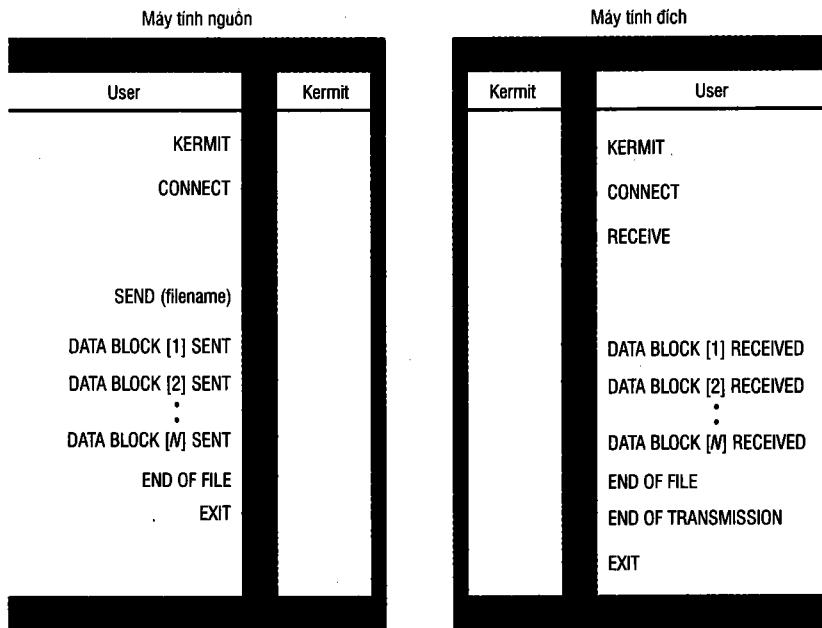
Lớp giao thức này là đơn giản nhất vì nó chỉ cho phép chuyển số liệu theo một hướng từ một máy tính (DTE) này đến một máy tính khác qua một liên kết số liệu điểm-nối-diểm. Nó được dùng với cấu hình trong hình 6.2(a). Một ứng dụng tiêu biểu là truyền tập tin dữ liệu từ máy tính này đến máy tính khác. Một trong những giao thức được dùng rộng rãi nhất là kermit.

Kermit được dùng rộng rãi để truyền nội dung của một hay nhiều tập tin từ một máy tính này đến máy tính kia thông qua một liên kết điểm-nối-diểm. Liên kết có thể là một kênh được thiết lập thông qua mạng điện thoại công cộng (chuyển mạch analog) sử dụng các modem hay một cặp dây xoắn đôi với các bộ điều khiển thu/phát thích hợp. Thường dùng truyền đồng bộ. Kermit là một ví dụ thực tế của giao thức idle RQ mà chúng ta đã mô tả trong chương 5.

Một số phiên bản của kermit cho phép nó truyền tập tin giữa hai máy tính cá nhân hoặc giữa một máy tính cá nhân với một máy tính server hay mainframe. Cơ cấu truyền tập tin cơ bản trong mỗi phiên bản là giống nhau. Các khác biệt chủ yếu là cách thức mà user của máy nguồn dùng chương trình kermit để truy nhập vào chương trình kermit ở máy tính đích ở thời điểm khởi đầu. Trước hết chúng ta sẽ xem xét phiên bản được dùng để truyền các tập tin giữa hai máy tính cá nhân.

Một tập lệnh đơn giản sẵn sàng cho cả hai user sau khi chương trình kermit đã được chạy ở cả hai hệ thống. Chúng được trình bày trong lược đồ tuần tự theo thời gian ở hình 6.3.

Nếu đang dùng modem thì một modem phải được đặt ở chế độ gọi và modem kia phải được đặt ở chế độ trả lời. Dĩ nhiên, cả hai modem phải được cài tốc độ hoạt động bằng nhau. Mỗi user chạy chương trình kermit và nhập lệnh CONNECT, lệnh này nếu thành công sẽ cho kết quả là một liên kết vật lý được thiết lập giữa hai hệ thống. Sau đó user trong hệ thống sẽ nhận tập tin nhập lệnh RECEIVE và user trong hệ thống truyền tập tin nhập vào lệnh SEND cùng với tên tập tin muốn truyền. Sau đó kermit trong hệ thống truyền sẽ chuyển các tập tin dưới dạng nguyên vẹn của chúng. Khi mỗi phân đoạn tập tin được truyền, một thông báo được xuất ra trên màn hình của cả hai user. Sau khi tất cả các phân đoạn của tập tin đã được truyền, cả hai user đều thoát ra khỏi kermit và trở về hệ điều hành cục bộ bằng lệnh EXIT. Để truyền tập tin theo hướng ngược lại, thứ tự của các lệnh được đảo lại giữa hai máy.



Hình 6.3 Các lệnh user của Kermit.

Chúng ta có thể thấy rằng kermit không đơn giản là một giao thức liên kết số liệu vì nó thực hiện một số các chức năng thêm vào như đọc ghi tập tin cũng như phân đoạn và tái thiết lập tập tin. Nó cũng có các loại frame (cũng gọi là gói) liên quan đến mỗi chức năng này như chúng ta có thể thấy từ dạng frame chuẩn ở hình 6.4(a).

Có hai khác biệt chính giữa dạng frame được dùng trong kermit và dạng frame được đề cập đến trong chương 5. Trước hết, một byte xác định chiều dài (length character) được dùng để chỉ chiều dài của mỗi frame thay vì dùng một ký tự điều khiển truyền ETX. Khác biệt thứ hai là các I-frame , ACK-frame và NAK-frame đều có cùng dạng cơ bản. Cũng có ký tự điều khiển dự phòng CR (carriage return) được dùng tại cuối của mỗi frame. Việc dùng ký tự chỉ chiều dài có lợi ích là nội dung frame có thể là dạng text hay dạng nhị phân vì máy thu chỉ làm công việc đơn giản là nhận và nối lại một số ký tự hay byte thích hợp (như đã được chỉ định trong phần header) khi tập tin đang được tái thiết lập. Thông thường user trong máy tính thu biết được dạng tập tin hay có thể suy ra nó từ tên tập tin.

Nội dung của tập tin dạng text được truyền theo tuần tự các khối 80 ký tự, mỗi khối được kết thúc bởi cặp ký tự CR/LF (carriage return/line feed). Tuy nhiên,

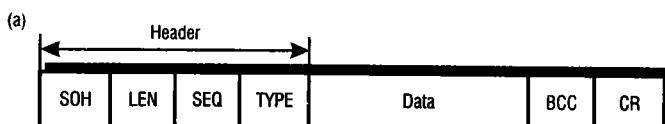
các tập tin nhị phân được truyền dưới dạng đơn giản hơn gồm một chuỗi các byte 8 bit. Bất cứ ký tự điều khiển định dạng nào nằm trong phần nội dung_text hay nhị phân_đều được mã hóa trước khi truyền nhằm đảm bảo không gây ảnh hưởng đến trạng thái của thiết bị thông tin trong khi truyền. Điều này là một đặc trưng của hoạt động điều khiển luồng trong một số modem. Mỗi ký tự điều khiển được phát hiện và được đổi thành tuần tự của hai ký tự in được bao gồm một ký tự tiền tố điều khiển _ # của ASCII kèm theo một ký tự ASCII có thể in được nằm cùng hàng và ở cột 4 hoặc cột 5 lần lượt tương ứng với cột 0 hoặc cột 1 trong bảng mã ASCII. Do đó Ctrl-A trở thành #A, CR trở thành #M và FS trở thành #\. Khi xuất hiện bất kỳ ký tự # nào, đều phải thêm một # phía trước.

Tuần tự trao đổi các frame bởi các thực thể giao thức của kermit để truyền một tập tin được trình bày trên hình 5.4(b). Frame được gửi trước tiên để khởi động truyền tập tin là frame gửi lời mời (S). Nó bao gồm một danh sách các tham số liên quan đến giao thức, như chiều dài frame tối đa và khoảng thời gian bắt khả dụng_timeout được dùng để truyền lại. Máy thu phục đáp bằng một frame chấp nhận (Y) với các tham số điều khiển truyền đã được thống nhất.

Kế tiếp máy phát xử lý truyền nội dung tập tin. Trước hết, một frame đầu tập tin có chứa tên tập tin được truyền, tiếp theo là tuần tự các frame dữ liệu (D) chứa nội dung của tập tin. Sau khi frame dữ liệu cuối cùng của tập tin đã được truyền, máy thu được thông báo bằng một frame kết thúc tập tin (Z). Sau đó các tập tin khác có thể được truyền theo cách tương tự. Cuối cùng, khi tất cả các tập tin đã được truyền, máy nguồn gửi một frame kết thúc giao tác (B) cho máy thu.

Kermit là một giao thức idle RQ. Do đó, sau khi truyền mỗi I-frame (I), máy phát đợi cho đến khi nhận được frame báo nhận (Y)_kiểm tra tổng khối đúng hoặc một frame từ chối (N)_BCC sai. Để dự phòng trường hợp các frame này bị hỏng, một bộ định thời được khởi động mỗi khi truyền một frame mới. Chỉ số tuần tự truyền trong mỗi I-frame tăng lên từng đơn vị theo modulo-64 và chỉ số tuần tự thu trong mỗi ACK-frame (Y) và NAK-frame (N) trùng với chỉ số tuần tự trong I-frame mà nó báo nhận hay từ chối.

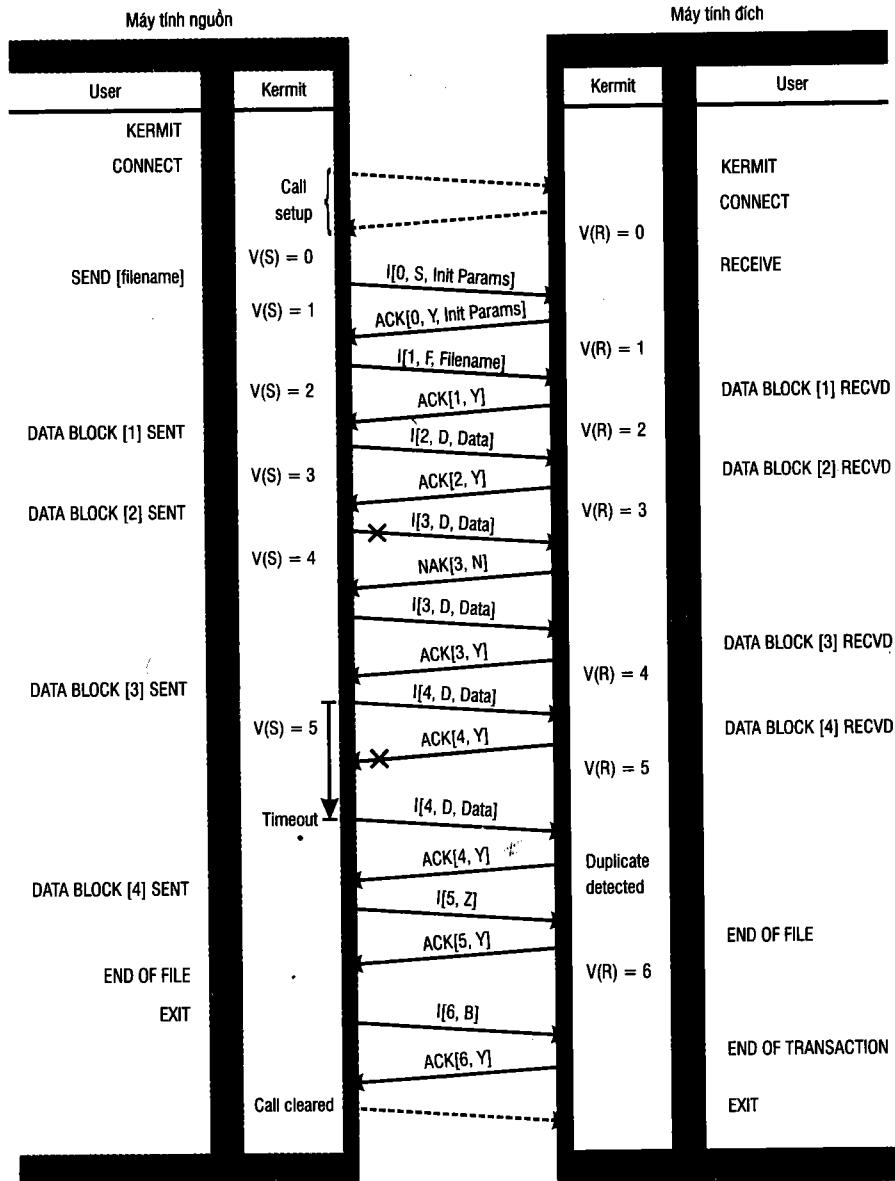
Các đặc trưng mà chúng ta vừa thảo luận là đặc trưng tối thiểu liên quan đến kermit. Chúng ta sẽ khảo sát thêm, chi tiết hơn về kermit trong mục 3.4.



Hình 6.4 Hoạt động của kermit (a) Dạng frame



(b)



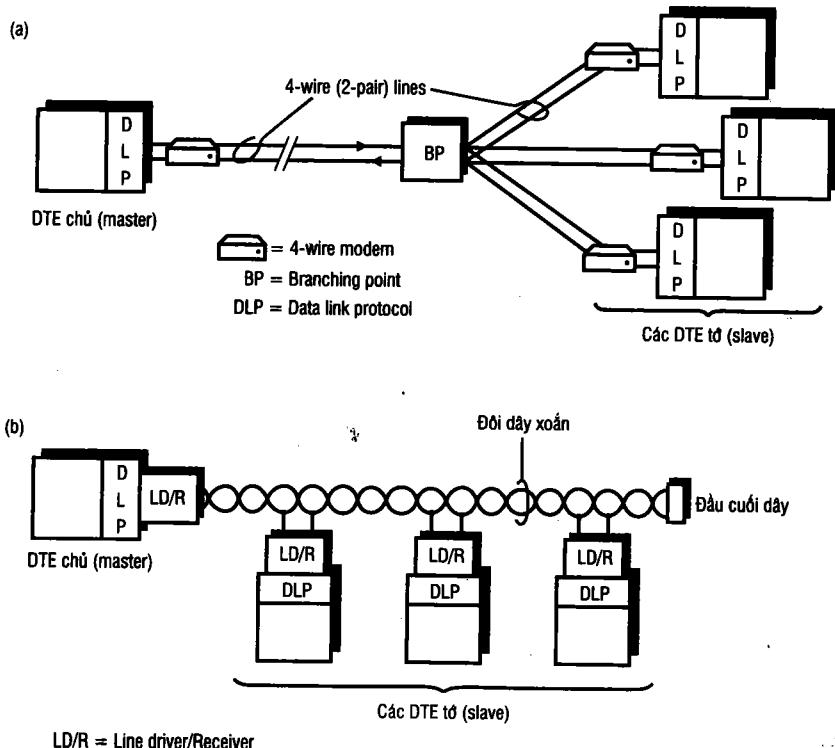
Hình 6.4 (tiếp theo) Hoạt động của kermit (b) Tuần tự frame.

3.2. Các giao thức bán song công

Hầu hết các giao thức thiêng kỵ tự đều hoạt động theo idle RQ, bán song công. Các nhà máy sản xuất máy tính lớn thường có các phiên bản riêng hơi

khác một tí. Có lẽ nổi tiếng nhất là một phiên bản được phát triển bởi IBM được gọi là *điều khiển đồng bộ nhị phân*, thường nói tắt là bisync hay BSC (Binary Synchronous Control). Vì nó là nền tảng của giao thức thiên hướng ký tự của ISO gọi là chế độ cơ bản (basic mode), nên chúng ta sẽ dùng BSC như là một ví dụ.

Như bao hàm trong tên của nó, BSC thường được dùng trong các lược đồ điều khiển truyền đồng bộ. Nó là giao thức có tạo cầu nối (connection-oriented) và được dùng chủ yếu trong các ứng dụng đa điểm, trong đó có một trạm (máy tính) chủ điều khiển tất cả các thông điệp truyền đến và đi từ một nhóm các trạm phụ thuộc (trạm tớ). Các trạm phụ thuộc được kết nối đến trạm chủ bằng các mạng đa điểm nếu tất cả các trạm đều tọa lạc tại những địa điểm cách xa nhau và dùng các modem, hoặc được kết nối đến trạm chủ thông qua mạng bus đa điểm nếu tất cả các trạm đều ở cùng một địa điểm và dùng các bộ thu/phát. Hai cấu hình được trình bày trên hình 6.5.



Hình 6.5 Các mạng thiên hướng ký tự: (a) Đa điểm (b) Bus đa điểm.

3.2.1. Các dạng frame

Để thực hiện các chức năng khác nhau liên quan đến quản lý liên kết, cần dùng thêm các frame điều khiển bên cạnh các frame mang thông tin. Ngoài ra, đối với truyền đồng bộ thiên hướng ký tự, máy thu cần phải đạt cho được cả đồng bộ ký tự và đồng bộ frame.

**Bảng 6.1 Các ký tự điều khiển được dùng với BSC**

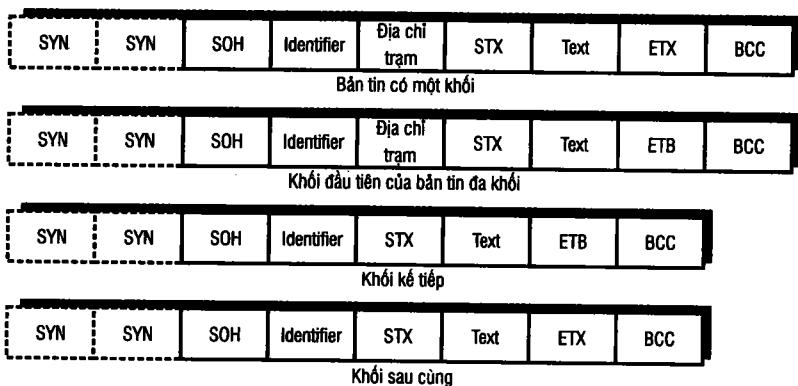
Ký tự	Chức năng
SOH(TC1)	Start of header: được dùng để chỉ bắt đầu của phần header trong một frame thông tin.
STX(TC2)	Start of text: được dùng với hai mục đích là kết thúc phần header và đánh dấu đầu của phần text
ETX(TC3)	End of text: được dùng để kết thúc một chuỗi text
EOT(TC4)	End of transmission: được dùng để chỉ định kết thúc hoạt động truyền của một hay nhiều khối văn bản và xóa cầu nối.
ENQ(TC5)	Enquiry: được dùng như là yêu cầu phúc đáp từ một trạm ở xa_phúc đáp có thể bao gồm thuộc tính và/hoặc trạng thái của trạm
ACK(TC6)	Acknowledge: báo nhận được truyền đi bởi máy thu khi phúc đáp cho một thông điệp mà máy phát đã gửi đến.
DLE(TC7)	Data link escape: được dùng để thay đổi ý nghĩa của các ký tự điều khiển truyền khác.
NAK(TC8)	Negative acknowledge: thông báo từ chối nhận, được gửi từ máy thu để phúc đáp cho một thông điệp được gửi tới từ máy phát nhưng không đạt yêu cầu.
SYN(TC9)	Synchronous idle: được dùng để cung cấp cho máy thu một phương tiện nhằm đạt được hay duy trì đồng bộ ký tự theo một lược đồ điều khiển truyền đồng bộ.
ETB(TC10)	End of transmission block: được dùng để chỉ điểm cuối của một khối dữ liệu khi một bản tin được chia thành một số khối.

Với BSC, các chức năng này được thực hiện dùng các ký tự điều khiển truyền EBCDIC hay ASCII/IA5 đã chọn. Chúng ta đã đề cập đến vai trò của vài ký tự điều khiển trong chương 5. Một danh sách đầy đủ hơn cùng với chức năng của chúng được trình bày trong bảng 6.1.

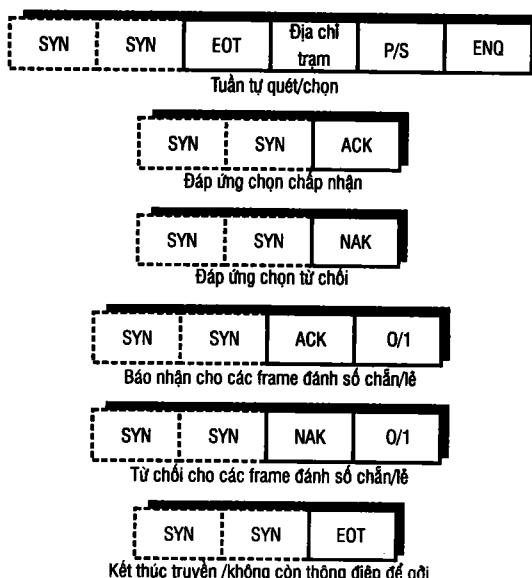
Các kiểu frame thông tin khác nhau_trong BSC được gọi là các khối dữ liệu (data block)_được trình bày trên hình 6.6(a). BSC dùng truyền đồng bộ thiên hướng ký tự và do đó tất cả các khối dữ liệu được truyền đều được dẫn đầu bởi ít nhất là hai ký tự SYN để máy thu đạt được sự đồng bộ ký tự. Các bản tin ngắn (ngắn hơn một chiều dài tối đa xác định) được truyền trong một khối đơn trong khi các bản tin dài hơn được truyền trong nhiều khối. Phần header _khi xuất hiện_ có mục đích tổng quát và thường định ra cách dịch phần dữ liệu như thế nào. Ngoài ra, đối với chế độ cơ bản còn có phần kiểm tra kiểu tổng đắt ngay sau ranh giới kết thúc khối (ETX hay ETB), nó là kiểm tra chẵn lẻ theo cột (xem chương 4). Hoạt động kiểm tra này bắt đầu từ ký tự STX cho đến ký tự định ranh

giới kết thúc khối. Vì kiểm tra chẵn lẻ chỉ có khả năng phát hiện lỗi, nên BSC thường dùng một mã CRC hai byte được tính theo CRC-16 thay cho BCC. Cả hai lược đồ đều giới hạn số ký tự trong mỗi khối dữ liệu truyền. Giới hạn này được xác định bởi tham số BER của liên kết đang được dùng, kích thước khối tối đa được chọn đảm bảo hầu hết các khối đều được nhận mà không bị lỗi. Các bản tin dài hơn được truyền thành một tuần tự gồm các khối ngắn hơn có kích thước cố định, mỗi khối được kết thúc bằng một ký tự điều khiển ETB. Khối dữ liệu sau cùng của tuần tự như vậy được kết thúc bằng một ký tự điều khiển ETX.

(a)



(b)



Hình 6.6 Các dạng frame của BSC : (a) Dữ liệu (b) Quản lý.

Các frame điều khiển khác nhau liên quan đến giao thức BSC được trình bày trên hình 6.6(b). Các ký tự điều khiển ACK và NAK có hai chức năng:



- Thông báo tình trạng nhận: ACK hay NAK được máy thu gửi lại cho máy phát trong khi đáp ứng một khối dữ liệu được truyền đến trước đó và do đó có chứa một chỉ số tuần tự.
- Phúc đáp cho một thông điệp điều khiển chọn: một ACK chỉ định rằng trạm được chọn có thể nhận một khối dữ liệu trong khi NAK chỉ ra sự từ chối từ trạm đã chọn.

Ký tự điều khiển ENQ được dùng trong cả hai frame điều khiển quét (poll) và chọn (select). Theo sau địa chỉ của trạm phụ thuộc (slave) được quét hay được chọn tương ứng là ký tự điều khiển P (poll) hay S (select), tiếp đến là ký tự ENQ.

Sau cùng là ký tự điều khiển EOT có hai chức năng:

- Đặt dấu hiệu kết thúc một tuần tự trao đổi bản tin hoàn chỉnh và xóa liên kết luận lý giữa hai chủ thể tham gia truyền tin.
- Cung cấp một phương tiện để trả liên kết về trạng thái nhàn rỗi (reset).

3.2.2. Sự trong suốt dữ liệu

Trong chương 3 Chúng ta đã đề cập đến việc dùng ký tự DLE để trong suốt dữ liệu khi truyền dữ liệu nhị phân thuần túy chứ không chỉ là các chuỗi ký tự. Về cơ bản thì các tuần tự của ký tự định dạng frame khác nhau trình bày trên hình 6.6 được hiệu chỉnh thành DLE/STX, DLE/ETX, v.v. Cũng vậy, bất cứ khi nào máy phát phát hiện một mẫu nhị phân tương ứng với một ký tự DLE trong phần nội dung, nó sẽ chèn thêm một ký tự DLE nữa. Máy thu sẽ thực hiện kiểm tra tương tự. Bất cứ khi nào máy thu phát hiện hai ký tự DLE liên tiếp, nó sẽ loại bỏ ký tự DLE chèn thêm trước khi chuyển dữ liệu đến nơi có chức năng xử lý tiếp. Một khác biệt nữa khi hoạt động trong chế độ trong suốt là kiểm soát lỗi. Thay vì kiểm tra chẵn lẻ ngang dọc 8 bit đơn giản cho một khối, chúng ta lại dùng mã đa thức phức tạp hơn, trong đó mỗi khối được kết thúc bởi một mã CRC 16 bit thay vì một BCC 8 bit.

3.2.3. Hoạt động của giao thức

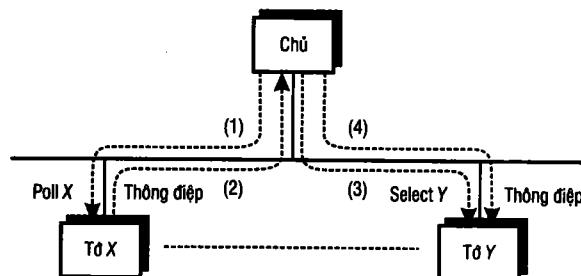
Như chúng ta đã thảo luận trước đây, máy tính chủ chịu trách nhiệm lập lịch cho tất cả các hoạt động truyền trên mỗi liên kết số liệu chia sẻ. Bản tin điều khiển quét được dùng để yêu cầu một máy phụ thuộc nào đó gửi bất kỳ số liệu đang đợi nào mà nó có; bản tin điều khiển chọn được dùng để hỏi máy phụ thuộc có sẵn sàng nhận số liệu hay không.

Hình 6.7(a) trình bày một tuần tự quét và chọn tiêu biểu. Một tuần tự của các frame trao đổi trên một đường dây đa nhánh được mô tả trên hình 6.7(b) và 6.7(c). Phần (b) trình bày cả tuần tự thành công và không thành công liên quan đến hoạt động chọn (select), trong khi phần (c) mô tả hai tuần tự liên quan đến hoạt động quét (poll).

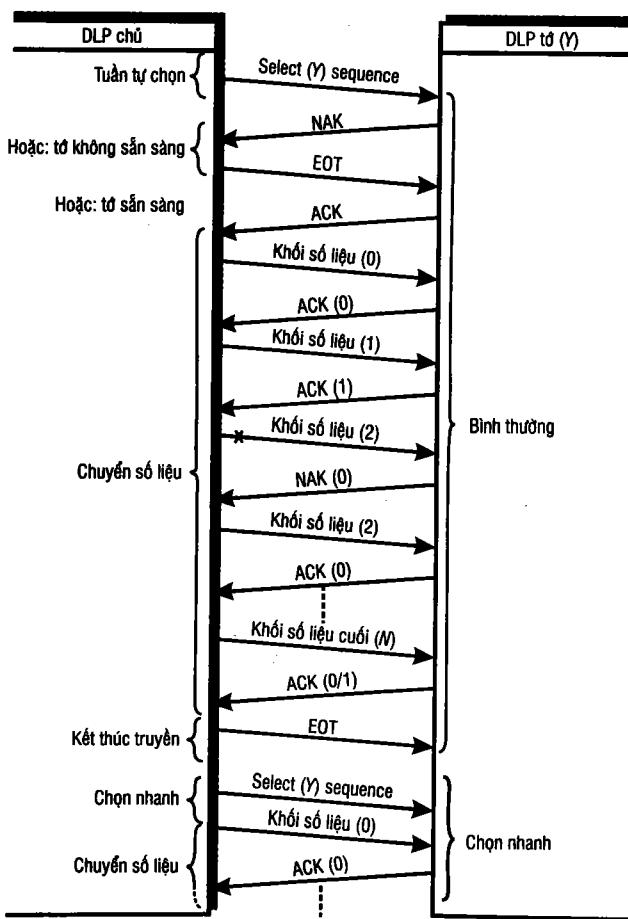
Để chọn một trạm phụ thuộc nào đó, trạm chủ gửi một bản tin điều khiển chọn ENQ, trong đó địa chỉ của trạm phụ thuộc được đặt trước ký tự ENQ. Giả sử trạm được chọn sẵn sàng nhận thông điệp, nó phúc đáp bằng một ACK. Sau đó trạm chủ gửi thông tin dưới dạng một khối số liệu đơn hoặc một tuần tự nhiều khối số liệu trong đó khối sau cùng được kết thúc bằng một ký tự ETX. Ngay khi

khối số liệu được nhận và lưu, trạm phụ thuộc sẽ tính toán lại tuần tự kiểm tra và giả sử không có lỗi nào xảy ra, nó sẽ đáp ứng bằng một ACK cho mỗi khối. Cuối cùng, sau khi bản tin đã được truyền hoàn tất, trạm chủ gửi một bản tin điều khiển EOT, nó có vai trò kết thúc hoạt động chuyển thông điệp và xóa cầu nối luận lý.

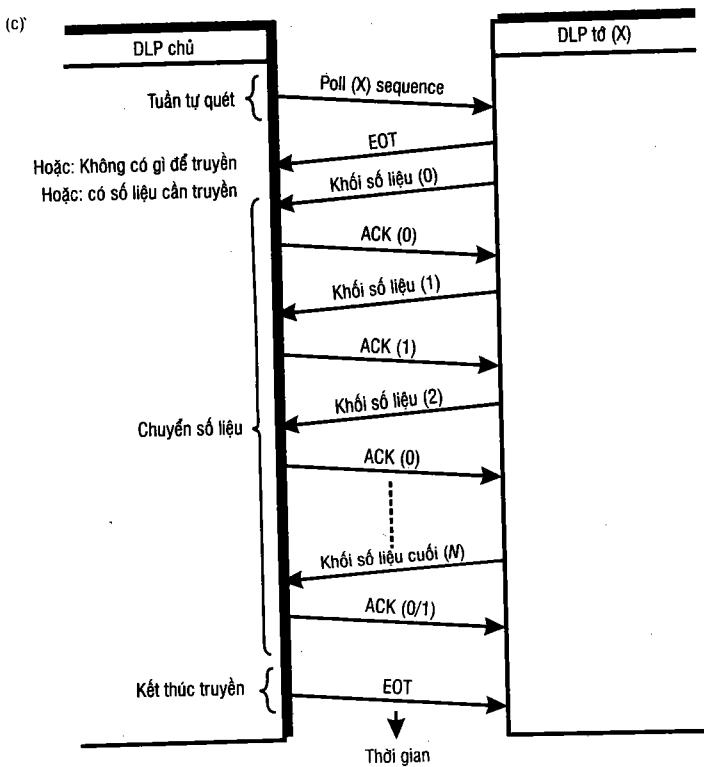
(a)



(b)



Hình 6.7 Các tuần tự frame của BSC: (a) Lược đồ quét-chọn (b) Chọn



Hình 6.7 (tiếp theo) Các tuần tự frame của BSC: (c) Quét.

Trong vài trường hợp, khi chọn một trạm không nhất thiết phải đợi một báo nhận cho thông điệp ENQ trước khi gửi một bản tin. Ví dụ, nếu một trạm đã được chọn trước đó và cầu nối luận lý chưa bị xóa. Trong trường hợp như vậy máy chủ gửi bản tin ngay sau thông điệp điều khiển chọn, không cần phải đợi một ACK (hay NAK). Điều này được gọi là **tuần tự chọn nhanh** (fast select sequence).

Trong pha quét, trước hết trạm chủ gửi một thông điệp điều khiển quét trong đó địa chỉ của trạm được quét ở trước ký tự ENQ. Sau đó, giả sử trạm được quét có một bản tin đang đợi truyền, nó đáp ứng bằng cách gửi bản tin này. Khi nhận khối dữ liệu, trạm chủ tính toán lại tuần tự kiểm tra, giả sử không có lỗi nó sẽ gửi thông báo chấp nhận (ACK). Cuối cùng, khi đã hoàn tất truyền bản tin và được báo nhận, cầu nối luận lý bị xóa bằng thông điệp điều khiển EOT.

Hình 6.7 mô tả BSC là một giao thức idle RQ bởi sau khi gửi một khối dữ liệu, máy phát phải đợi một ACK hoặc NAK trước khi gửi khối kế tiếp, và nếu nhận được NAK thì phải truyền lại khối bị hỏng. Việc dùng thông điệp điều khiển NAK đảm bảo rằng một khối dữ liệu bị hỏng sẽ được truyền lại khi nhận thông điệp NAK thay vì phải đợi sau khoảng thời gian timeout. Như chúng ta đã đề cập



đến ở chương 5, nếu khôi dữ liệu truyền bị hỏng hoàn toàn, một cơ cấu timeout được tăng cường nhằm đảm bảo cho sự truyền lại khôi bị hỏng này. Dùng danh định hay chỉ số tuần tự giúp máy thu có thể phát hiện dễ dàng sự trùng khôi dữ liệu.

Lưu ý rằng trong BSC chỉ số tuần tự gửi tăng lên từng đơn vị theo modulo một giá trị thống nhất trước, trong khi chỉ số tuần tự thu trong ACK hay NAK chỉ gia tăng theo modulo-2. Kết quả là chỉ số tuần tự thu 0 liên hệ đến các frame được đánh số chẵn trong khi 1 liên hệ đến các frame được đánh số lẻ. Vì chỉ có thể có một frame trùng, nên điều này cũng đủ cho phép máy thu phát hiện việc trùng này.

3.2.4. Giao tiếp user

Điều quan trọng là phải phân biệt sự khác nhau giữa các dịch vụ được cung cấp bởi lớp liên kết và hoạt động chi tiết của thực thể giao thức lớp liên kết. Để minh họa điều này, mối quan hệ giữa các dịch vụ user và các khôi thông điệp khác (dữ liệu và điều khiển) liên quan đến giao thức BSC được trình bày trên hình 6.8. Như chúng ta có thể thấy, thông điệp điều khiển chọn ban đầu được báo nhận và điều này tương đương như một xác nhận của trạm ở xa rằng sẵn sàng tiếp nhận dữ liệu. Tuy nhiên, đối với tuần tự quét (poll), không có ACK gửi lại trong việc đáp ứng cho thông điệp quét ban đầu và vì vậy hàm thực thể xác nhận phải được gọi bởi thực thể giao thức cục bộ sau khi thông điệp quét đã được truyền. Một thủ tục tương tự cũng được thực hiện khi xóa liên kết.

Cũng giống như giao thức kermit, BSC thực hiện một chức năng phân đoạn và tái thiết lập. Do đó, khi nhận một L_DATA.request (với tham số là bản tin), thực thể giao thức truyền phân đoạn bản tin thành một tuần tự các khôi dữ liệu để truyền. Tương tự, thực thể thu tái thiết lập các khôi thành một bản tin hoàn chỉnh trước khi chuyển nó đến lớp user cao hơn bằng hàm thực thể L_DATA.indication.

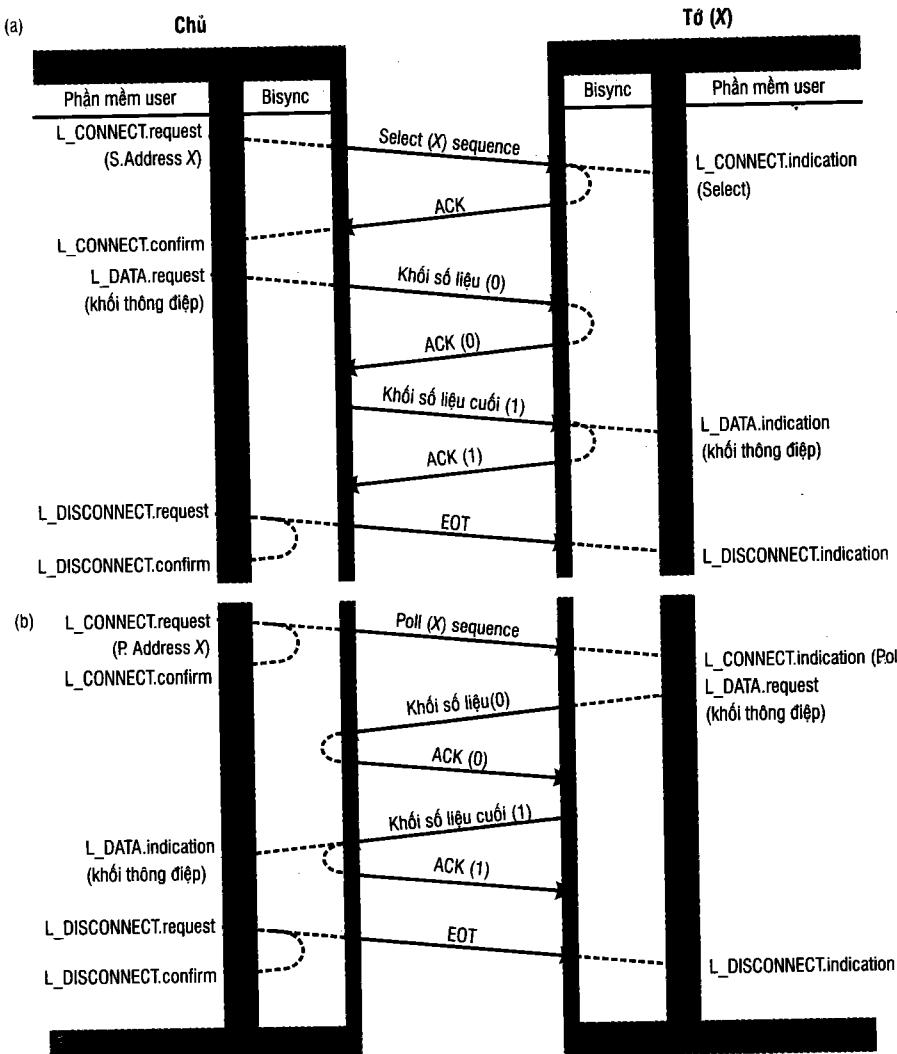
Vì BSC là một giao thức bán song công hiệu quả, nên BSC không thể khai thác chế độ thông tin song công hoàn toàn ngay cả khi được hỗ trợ bởi liên kết vật lý. Tuy nhiên, vì nhu cầu bộ đệm của nó là tối thiểu nên BSC vẫn còn được dùng rộng rãi trong các mạng mà kích thước bộ đệm là một tiêu chí cần xem xét. Tuy vậy, trong những năm gần đây, đã có xu thế tiếp cận các giao thức thiên hướng bit hiệu quả và linh hoạt hơn. Đó là trường hợp của các mạng máy tính, nơi yêu cầu cao về hoạt động trong suốt.

3.2.5. Hiệu suất của giao thức

Chúng ta đã thảo luận về hiệu suất liên kết đạt được đối với giao thức idle RQ trong chương 5. Tuy nhiên, ở đây chúng ta lại tiếp tục với đề tài này bởi mục tiêu chủ yếu trong việc sử dụng BSC là dùng cho các ứng dụng trong đó có một trạm chủ gửi và nhận dữ liệu từ nhiều trạm phụ thuộc. Khi đó còn có một tham số đánh giá hiệu suất khá quan trọng nữa là thời gian trung bình để quét hoặc chọn tất cả các trạm phụ thuộc trên liên kết.



Trong thực tế, vì idle RQ có hiệu suất liên kết thấp hơn so với continuous RQ, nên giao thức idle RQ được dùng chủ yếu trong các liên kết đa nhánh (multidrop) hoạt động với tốc độ đến 64Kbps (giả sử). Trong các liên kết như vậy, thời gian cần truyền một bản tin sẽ là khoảng thời gian nào vượt trội trong tuần tự quét hay chọn. Ví dụ, nếu một bản tin trung bình là 1000 bit và tốc độ liên kết là 10Kbps, thì thời gian cần truyền một bản tin là 0,1s. Ngược lại, thông điệp điều khiển liên quan đến tuần tự quét hay chọn là ngắn (giả sử 30bit), vì vậy thời gian cần truyền các thông điệp này cũng ngắn (0,003s đối với tốc độ 10kbps). Ngay cả tính luôn khoảng thời gian 0,001s để xử lý các thông điệp này thì tổng thời gian cho mỗi tuần tự quét hay chọn (0,004s) cũng rất nhỏ so với thời gian truyền bản tin.



Hình 6.8 Tương tác lớp liên kết/user: (a) Chọn (b) Quét.

Khi không có bản tin nào cần truyền, thời gian tối thiểu cần để quét tất cả các trạm phụ thuộc gấp N lần thời gian cần để quét một trạm, trong đó N là số trạm phụ thuộc tham gia vào liên kết. Khi có bản tin chuẩn bị truyền, thời gian trung bình quét tất cả các trạm phụ thuộc tăng lên và tùy vào tốc độ trung bình mà bản tin được phát ra. Thời gian quét lớn nhất xảy ra khi tốc độ trung bình phát bản tin đạt đến tốc độ bit của liên kết, vì quá tốc độ này liên kết trở nên quá tải và thời gian trễ tăng dần.

Tổng quát, thời gian trung bình để quét mỗi trạm thứ cấp có thể tính theo công thức:

$$T_{avr} = T_{min} / (1 - M_r T_{ix})$$

Trong đó T_{min} là thời gian tối thiểu để quét tất cả các trạm thứ cấp (phụ thuộc), M_r là tốc độ trung bình phát bản tin và T_{ix} là thời gian cần thiết để truyền một bản tin có kích thước trung bình. Nếu M_r là thấp so với T_{ix} , thì T_{avr} xấp xỉ T_{min} ; tuy nhiên khi M_r tăng thì T_{avr} sẽ tăng theo.

Ví dụ

Một giao thức BSC được dùng để điều khiển luồng thông điệp giữa một máy tính và 10 đầu cuối làm việc theo chế độ ký tự qua một liên kết số liệu đa điểm. Tốc độ dữ liệu là $R=10\text{Kbps}$ và chiều dài trung bình của thông điệp là $N_i = 1000$ bit. Nếu thông điệp quét và ACK liên hệ với nó là 30 bit và tổng thời gian để xử lý thông điệp này là 1ms, xác định thời gian trung bình mà mỗi đầu cuối sẽ được quét nếu tốc độ phát tin trung bình là:

- (a) 1 thông điệp trong một phút
- (b) 8 thông điệp trong một giây

Bỏ qua tỉ lệ lỗi bit (BER) và các khoảng thời gian trễ do lan truyền tín hiệu.

Tóm tắt lời giải :

Thời gian để truyền một thông điệp trung bình là:

$$T_{ix} = 1000/10^4 = 100\text{ms}$$

Thời gian truyền một thông điệp quét và ACK của nó là:

$$30/10^4 = 3\text{ms}$$

Thời gian quét một trạm thứ cấp là:

$$3 + 1 = 4\text{ms}$$

Thời gian tối thiểu để quét tất cả các trạm thứ cấp là:

$$T_{min} = 10 \times 4 = 40\text{ms}$$

Vậy $T_{avr} = T_{min} / (1 - M_r T_{ix})$

(a) $M_r = 1$ thông điệp trong một phút = $10^3/60$ thông điệp ms^{-1} . Do đó:

$$T_{avr} = 40/(1 - 10^{-3} \cdot 10^2/60) = 40\text{ms}$$

(b) $M_r = 8 \text{ thông điệp trong một giây} = 8 \cdot 10^3 \text{ thông điệp ms}^{-1}$. Do đó:

$$T_{avr} = 40 / (1 - 8 \cdot 10^3 \cdot 10^2) = 40/0,2 = 200\text{ms}$$

3.3. Các giao thức song công hoàn toàn

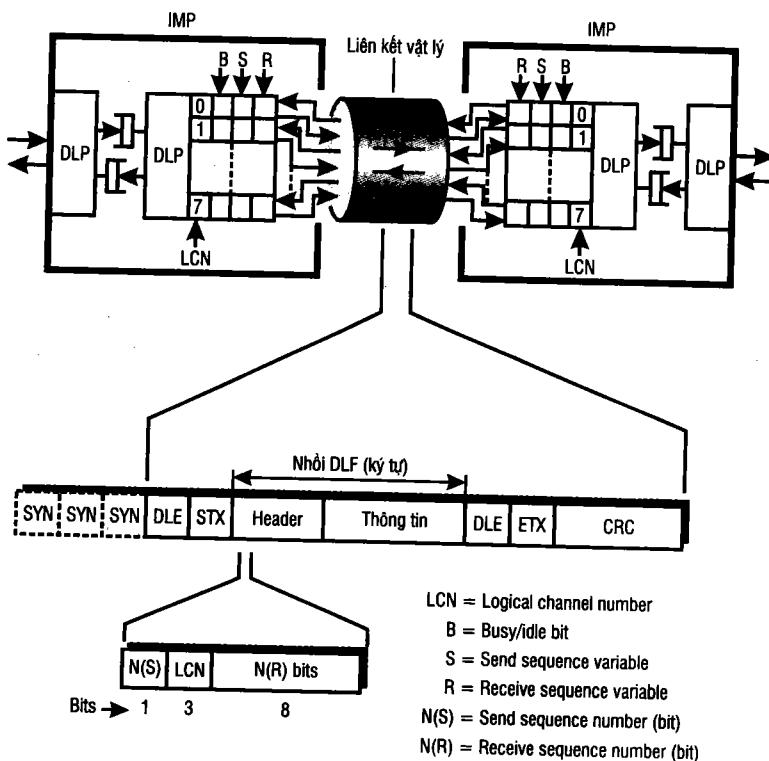
Có một vài giao thức thiên hướng ký tự hoạt động trong chế độ truyền tin song công hoàn toàn (full-duplex). Để minh họa chúng ta sẽ xem xét giao thức liên kết số liệu được dùng sớm nhất trong mạng ARPANET để điều khiển luồng frame thông tin xuyên qua các liên kết đến các node chuyển mạch trong một mạng, các node chuyển mạch như vậy cũng được gọi là các bộ xử lý thông điệp giao tiếp viết tắt bằng từ tiếng Anh là IMP (Interface Message Processor). Giao thức này hoạt động thông qua các liên kết song công điểm-nối-điểm kết nối hai node chuyển mạch với nhau.

Giao thức này giúp truyền các frame thông tin theo cả hai hướng một cách đồng thời và dùng lược đồ điều khiển truyền continuous RQ cho cả hai hướng. Giao thức hoạt động hiệu quả với cửa sổ truyền K=8 cho các liên kết mặt đất hoặc K=16 cho các liên kết vệ tinh. Để đảm bảo một luồng frame liên tục, cho phép 8 (hay 16 đối với vệ tinh) luồng thông tin dừng-và-chờ (stop-and-wait) riêng biệt tại bất cứ thời điểm nào trong tiến trình.

Để đạt được điều này, một liên kết vật lý được phép hoạt động như là 8 (hay 16) liên kết luận lý riêng biệt, luồng frame đi qua mỗi liên kết được điều khiển bởi cơ cấu giao thức dừng-và-chờ của nó. Chỉ số tuần tự truyền trong header của mỗi frame là hợp của hai field: một số tuần tự một bit _0 hay 1_ là chỉ số tuần tự truyền bình thường liên hệ với giao thức idle RQ, và chỉ số kênh luận lý LCN (logical Channel Number) chỉ định kênh luận lý mà frame được gán vào đó.

Tương tự như hầu hết các lược đồ song công hoàn toàn, với thiên hướng bit và thiên hướng ký tự, thông tin báo nhận liên hệ với luồng frame thông tin (I-frame) chạy theo một hướng được chứa trong phần header của frame thông tin chạy theo hướng ngược lại. Do đó chỉ dùng một loại frame và các field khác nhau trong phần header của nó liên hệ với chức năng đặc biệt. Dạng frame tổng quát và các field trong phần header liên quan đến giao thức liên kết số liệu được trình bày trên hình 6.9, cùng với hoạt động tổng quát của giao thức này.

Để phục vụ luồng frame song công hoàn toàn, các liên kết vật lý trong cả hai chiều xuôi và ngược đều hỗ trợ 8 (hay 16) liên kết luận lý. Do đó, trong mỗi kênh luận lý, giao thức liên kết dữ liệu tại mỗi phía của liên kết đều duy trì các biến tuần tự truyền và nhận riêng biệt. Biến tuần tự truyền_0 hay 1_ là một chỉ số tuần tự truyền sẽ được gán vào frame mới kế tiếp để truyền lên kênh đường ra, trong khi biến tuần tự thu là chỉ số tuần tự của frame thông tin kế tiếp mong nhận được trên kênh ngược lại. Ngoài ra, nhằm đảm bảo mỗi kênh đường ra hoạt động trong chế độ dừng-và-chờ, mỗi bên cũng có một bit báo bận hay rảnh rỗi (busy/idle bit) liên hệ đến mỗi kênh, nó là dấu hiệu báo kênh bận hay không, nghĩa là có một báo nhận vẫn còn đang phát trên kênh này hay không.



Hình 6.9 Chi tiết về giao thức liên kết dữ liệu IMP-to-IMP của ARPANET.

Bên trong mạng ARPANET, mỗi frame được xem như một thực thể riêng biệt, nghĩa là các frame của cùng một bản tin được truyền nhận một cách độc lập nhau. Khi tiếp nhận một frame được gửi tới, giao thức liên kết số liệu chỉ cần quét bit busy/idle liên hệ với mỗi kênh luận lý để xác định một kênh có rảnh hay không, và nếu rảnh sẽ chèn chỉ số tuần tự truyền thích hợp _0 hay 1_ và chỉ số kênh luận lý vào phần header của frame này, khởi động bộ định thời (timer) cho frame và bắt đầu hoạt động truyền. Nếu kênh không rảnh, frame được xếp vào hàng đợi nhập chờ kênh rảnh.

Giao thức này dùng CRC 24 bit để phát hiện lỗi. Sửa lỗi dùng một byte báo nhận trong phần header của frame. 8 bit này là nối tiếp với 8 bit của chỉ số tuần tự thu liên hệ với luồng frame trong 8 kênh luận lý theo hướng ngược lại. Khi tiếp nhận một frame, giao thức liên kết số liệu đọc byte báo nhận cho tất cả các kênh tích cực, và dịch ra bit tương ứng tùy theo giao thức idle RQ đã được mô tả trong chương 5. Bằng cách này sẽ thu được thông tin báo nhận liên quan đến tất cả các kênh mỗi khi nhận được một frame mới. Điều này có hiệu quả tương tự như khi dùng một cửa sổ truyền K=8 (hay 16) cho liên kết. Nó cũng mang ý nghĩa như sử dụng lược đồ báo nhận ngầm định (implicit acknowledgment), chỉ có ACK.



Tóm lại, mặc dù nhiều giao thức thiên hướng ký tự vẫn còn được sử dụng rộng rãi, nhưng với sự xuất hiện các vi mạch tích hợp rẻ tiền đã tạo điều kiện hiện thực nhiều giao thức thiên hướng bit hiệu quả hơn, có nghĩa là tất cả các giao thức mới và nhiều giao thức hiện có thuộc loại thiên hướng bit.

3.4. Ví dụ về các giao thức thiên hướng ký tự thường gấp

3.4.1. Giao thức XMODEM

Năm 1977 một nhà tháo chương Ward Christensen đã đưa ra một giao thức truyền nhận tập tin gọi là XMODEM, đây là một giao thức đơn giản dùng thủ tục idle RQ (stop-and-wait) cùng với kiểu gói dữ liệu có độ dài cố định. Giao thức XMODEM được sử dụng rất rộng rãi và trở thành một trong những giao thức truyền nhận tập tin chuẩn mà tất cả các chương trình truyền số liệu phải có.

Gói dữ liệu của XMODEM

Trong giao thức XMODEM tất cả các field ngoại trừ field thông tin, đều có độ dài là một byte như sau:

SOH	Chỉ số thứ tự	Bù 1 của số thứ tự	Thông Tin	Kiểm Tra
-----	---------------	--------------------	-----------	----------

- Chỉ số thứ tự của gói truyền đầu tiên là 1 và các số tiếp theo được tăng dần, modulo với 256. Ngoài ra còn có field để chứa giá trị bù 1 của số thứ tự gói hiện hành trong field số thứ tự trước đó.
- Vùng thông tin có độ dài cố định là 128 bytes, thông tin có thể ở dạng text hoặc nhị phân.
- Vùng kiểm tra là một byte, dùng phương pháp kiểm tra kiểu tổng BSC và chỉ tính theo nội dung của vùng thông tin.

Sau này giao thức XMODEM-CRC sử dụng field kiểm tra có độ dài 2 byte và dùng phương pháp kiểm tra CRC với đa thức sinh theo CCITT là $X^{16} + X^{12} + X^5 + 1$.

Truyền tập tin bằng giao thức XMODEM

Hình 6.10 là lưu đồ mô tả quá trình truyền mỗi tập tin của giao thức XMODEM. Quá trình này có thể chia ra làm 3 giai đoạn :

(i) Giai đoạn khởi động

Như mọi giao thức khác, để khởi động việc truyền phải có sự hợp đồng giữa máy phát và máy thu. Máy phát chờ tín hiệu NAK từ máy thu. Khi có một tín hiệu NAK đầu tiên thì máy phát sẽ bắt đầu thực hiện việc truyền gói đầu tiên.

(ii) Giai đoạn giữa

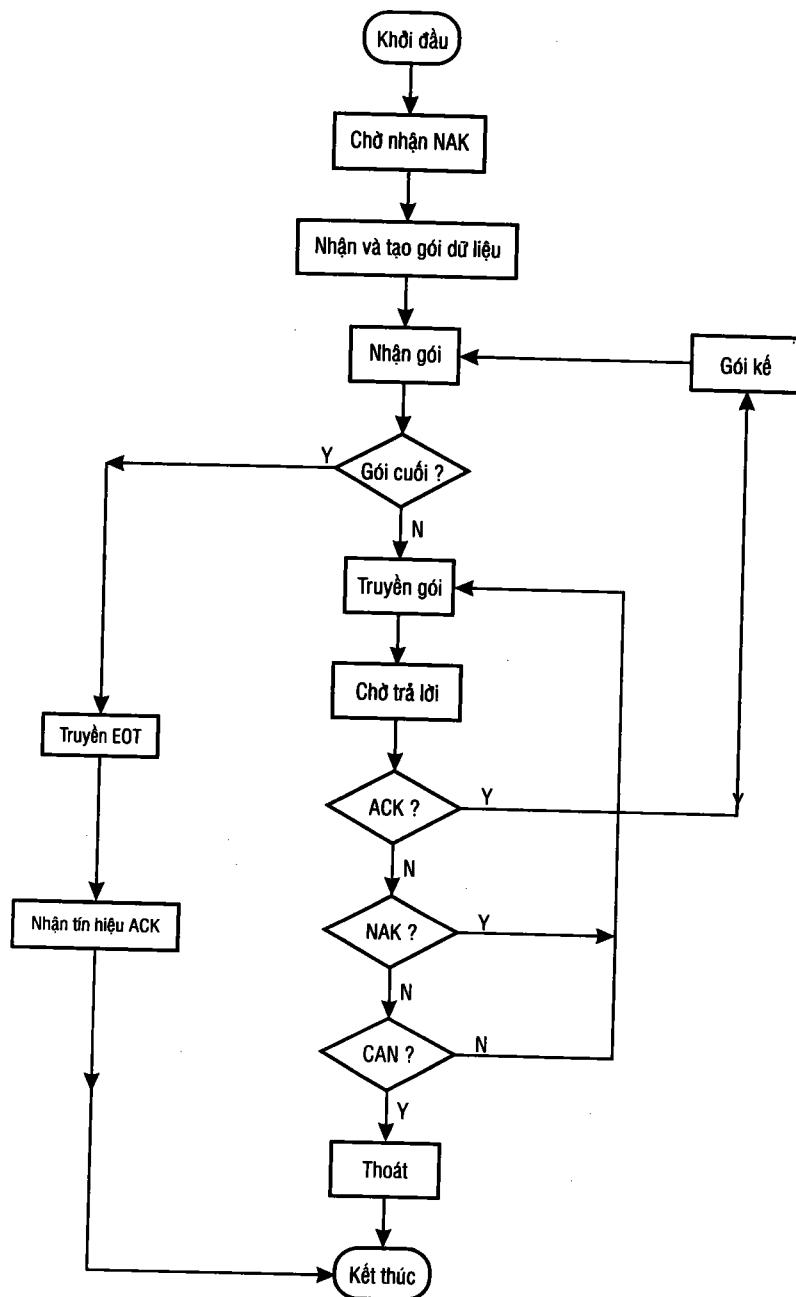
Máy phát chia tập tin thành từng khung 128 bytes để truyền đi và chờ tín hiệu báo nhận của máy thu đối với từng gói sau khi truyền. Nếu nhận được tín hiệu ACK máy phát sẽ xác nhận truyền gói kế tiếp. Nếu nhận được tín hiệu NAK máy phát sẽ truyền lại gói trước đó. Nếu nhận được tín hiệu CAN thì máy phát sẽ kết thúc truyền không điều kiện.

(iii) Giai đoạn kết thúc

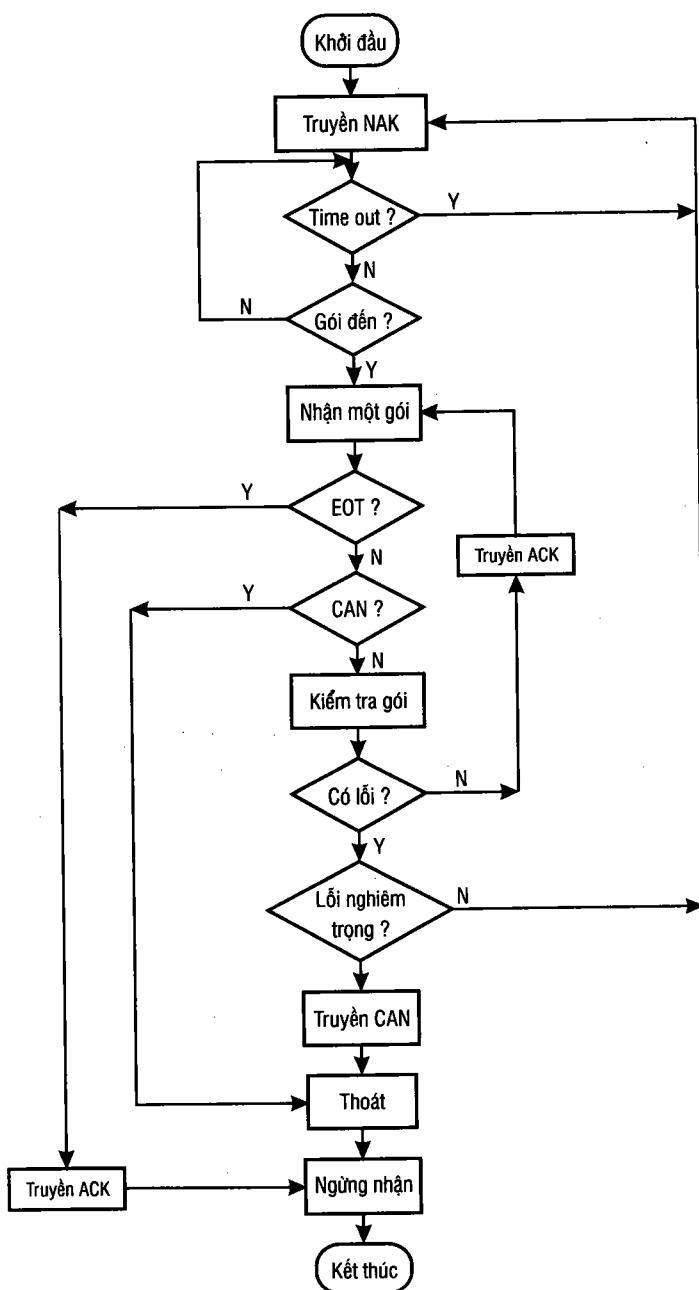
Nếu giai đoạn giữa diễn ra bình thường thì máy phát sẽ truyền thông điệp EOT (End of Transmission) để báo cho máy thu biết không còn truyền tập tin nào nữa. Lúc này máy thu sẽ đáp lại bằng tín hiệu ACK để cho biết đã nhận được EOT.

Sau đó là các công việc kết thúc việc truyền tập tin như đóng tập tin, giải phỏng vùng nhớ chứa tập tin .v.v.

Nếu giai đoạn giữa kết thúc không bình thường, do có tín hiệu CAN từ máy thu chẳng hạn thì máy phát kết thúc truyền tập tin mà không gửi tín hiệu EOT.



Hình 6.10 Lưu đồ truyền tập tin của giao thức XMODEM



Hình 6.11 Lưu đồ nhận tập tin của giao thức XMODEM

Nhận tập tin bằng giao thức XMODEM

Trong quá trình nhận tập tin, ngoài việc nhận các gói dữ liệu còn phải kiểm tra các field chứa thông tin phục vụ truyền để đảm bảo gói vừa nhận được là đúng số thứ tự cần và không bị sai. Do đó quá trình nhận sẽ phức tạp hơn quá trình truyền. Lưu đồ nhận được mô tả trong hình 6.11.

Quá trình nhận tập tin cũng được xúc tiến qua 3 giai đoạn:

(i) Giai đoạn khởi động

Sau khi thực hiện các công việc chuẩn bị như cấp phát vùng nhớ để chứa tạm (đệm) các gói nhận, mở các tập tin, đặt tên tập tin, chỉ đường dẫn .v.v. để sẵn sàng nhận các gói dữ liệu.

(ii) Giai đoạn giữa

Máy thu sẽ chuyển vào một vòng lặp trong đó thực hiện các thao tác như chờ một gói dữ liệu đến và truyền tín hiệu NAK nếu không có gói dữ liệu đến trong vòng 10 giây.

Khi nhận được byte đến thì máy thu biết là có một gói dữ liệu đến, và nó sẽ thực hiện các thao tác sau:

- Nếu byte nhận được không phải là SOH mà là EOT thì sẽ kết thúc giai đoạn giữa.
- Nếu byte nhận được là CAN sẽ kết thúc nhận không điều kiện
- Máy thu kiểm tra gói đến. Nó phải biết chắc là vùng thứ 2 và thứ 3 trong gói không sai lệch, bằng cách lấy bù 1 của giá trị ở một vùng nào đó (thứ 2 hoặc thứ 3) rồi EXOR với giá trị của vùng kia. Kết quả phải là 0, nếu khác 0 thì máy thu sẽ truyền tín hiệu NAK về cho máy phát.
- Kế đến máy thu sẽ kiểm tra số thứ tự của gói đến có phải là gói đang muốn nhận. Nếu không phải thì có nghĩa là có 1 sự cố sai sót xảy ra. Vì máy thu không thể biết sự cố này, nên không thể khắc phục và sẽ truyền tín hiệu CAN để yêu cầu máy phát kết thúc việc truyền. Chỉ có trường hợp số thứ tự của gói đến trùng với số thứ tự của gói trước đó thì nó giả thiết là do máy phát không nhận được ACK của gói này. Lúc này máy thu loại bỏ gói mới đến và truyền một tín hiệu ACK khác và quay lại chờ gói kế tiếp truyền đến.
- Cuối cùng máy thu sẽ tính vùng thông tin theo phương pháp tổng kiểm tra hoặc CRC để so sánh với giá trị trong vùng kiểm tra của gói vừa nhận. Nếu hai giá trị trùng nhau thì xem như không có lỗi và máy thu sẽ truyền tín hiệu ACK, ngược lại sẽ truyền tín hiệu NAK.

(iii) Giai đoạn kết thúc

Nếu quá trình nhận kết thúc bình thường thì máy phát sẽ truyền cho máy thu thông điệp EOT, máy thu sẽ đáp lại bằng một ACK và tiến hành các công việc sau cùng như đóng các tập tin, giải phóng vùng nhớ .v.v.

Đối với giao thức XMODEM-CRC các quá trình truyền và nhận cũng tương tự XMODEM, chỉ khác là ở giai đoạn khởi động, máy thu sẽ không truyền tín hiệu NAK mà truyền ký tự ‘C’ và máy phát cũng chờ nhận ký tự ‘C’ để khởi động. Còn sau đó vẫn dùng tín hiệu ACK và NAK để đồng bộ quá trình truyền nhận các gói.

3.4.2. Giao thức YMODEM

Giao thức YMODEM là sự mở rộng của giao thức XMODEM, được đưa ra đầu tiên vào năm 1981 để chạy trên hệ điều hành CP/M.

YMODEM có hai điểm cải tiến so với XMODEM là nó cho phép truyền nhận các khối dữ liệu có độ dài 1024 bytes (gấp 10 lần khối dữ liệu của XMODEM). Nếu đường truyền không tốt, YMODEM tự động giảm độ dài khối xuống 128 bytes để giảm bớt số byte phải truyền lại khi phát hiện sai. Như vậy trong trường hợp xấu nhất thì độ dài khối và chất lượng truyền sẽ bằng với giao thức XMODEM. Bên cạnh đó, YMODEM còn có một cải tiến nhằm giảm khả năng hư hỏng khi kết thúc truyền một tập tin. Như đã biết XMODEM sẽ hủy bỏ việc truyền nhận một tập tin ngay khi nhận được mẫu bit của ký tự Ctrl-X (ký tự ASCII là CAN) mẫu bit này rất dễ được tạo ra bởi nhiễu trên đường truyền. Trong khi YMODEM yêu cầu phải có 2 ký tự CAN liên tiếp nhau mới hủy bỏ việc truyền nhận tập tin.

YMODEM còn cung cấp 2 tính năng có ý nghĩa cho người sử dụng, đó là việc sử dụng phương pháp phát hiện và kiểm tra sai CRC và truyền các thông tin liên quan đến tập tin truyền cho máy thu. Các thông tin này gồm có tên tập tin, ngày, giờ và kích thước tập tin được truyền, các thông tin này ở trong khối đầu tiên khi bắt đầu việc truyền.

Ưu điểm sau cùng của YMODEM so với XMODEM là tính năng truyền nhóm tập tin. Có thể dùng đường dẫn và ký tự đại diện để chọn một nhóm các tập tin truyền cho các máy tính chạy hệ điều hành PC-DOS, MS-DOS, CP/M, UNIX hoặc XENIX. Có thể dùng kỹ thuật này cho việc lưu trữ dự phòng các tập tin (backup) nếu máy đầu xa có đủ dung lượng đĩa.

Nhưng YMODEM cũng có một số hạn chế so với XMODEM, YMODEM chỉ truyền ký tự dữ liệu 8 bit. YMODEM cũng dùng thủ tục idle RQ truyền và chờ, sau mỗi gói được truyền đi máy phát sẽ chờ tín hiệu báo nhận tốt (positive acknowledge). Vì thế cả YMODEM và XMODEM đều phải thêm thời gian đáng kể do việc chờ tín hiệu chấp nhận của máy thu đối với từng gói truyền. Tuy nhiên, nhược điểm này không quan trọng khi truyền theo phương thức truyền nhận tập tin qua hệ thống thông tin vệ tinh hoặc các mạng số liệu công cộng. Nhược điểm này sẽ được khắc phục ở giao thức kermit.

3.4.3. Giao thức kermit

Giao thức kermit xuất phát từ Đại học Columbia vào năm 1981. Giao thức kermit không chỉ cho phép thực hiện truyền và nhận tập tin giữa các máy tính nhỏ như giao thức XMODEM mà còn cho phép truyền và nhận tập tin giữa các hệ thống máy tính lớn như DECSYSTEM-20 và IBM-370. Do đó giao thức kermit rất

phức tạp. kermit cũng sử dụng thủ tục idle RQ (dừng và chờ) như XMODEM, nhưng có một số điểm khác biệt quan trọng so với XMODEM như sau:

- Kermit cho phép truyền và nhận nhiều tập tin cùng lúc.
- Kermit yêu cầu đối với kênh truyền rất tối thiểu, như kênh chỉ có thể truyền mọi ký tự mã ASCII, ký tự điều khiển SOH.
- Gói dữ liệu của kermit có chiều dài thay đổi được.
- Các tín hiệu trả lời của máy thu phải là những gói (trong khi XMODEM chỉ dùng các ký tự). Kermit cũng có thể dùng thủ tục continuous RQ nhờ trong gói ACK và NAK có vùng (field) chứa chỉ số thứ tự truyền (hay nhận) của gói.
- Kermit có các gói “đưa tin” nhờ đó có thể mở rộng các chức năng của giao thức mà không ảnh hưởng đến hoạt động của các phiên bản trước, và có thể bỏ qua một số thông số quan trọng.

Gói kermit tổng quát

MARK	LEN	SEQ	TYPE	DATA	CHK1	CHK2	CHK3
------	-----	-----	------	------	------	------	------

Gói của kermit bao gồm 6 vùng (field): vùng thông tin (data) có độ dài thay đổi tùy theo kiểu gói, các vùng còn lại được gọi là vùng điều khiển (hoặc là vùng phục vụ). Kermit qui định dữ liệu truyền đều là các ký tự mã ASCII in được (từ 20H-7EH) ngoại trừ vùng đánh dấu (MARK), như vậy đối với các vùng chứa giá trị số như vùng chỉ số thứ tự thì phải được chuyển đổi sang mã in được (printable character) trước khi truyền bằng cách cộng thêm 20H. Sau khi nhận sẽ được chuyển trở lại bằng cách trừ bớt 20H. Như vậy, giá trị số của các vùng này không được lớn hơn 94 (5EH), vì nếu nó bằng 5FH thì sau khi đổi sẽ trở thành 7FH là ký tự điều khiển DEL chứ không phải ký tự in được, còn nếu lớn hơn nữa thì sau khi đổi sẽ không còn ở trong phạm vi của bảng mã ASCII.

Điều này không thể áp dụng ở vùng thông tin vì byte dữ liệu cần truyền có thể có giá trị bất kỳ. Nếu có các ký tự điều khiển trong thông tin, thì kermit sẽ chuyển đổi bằng cách:

- Chèn thêm 1 ký tự in được ở phía trước (thường dùng ký tự # như đã nói trong mục 3.1, mã ASCII là 23H)
- EXOR ký tự điều khiển với 40H.

Ví dụ truyền 2 hàng chữ soạn bằng Wordstar sau:

Mary

a little

Chuỗi giá trị cần truyền là



4D	61	72	79	0A	0D					
M	a	r	y	LF	CR					
61	20	13	6C	69	74	74	6C	65	13	
a	SP	DC3	l	i	t	t	l	e	DC3	

(Ghi chú : Wordstar sử dụng ký tự điều khiển DC3 để đánh dấu gạch dưới từ little)

Kermit sẽ chuyển đổi các ký tự điều khiển trước khi truyền.

LF: 0AH → # J : 23H 4AH

CR: 0DH → # M : 23H 4DH

DC3: 13H → # S : 23H 53H

Như vậy chuỗi giá trị truyền sẽ là

Mary # J # M a SP # S little # S

Đối với các hệ thống sử dụng ký tự dài 8 bit, thì những ký tự có bit có ý nghĩa lớn nhất (MSB bit) bằng 1 sẽ được kermit đổi sang mã ASCII in được bằng cách thêm vào trước một ký tự (&).

Ý nghĩa của các vùng trong gói được mô tả như sau:

- Vùng đánh dấu (MARK) : để đánh dấu bắt đầu của gói là ký tự SOH mã 01H
- Vùng độ dài (LEN) : số bytes trong gói tính từ sau byte này (tức là độ dài của gói trừ 2), giá trị tối đa là 94 như vậy độ dài tối đa một gói là 96 bytes.
- Vùng số thứ tự (SEQ) : số thứ tự của gói, modulo với 64. Gói truyền đầu tiên (gói S) sẽ có số thứ tự là 0, số thứ tự của gói kế tiếp sau gói có số thứ tự là 63 sẽ trở lại là 0.
- Vùng kiểu gói (TYPE) : để phân biệt các kiểu gói khác nhau. Mỗi kiểu gói sẽ có nội dung và nhiệm vụ khác nhau.
- Vùng thông tin (DATA) : nội dung của tập tin cần truyền được chứa trong gói 'D', còn trong gói 'F' chứa tên tập tin. Trong một số kiểu gói khác, vùng này không chứa gì.
- Vùng kiểm tra (CHECK): có thể chọn vùng 1 byte tổng kiểm tra hoặc 2 byte tổng kiểm tra, hoặc 3 byte CRC. Giá trị kiểm tra được tính từ vùng độ dài (không tính vùng đánh dấu). Vì các byte này cũng phải được đổi ra ký tự in được, nhưng chúng lại có giá trị bất kỳ không thể giới hạn ở giá trị nhỏ hơn 5EH nên trước khi đổi thì 2 bit 6 và 7 sẽ được trích ra và cộng vào 6 bit thấp còn lại. Nhờ đó ở đầu thu có

thể lấy lại được hai giá trị này, còn ở đầu phát sau khi biến đổi ta sẽ có 2 bit 6 và 7 đều là 0.

Có thể áp dụng công thức sau để biến đổi 1 byte kiểm tra x trước khi chuyển sang mã ký tự in được :

$$(x + (x \text{ AND } C0H) / 40H) \text{ AND } 3FH$$

Các kiểu gói của kermit

Trong giao thức kermit một gói rỗng (empty packet) được định nghĩa là một gói mà trong vùng thông tin không có ký tự nào và gói trống (blank packet) là một gói không rỗng với vùng thông tin chứa một hoặc nhiều ký tự khoảng trắng SP.

Một số kiểu gói của kermit có chứa thông tin như gói 'D' ngược lại một số kiểu luôn luôn rỗng như gói 'Z'. Còn lại một số kiểu có thể rỗng ở trong một khoảng thời gian nhất định như gói ACK. Phần sau đây sẽ mô tả chi tiết hơn các kiểu gói khác nhau của kermit.

Gói 'S'

Đây là một kiểu gói lý thú nhất. Chức năng của nó là thông báo cho máy đối phương (máy nhận gói này) biết một số tham số truyền quan trọng. Vùng số thứ tự của gói này luôn là 0 và vùng thông tin của nó sẽ gồm các thành phần sau:

MAXL	TIME	NPAD	PADC	EOL	QCTL	QBIN	CHKT	REPT	CAPAS	Dành riêng
------	------	------	------	-----	------	------	------	------	-------	------------

- **MAXL** : Cho biết kích thước gói tối đa mà máy có thể nhận. Mặc định là none.
- **TIME** : Số giây tối đa mà máy đối phương phải chờ một gói trước khi kết thúc thời gian chờ (timeout). Dùng để phối hợp với các hệ thống tốc độ chậm. Mặc nhiên là none.
- **NPAD** : Số ký tự đóng gói (pad) mà máy đối phương phải truyền trước mỗi gói. Được dùng cho các hệ thống bán song công khi chuyển từ phát sang thu hoặc ngược lại. Mặc định là 0 (không đóng gói).
- **PADC** : Định dạng ký tự đóng gói, thường dùng ký tự NUL hoặc DEL. Mặc định là NUL _ Vùng này không cần nếu NPAD là 0.
- **EOL** : Cho biết ký tự mà máy cần dùng để kết thúc một gói. Máy chỉ nhận một gói khi có ký tự này theo sau. Mặc định là CR. Cần lưu ý ký tự này không phải thuộc về gói truyền mà chỉ được truyền kế tiếp sau gói này.
- **QCTL** : Ký tự mà máy phát dùng để chèn vào khi chuyển đổi các ký tự điều khiển ra mã ASCII in được. Mặc định là #.

Các thành phần còn lại có thể có hoặc không.

- **QBIN** : Ký tự mà máy phát dùng để chèn khi chuyển đổi các ký tự ở hệ thống 5 bit chỉ được dùng là ký tự trong khoảng 21H-3FH hoặc 60H-7FH. Mặc định là ký tự SP (không chèn ký tự).
- **CHKT** : Cho biết phương thức kiểm tra được sử dụng trong các gói
 - = 1 → tổng kiểm tra bằng 1 byte.
 - = 2 → tổng kiểm tra bằng 2 bytes.
 - = 3 → CRC.

Mặc định là 1.

- **REPT** : Ký tự mà máy phát dùng để chèn vào khi nén các ký tự liên tiếp giống nhau trong chuỗi dữ liệu truyền. Cũng chỉ dùng ký tự trong khoảng 21H-3FH hoặc 60H-7EH. Nếu là SP thì cho biết là không thực hiện việc nén.
- **CAPAS** : Đây là 1 vùng mở rộng không giới hạn. Có thể có nhiều byte kế tiếp nhau hoặc chỉ có 1 byte. Bit 0 của byte sẽ cho biết ở sau nó còn có byte tiếp theo không.

Bit 0 = 0 → không có byte tiếp theo.

Bit 0 = 1 → còn có byte đi sau.

Như vậy trong 1 byte chỉ còn 6 bit có ý nghĩa. Vào giữa năm 1986 thì bit 5 và 4 vẫn còn dành riêng. Bit 3 dùng để xác định thuộc tính của gói, bit 2 dùng để cài đặt việc sử dụng thủ tục continuous RQ (RQ liên tục) thay cho idle RQ (dừng và chờ). Bit 1 để cài đặt việc sử dụng gói có kích thước lớn hơn 94 bytes.

Gói 'Y'

Còn gọi là gói ACK. Được truyền đi khi máy thu đã chấp nhận một gói. Gói này cũng gồm đầy đủ các vùng SEQ, LEN và CHECK. Vùng thông tin của gói này thường là rỗng, nhưng trong trường hợp trả lời cho gói 'S' thì vùng thông tin sẽ chứa các thông số khởi tạo của máy thu với định dạng giống như vùng thông tin của gói 'S'.

Gói 'F'

Chứa tên của tập tin truyền. Để tránh trường hợp có các ký tự đặc biệt mà hệ thống ở nơi nhận có thể hiểu lầm khi tên tập tin giới hạn chỉ là các ký số và các ký tự chữ hoa (30H-39H, 41H-5AH). Mặc dù gói có độ dài không hạn chế nhưng vùng thông tin của gói này chỉ chứa tên tập tin chứ không có đường dẫn.

Trong gói 'Y' mà máy thu dùng để thông báo chấp nhận cũng có thể có tên tập tin mà máy thu dùng để lưu trữ tập tin truyền.

Gói 'N'

Khi máy thu không chấp nhận gói vừa thu được thì nó sẽ gửi cho máy phát gói này. Gói này cũng có đầy đủ các vùng SEQ, LEN và CHECK. Vùng thông tin của gói luôn rỗng.

Gói 'D'

Là gói chứa thông tin dữ liệu của tập tin truyền và dữ liệu trong vùng thông tin tất nhiên cũng phải là mã ASCII in được.

Gói 'Z' và 'B'

Gói 'Z' còn gọi là gói EOF được máy phát truyền đi sau khi nó đã truyền xong một tập tin.

Gói 'B' còn gọi là gói EOT được máy phát truyền đi khi tất cả các tập tin đã được truyền. Trong cả hai trường hợp máy thu đều trả lời bằng gói 'Y'.

Gói 'E'

Được dùng để máy phát hoặc máy thu truyền thông báo khi phát hiện một sai sót, gói này không chỉ ra hoặc chú ý về lỗi sai.

Gói 'A'

Chỉ được dùng khi có cài đặt trong vùng mở rộng (CAPAS) của gói 'S'. Gói này được dùng để cho biết các thuộc tính của tập tin truyền như kích thước (theo kbyte), nội dung (text, nhị phân, hình ảnh...), ngày, giờ, đường dẫn, thông tin về độ an toàn và khả năng truy xuất (password, mức bảo vệ, ...), tập tin được tạo ra trên máy tính nào hoặc trên hệ điều hành nào, kiểu mã hóa (ASCII, EBCDIC, ...).

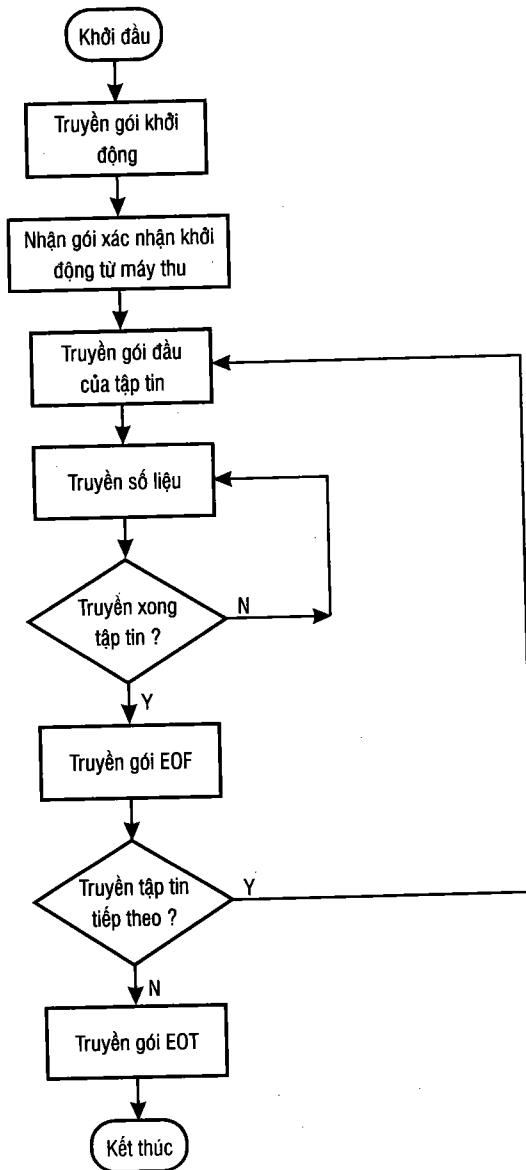
Truyền và nhận tập tin bằng giao thức kermit

Kermit là một giao thức cho phép truyền và nhận tập tin giữa các loại máy tính khác nhau, với tốc độ cao, truyền trong một khoảng cách xa và đặc biệt là cho phép truyền cả một nhóm các tập tin (người sử dụng có thể dùng những dấu hiệu đại diện để yêu cầu truyền nhiều tập tin mà không phải dừng sau mỗi tập tin). Kermit có thể truyền cả tập tin văn bản (text) và nhị phân, nhưng mọi dữ liệu đều được đổi ra dạng văn bản (text) trước khi truyền. Ngoài ra nó có thể truyền cả những ký tự dữ liệu 8 bit như mã ASCII mở rộng, hoặc các ký tự đồ họa bằng cách chuyển sang các ký tự 7 bit trước khi truyền.

Với việc trao đổi các gói khởi tạo 'S' giữa hai máy khi bắt đầu quá trình truyền nhận, kermit cho phép 2 máy so sánh các gói này và chọn lựa những thông số mà cả 2 máy đều có thể thay đổi sao cho việc truyền tin được tối ưu.

Lưu đồ hình 6.12 mô tả một cách đơn giản quá trình truyền một hoặc nhiều tập tin theo giao thức kermit.

Khả năng truyền một nhóm tập tin cùng lúc của kermit giúp tiết kiệm được thời gian truyền và có thể khởi tạo cùng các thông số truyền cho nhiều tập tin. Ngoài ra nó làm giảm bớt một số gói truyền không cần thiết.



Hình 6.12 Lưu đồ tóm tắt quá trình truyền của kermit

4. CÁC GIAO THỨC THIỀN HƯỚNG BIT

Tất cả các giao thức liên kết số liệu mới đều là các giao thức thiên hướng bit. Lưu ý rằng các giao thức như vậy sử dụng các mẫu bit đã được định nghĩa thay cho các ký tự điều khiển truyền để đánh dấu mở đầu và kết thúc một frame. Máy thu duyệt luồng bit thu theo từng bit một để tìm mẫu bit đầu và cuối frame. Ba phương pháp báo hiệu bắt đầu và kết thúc một frame được gọi là phân định ranh giới frame (frame delimiting) được trình bày trên hình 3.13, gồm có:

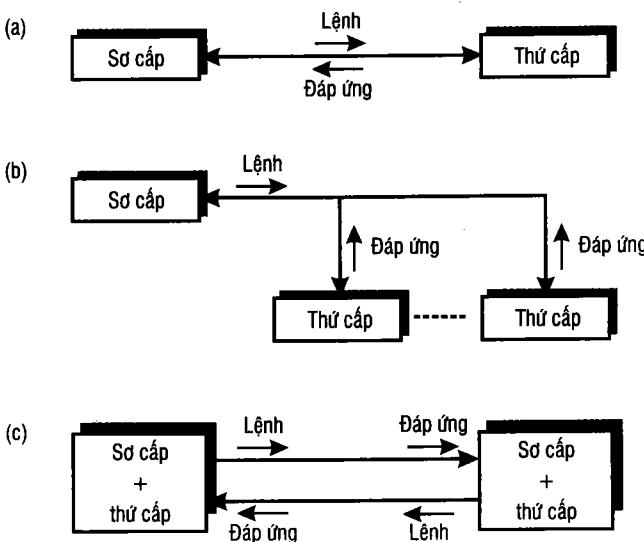
- Mẫu bit duy nhất (không trùng với mẫu nào) bắt đầu và kết thúc một frame, được gọi là cờ (01111110), kết hợp với kỹ thuật nhồi các bit 0.
- Một mẫu bit duy nhất đánh dấu đầu frame, được gọi là ranh giới đầu frame (10101011), và một byte chỉ chiều dài (đơn vị là byte) trong phần header của frame.
- Mẫu xác định ranh giới đầu và cuối frame duy nhất gồm các bit được tạo ra do cưỡng bức mã hóa.

Nhìn chung, phương pháp đầu tiên được dùng với giao thức điều khiển liên kết số liệu mức cao (HDLC), trong khi đó hai phương pháp còn lại được dùng với giao thức LLC. Trong thực tế, hầu hết các giao thức thiên hướng bit đều là dẫn xuất từ giao thức HDLC, do đó chúng ta sẽ bàn đến giao thức này trước tiên.

4.1. Giao thức điều khiển liên kết số liệu mức cao HDLC (High-level Data Link Control)

Giao thức HDLC là một giao thức chuẩn hóa quốc tế và đã được định nghĩa bởi ISO để dùng cho cả liên kết điểm-nối-điểm và đa điểm. Nó hỗ trợ hoạt động ở chế độ trung suốt, song công hoàn toàn và ngày nay được dùng một cách rộng rãi trong các mạng đa điểm và trong các mạng máy tính. Mặc dù từ viết tắt HDLC khá phổ biến, nhưng một số nhà máy lớn và các tổ chức tiêu chuẩn khác vẫn còn gọi giao thức này bằng các tên riêng của họ. Ví dụ SDLC (Synchronous Data Link Control) của IBM, đây là tiền thân của HDLC, và ADCCP (Advanced Data Communications Control Procedure) là tên được dùng bởi tổ chức ANSI (American National Standards Institute).

Bởi HDLC đã được định nghĩa như là một giao thức điều khiển liên kết số liệu tổng quát, nên chúng ta có thể dùng nó trong một số cấu hình mạng khác nhau, như trình bày trên hình 6.13. Trong HDLC, các frame được gửi từ trạm sơ cấp đến trạm thứ cấp được gọi như là các lệnh (command) và các frame được gửi từ thứ cấp đến sơ cấp được gọi là các đáp ứng (response). Hai cấu hình được trình bày trong phần (a) và (b) chỉ có một trạm sơ cấp được gọi là *cấu hình không cân bằng*, trong khi đó phần (c) có hai trạm sơ cấp và được gọi là *cấu hình cân bằng*. Trong cấu hình cân bằng, vì mỗi trạm đều có phần sơ cấp và cả thứ cấp, nên chúng được gọi là các trạm kết hợp.



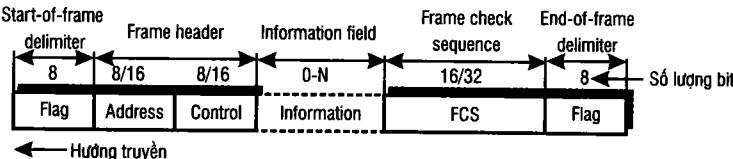
Hình 6.13 Cấu hình mạng dùng HDLC: (a) Điểm-nối-điểm với một sơ cấp và một thứ cấp (b) Đa điểm với một sơ cấp và nhiều thứ cấp (c) Điểm-nối-điểm với hai sơ cấp và hai thứ cấp.

HDLC có 3 chế độ hoạt động:

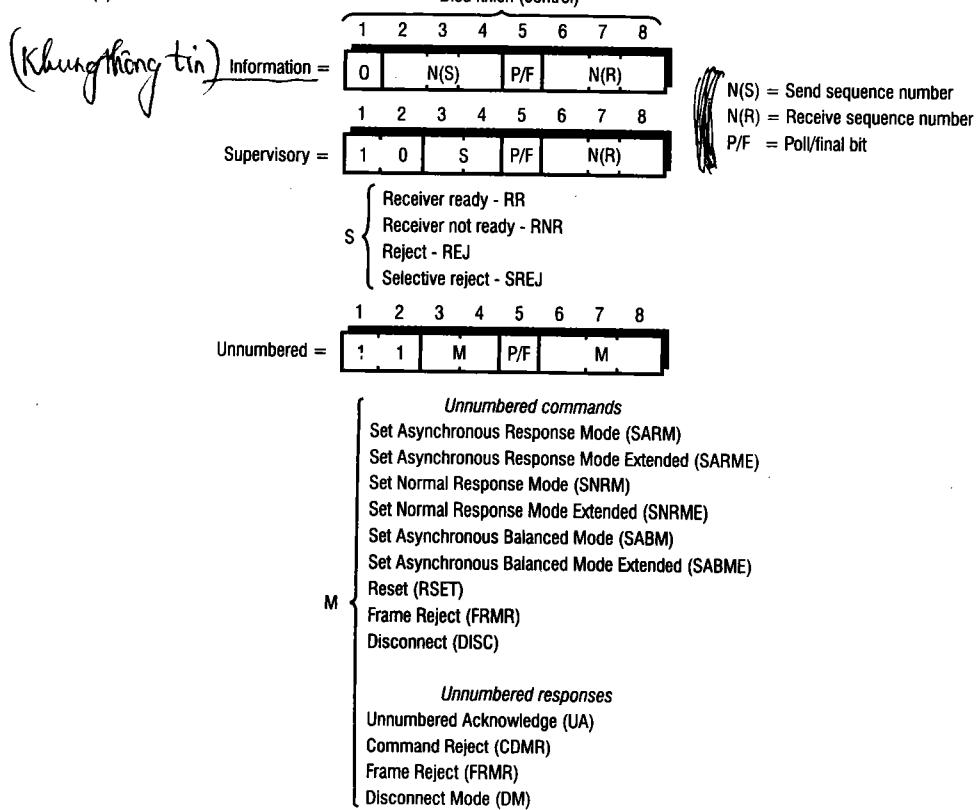
- Chế độ đáp ứng thông thường NRM (Normal Response Mode): Chế độ này được dùng trong cấu hình không cân bằng. Trong chế độ này, trạm thứ cấp chỉ có thể truyền khi nhận được chỉ thị đặc biệt của trạm sơ cấp. Liên kết này có thể là điểm-nối-điểm hay đa điểm. Trường hợp đa điểm chỉ cho phép một trạm sơ cấp.
- Chế độ đáp ứng bất đồng bộ ARM (Asynchronous Response Mode): Chế độ này cũng được dùng trong cấu hình không cân bằng. Nó cho phép một trạm thứ cấp xúc tiến một hoạt động truyền mà không cần sự cho phép từ trạm sơ cấp. Chế độ này thường được dùng trong các cấu hình điểm-nối-điểm và các liên kết song công và cho phép thứ cấp truyền các frame một cách bất đồng bộ với sơ cấp.
- Chế độ cân bằng bất đồng bộ ABM (Asynchronous Balanced Mode): Chế độ này được dùng chủ yếu trên các liên kết song công điểm-nối-điểm cho ứng dụng truyền số liệu máy tính-đến-máy tính và cho các kết nối giữa máy tính và mạng số liệu công cộng (PSDN). Trong chế độ này, mỗi trạm có một trạng thái như nhau và thực hiện cả hai chức năng sơ cấp và thứ cấp. Nó là chế độ được dùng trong giao thức nối tiếng X.25.

4.1.1. Các khuôn dạng của frame

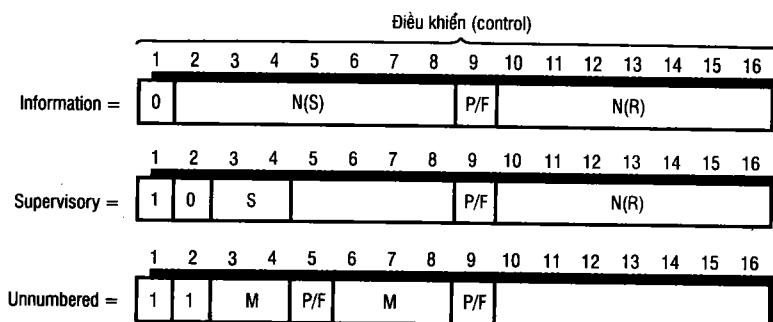
(a)



(b)



(c)



Hình 6.14 Dạng frame HDLC và các kiểu frame: (a) Dạng chuẩn/mở rộng (b) Ý nghĩa của các bit trong vùng điều khiển chuẩn (c) Ý nghĩa của các bit trong vùng điều khiển mở rộng.

Không giống như BSC, trong HDLC cả số liệu và thông tin điều khiển đều được tải trong cùng một dạng frame chuẩn. Dạng frame này được trình bày trên hình 6.14 cùng với các kiểu frame khác nhau được xác định trong vùng điều khiển của phần header. Có 3 nhóm frame được dùng trong HDLC:

- 1) Các frame không đánh số: Các frame này được dùng cho các chức năng như thiết lập liên kết và xóa liên kết. Tên được gọi là *không đánh số* vì chúng không chứa bất kỳ thông tin báo nhận nào, nên cũng không có chỉ số tuần tự.
- 2) Các frame thông tin: Các frame này mang các thông tin thực hay số liệu và thường được xem như là các I-frame. Các I-frame cũng có thể được dùng để tải thông tin báo nhận liên hệ đến luồng I-frame theo hướng ngược lại khi liên kết đang hoạt động ở chế độ ABM hay ARM.
- 3) Các frame quản lý: Các frame này được dùng để kiểm soát lỗi và điều khiển luồng, do đó chứa các số tuần tự truyền và nhận.

Việc dùng các cờ như là các dấu hiệu chỉ đầu và cuối frame, cùng với việc chèn và xóa các bit 0 nhằm đạt được sự trong suốt dữ liệu đều đã được mô tả trong chương 3, ở đây sẽ không lặp lại.

Tuần tự kiểm tra frame FCS (Frame Check Sequence) là một mã CRC 16 bit cho toàn bộ phần nội dung của frame được đóng bởi hai cờ. Đa thức sinh được dùng với HDLC thường là CRC-CCITT:

$$X^{16} + X^{12} + X^5 + 1$$

FCS được tạo ra bởi thủ tục được mô tả trong chương 4 và được tăng cường bởi các thủ tục khác nhằm làm cho sự kiểm tra tinh vi hơn. Ví dụ như thêm 6 bit 1 vào đuôi của số bị chia trước khi chia (thay vì bit 0) và đảo ngược số dư. Điều này có tác dụng là làm cho số dư được tính lại bởi máy thu không phải tất cả đều là 0 mà là một mẫu bit đặc biệt _0001 1101 0000 1111.

Nội dung của vùng địa chỉ phụ thuộc vào chế độ hoạt động. Trong NRM, ví dụ trên đường dây đa nhánh, mỗi trạm thứ cấp đều được gán một địa chỉ duy nhất. Bất cứ khi nào trạm sơ cấp thông tin với một trạm thứ cấp, thì vùng địa chỉ này chứa địa chỉ của trạm thứ cấp. Có thể có địa chỉ được xem như là địa chỉ nhóm và được gán cho một nhóm các trạm thứ cấp. Tất cả các frame được truyền với địa chỉ nhóm kèm theo sẽ được tất cả các trạm thứ cấp trong nhóm đó nhận. Tương tự, một địa chỉ broadcast (quảng bá) có thể được dùng để truyền cho tất cả các trạm thứ cấp trên liên kết.

Khi một trạm thứ cấp gửi một thông điệp phúc đáp cho sơ cấp, vùng địa chỉ luôn chứa địa chỉ duy nhất của thứ cấp này. Trong trường hợp các mạng lớn có nhiều trạm thứ cấp, vùng địa chỉ có thể được mở rộng hơn 8 bit. Bit có ý nghĩa nhỏ nhất trong mỗi vùng 8 bit được dùng để chỉ định phía sau nó có một octet (byte) nào khác không. Nếu bit này là 0 thì phía sau có một octet khác, ngược lại thì đây là octet sau cùng (hay chỉ có một octet). Lưu ý rằng trong ABM vùng địa chỉ không được dùng theo cách này bởi chỉ có các liên kết điểm-nối-diểm trực tiếp. Thay vì vậy, nó được dùng để định hướng cho các lệnh và các đáp ứng liên hệ với chúng.

Các ý nghĩa của các bit trong vùng điều khiển được trình bày trên hình 6.14(b). Vùng S trong các frame quản lý và vùng M trong các frame không đánh số được dùng để định nghĩa kiểu frame đặc biệt. Các chỉ số tuần tự truyền và nhận_N(S) và N(R) được dùng phối hợp với các thủ tục kiểm soát lỗi và điều khiển luồng.

Bit P/F được gọi là bit quét/kết (poll/final). Một frame của bất kỳ loại nào được gọi là frame lệnh (command frame) nếu nó được gửi bởi trạm sơ cấp và là frame đáp ứng (response frame) nếu nó được gửi bởi một trạm thứ cấp. Bit P/F được gọi là bit quét khi được dùng trong một frame lệnh, nếu được set sẽ chỉ thị cho máy thu phải báo nhận cho frame này. Máy thu báo nhận cho frame này bằng cách gửi lại một frame đáp ứng trong đó bit P/F được set, lúc này bit P/F được gọi là bit kết.

Việc dùng 3 bit cho mỗi N(S) và N(R) có nghĩa là chỉ số tuần tự có thể nằm trong dải từ 0 đến 7. Điều này cũng có nghĩa là kích thước tối đa của cửa sổ được chọn là 7. Mặc dù như vậy là đủ lớn cho nhiều ứng dụng, nhưng đối với những liên kết rất xa (ví dụ như liên kết vệ tinh) hay tốc độ bit rất cao, nếu muốn đạt được hiệu suất liên kết cao thì phải mở rộng cửa sổ hơn nữa. Dạng mở rộng dùng 7 bit (từ 0 đến 127), do đó giá tăng kích thước cửa sổ lên đến 127.

Vùng địa chỉ nhận dạng trạm thứ cấp đã gửi frame, và là vùng không cần đến trong các liên kết điểm-nối-diểm. Tuy nhiên, đối với liên kết đa điểm, thì vùng địa chỉ có thể là 8 bit_chế độ thường hoặc là bộ số của 8 bit_chế độ mở rộng. Trong trường hợp sau, bit 1 của octet địa chỉ có giá trị thấp nhất được set là 0 và bit 1 trong octet sau cùng được set là 1. Các bit còn lại hình thành nên địa chỉ. Trong cả hai chế độ, địa chỉ có tất cả các bit đều là 1 được dùng làm địa chỉ broadcast.

4.1.2. Các kiểu frame

Trước khi mô tả hoạt động của giao thức HDLC, có lẽ nên liệt kê vài kiểu frame và tìm hiểu chức năng của chúng. Có ba chủng loại frame được dùng. Vài kiểu khác nhau trong mỗi loại được liệt kê trong hình 6.14(b).

Các frame không đánh số được dùng để quản lý liên kết. Ví dụ, SNRM-frame và SABM-frame, cả hai được dùng để thiết lập một liên kết luân lý giữa trạm sơ cấp và trạm thứ cấp và để thông báo cho trạm thứ cấp về chế độ hoạt động đang được dùng. Một liên kết luận lý được xóa khi trạm sơ cấp gửi một DISC-frame. UA-frame được dùng như một báo nhận cho các frame khác trong chủng loại này.

Mặc dù có bốn loại frame quản lý, nhưng chỉ RR và RNR được dùng trong cả NRM và ABM. Cả hai frame này được dùng để cho biết một trạm thứ cấp có sẵn sàng thu hay không đối với một hay nhiều frame được gửi từ sơ cấp, và cho mục đích báo nhận. REJ-frame và SREJ-frame chỉ được dùng trong ABM, cho phép truyền tin theo hai hướng đồng thời xuyên qua một liên kết điểm-nối-diểm. Hai frame này được dùng để chỉ đến một trạm khác đã xảy ra lỗi tuần tự, nghĩa là đã nhận một I-frame chứa một số tuần tự N(S) không đúng thứ tự. SREJ-frame được dùng cùng với thủ tục truyền lại có chọn lựa (seletive repeat), ngược lại REJ-frame được dùng với thủ tục truyền lại một nhóm (go-back-N).

4.1.3. Hoạt động của giao thức

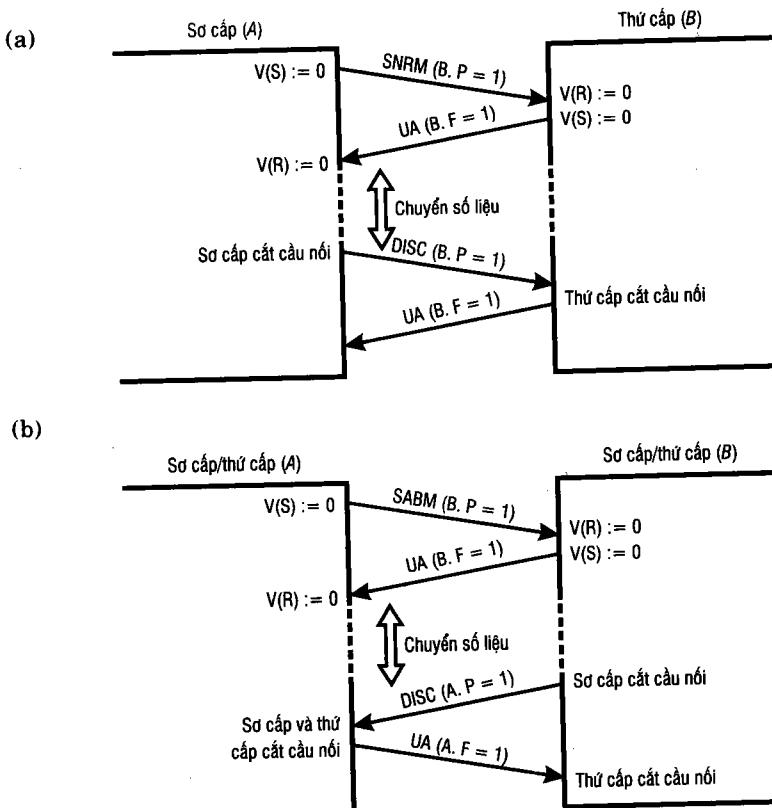
Phần này làm sáng tỏ vài đặc trưng quan trọng của giao thức HDLC, thay vì cung cấp một mô tả đầy đủ hoạt động của nó. Có hai chức năng cơ bản là quản lý liên kết và chuyển số liệu (bao gồm kiểm soát lỗi và điều khiển luồng).



- Quản lý liên kết

Trước khi truyền bất kỳ thông tin số liệu nào giữa sơ cấp và một trạm thứ cấp trên một liên kết đa điểm hoặc giữa hai trạm được nối với nhau qua một liên kết điểm-nối-diểm, một cầu nối luận lý giữa hai chủ thể truyền tin phải được thiết lập. Điều này được hoàn thành nhờ sự trao đổi hai frame không đánh số, được trình bày trên hình 6.15.

Trong một liên kết đa điểm (hình 6.15(a)), trước tiên trạm sơ cấp truyền một SNRM-frame với bit P/F được set là 1 và địa chỉ của thứ cấp được đặt trong vùng địa chỉ. Thứ cấp đáp ứng bằng một UA-frame với bit kết (final) được set là 1 cùng với địa chỉ của nó trong vùng địa chỉ. Như chúng ta có thể thấy, thủ tục thiết lập có tác dụng khởi động các biến tuần tự trong mỗi trạm. Các biến này được dùng trong các thủ tục điều khiển luồng và kiểm soát lỗi. Cuối cùng, sau khi tất cả số liệu đã được truyền, liên kết bị xóa khi sơ cấp gửi một DISC-frame và thứ cấp đáp ứng một UA-frame.



Hình 6.15 Thủ tục quản lý liên kết: (a) Chế độ đáp ứng thông thường trên liên kết đa điểm (b) Chế độ cân bằng bắt đồng bộ trên liên kết điểm-nối-diểm.

Thủ tục thiết lập một liên kết điểm-nối-diểm giống như thủ tục thiết lập liên kết được dùng trong liên kết đa điểm. Tuy nhiên, trong ví dụ được trình bày trên hình 6.15(b) thì ABM được chọn và do đó một SABM-frame được truyền trước tiên. Trong chế độ này, cả hai phía của liên kết đều có thể khởi động truyền các I-frame một cách độc lập, vì vậy mỗi trạm thường là một trạm kết hợp cả sơ cấp và thứ cấp. Cả hai trạm đều có thể khởi động và xóa liên kết trong chế độ này. Trong hình 6.15(b), trạm A khởi động thiết lập liên kết còn trạm B khởi động xóa cầu nối luận lý này (sau khi đã truyền xong số liệu). Chỉ cần một lần trao đổi các frame này đã thiết lập một liên kết theo cả hai hướng. Như chúng ta có thể thấy, vùng địa chỉ được dùng để định hướng cho frame lệnh (SABM/DISC) và các đáp ứng liên hệ với nó.

Nếu máy thu từ chối lệnh thiết lập cầu nối trong cả hai chế độ, nó sẽ phúc đáp một DM-frame (disconnected mode-frame) khi nhận frame xác lập chế độ (SNRM hoặc SABM). DM-frame chỉ ra rằng trạm đang đáp ứng cắt cầu nối luận lý.

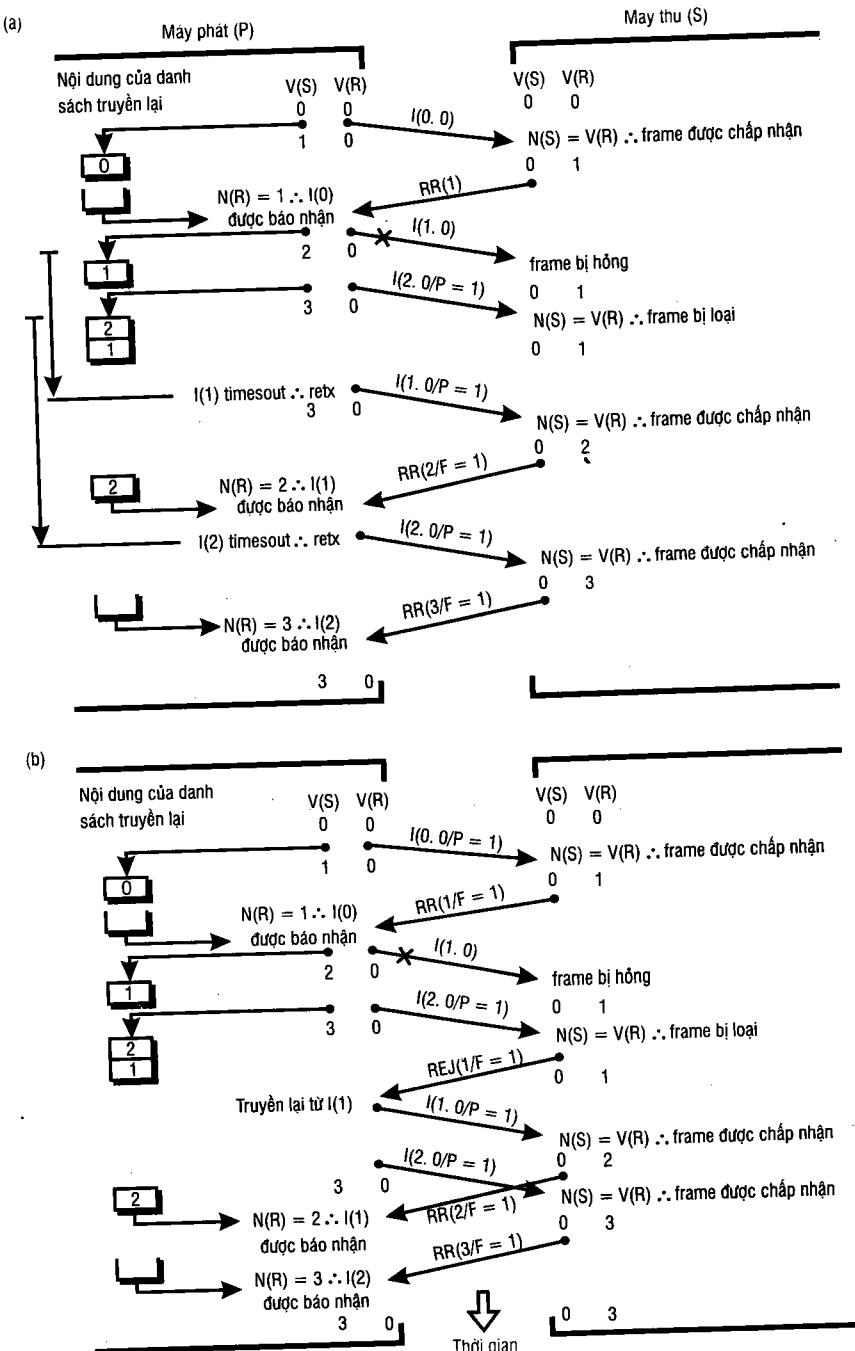
• Chuyển số liệu

Trong NRM, tất cả các I-frame được truyền dưới sự điều khiển của trạm sơ cấp. Các frame quét không đánh số *UP-frame* (*Unnumbered Poll-frame*) với bit P được set là 1, thường được trạm sơ cấp dùng để quét một trạm thứ cấp. Nếu thứ cấp không có số liệu truyền, nó gửi đáp một RR-frame trong đó bit F được set. Nếu có số liệu đang đợi truyền, nó sẽ truyền ngay dưới dạng các I-frame, trong I-frame cuối cùng bit F được set là 1.

Hai khía cạnh quan trọng nhất trong cung đoạn chuyển số liệu đó là điều khiển luồng và kiểm soát lỗi. Kiểm soát lỗi dùng thủ tục continuous RQ theo chiến lược truyền lại có chọn lựa hay truyền một nhóm, trong khi đó điều khiển luồng dựa vào cơ cấu cửa sổ. Chúng ta đã nói về hai thủ tục này trong chương 5, do đó ở đây chúng ta chỉ trình bày một tuần tự frame nhằm minh họa việc sử dụng các kiểu frame khác nhau.

Hình 6.16(a) mô tả thủ tục báo nhận và truyền lại cơ bản; trong ví dụ này chỉ dùng các RR-frame và giả sử áp dụng chiến lược truyền lại một nhóm (go-back-N). Trong hình chỉ trình bày luồng frame theo một chiều, vì vậy tất cả các thông tin báo nhận phải được gửi thông qua các frame quản lý báo nhận đặc biệt. Như chúng ta có thể thấy, mỗi phía của liên kết đều duy trì một biến tuần tự truyền và biến tuần tự thu. V(S) chỉ định số tuần tự truyền kế tiếp N(S), được trạm này phân phối cho một I-frame truyền, và V(R) là chỉ số tuần tự truyền của I-frame mà trạm này đang mong đợi theo tuần tự.

Mỗi frame quản lý RR (báo nhận tốt) chứa một số tuần tự thu N(R), nó thừa nhận tất cả các I-frame đã được truyền từ thời điểm đó trở về trước và bao gồm I-frame có N(S) bằng $[N(R) - 1]$. Tương tự, hình 6.16(b) cho thấy mỗi frame quản lý REJ (thông báo từ chối) chứa một N(R) chỉ ra I-frame vừa nhận không đúng theo thứ tự và máy phát phải có trách nhiệm khởi động truyền lại các I-frame bắt đầu từ chỉ số N(S) bằng N(R).



Hình 6.16 Sử dụng các frame báo nhận: (a) Chỉ có báo nhận tốt (RR)
(b) Thông báo từ chối (REJ).

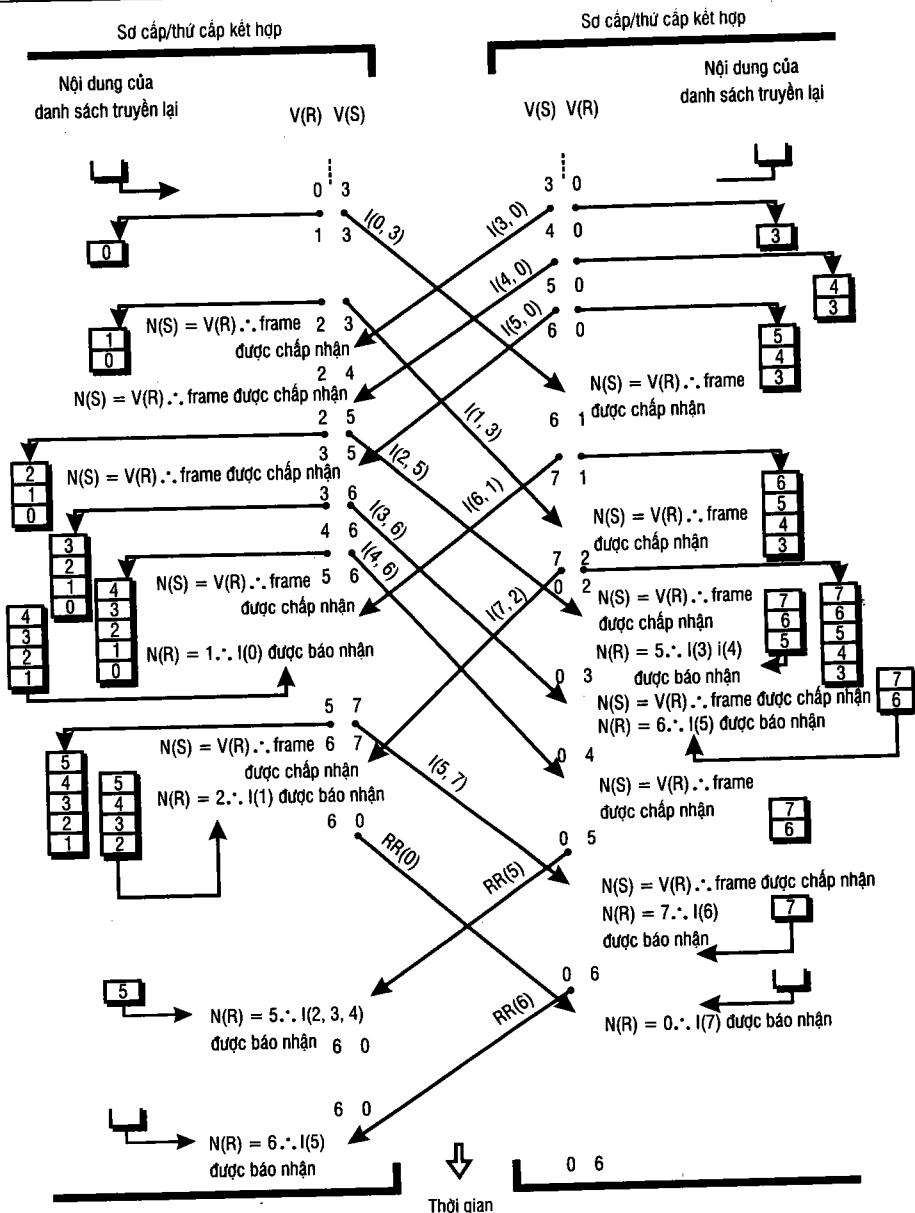
Hình 6.16(a) chỉ ra rằng bất kỳ một I-frame nào được nhận không đúng theo thứ tự đều bị loại. Do đó, khi tiếp nhận frame I(2,0/p = 1)_bit P được set vì frame trước đó không được báo nhận_frame này bị loại và máy thu không làm gì cả. Khi không có các báo nhận thì bộ định thời liên hệ với các frame I(1) và I(2) đều hết hạn và cả hai I-frame đều được truyền lại. Trong ví dụ này giả sử mỗi I-frame được nhận tốt và được báo nhận bằng RR-frame.

Trong hình 6.16(b), có sử dụng các frame thông báo từ chối (REJ). Khi máy thu phát hiện rằng frame I(2,0/P= 1)_ đó là frame cuối cùng trong tuần tự với P = 1_là không đúng tuần tự, nó gửi một REJ-frame trong đó bit F được set. Sau đó máy phát phải phát lại các frame I(1,0) và I(2,0), trong đó bit P một lần nữa được set bằng 1 trong frame I(2,0). Máy thu thông báo nhận tốt cho mỗi frame với bit F được set bằng 1 trong RR-frame sau cùng. Nếu dùng chiến lược truyền lại có chọn lựa thì frame I(2,0/P = 1) sẽ được chấp nhận và một SREJ-frame được gửi lại để yêu cầu truyền lại frame I(1,0).

Tuần tự frame trong hình 6.16 là ví dụ tiêu biểu về công việc chuyển thông tin qua một liên kết đa điểm hoạt động theo chế độ NRM. Tuy nhiên, đối với liên kết điểm-nối-diểm trong ABM, có thể cho phép một luồng I-frame song song. Giống như trong NRM, thông tin báo nhận liên hệ với luồng I-frame theo một hướng có thể được tải trong các I-frame chạy theo hướng ngược lại. Hình 6.17 là một ví dụ. Để việc trình bày được rõ ràng, ở đây không trình bày các lỗi truyền.

Khi nhận mỗi frame, cả N(S) và N(R) đều được đọc. Trước hết N(S) được so sánh với biến V(R) của máy thu. Nếu bằng nhau thì frame vừa nhận là đúng tuần tự và được chấp nhận; nếu chúng không bằng nhau, frame bị loại và một REJ-frame hay SREJ-frame được gửi trả lại. N(R) được kiểm tra và được dùng để báo nhận cho bất kỳ frame còn tồn tại nào trong danh sách truyền lại. Sau cùng, khi không còn I-frame nào đang đợi truyền, một RR-frame được dùng để báo nhận cho các I-frame chưa được báo nhận còn tồn tại trong mỗi danh sách truyền lại.

Điều khiển luồng là đặc biệt quan trọng khi truyền song song hoàn toàn và liên kết đang hoạt động trong chế độ ABM. Đối với NRM, nếu sơ cấp bị quá tải tạm thời, nó có thể làm công việc đơn giản là ngưng quét (polling), do đó tạo điều kiện cho quá tải lắng dịu. Tuy nhiên, khi cả hai phía của liên kết hoạt động một cách độc lập với nhau, chúng ta có thể dùng một cơ cấu khác. Thủ tục điều khiển luồng được dùng trong HDLC dựa vào *cơ cấu của sổ trượt* đã được trình bày trong chương 5.

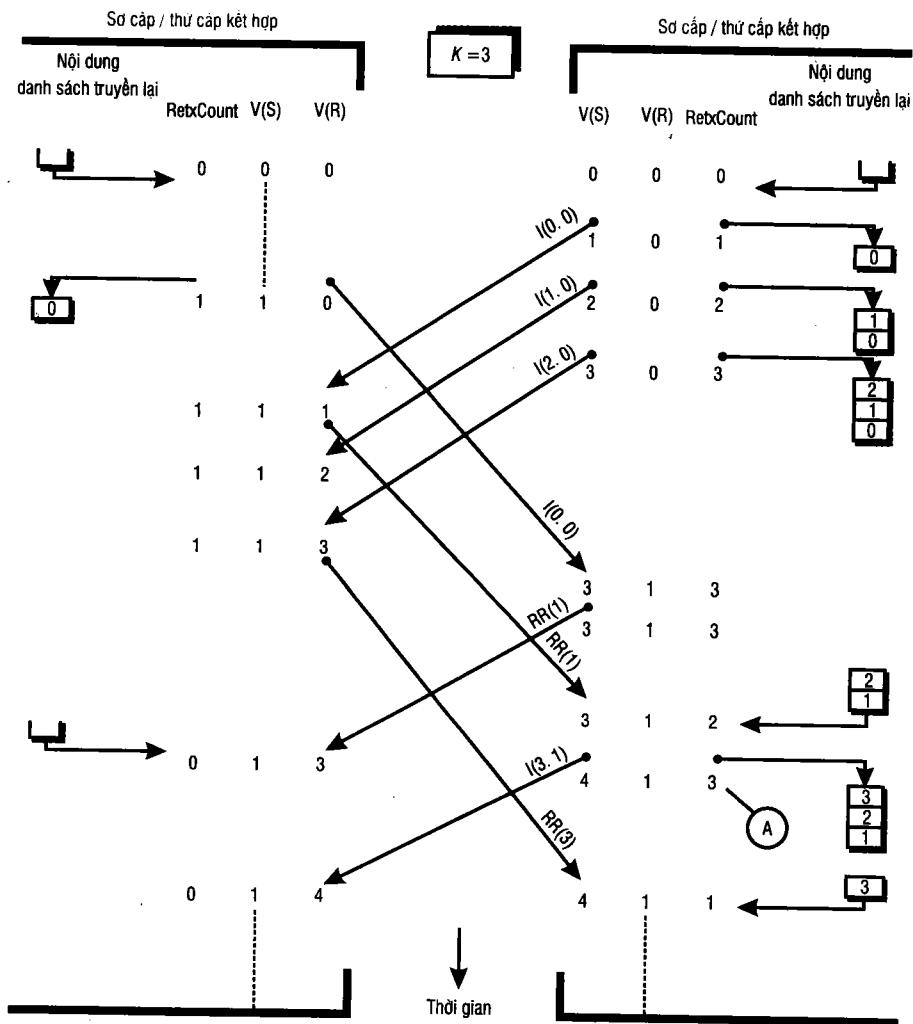


Hình 6.17 Thủ tục với báo nhận được gửi trong các I-frame theo hướng ngược lại.

Như chúng ta đã thấy trong hình 6.16 và 6.17, các chỉ số tuần tự truyền và nhận được tăng theo modulo-8, vì vậy kích thước cửa sổ truyền được dùng là K=7. Do đó, 7 là số lượng I-frame tối đa được phép đợi báo nhận trong danh sách truyền lại tại bất kỳ thời điểm nào. Mỗi phía của liên kết duy trì một biến riêng biệt được gọi là biến đếm truyền lại (RetxCount), nó được gán bằng 0 ngay khi

thiết lập liên kết. Biến này được tăng một đơn vị mỗi khi truyền một I-frame, do đó cũng có nghĩa là được tăng mỗi khi có một I-frame được đặt vào danh sách truyền lại. Biến này được giảm một đơn vị bất cứ lúc nào nhận được một thông báo nhận tốt, cũng có nghĩa là được giảm mỗi khi mỗi I-frame được xóa khỏi danh sách truyền lại. Sơ cấp ngưng truyền các I-frame khi biến truyền lại đạt đến K và không quay trở lại cho đến khi nhận được một thông báo nhận tốt thông qua RR-frame hoặc thông tin này được mang trong một I-frame chạy theo hướng ngược lại. Chúng ta có thể kết luận rằng sẽ ngưng truyền các I-frame khi:

$$V(S) = N(R)_{\text{nhận sau cung}} + K$$



(A) = Cửa sổ hoạt động

Hình 6.18 Thủ tục điều khiển luồng theo phương pháp cửa sổ trượt.

Lưu ý rằng cơ cấu cửa sổ chỉ điều khiển luồng I-frame theo một hướng và các frame quản lý cũng như các frame không đánh số không bị ảnh hưởng bởi cơ cấu này. Do đó các frame này vẫn có thể được truyền trong khi cửa sổ đã mở hết. Hình 6.18 là một ví dụ, để rõ ràng ở đây chỉ trình bày ảnh hưởng lên luồng I-frame theo một hướng.

Việc dùng cơ cấu cửa sổ có nghĩa là các chỉ số tuần tự trong tất cả các I-frame đến đều nằm trong các ranh giới nào đó. Khi nhận mỗi I-frame, thứ cấp phải kiểm tra xem có đúng như vậy không, và nếu không đúng thì phải tiến hành thao tác hiệu chỉnh. Do đó, mỗi N(S) và N(R) thu được phải thỏa mãn các điều kiện sau đây:

$$(1) V(R) \leq N(S) \leq V(R) + K$$

$$(2) V(S) \geq N(R) \geq V(S) - RetxCount \text{ (biến đếm truyền lại).}$$

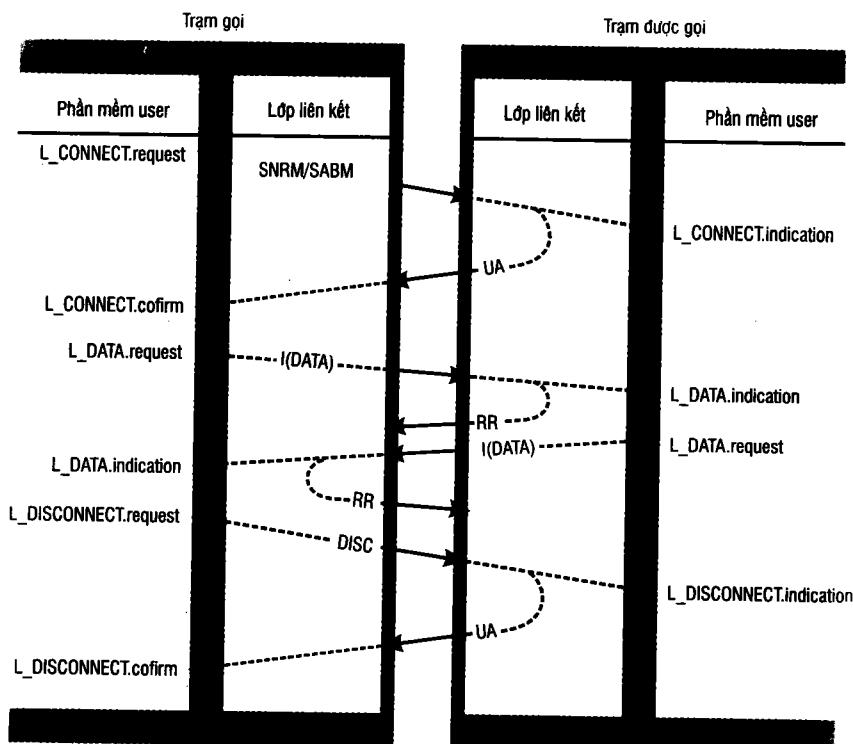
Nếu $N(S)$ bằng với $V(R)$, thì tất cả là tốt và I-frame được chấp nhận. Nếu $N(S)$ không bằng $V(R)$ nhưng vẫn còn trong dài, thì đơn giản là frame đã bị hỏng và một REJ-frame (go-back-N) hoặc SREJ-frame (selective repeat) được gửi đi để thông báo với sơ cấp rằng đã xảy ra lỗi tuần tự và truyền lại từ I-frame này. Điều này đã được mô tả trong hình 6.16.

Nếu $N(S)$ hoặc $N(R)$ nằm ngoài dài, thì chỉ số tuần tự tại cả hai đầu của liên kết trở nên bất đồng bộ và liên kết phải được khởi động lại. Điều này được thực hiện khi thứ cấp phát hiện chỉ số tuần tự nằm ngoài dài, nó loại bỏ frame vừa nhận và gửi một thông báo từ chối FRMR (chế độ ABM) hoặc một lệnh từ chối CMDR (chế độ NRM) cho sơ cấp. Sơ cấp loại bỏ tất cả các I-frame đang đợi và xúc tiến thiết lập liên kết trở lại bằng cách gửi một SAMB/SNRM và chờ đáp ứng UA. Khi nhận được đáp ứng này, cả hai phía của liên kết đều phục hồi (reset) các biến cửa sổ và biến tuần tự của chúng để cho phép luồng I-frame được tiếp tục cháy. Trên thực tế, đây chỉ là một nguyên nhân buộc phải thiết lập lại liên kết; còn có nguyên nhân khác là trong cung đoạn chuyển giao số liệu, lại nhận được một frame không đánh số chỉ ra rằng sơ cấp và thứ cấp đã ở trong tình trạng bất đồng bộ.

Thủ tục điều khiển luồng vừa trình bày được điều khiển bởi phía sơ cấp của liên kết, thủ tục này điều khiển luồng I-frame tùy theo cửa sổ truyền. Ngoài ra, cũng cần thiết để thứ cấp dừng luồng I-frame khi có biến cố nào đó xảy ra tại đó. Ví dụ đối với chiến lược truyền lại một nhóm (go-back-N), cửa sổ thu là 1 và là lý do dễ hiểu nhằm đảm bảo có đủ bộ nhớ tại đầu thu. Tuy nhiên, nếu chiến lược truyền lại có chọn lựa (selective repeat) được dùng thì hoàn toàn có thể xảy ra tình trạng thứ cấp không còn bộ đệm rỗng để chứa bất kỳ một I-frame mới nào. Do đó, đến một thời điểm mà tất cả các bộ đệm của thứ cấp gần như đầy, nó gửi một frame quản lý RNR-frame đến sơ cấp để yêu cầu ngừng ngay việc truyền I-frame mới. Dĩ nhiên, các frame báo nhận không bị ảnh hưởng. Khi số lượng bộ đệm bị đầy xuống thấp hơn một mức qui định trước, thứ cấp lại gửi một RR-frame cho sơ cấp, $N(R)$ trong đó chỉ ra I-frame mà từ đó nên bắt đầu truyền tiếp.

4.1.4. Giao tiếp user

Bây giờ chúng ta có thể xây dựng mối liên hệ giữa các dịch vụ user thuộc lớp liên kết được trình bày trong hình 6.1 với các đơn vị thông điệp của giao thức liên quan đến HDLC. Các liên hệ được trình bày trên hình 6.19.

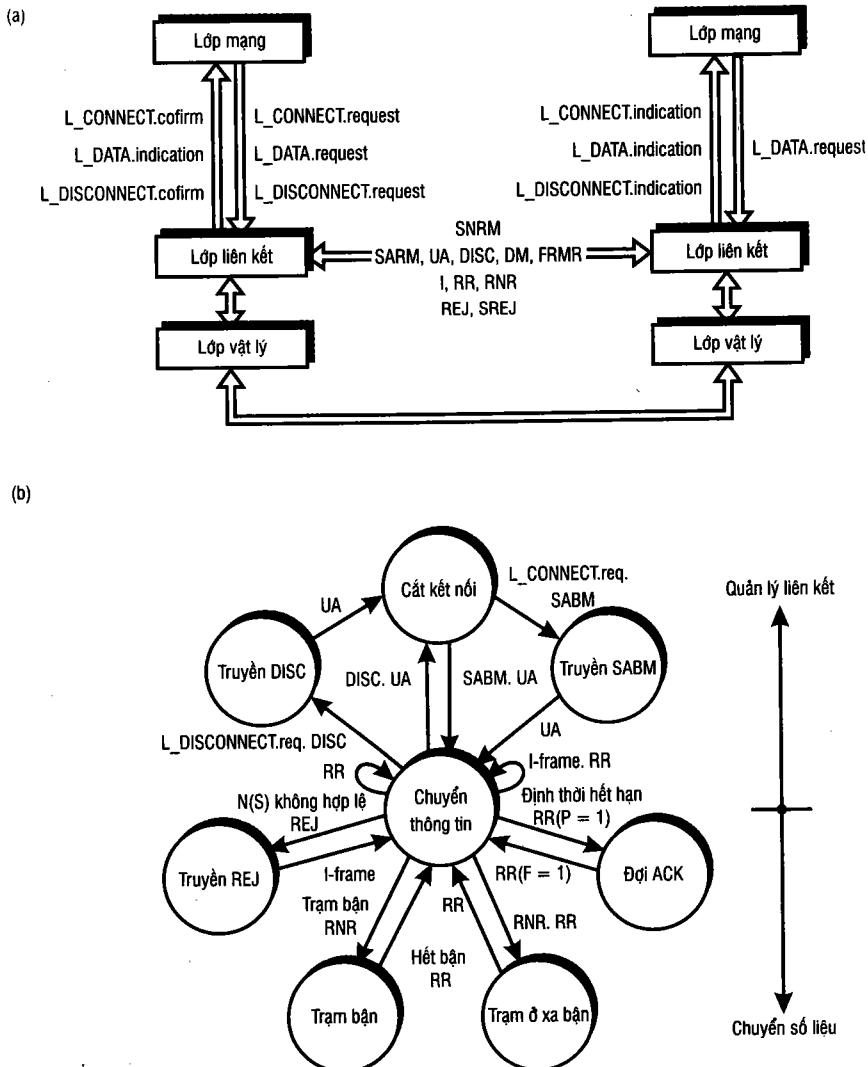


Hình 6.19 **Hoạt động giao tiếp giữa user và lớp liên kết.**

Khi tiếp nhận hàm thực thể khởi động L_CONNECT.request từ user, trước hết thực thể giao thức lớp liên kết trong hệ thống gọi (calling) gửi một frame quản lý SNRM/SABM cho thực thể giao thức lớp liên kết ở hệ thống được gọi (called). Khi nhận frame này, giao thức lớp liên kết tạo ra và chuyển một hàm thực thể L_CONNECT.indication cho user được gọi. Ngoài ra nó tạo ra một UA-frame và gửi frame này để đáp ứng cho phía gọi. Khi phía gọi nhận được UA-frame, giao thức lớp liên kết tại đây tạo ra hàm thực thể L_CONNECT.confirm và chuyển hàm này cho user để cho công việc truyền số liệu của user có thể bắt đầu bằng cách dùng dịch vụ L_DATA. Cuối cùng, khi tất cả các số liệu đã được truyền xong, liên kết được xóa bằng các frame quản lý DISC và UA.

Một tóm tắt về các hàm thực thể dịch vụ và các loại frame khác nhau (còn gọi là các đơn vị số liệu giao thức, viết tắt bằng từ tiếng Anh là PDU_Protocol Data Unit) liên hệ với HDLC được trình bày trên hình 6.20(a). Trong thực tế, còn

có nhiều frame không đánh số liên hệ với HDLC so với những gì trình bày trên hình này, mục tiêu ở đây nhằm đơn giản cho việc làm sáng tỏ các khía cạnh hoạt động của HDLC đang muốn trình bày. Một sơ đồ chuyển trạng thái đã được đơn giản hóa ở hình 6.20(b) giúp bạn đọc hiểu vấn đề chắc chắn hơn. Những gì ghi kèm theo mỗi mũi tên chính là sự kiện đến gây ra sự chuyển trạng thái (nếu có). Lưu ý rằng một sơ đồ chuyển trạng thái chỉ trình bày hoạt động đúng của thực thể giao thức; thông thường nó được kèm theo một định nghĩa hoàn chỉnh hơn dưới dạng một bảng sự kiện-trạng thái và (hoặc) mã giả (pseudocode).



Hình 6.20 *Tóm tắt LAP-B (a) Các hàm thực thi dịch vụ (b) Sơ đồ chuyển trạng thái (ABM).*

4.2. Thủ tục truy xuất liên kết LAPB

Thủ tục truy xuất liên kết phiên bản B còn được gọi là LAPB (Link Access Procedure version B) là một tập con của HDLC, nó được dùng để điều khiển truyền các I-frame qua một liên kết số liệu song công diêm-nối-diêm nối giữa một máy tính với một mạng chuyển mạch gói công cộng (hay tư nhân). Các mạng như vậy trước đây thường là mạng X.25. Thực ra LAPB là một phiên bản mở rộng của một tập con nổi tiếng của HDLC đó là thủ tục truy xuất liên kết phiên bản A hay còn gọi là LAPA.

Bảng 6.2

Loại	LAPA		LAPB	
	Các lệnh	Các đáp ứng	Các lệnh	Các đáp ứng
Quản lý	RR	RR RNR REJ	RR	RR RNR REJ
Không đánh số	SARM DISC	UA CMDR	SABM DISC	UA DM FRMR
Thông tin	I		I	

Khả năng ứng dụng của LAPB được trình bày trên hình 6.2(c). Máy tính là DTE và tổng đài chuyển mạch gói là thiết bị kết cuối mạch số liệu (DCE). LAPB được dùng để điều khiển truyền các frame thông tin qua giao tiếp DTE-DCE cục bộ và được gọi là *có ý nghĩa cục bộ*.

LAPB dùng chế độ cân bằng bất đồng bộ trong đó DTE và DCE như là các trạm kết hợp (sơ cấp/thứ cấp) và tất cả các I-frame được xem như các frame lệnh. Giao thức LAPA trước đây dùng chế độ đáp ứng bất đồng bộ và không dùng các REJ-frame hay RNR-frame như là các frame lệnh. Một tóm tắt các frame được dùng trong LAPA và LAPB được trình bày trong bảng 6.2. RR-frame và REJ-frame được dùng để kiểm soát lỗi và RNR được dùng để điều khiển luồng. Các frame này không hỗ trợ chiến lược truyền lại có chọn lựa (SREJ). Ví dụ về tuân tự frame được trình bày trong các hình trước đây liên quan đến HDLC đều đúng cho LAPB. Như chúng ta đã nói, truyền một frame thông tin (lệnh) với bit P được set thì kết quả là trạm nhận sẽ đáp ứng một frame quản lý với bit F cũng được set. Cả hai trạm đều có thể thiết lập liên kết. Để phân biệt giữa hai trạm, các địa chỉ DTE và DCE được dùng theo bảng 6.3. Nếu một DTE không hoạt động về mặt luận lý mà tiếp nhận một frame yêu cầu xây dựng liên kết (SABM/SABME), nó phải phúc đáp bằng một frame DM.

**Bảng 6.3**

HƯỚNG	CÁC ĐỊA CHỈ	
	Các lệnh	Các đáp ứng
DTE → DCE	01 Hex (B)	03 Hex(A)
DCE → DTE	03 Hex(A)	01 Hex(B)

Tóm tắt việc dùng bit P/F trong LAPB được trình bày trong bảng 6.4.

Nhớ rằng trong chế độ SABM dùng một octet (8 bit) cho field điều khiển. Còn các chỉ số tuần tự truyền và nhận, mỗi chỉ số chiếm 3 bit_có 8 chỉ số tuần tự_cho phép cửa sổ truyền có kích thước tối đa là 7. Nếu dùng chế độ mở rộng (SABME), thì field điều khiển có 2 octet. Do đó các chỉ số tuần tự truyền và nhận được mở rộng đến 7 bit_có 128 số tuần tự_lúc đó cho phép kích thước tối đa của cửa sổ truyền là127. Chế độ này được dùng cho các liên kết rất xa và các liên kết có tốc độ cao.

Ngày nay có sẵn các vi mạch tích hợp cỡ lớn (LSI) trong đó có thể hiện thực LAPB và ghi trong bộ nhớ đặc biệt, được gọi là phần mềm LAPB. Các vi mạch này thường được xem là các mạch X.25 mặc dù trong đó chỉ hiện thực LAPB thay vì hiện thực đầy đủ tập giao thức X.25. Sự xuất hiện các vi mạch này làm gia tăng đáng kể việc sử dụng LAPB trong các ứng dụng có sử dụng truyền tin giữa máy tính với máy tính.

Bảng 6.4

Frame lệnh được gửi với P=1	Frame đáp ứng được gửi với F=1
SABM/SABME	UA/DM
I-frame	RR,REJ,RNR
RR,REJ,RNR	FRMR
DISC	RR,REJ,RNR
	FRMR
	UA/DM

4.3. Thủ tục đa truy xuất

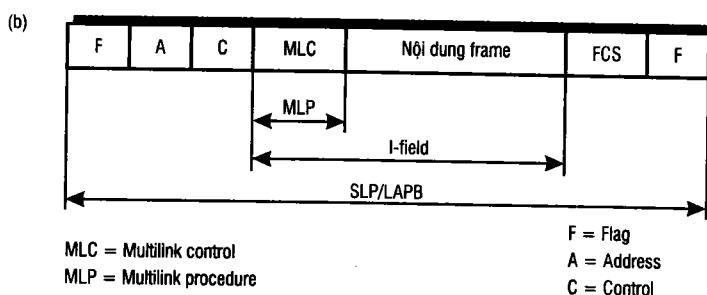
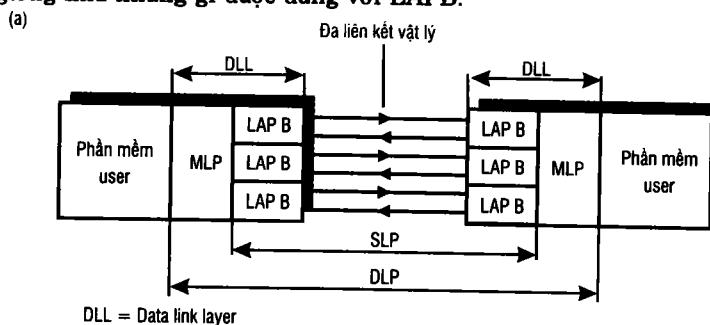
Chúng ta đã mô tả việc sử dụng HDLC để điều khiển truyền các frame số liệu qua một liên kết song công. Vì HDLC điều khiển truyền qua liên kết đơn như vậy nên còn được gọi là thủ tục liên kết đơn SLP (Single Link Procedure). Tuy nhiên, trong một vài trường hợp, thông lượng có sẵn của một liên kết đơn như vậy không đủ đáp ứng nhu cầu của các ứng dụng, vì vậy chúng ta phải dùng đến đa

liên kết. Để phục vụ cho điều này, một thủ tục mở rộng của LAPB đã được định nghĩa và được gọi là thủ tục đa liên kết MLP (Multilink Procedure).

Hình 6.21(a) trình bày hoạt động chuyển các frame số liệu qua mỗi liên kết vật lý được điều khiển bởi một thủ tục SLP theo như cách đã được mô tả. Một MLP hoạt động trên một tập các thủ tục liên kết và xem chúng đơn giản như là một nhóm các liên kết có sẵn để truyền thông tin của user. Điều này có nghĩa là phần mềm user không hề biết có nhiều liên kết vật lý đang được dùng và cùi xúi như một giao tiếp liên kết luận lý trước đây.

MLP xem tập các thủ tục liên kết như là một nhóm liên kết mà qua đó sẽ truyền các frame của user. Nó hoạt động với tập các số tuần tự riêng và các thủ tục điều khiển luồng cũng như kiểm soát lỗi là độc lập trong từng SLP. Do đó nếu một SLP tự dừng không hoạt động, thì MLP sẽ khởi động lại các frame theo cách thức thông thường nhưng dùng tập liên kết có sẵn nhỏ hơn.

Để thực hiện lược đồ này, MLP thêm một field điều khiển vào đầu của mỗi frame mà nó tiếp nhận để truyền trước khi chuyển frame này cho một SLP. Vùng này được gọi là vùng *điều khiển đa liên kết* MLC (Multilink Control) và khá trong suốt đối với một SLP. SLP xem phần MLC kết hợp và phần nội dung frame như là vùng thông tin thống nhất và xử lý thêm vùng địa chỉ (A) và vùng điều khiển (C) như trên hình 6.21(b). Các cơ cấu điều khiển luồng và kiểm soát lỗi liên hệ với MLP rất giống như những gì được dùng với LAPB.



Hình 6.21 Thủ tục đa liên kết: (a) Vị trí của thủ tục này trong mỗi liên hệ với lớp liên kết số liệu (b) Khuôn dạng của frame.

Vùng MLC bao gồm hai octet và chứa một chỉ số tuần tự 12 bit. Điều này cung cấp 4096 (0 đến 4095) số tuần tự và do đó kích thước tối đa của cửa sổ truyền là 4095, cho phép một số lượng liên kết đáng kể, mỗi liên kết có khả năng hoạt động với tốc độ cao. Ví dụ như khi hai mạng chuyển mạch gói X.25 được nối với nhau.

4.4. Thủ tục truy xuất liên kết LAPM

Các modem có khả năng khắc phục lỗi ngày nay sử dụng một thủ tục được gọi là LAPM (Link Access Procedure for Modem). Thông qua thủ tục này chúng có thể chấp nhận số liệu được truyền bắt đồng bộ từ DTE nhưng sẽ truyền số liệu đi theo chế độ đồng bộ thiên hướng bit (bit-oriented) và dùng một giao thức khắc phục lỗi dựa trên HDLC.

Ứng dụng của LAPM được minh họa trên hình 6.22.

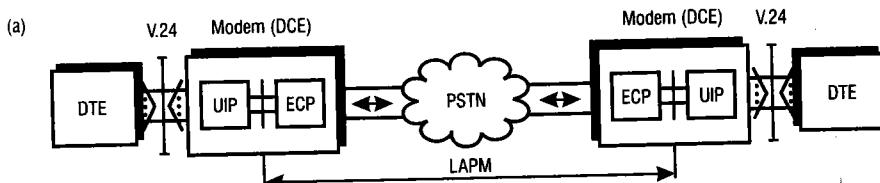
Mỗi modem sẽ gồm hai đơn vị chức năng :

- Phần giao tiếp với user UIP: User Interface Part.
- Phần sửa lỗi ECP :Error Correcting Part.

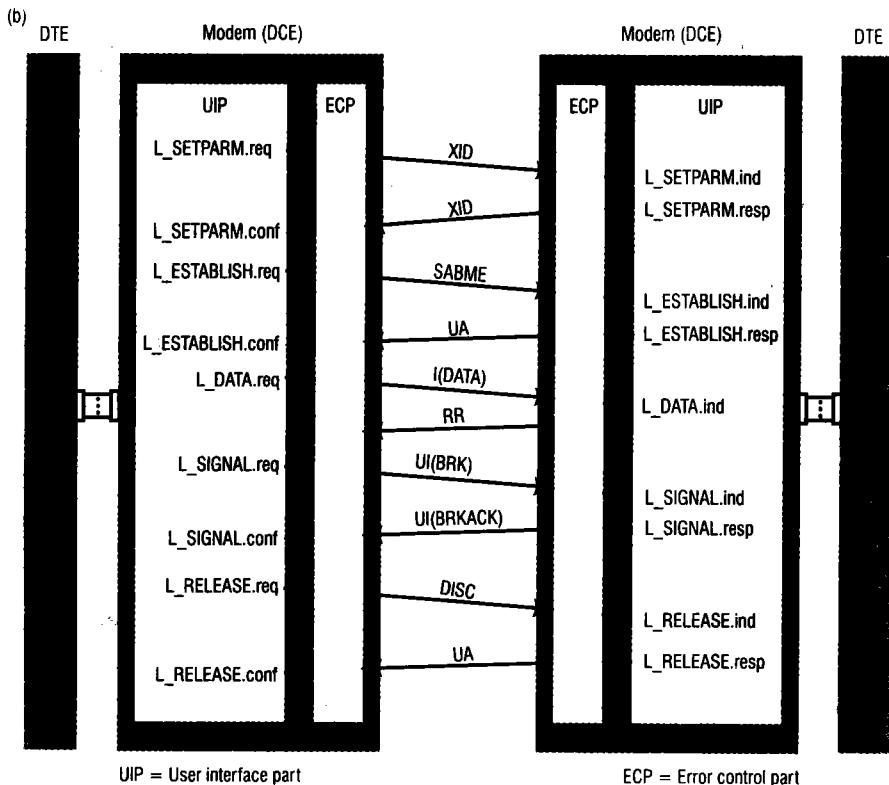
Thủ tục LAPM có liên hệ với phần thứ hai ECP trong khi UIP liên quan đến việc truyền các byte hay ký tự dữ liệu xuyên qua giao tiếp V.24 và phiên dịch bất cứ tín hiệu điều khiển luồng nào xuyên qua giao tiếp.

Thành phần UIP trao đổi với ECP bằng cách dùng tập các hàm thực thể dịch vụ như minh họa trên hình 6.22. Ở đây các loại frame của HDLC được sử dụng để thực hiện dịch vụ.

Trước khi thiết lập một liên kết luận lý, các ECP phải thống nhất các tham số hoạt động được dùng với giao thức. Các tham số bao gồm số byte tối đa trong frame thông tin (I-frame), cài đặt bộ định thời báo nhận, số lần truyền lại tối đa và kích thước cửa sổ (window size). Thông thường mỗi tham số đều có giá trị mặc định cho trước, nhưng nếu vì mục đích nào đó cần thay đổi chúng thì UIP của modem phát phải công bố một thực thể yêu cầu L_setparm kèm theo các giá trị của từng tham số hoạt động. Các giá trị này được trao đổi khi hai ECP trao đổi hai frame không đánh số đặc biệt, được gọi là XID (exchange identification), một cho ý nghĩa lệnh (command) và một cho ý nghĩa đáp ứng (response).



Hình 6.22 LAPM: (a) Lược đồ tổng quát



Hình 6.22 (tiếp theo) LAPM: (b) Các thực thể dịch vụ và các loại frame tương ứng.

Khi các tham số hoạt động đã thông nhất, một liên kết luận lý có thể được thiết lập khi UIP của modem gọi phát ra hàm thực thể L_ESTABLISH.request. Điều này khiến cho ECP phát frame điều hành SABM hay SABME. ECP thu sau khi nhận SABM sẽ đáp ứng bằng hàm thực thể L_ESTABLISH.indication đến UIP nội bộ, trên thực thể đáp ứng của mình ECP thu sẽ gửi một UA-frame. Khi nhận được frame này, ECP gọi sẽ phát ra hàm thực thể xác nhận và liên kết luận lý được thiết lập. Việc truyền số liệu có thể được bắt đầu bằng cách dùng dịch vụ L_DATA.

Đầu tiên UIP đóng gói dữ liệu nhận được qua giao tiếp V.24, sau đó chuyển gói hoàn chỉnh này đến ECP bằng cách dùng hàm thực thể L_DATA.request. ECP sẽ gói dữ liệu này trong vùng thông tin của I-frame và truyền chúng đi dưới sự giám sát của thủ tục kiểm soát lỗi trong HDLC. Sau đó ECP ở đầu thu sẽ chuyển khỏi dữ liệu đến UIP nội bộ và chúng sẽ tiếp tục được truyền qua giao tiếp V.24.

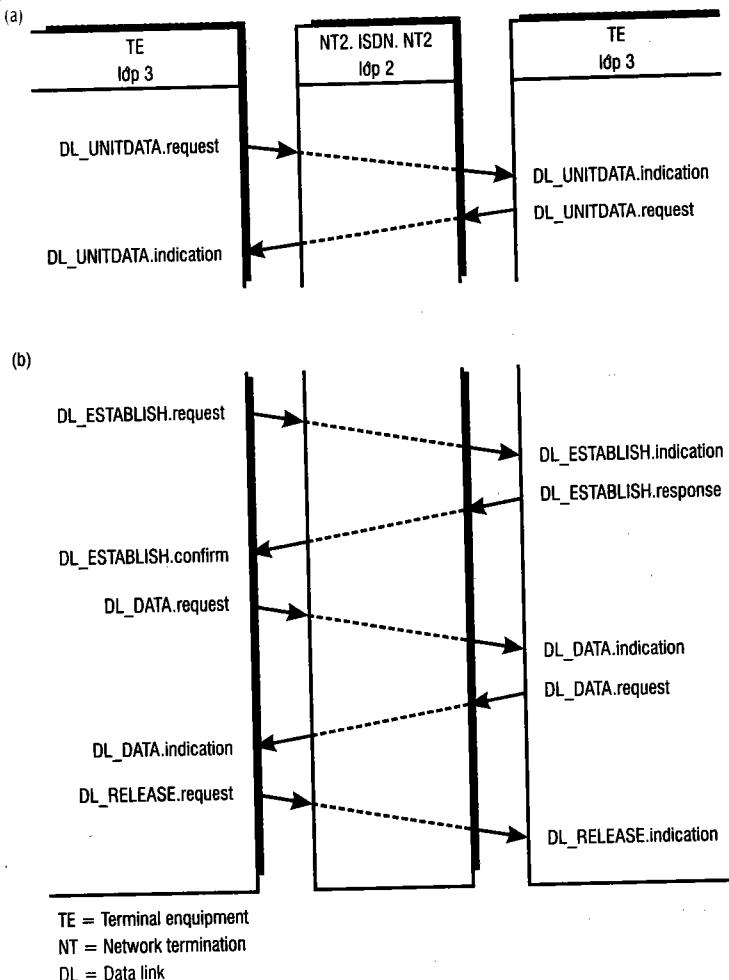
Nếu một tác nhân điều khiển luồng (flow control) được phát hiện trong cung đoạn chuyển dữ liệu ví dụ có ký tự X-OFF nhận được hay tín hiệu trên đường DTR chuyển về mức 'không tích cực' thì UIP ngưng đổ dữ liệu đến DTE cục bộ và phát hàm thực thể L_SIGNAL.request ngay tức thì đến ECP cục bộ. ECP cục bộ sau đó thông báo với ECP đầu xa bằng báo hiệu BRK để ngưng truyền tạm thời. BRK là frame thông tin đặc biệt được xem như là một UI vì không có số tuần tự cho cơ cấu kiểm soát lỗi. Sau khi nhận BRK, ECP sẽ phát L_SIGNAL.indication đến UIP nội bộ của nó và đáp ứng lại bằng một frame UI khác có tên là BRKACK. Sau đó

UIP cũng sẽ kích hoạt tín hiệu điều khiển luồng tương tự xuyên qua giao tiếp V.24 của nó.

Sau cùng, khi tất cả số liệu đã được truyền, liên kết sẽ bị xoá do UIP nguồn truyền hàm thực thể L_RELEASE.request. Đây là một dịch vụ có xác nhận và các frame của LAPM có liên hệ gồm DISC và UA.

4.5. Thủ tục truy xuất liên kết LAPD

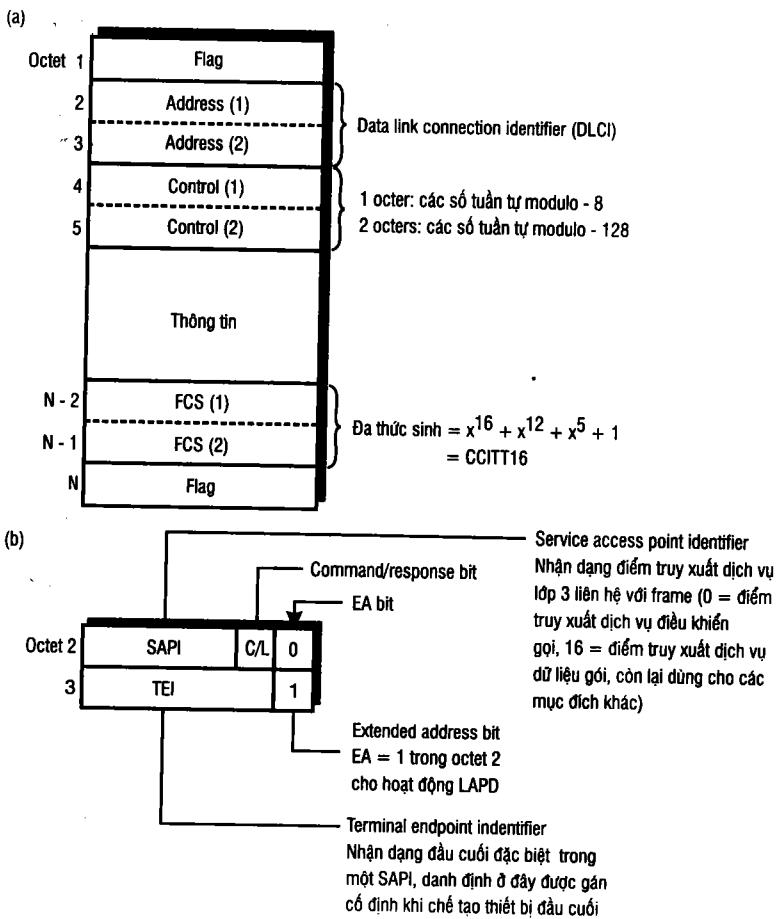
Thủ tục truy xuất liên kết kênh D gọi tắt là LAPD (Link Access Procedure D-channel) là một tập con của HDLC dùng cho ISDN. Nó được định nghĩa để điều khiển luồng I-frame liên quan mật thiết với kênh báo hiệu. Kênh báo hiệu được gọi là kênh D. LAPD còn được dùng dưới dạng mở rộng để điều khiển luồng I-frame qua một kênh thuê bao liên quan đến một dịch vụ được gọi là *tiếp frame (frame relay)*. Chúng ta sẽ có dịp bàn đến ISDN trong phần mạng diện rộng _WAN. Ở đây chúng ta chỉ đề cập đến các hoạt động cơ bản của LAPD và xem nó liên hệ với HDLC như thế nào.



Hình 6.23 Các hàm thực thể dịch vụ user của LAPD:

(a) Không tạo cầu nối (b) Có tạo cầu nối.

Có hai loại dịch vụ đã được định nghĩa để dùng với LAPD. Một sơ đồ tuân tự theo thời gian mô tả hai tập hàm thực thể dịch vụ được trình bày trên hình 6.23. Như chúng ta có thể thấy, cả hai loại dịch vụ có tạo cầu nối và không tạo cầu nối đều được hỗ trợ. Giống như PSTN analog, ISDN là mạng chuyển mạch-mạch (circuit-switching network) có nghĩa là cần thiếp lập một mạch trước khi bất kỳ thông tin nào được truyền. Điều này được thực hiện bằng cách dùng kênh báo hiệu riêng_kênh D_ có một tập giao thức riêng trong đó LAPD là một phần cấu thành.



Hình 6.24 LAPD: (a) Khuôn dạng frame (b) Cách dùng field địa chỉ.

Dịch vụ tạo cầu nối được dùng để truyền các thông điệp thiết lập gọi giữa một thiết bị đầu cuối và tổng đài cục bộ. Giao thức liên hệ có kết hợp với kiểm soát lỗi. Dịch vụ không tạo cầu nối được dùng để truyền các bản tin liên quan đến quản lý và giao thức liên quan dùng tiếp cận tổng lực (best-try, cách diễn tả khác của dịch vụ không tạo cầu nối) không báo nhận.

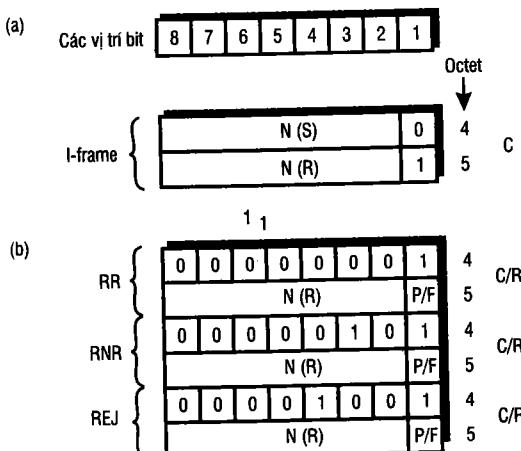


Chúng ta sẽ thấy, có đến 8 thiết bị đầu cuối có thể chia sẻ một mạch truy xuất cơ bản (do đó cũng chia sẻ cùng kênh D) giữa một vị trí khách hàng và tổng đài cục bộ ISDN. Tuy nhiên, tất cả các bản tin thiết lập gọi đều được gửi đến thiết bị kết cuối đặc biệt được chỉ ra trong vùng địa chỉ LAPD. Điều này giống với nguyên lý địa chỉ hóa được dùng trong chế độ NRM, ngoại trừ LAPD không có máy chủ (master) và kiến trúc bus vật lý của nó cho phép các đầu cuối được nối vào truy xuất bus theo một phương thức bình đẳng. Cấu trúc tổng quát của mỗi frame LAPD được trình bày trên hình 6.24.

Hai octet được dùng cho vùng địa chỉ. Chúng bao gồm hai địa chỉ thành phần: một danh định của điểm truy xuất dịch vụ SAPI (Service Access Point Identifier) và một danh định của thiết bị cuối TEI (Terminal Endpoint Identifier). Về cơ bản, SAPI định danh một lớp dịch vụ mà đầu cuối liên hệ thoại, số liệu, vừa thoại vừa số liệu_ và TEI định danh duy nhất cho thiết bị cuối trong lớp dịch vụ đó. Cũng có một địa chỉ broadcast (tất cả các bit đều là 1) cho phép gửi thông điệp đến tất cả các đầu cuối trong một lớp. Ví dụ, có thể sử dụng để tất cả các máy điện thoại đều nhận một thông điệp yêu cầu thiết lập gọi được gửi đến.

Các bố trí khác của vùng điều khiển (có 4 và 5 octet) liên hệ với LAPD được tóm tắt trong hình 6.25, ở đây cũng trình bày các frame có thể được truyền như là các lệnh và các frame có thể được truyền như là các đáp ứng.

Cũng như LAPM, trong LAPD các frame không đánh số thường được xem như thông tin không đánh số UI (unnumbered information) cũng được dùng. LAPD dùng UI với dịch vụ không tạo cầu nối. Vì không có kiểm soát lỗi đi kèm với dịch vụ này, nên tất cả các thông tin được gửi đi với một field điều khiển không có N(S) cũng không có N(R). Các frame này có một vùng FCS; nếu bị hỏng, thì đơn giản là loại bỏ frame. Thông thường với dịch vụ này, sau đó lớp cao hơn phải phát hiện frame bị loại bỏ ví dụ như thiếu một đáp ứng thích hợp (cũng trong một UI-frame)_và thực hiện một nỗ lực khác. Như chúng ta sẽ thấy trong mục 4.6, thủ tục này cũng được dùng trong các mạng LAN.



Hình 6.25 Định nghĩa các vùng bit điều khiển trong LAPD:
(a) Thông tin (b) Quản lý

(c)

SABME	0	1	1	P	1	1	1	1	4	C
DM	0	0	0	F	1	1	1	1	4	R
UI	0	0	0	P	0	0	1	1	4	C
DISC	0	1	0	P	0	0	1	1	4	C
UA	0	1	1	F	0	0	1	1	4	R
FRMR	1	0	0	F	0	1	1	1	4	R

C = Command
R = Response

Hình 6.25 (tiếp theo) Định nghĩa các vùng bit điều khiển trong LAPD:
(c) Không đánh số.

Đặc tả giao thức và định nghĩa dịch vụ LAPD lần lượt được đặc tả trong các khuyến nghị I.441 và I.440 của ITU-T, tương tự như các khuyến nghị Q.921 và Q.920.

4.6. Điều khiển liên kết luận lý

Điều khiển liên kết luận lý LLC (Logical Link Control) là một dẫn xuất của HDLC được dùng trong các mạng LAN. Các mạng LAN được xem xét chi tiết trong chương 7, thế nhưng tổ chức tổng quát của hai loại topo cơ bản là bus và ring cùng với phạm vi của DLP (LLC) đều đã được trình bày sớm trong hình 6.2.

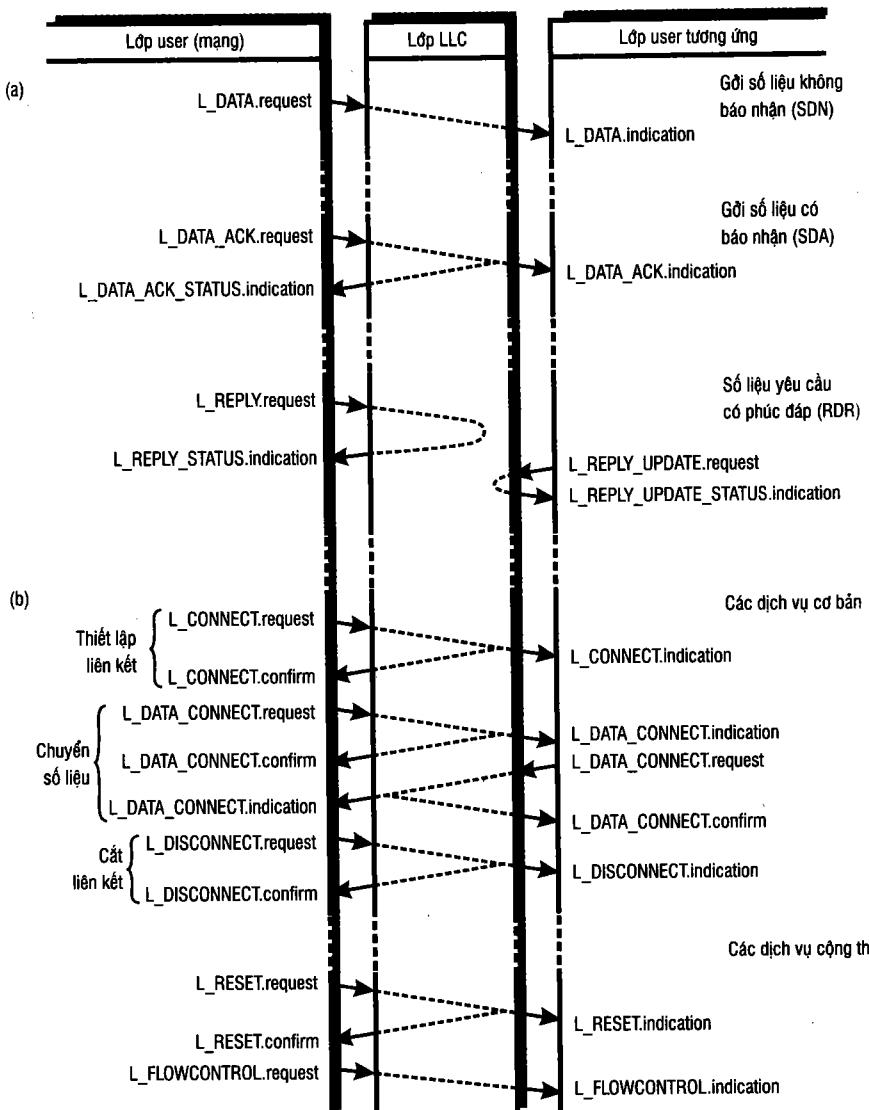
Cả hai topo đều dùng một môi trường chia sẻ_bus hay ring_là nơi diễn ra tất cả các hoạt động truyền frame. Giống như một mạng đa điểm, chúng ta cần một phương thức điều khiển truyền frame có trật tự. Không giống như các mạng đa điểm, không có máy tính chủ, vì vậy cần một giải thuật phân tán đảm bảo rằng môi trường được sử dụng theo một phương thức bình đẳng bởi tất cả các DTE kết nối vào đó. Đối với LAN, lớp liên kết số liệu bao gồm có hai lớp con: lớp con điều khiển truy xuất môi trường MAC (Medium Access Control), ở đó hiện thực giải thuật điều khiển truy xuất phân tán, và lớp con LLC. Hoạt động chi tiết của các lớp MAC khác nhau sẽ được mô tả trong chương 7; phần này chỉ tập trung vào hoạt động của lớp LLC. Lưu ý rằng đối với một LAN, vì không có các tổng đài chuyển mạch trong mạng, nên lớp LLC hoạt động dựa theo giao tiếp ngang hàng (peer basis), nghĩa là trao đổi diễn ra giữa hai lớp LLC trong hai đầu cuối đang truyền tin cho nhau.

4.6.1. Các dịch vụ user

Hai loại dịch vụ user được LLC cung cấp là: dịch vụ *không tạo cầu nối không báo nhận* và *tập các dịch vụ có tạo cầu nối*. Dịch vụ *không tạo cầu nối không báo nhận* cho phép user khởi động truyền các đơn vị số liệu dịch vụ với một lượng thông tin giao thức (overhead) tối thiểu. Thông thường dịch vụ này được dùng khi mà các chức năng như khắc phục lỗi và giám sát tuần tự được cung cấp trong một lớp giao thức cao hơn và không cần phải lặp lại trong lớp LLC. Ở phương diện khác, các dịch vụ có tạo cầu nối cho phép user thiết lập một cầu nối luận lý mức liên kết trước khi bắt đầu cung đoạn chuyển số liệu, và nếu cần, thực hiện luôn chức năng khắc phục lỗi và thứ tự hóa luồng thông tin qua một cầu nối đã được thiết lập.

Trong các ứng dụng mạng LAN theo thời gian thực nào đó, ví dụ như trong công tác điều khiển quá trình cho các trang thiết bị có thành phần chủ yếu là máy tính, thực hiện chức năng tạo kết nối thống nhất phân bố xung quanh một nhà máy hóa chất, thì sự xuất hiện lượng thời gian ban đầu để thiết lập một cầu nối

luận lý trước khi truyền số liệu là không thể chấp nhận được. Tuy nhiên, thông thường cũng cần đến vài báo nhận thành công cho một đơn vị số liệu nào đó đã truyền, vì vậy dịch vụ không tạo cầu nối không báo nhận cũng không thể áp dụng được. Một dịch vụ tăng cường, được gọi là dịch vụ *không tạo cầu nối có báo nhận* được định nghĩa nhằm phục vụ cho loại nhu cầu này. Tương tự, còn có một dịch vụ được gọi là *dịch vụ lấy phúc đáp* (obtain reply service) cho phép user ở xa yêu cầu số liệu mà không cần phải thiết lập cầu nối trước.



Hình 6.26 Các hàm thực thể dịch vụ user của LLC:
(a) Không tạo cầu nối (b) Có tạo cầu nối.

Các hàm thực thể khác nhau liên kết với hai tập dịch vụ được trình bày một cách tuần tự theo thời gian trên hình 6.26. Mỗi hàm thực thể được mô tả đều có các tham số liên hệ với nó. Chúng bao gồm một đặc tả địa chỉ nguồn và địa chỉ đích. Hai địa chỉ này có mặt trong tất cả các hàm thực thể, chỉ rõ phần tối thiểu của các địa chỉ vật lý đang được dùng trên môi trường mạng. Tuy nhiên, thông thường cả hai địa chỉ này là sự phối hợp của các địa chỉ được dùng trên môi trường vật lý và danh định của điểm truy xuất dịch vụ cục bộ (LLC-SAP). Chúng ta sẽ xem xét LLC-SAP sâu hơn trong chương 7.

Với dịch vụ không tạo cầu nối không báo nhận, khi thực thể giao thức LLC nhận một hàm thực thể yêu cầu truyền số liệu (L_DATA.request), nó cố gắng tối đa để truyền số liệu bằng cách dùng lớp con MAC. Không có xác nhận nào về cuộc truyền thành công hay thất bại. Tuy nhiên, với dịch vụ không tạo cầu nối có báo nhận, user được thông báo thành công hay thất bại của việc chuyển L_DATA_ACKNOWLEDGE.indication đến user ở xa nhờ phương tiện là hàm thực thể L_DATA_ACKNOWLEDGE_STATUS.indication.

Các hàm thực thể khác nhau phối hợp với *dịch vụ lấy phúc đáp* cho phép một user:

- Yêu cầu nội dung của một bộ đệm thông điệp đang được duy trì bởi một thực thể LLC ở xa (L_REPLY.request/indication)
- Cập nhật nội dung của một bộ đệm thông điệp được duy trì bởi thực thể LLC cục bộ của nó (L_REPLY_UPDATE.request và L_REPLY_UPDATE_STATUS.indication)

Với dịch vụ có tạo cầu nối, một cầu nối luận lý phải được thiết lập bằng cách dùng các hàm thực thể L_CONNECT trước khi truyền bất kỳ số liệu nào. Tương tự, sau khi tất cả số liệu đã được truyền, cầu nối phải được xóa bởi hàm thực thể L_DISCONNECT. Trong cung đoạn truyền số liệu, việc tiếp nhận một đơn vị số liệu tốt đều được báo nhận thành công bởi thực thể LLC ở xa, được thực thể nội bộ đổi thành một hàm thực thể L_DATA_CONNECT.confirm và chuyển cho user.

Các hàm thực thể RESET và FLOWCONTROL cho phép user điều khiển luồng số liệu dịch vụ qua một cầu nối đã được thiết lập. Dịch vụ RESET có một hành động chịu thất bại sớm (abortive) được xem như là kết cục của sự kiện khi có bất kỳ số liệu không được báo nhận nào đang bị loại bỏ. Do đó, nó chỉ được dùng nếu thực thể giao thức lớp mạng không bám được tuân tự của các đơn vị số liệu đang được truyền.

Hai hàm thực thể điều khiển luồng chỉ có ý nghĩa cục bộ là: L_FLOWCONTROL.request chỉ ra lượng số liệu mà user được chỉ định để chấp nhận từ thực thể giao thức LLC cục bộ của nó và hàm thực thể L_FLOWCONTROL.indication chỉ ra lượng số liệu mà thực thể giao thức LLC chuẩn bị chấp nhận từ user, cả hai đều liên kết với một cầu nối xác định. Nếu số lượng được chỉ định bằng 0 thì luồng dữ liệu sẽ dừng; nếu số lượng là không xác định, có nghĩa là không có điều khiển luồng trên cầu nối này. Số lượng cho phép được cập nhật một cách linh động tùy vào mỗi yêu cầu.

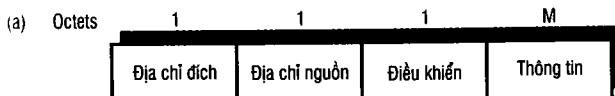


4.6.2. Hoạt động của giao thức

Khuôn dạng của mỗi frame LLC được trình bày trên hình 6.27(a). Cả địa chỉ nguồn và đích chỉ liên quan đến điểm truy xuất dịch vụ LLC; chúng không chứa các địa chỉ được dùng trên môi trường vật lý. Không có vùng FCS. Về cơ bản, frame LLC hoàn chỉnh được chuyển đến lớp con MAC dưới dạng một hàm thực thể, trong đó gồm có frame này và địa chỉ được dùng trên môi trường vật lý và được xem như các tham số. Lớp con MAC nắm giữ các chức năng địa chỉ hóa cho mạng và phát hiện lỗi. Do đó, trong ngữ cảnh của mô hình tham chiếu ISO, lớp liên kết số liệu (data link) tương đương với một phần của lớp MAC cộng với lớp LLC.

Field điều khiển trong mỗi frame có một octet. Nó định nghĩa loại frame, và ở những nơi thích hợp nó còn có chỉ số tuần tự truyền và nhận phục vụ cho điều khiển luồng và kiểm soát lỗi. Việc sử dụng các bit khác nhau trong field này được trình bày chi tiết trên hình 6.27(b).

Thực thể giao thức LLC hỗ trợ hai loại hoạt động: loại 1 hỗ trợ dịch vụ không tạo cầu nối không báo nhận và loại 2 phục vụ cho dịch vụ tạo cầu nối. Trong thực tế, loại 2 rất giống với giao thức HDLC ngoại trừ các chức năng như phát hiện lỗi và định dạng frame đều được cung cấp bởi lớp MAC.



(b)

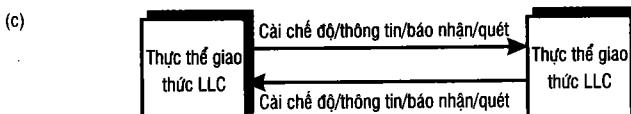
Bits	1	2	3	4	5	6	7	8
0	N(S)			P/F	N(R)			Các frame thông tin
1	0	S		P/F	N(R)			Các frame quản lý
1	1	M		P/F	M			Các frame không đánh số

N(S), N(R) = Số tuần tự truyền và nhận

S = Xác định chức năng quản lý

M = Xác định chức năng hiệu chỉnh

P/F = Poll/Final bit



Hình 6.27 Các khía cạnh của giao thức LLC:

- (a) Khuôn dạng frame
- (b) Các định nghĩa bit trong field điều khiển
- (c) Chức năng điều khiển liên kết dữ liệu (loại 2).



Các chức năng điều khiển liên kết số liệu cho hoạt động của loại 2 được trình bày trên hình 6.27(c). Khác nhau chủ yếu giữa giao thức LLC và HDLC là sự cung cấp dịch vụ không tạo cầu nối không báo nhận. Tập các lệnh và các đáp ứng trong loại 1 được liệt kê trong bảng 6.5.

Frame lệnh UI được dùng để gửi một khối số liệu đến một hay nhiều LLC. Vì không có báo nhận hay không có điều khiển tuần tự liên kết với hoạt động của loại 1 nên UI-frame không chứa N(S) và N(R). Cũng không có đáp ứng cho một UI.

Các frame lệnh XID và TEST là tùy chọn. Tuy nhiên, nếu chúng được gửi đi thì LLC nhận biết phải đáp ứng. Việc sử dụng các lệnh này như sau:

- Lệnh XID với một địa chỉ nhóm xác định thành viên hiện hành của nhóm này. Mỗi thành viên của nhóm đáp ứng cho lệnh bằng cách gửi một frame đáp ứng XID trực chỉ đến thực thể LLC gốc.
- Một thực thể LLC có thể dùng một lệnh XID với địa chỉ đích là địa chỉ broadcast để thông báo sự hiện diện của nó trên môi trường mạng.
- Lệnh TEST cung cấp một phương tiện kiểm thử vòng (loopback test) trên mỗi đường truyền dẫn LLC-đến-LLC.

Bảng 6.5

Các lệnh	Các đáp ứng
UI	-
XID	XID
TEST	TEST

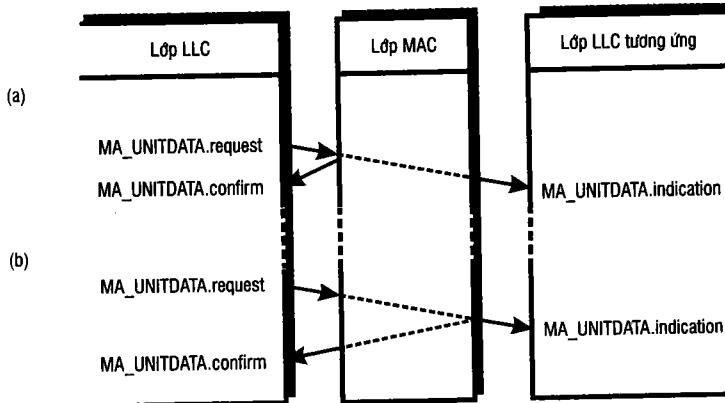
4.6.3. Các dịch vụ MAC

Bất chấp chế độ hoạt động nào ở lớp MAC bên dưới, luôn có một tập các dịch vụ user chuẩn được định nghĩa cho lớp LLC dùng để truyền các frame LLC đến lớp LLC tương ứng. Các hàm thực thể dịch vụ user được hỗ trợ gồm:

- MA_UNITDATA.request
- MA_UNITDATA.indication
- MA_UNITDATA.confirmation

Một sơ đồ tuần tự theo thời gian mô tả việc sử dụng chúng được trình bày trên hình 6.28. Trong vài mạng LAN, hàm thực thể xác nhận chỉ ra rằng yêu cầu đã được **chuyển đi** thành công (hay thất bại)_phần (a)_trong khi đó đối với vài

mạng LAN khác thì chỉ ra rằng yêu cầu đã được **phân phối** thành công (hay thất bại)_phần (b).

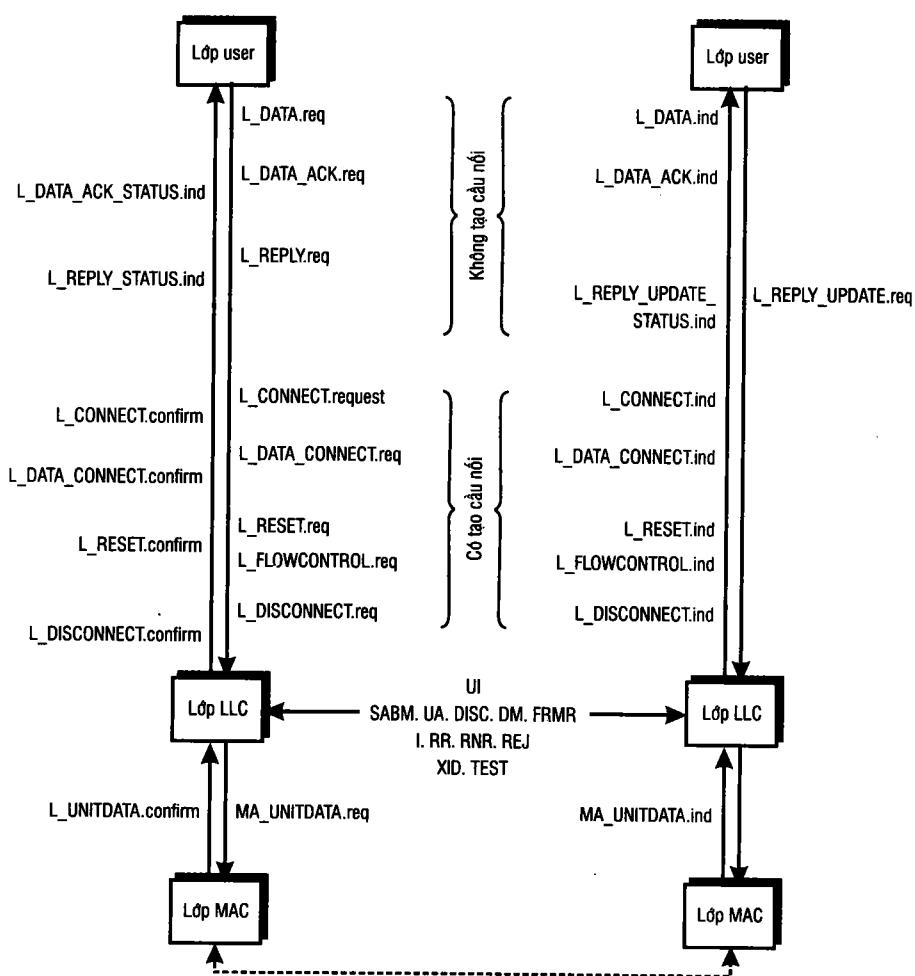


Hình 6.28 Các hàm thực thể dịch vụ user của lớp MAC:
(a) Xác nhận cục bộ (b) Xác nhận từ xa.

Mỗi hàm thực thể dịch vụ đều có các tham số liên hệ. Bao gồm trong hàm thực thể MA_UNITDATA.request là địa chỉ đích được yêu cầu (có thể là một địa chỉ cá nhân, địa chỉ một nhóm hay địa chỉ broadcast), một đơn vị số liệu dịch vụ (chứa frame LLC) và lớp dịch vụ được yêu cầu liên hệ với frame này. Tham số sau cùng được dùng trong vài loại LAN khi mà giao thức điều khiển truy xuất môi trường có ưu tiên được sử dụng.

Hàm thực thể MA_UNITDATA.confirm gồm một tham số chỉ ra sự thành công hay thất bại của hàm thực thể MA_UNITDATA.request liên hệ. Tuy nhiên, từ hình 6.28 chúng ta có thể thấy rằng hàm xác nhận không được tạo ra bởi đáp ứng từ lớp LLC ở xa mà lại là do thực thể MAC cục bộ. Nếu tham số này là thành công, thì chỉ đơn giản là thực thể giao thức MAC đã thành công trong việc truyền đơn vị số liệu dịch vụ lên môi trường mạng; nếu không thành công, tham số này chỉ ra tại làm sao nỗ lực truyền bị thất bại. Chúng ta sẽ thảo luận vấn đề này và các chủ đề khác liên quan đến MAC trong chương 7.

Hình 6.29 tóm tắt các dịch vụ khác nhau liên hệ với các lớp LLC và MAC, đồng thời liệt kê các loại frame LLC trao đổi giữa hai thực thể giao thức LLC. Như chúng ta có thể thấy, nếu tách UI, XID và TEST ra thì phần còn lại rất giống như những gì đã được dùng ở HDLC.



Hình 6.29 Tóm tắt lớp LLC và MAC.

CHƯƠNG 7

KỸ THUẬT TRUYỀN SỐ LIỆU TRONG MẠNG MÁY TÍNH CỤC BỘ

TỔNG QUAN

Các mạng số liệu cục bộ thường được gọi theo cách đơn giản là các mạng cục bộ và gọi tắt là LAN (Local Area Network) chúng được dùng để liên kết các đầu cuối thông tin phân bố trong một tòa nhà hay một cụm công sở nào đó. Ví dụ có thể dùng LAN để liên kết các máy trạm (Workstation) phân bố ở các văn phòng trong một cao ốc hay trong một khuôn viên của trường Đại học, cũng có thể liên kết các trang thiết bị mà nền tảng cấu tạo của chúng là máy tính phân bố xung quanh một nhà máy hay một bệnh viện. Vì tất cả các thiết bị tọa lạc trong một phạm vi hẹp nên các LAN thường được xây dựng và quản lý bởi một tổ chức hay cơ sở nào đó. Chính vì lý do này mà các LAN còn được xem là các mạng số liệu tư nhân.

Điểm khác biệt chủ yếu giữa một đường truyền thông tin được thiết lập bằng LAN và một cầu nối được thực hiện qua mạng số liệu công cộng là một LAN thường cho tốc độ truyền số liệu nhanh hơn do đặc trưng phân cách về mặt vật lý và cự ly ngắn. Trong ngữ cảnh của mô hình tham chiếu OSI thì khác biệt này chỉ tự biểu lộ tại các lớp phụ thuộc mạng. Trong nhiều trường hợp các lớp giao thức cấp cao hơn trong mô hình tham chiếu giống nhau trong cả LAN và mạng số liệu công cộng. Có hai loại LAN hoàn toàn khác nhau: LAN nối dây (wired LAN) và LAN không dây (wireless LAN). Như bao hàm trong tên của từng loại, LAN nối dây dùng các dây nối cố định thực, như cáp xoắn, cáp đồng trực để làm môi trường truyền dẫn trong khi đó các LAN không dây dùng sóng vô tuyến hay sóng ánh sáng làm môi trường truyền. Cách tiếp cận với hai loại là khác nhau, do đó mỗi loại phải được xem xét một cách riêng biệt.

1. CÁC MẠNG LAN NỐI DÂY

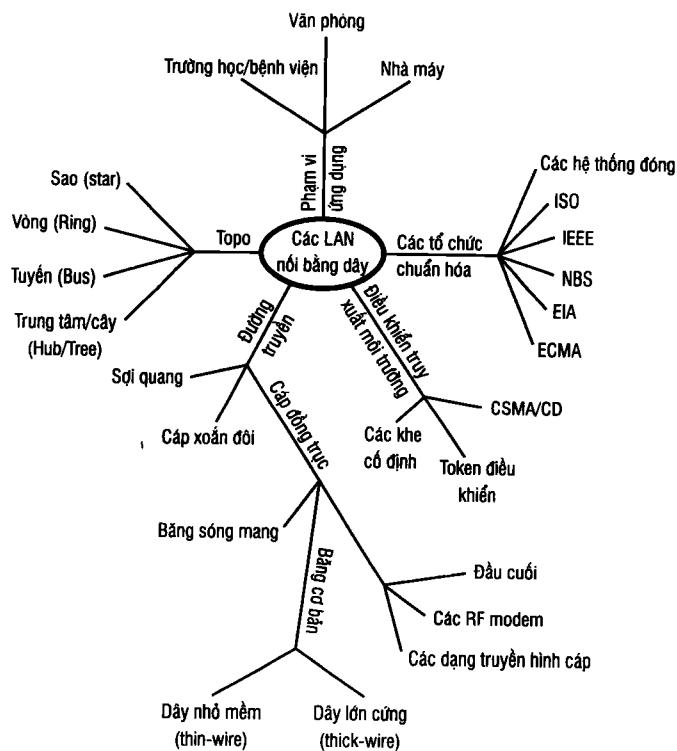
Trước khi nghiên cứu cấu trúc và hoạt động của các kiểu LAN nối dây cần nhận diện một vài yếu tố cần chọn lựa trong xây dựng LAN. Hình 7.1 tóm tắt các yếu tố này. Cần lưu ý rằng trên hình chỉ là tóm tắt, ngoài ra còn có nhiều liên kết có thể dùng giữa các nhánh trong hình.

1.1. Các Topo

Hầu hết các mạng diện rộng WAN (Wide Area Network) chẳng hạn như mạng điện thoại công cộng PSTN (Public Switching Telephone Network), dùng topo dạng lưới. Tuy nhiên, do đặc thù phạm vi vật lý giới hạn của các thuê bao (DTE) trên LAN

nên cho phép dùng các topo đơn giản hơn. Có 4 topo thông dụng là **star**, **bus**, **ring** và **hub** như trình bày trên hình 7.2.

Có lẽ ví dụ minh họa tốt nhất cho topo dạng star là tổng đài PABX (Private Automatic Branch eXchange) kỹ thuật số. Một cầu nối được thiết lập xuyên qua một tổng đài PABX analog truyền thống bằng nhiều phương pháp giống với một cầu nối được thực hiện qua mạng PSTN analog, trong đó tất cả các con đường xuyên qua mạng đều được thiết kế chỉ để mang tín hiệu thoại có băng thông giới hạn. Do đó, muốn dùng các cầu nối này để truyền số liệu, phải dùng các modem. Tuy nhiên, hầu hết các PABX hiện đại dùng các kỹ thuật chuyển mạch số và do đó cũng được gọi là các tổng đài số cá nhân PDX (Private Digital eXchange). Hơn nữa, với sự xuất hiện các IC giá rẻ thực hiện các chức năng chuyển đổi từ analog sang digital và ngược lại, làm cho việc mở rộng chế độ làm việc digital đến tận kết cuối thuê bao nhanh chóng trở thành thực tế và phổ biến. Điều này có nghĩa là những đường chuyển mạch 64 Kbps, thường được dùng cho điện thoại số sẽ luôn có sẵn tại mỗi kết cuối thuê bao, do đó có thể được dùng cho cả thoại và số liệu.



Hình 7.1

Các tùy chọn trên LAN.

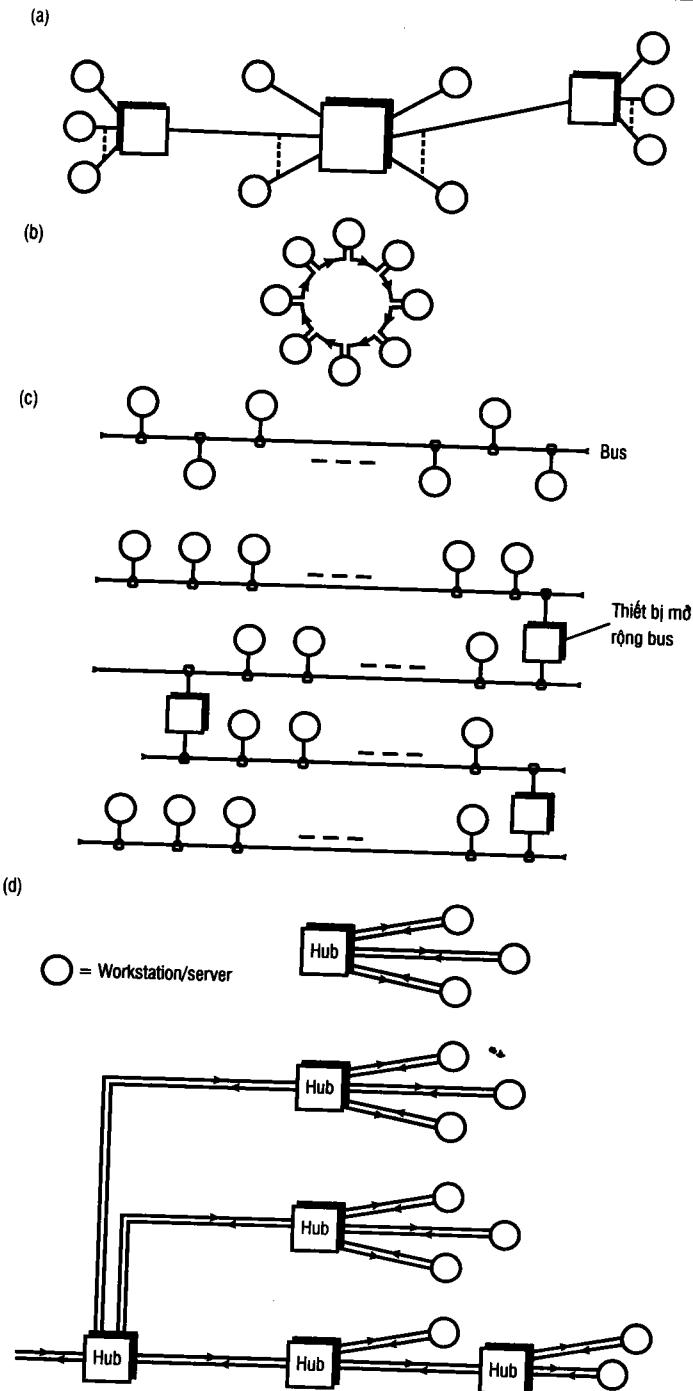
Tuy nhiên, ứng dụng chủ yếu của PDX gần như chuyên cung cấp một đường dẫn chuyển mạch cho phiên thông tin cục bộ giữa các đầu cuối tích hợp thoại và số liệu, phục vụ trao đổi thư điện tử, truyền tập tin... bên cạnh dịch vụ thông thoại truyền thống. Hơn thế nữa, kỹ thuật số trong PDX cho phép cung cấp các dịch vụ như voice store-and-forward và teleconferencing (nhiều thuê bao tham gia vào cuộc gọi đơn).

Các topo thích hợp hơn đối với các LAN đã được thiết kế để thực hiện chức năng của các mạng truyền số liệu nhỏ nhằm liên kết các thiết bị máy tính cục bộ, đó là topo dạng bus và topo dạng ring. Trong thực tế, các mạng bus thường được mở rộng thành một tập liên bus. Thông thường, trong topo dạng bus cáp mạng được dẫn qua các vị trí có DTE cần nối vào trong mạng, và một kết nối vật lý được thực hiện tại đó để cho phép các DTE truy xuất các dịch vụ mạng. Kế đến một mạch điều khiển truy xuất MAC và các giải thuật được dùng để chia sẻ băng thông truyền dẫn có sẵn cho nhóm DTE được nối vào mạng.

Với topo ring, cáp mạng đi từ một DTE đến một DTE khác cho đến khi các DTE được nối với nhau thành một vòng hay ring. Đặc trưng của ring là một liên kết điểm nối điểm trực tiếp với mỗi DTE lâng giêng hoạt động theo một chiều. Cần các giải thuật thích hợp làm nhiệm vụ chia sẻ việc sử dụng ring giữa các user trong nhóm.

Tốc độ truyền số liệu được dùng trong bus và ring vào khoảng từ 1 đến 100 Mbps, điều này nói lên rằng chúng khá phù hợp để liên kết nhóm các thiết bị cục bộ dựa trên nền máy tính, chẳng hạn như các workstation trong các văn phòng hay các bộ điều khiển thông minh xung quanh một hệ xử lý nào đó.

Một dạng topo khác được gọi là hub, hình 7.2(d). Mặc dù các mạng này giống như dạng star nhưng hub chỉ đơn giản là kết nối dạng bus hay ring được tập trung lại tại một đơn vị trung tâm. Các dây dẫn được dùng để kết nối mỗi DTE vào bus hay ring được mở rộng ra từ hub. Do đó không giống như PDX, hub không thực hiện bất kỳ hoạt động chuyển mạch nào, nó chỉ làm chức năng của một tập các bộ lắp truyền lại tất cả các tín hiệu nhận được từ các DTE đến tất cả các DTE khác theo phương pháp như trong các mạng bus và ring. Hub cũng có thể được kết nối theo dạng phân cấp hình cây. Dạng topo hỗn hợp dựa vào các chức năng của mạng ring và bus hay một tập liên kết các mạng như vậy sẽ được xem xét ở mục tiếp theo.



Hình 7.2 Các loại topo mạng: (a) Star (b) Bus (c) Ring (d) Hub/tree.



1.2. Môi trường truyền dẫn

Cáp xoắn, cáp đồng trục và cáp quang là các loại môi trường truyền chủ yếu của mạng LAN.

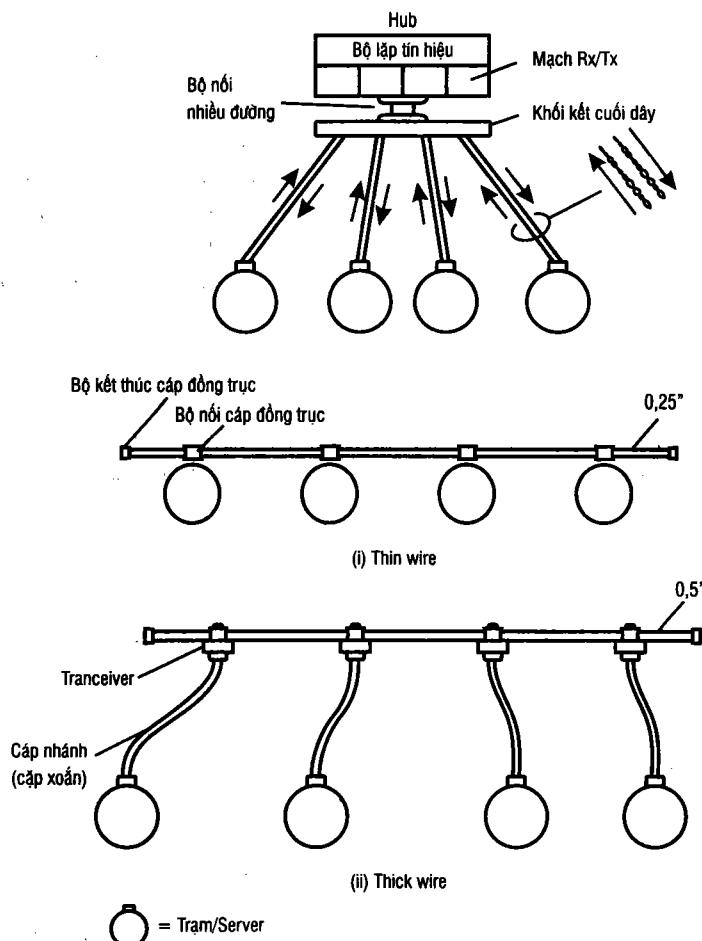
Cáp xoắn UTP và STP được dùng chủ yếu trong các mạng star và hub. Vì nó mềm hơn các loại khác, nên rất dễ lắp đặt. Ngoài ra, vì các ống dẫn dây phù hợp với cáp xoắn đang sẵn có ở hầu hết các văn phòng cho điện thoại, nên sẽ tiết kiệm khi cho thêm cáp xoắn vào cho các mục đích truyền số liệu thay vì phải lắp đặt các ống dẫn mới cho cáp đồng trục hay cáp quang. Sơ đồ chung được trình bày trên hình 7.3(a).

Có một giới hạn trên chiều dài của cáp xoắn đối với tốc độ bit đang dùng. Thông thường giới hạn là 100 mét dài cho tốc độ 1Mbps, nếu có trang bị thêm các mạch hỗ trợ triết nhiễu xuyên âm thì có thể đạt đến 10Mbps trên 100m, hiện nay có thể đạt đến 100Mbps. Một sắp xếp thông thường là dùng cáp xoắn nối giữa mỗi DTE và buồng nối dây gần nhất trên một tầng nhà, sau đó dùng cáp đồng trục làm liên kết nối buồng nối dây của tầng với hub chính của tòa nhà. Trong thiết lập mạng nối liền nhiều tòa nhà, cáp quang có thể được dùng để nối mỗi hub trong một tòa nhà đến hub trung tâm. Trường hợp này tốc độ truyền dẫn cao hơn và được cấu hình luận lý theo dạng ring. Tổ chức mạng này thường được gọi là **structured wiring**.

Cáp đồng trục cũng được dùng rộng rãi trong LAN và chủ yếu ở dạng bus, hoạt động truyền theo băng cơ bản hay băng rộng. Hai loại cáp được dùng với băng cơ bản là *thin wire* và *thick wire*. Các thuật ngữ này có liên hệ đến đường kính của cáp, thin wire có đường kính vào khoảng 0,25 inch và thick wire có đường kính 0,5inch. Thông thường, cả hai có cùng tốc độ hoạt động, nhưng thin wire có suy giảm lớn hơn thick wire; chiều dài tối đa của thin wire giữa các repeater vào khoảng 200m so với 500m của thick wire. Repeater được dùng để tái sinh tín hiệu nhận được thành tín hiệu gốc (do bị tổn thương trên môi trường truyền). Cả hai chế độ hoạt động của cáp thin wire và thick wire lần lượt được tham chiếu đến như là 10BASE2 _ngu ý 200m _và 10BASE5, ngu ý 500m.

Cáp đồng trục dạng thin wire thường được dùng để liên kết các workstation trong cùng một văn phòng hay phòng thí nghiệm. Bộ nối vật lý cáp được gắn trực tiếp vào card giao tiếp NIC (network interface card) trong workstation. Tuyến cáp hình thành một chuỗi mắc xích khi chạy qua DTE này đến DTE kế.

Ngược lại, thick wire vì cứng hơn nên thường dùng để cài đặt ở những vị trí cách xa workstation, ví dụ như chạy dọc hành lang. Thêm vào dây dẫn còn có một bộ thu phát điện được gọi là *transceiver* tọa lạc giữa điểm rẽ cáp (đồng trục) chính _được gọi là AUI (attachment unit interface)_và điểm kết nối vào workstation. Kết cấu này khá đắt tiền, do đó được dùng chủ yếu khi các workstation được đặt tại các văn phòng khác nhau hay khi liên kết các đoạn thin wire. Cả hai được miêu tả trên hình 7.3(b).



Hình 7.3 Môi trường truyền: (a) Cáp xoắn (b) Cáp đồng trục.

Với truyền dẫn băng rộng, thay vì truyền thông tin trên cáp dưới dạng hai mức điện áp tương ứng với luồng bit đang truyền, toàn bộ băng thông có sẵn của cáp được chia thành một số các băng tần con hay kênh, mỗi băng tần con được trang bị một cặp modem đặc biệt được dùng để cung cấp một kênh truyền số liệu riêng biệt. Dạng này được gọi là ghép kênh theo tần số, và vì tần số được dùng là tần số radio nên các modem dùng ở đây được gọi là các RF modem (radio frequency modem). Nguyên lý này, cũng thường gọi là *công tác băng rộng*, cũng được dùng rộng rãi trong công nghệ truyền hình CATV (communication antenna television) để ghép một số kênh truyền hình ra một cáp đồng trục đơn.

Một hệ thống CATV được trình bày trên hình 7.4. Mỗi kênh TV được phân bổ một băng tần đặc biệt, thông thường có dải thông là 6MHz. Mỗi tín hiệu hình nhận được (từ các antenna khác nhau) được dùng để điều chế lên sóng mang nằm trong băng tần đã chọn. Tín hiệu sóng mang được điều chế được truyền qua mạng cáp và hiện diện tại các kết cuối thuê bao. Thuê bao chọn một kênh truyền hình đặc biệt nào đó bằng cách điều hướng đến băng tần thích hợp.



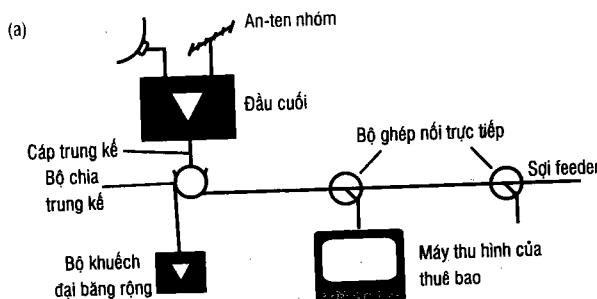
Theo cách thức tương tự, chúng ta có thể suy ra các kênh truyền số liệu tương ứng từ một cáp đơn bằng cách phân phôi cho mỗi kênh một phần băng thông tổng, băng thông cho mỗi kênh được xác định bởi tốc độ số liệu cần. Tuy nhiên, trong truyền số liệu thường yêu cầu khả năng song công, có thể thực hiện theo hai cách:

- 1) Hệ thống cáp đơn: Các đường dẫn truyền và nhận được phân phôi hai băng tần khác nhau trên cùng một cáp.
- 2) Hệ thống cáp đôi: Hai cáp tách biệt được dùng, một đường phục vụ truyền và một đường phục vụ nhận.

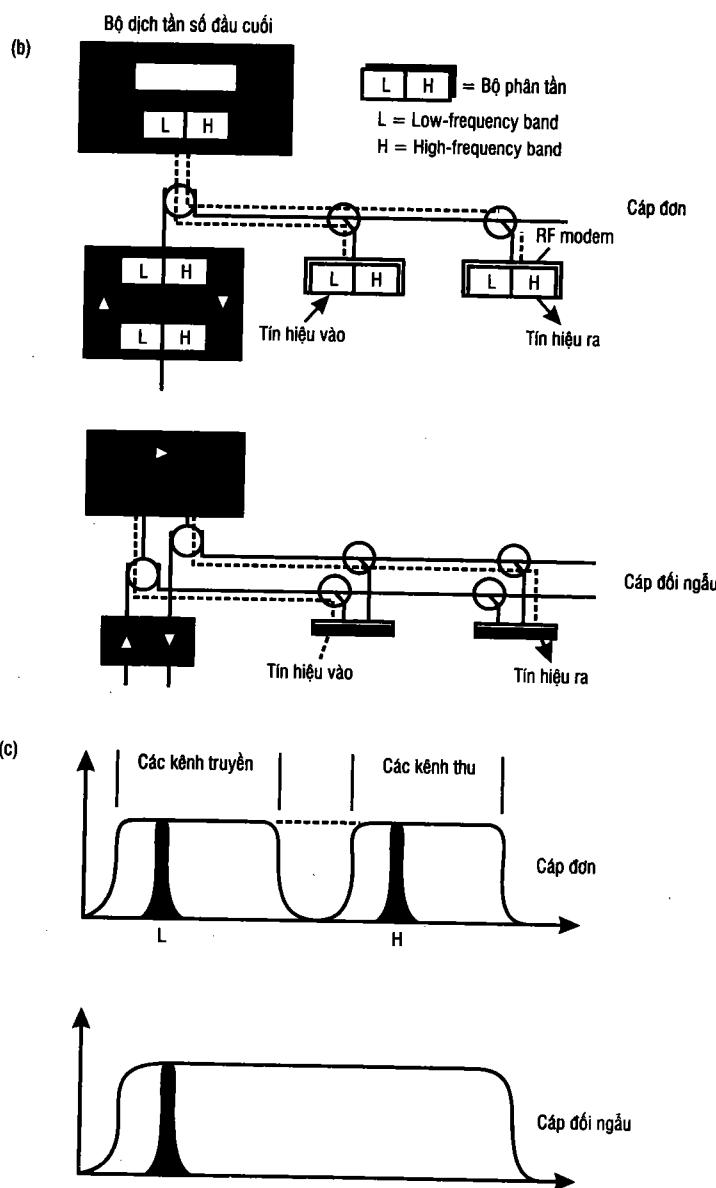
Lực đồ cho mỗi hệ thống được trình bày trên hình 7.4(b). Khác biệt chính giữa chúng là đối với hệ thống cáp đôi cần lượng cáp và đầu nối gấp đôi so với hệ thống cáp đơn. Đối với loại này tổng băng thông là có sẵn trên mỗi hướng (thông thường là từ 5 đến 450MHz). Hơn nữa, thiết bị HE (headend) chỉ đơn giản là bộ khuếch đại, trong khi đối với hệ thống cáp đơn thiết bị này được gọi là *bộ dịch tần số*, được yêu cầu để dịch tín hiệu tần số đến liên quan với các đường dẫn thu khác nhau thành ra các tần số ra tương ứng được dùng cho các đường truyền.

Một tín hiệu dạng sin trong băng tần được chọn theo hướng ngược lại (đó là đến HE) trước hết được điều chế bởi số liệu truyền dùng RF modem. Tín hiệu này được nạp vào trong cáp dùng bộ ghép đặc biệt, được thiết kế sao cho hầu hết các tín hiệu được truyền theo hướng ngược lại đi vào HE. Sau đó dùng một bộ dịch để chuyển các tín hiệu nhận được trên các băng tần thu khác nhau thành một tập tần số tương ứng. Các tín hiệu điều chế thu được do đó được dịch ra tần số bởi HE, và RF modem bên máy thu được điều hướng vào băng tần tín hiệu được dịch này. Số liệu truyền được giải điều chế từ tín hiệu thu bởi modem thu và chuyển cho DTE của nó.

Chúng ta có thể suy ra rằng một cặp tần số chỉ cung cấp một đường truyền số liệu đơn công (simplex) giữa hai DTE. Như vậy, phải cần hai cặp tần số để cung cấp một đường truyền số liệu song công (duplex). Tuy nhiên, một kênh số liệu đơn công 9,6Kbps chỉ yêu cầu mức 20KHz trong toàn bộ băng thông, do đó một cặp băng tần 6MHz có thể dùng để cung cấp đến 300 kênh đơn công hay 150 kênh song công. Các kênh tốc độ cao hơn cần nhiều băng thông hơn, ví dụ hai băng 6MHz cho một kênh song công 5Mbps hay 3 băng 6MHz cho một kênh song công 10Mbps.



Hình 7.4 Các hệ thống cáp đồng trục băng rộng:
(a) Các thành phần trong hệ thống CATV cơ bản



Hình 7.4 (tiếp theo) Các hệ thống cáp đồng trục băng rộng:
(b) Mạng số liệu (c) Sử dụng tần số.

Giá phải trả để ghép được nhiều kênh số liệu khác nhau từ một cáp đơn có liên quan đến giá khá cao của mỗi cáp RF modem. Tuy nhiên, một cáp đồng trục băng rộng có thể được dùng qua những cự ly xa hơn cáp băng cơ bản. Do đó, việc dùng



cáp đồng trục băng rộng có khuynh hướng phổ dụng và là một môi trường truyền linh hoạt được sử dụng trong công nghiệp chế tạo hay trong xây dựng, đặc biệt khi các tòa nhà trải rộng trên phạm vi hàng chục km. Khi được dùng theo cách này, các dịch vụ khác như truyền hình mạch kín, thoại có thể dễ dàng tích hợp vào trong cáp đang dùng cho truyền số liệu. Từ đó băng rộng là một chọn lựa chắc chắn hơn so với băng cơ bản cho các mạng cung cấp nhiều dịch vụ cùng một lúc.

Cáp quang với đặc điểm vốn có, có thể hoạt động với tốc độ khá cao vượt xa tốc độ của cáp xoắn và cáp đồng trục. Vì số liệu được truyền thông qua một chùm ánh sáng, nên tín hiệu không bị ảnh hưởng bởi các xuyên nhiễu điện từ. Do đó cáp quang rất thích hợp đối với các ứng dụng cần tốc độ truyền cực cao hoặc có khả năng loại bỏ ảnh hưởng của nhiễu cao, như trong các công sở có nhiều thiết bị điện công suất lớn. Ngoài ra, vì cáp quang không bức xạ ra sóng điện từ, là cơ sở chủ yếu cho các đối tượng thám mã, nên rất phù hợp cho các ứng dụng yêu cầu mức bảo mật cao.

Vì số liệu được truyền thông qua các búp sóng, nên phải dùng các thiết bị thu phát biến đổi quang thành điện và điện thành quang. Các bộ kết nối vật lý được dùng với cáp quang khá đắt tiền so với các bộ nối của cáp xoắn và cáp đồng trục, cũng rất khó thực hiện các nối rẽ vào một cáp quang. Chính vì các lý do này, nên thường dùng cáp quang trong cấu hình hub hay ring tốc độ cao cũng như các mạng dùng các đường truyền điểm nối điểm. Cụ thể là các mạng FDDI (Fiber Distributed Data Interface) và DQDB (Distributed- Queue, Dual- Bus).

1.3. Các phương pháp điều khiển truy xuất môi trường

Khi một đường thông tin được thiết lập giữa hai DTE qua một mạng star, phần tử điều khiển trung tâm đảm bảo rằng đường truyền giữa hai DTE được giữ vững trong suốt cuộc gọi. Tuy nhiên đối với topo dạng bus và ring, chỉ có một đường dẫn luân lý liên kết tất cả các DTE. Do đó, tất cả các DTE trên mạng phải tuân thủ một nguyên tắc nhất định để đảm bảo môi trường truyền được truy xuất và được dùng theo một phương pháp tối ưu. Có hai kỹ thuật được chấp nhận và được công bố trong nhiều văn bản tiêu chuẩn khác nhau đó là: *đa truy xuất cảm nhận sóng mang có phát hiện đụng độ CSMA/CD* (Carrier Sense Multiple-Access With Collision Detection) dùng cho topo dạng bus, và *thẻ điều khiển* (control token) cho mạng bus hoặc ring. Một phương pháp truy xuất khác dựa trên một *vòng phân khe* (slotted ring) cũng được dùng rộng rãi với mạng ring.

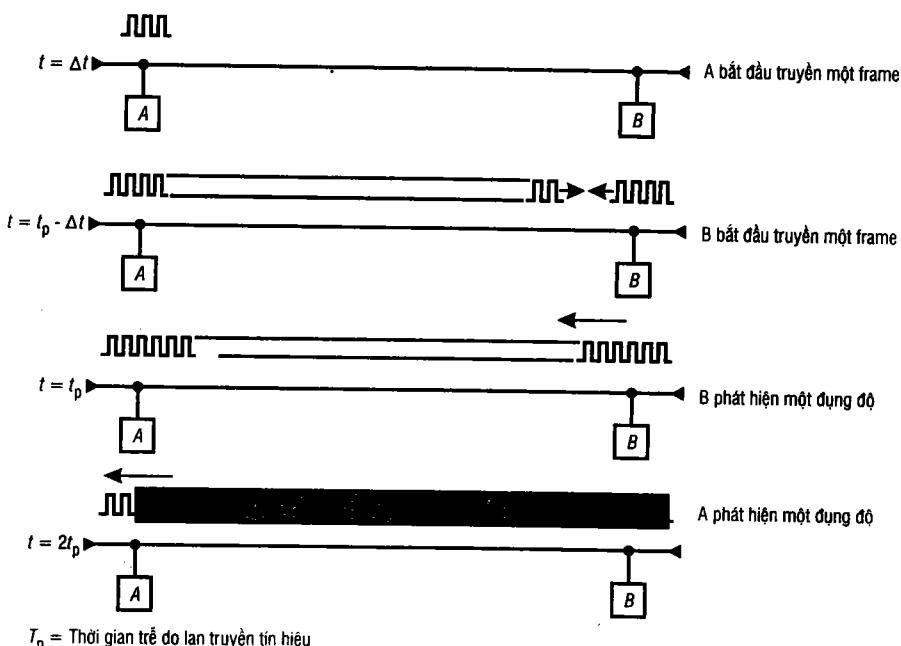
1.3.1. Đa truy xuất cảm nhận sóng mang có phát hiện đụng độ CSMA/CD

Phương pháp CSMA/CD chỉ được dùng cho mạng bus. Với cấu hình mạng này tất cả các DTE được nối một cách trực tiếp vào cùng một cáp, cáp này được dùng để truyền tất cả số liệu cho bất kỳ một cặp DTE nào trên mạng này. Tổ chức hoạt động như vậy của cáp được gọi là làm việc theo *chế độ đa truy xuất* (multiple access mode). Để truyền số liệu, DTE truyền trước hết phải đóng gói số liệu thành một



frame với địa chỉ DTE đích được đặt tại đầu frame. Sau đó frame này được truyền trên cáp, thao tác truyền này mang tính quảng bá (broadcast). Tất cả các DTE trên mạng đều phát hiện được frame bắt đầu khi nào nó được truyền trên cáp. Khi DTE đích phát hiện được frame hiện hành đang được truyền có địa chỉ ở đầu frame trùng với địa chỉ của nó, nó tiếp tục đọc số liệu chứa trong frame và đáp ứng tùy vào giao thức liên kết được định nghĩa. Địa chỉ DTE nguồn cũng là một phần trong header của frame, nhờ đó DTE đích có thể đáp ứng trực tiếp với DTE nguồn này.

Với cơ chế hoạt động này hai DTE nào đó có thể cùng nỗ lực truyền một frame lên cáp cùng lúc vào bất cứ thời điểm nào, khiến cho số liệu từ hai nguồn va chạm nhau gây hư hỏng. Để giảm thiểu khả năng này, trước khi DTE nguồn truyền một frame nó phải lắng nghe xem trên cáp có một frame nào đang được truyền hay không. Nếu có một tín hiệu sóng mang được cảm nhận CS (carrier sense), thì DTE sẽ dừng quá trình truyền của nó cho đến khi frame đang chuyển được truyền xong, và chỉ nỗ lực truyền frame vào thời điểm sau đó. Ngay cả như vậy, cũng có hai DTE muốn truyền frame cùng đồng thời xác định không có hoạt động nào trên bus, và cả hai sẽ bắt đầu truyền frame một cách đồng thời. Một đụng độ sẽ xảy ra, nội dung của cả hai frame sẽ va chạm nhau và hư hỏng. Điều này được minh họa trên hình 7.5.



Hình 7.5

Đụng độ trên CSMA/CD.

Một DTE sẽ giám sát tín hiệu số liệu trên cáp đồng thời với thao tác truyền nội dung frame lên cáp. Nếu tín hiệu được giám sát và truyền là khác nhau, xem như có



một dung độ xảy ra_CD (collision detect). Để đảm bảo các DTE liên quan đến sự dung độ đều được cảnh báo về sự cố dung độ này, DTE đầu tiên làm rõ thêm sự dung độ này bằng cách gửi đi một mảng bit ngẫu nhiên trong một thời gian ngắn. Mảng này được gọi là *dãy nhồi* (jam sequence). Hai (hay nhiều hơn) DTE liên quan, sau khi phát hiện điều này sẽ phải đợi trong một khoảng thời gian ngắn ngẫu nhiên trước khi truyền lại các frame bị ảnh hưởng của nó. Có thể kết luận rằng việc truy xuất vào bus CSMA/CD mang tính xác suất và phụ thuộc vào tải trên mạng. Vì tốc độ bit trên mạng là khá cao (đến 10Mbps) tải mạng có khuynh hướng giảm. Vì khởi sự truyền frame chỉ bắt đầu nếu cáp không ở trạng thái truyền frame nên xác suất xảy ra dung độ là nhỏ.

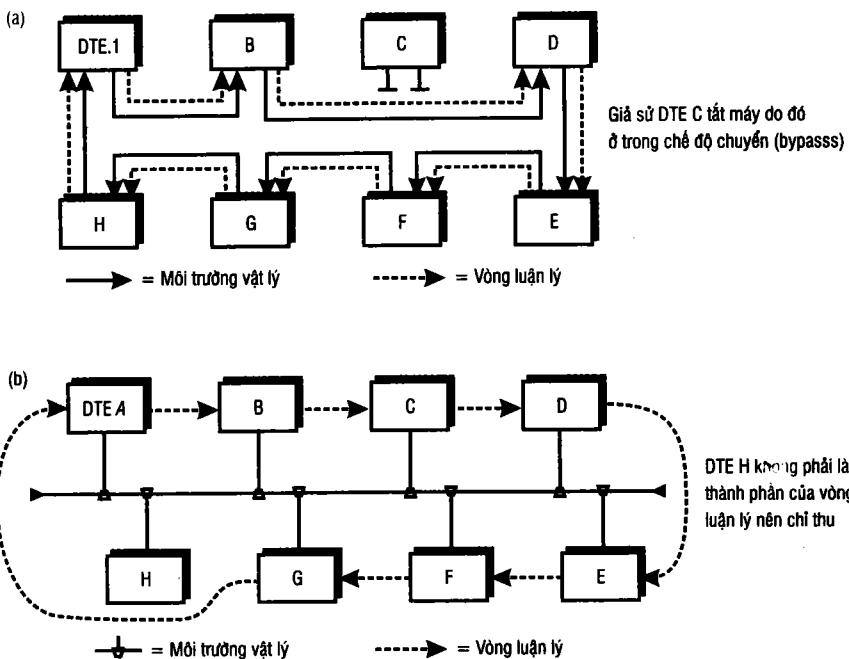
1.3.2. Token (thẻ) điều khiển

Một phương pháp khác cũng nhằm truy xuất vào môi trường chia sẻ là phương pháp dùng token điều khiển. Token này được chuyển từ một DTE này sang một DTE khác tùy vào qui tắc được định nghĩa, được tất cả các DTE nối vào môi trường hiểu và tuân theo. Một DTE có thể truyền một frame chỉ khi nó ở trong trạng thái sở hữu token, sau khi đã truyền frame, nó chuyển token đi tiếp để cho phép DTE khác truy cập vào môi trường truyền. Tuần tự hoạt động có thể tóm tắt như sau:

- Trước hết một vòng luận lý được thiết lập để liên kết tất cả các DTE nối vào môi trường vật lý này, và một token điều khiển được tạo ra.
- Token này được chuyển từ DTE này sang DTE khác xung quanh vòng luận lý cho đến khi nó được tiếp nhận bởi một DTE đang đợi truyền frame.
- Sau đó DTE đang đợi truyền, truyền các frame của nó qua môi trường vật lý này, sau khi đã kết thúc việc truyền, nó sẽ chuyển token điều khiển đến DTE kế trong vòng luận lý.

Chức năng giám sát trong các DTE kết nối vào môi trường vật lý này là cơ sở cho việc khởi tạo và khôi phục cầu nối của vòng luận lý và cả trong trường hợp đánh mất token. Mặc dù các chức năng giám sát thường được gán vào tất cả các DTE trên môi trường, nhưng tại một thời điểm chỉ có một DTE thực hiện khả năng này nhằm đáp ứng việc phục hồi và khởi tạo.

Môi trường vật lý không nhất thiết phải là topo dạng ring, một token cũng có thể được dùng để điều khiển truy xuất vào một mạng bus. Sự thiết lập vòng luận lý trên hai loại mạng được trình bày trên hình 7.6.



Hình 7.6 Điều khiển truy xuất môi trường bằng token điều khiển:
(a) Token ring (b) Token bus.

Đối với một vòng vật lý thì cấu trúc luân lý của vòng chuyển token giống với cấu trúc vòng vật lý, với thứ tự chuyển token trùng với thứ tự vật lý của các DTE nối vào trong ring. Tuy nhiên, với mạng bus thứ tự của vòng luân lý hoàn toàn không giống với thứ tự vật lý của các DTE trên cáp. Hơn nữa, với phương pháp truy xuất token trên một mạng bus, tất cả các DTE không cần nối vào vòng luân lý. Ví dụ, DTE H không phải là một phần của vòng luân lý được trình bày trên hình 7.6(b). Điều này có nghĩa là DTE H chỉ hoạt động trong chế độ nhận, vì nó chẳng bao giờ sở hữu một token điều khiển. Một đặc tính khác của phương pháp điều khiển truy xuất bằng token là có thể phối hợp token với mức ưu tiên, do đó cho phép các frame có tính ưu tiên cao hơn được truyền trước. Một số khía cạnh khác của phương pháp truy xuất token sẽ được mở rộng trong các phần tiếp theo.

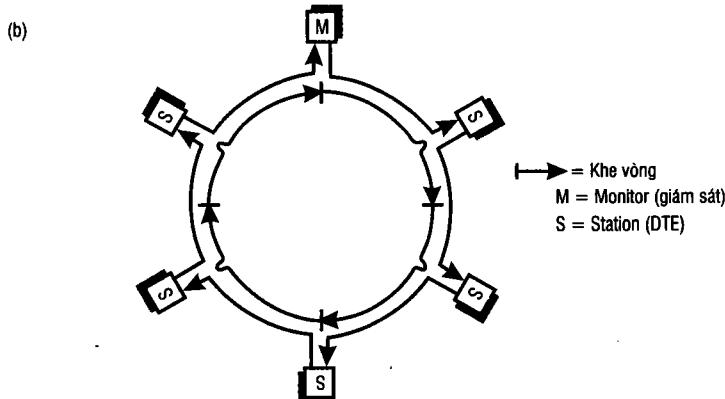
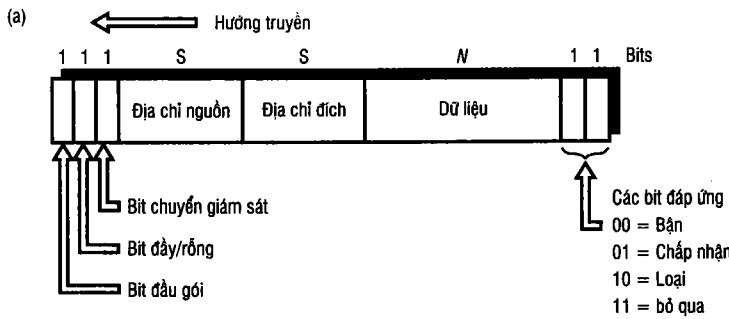
1.3.3. Vòng được phân chia khe (slotted ring)

Các vòng phân khe được dùng chủ yếu cho điều khiển truy xuất vào một mạng ring. Trước hết ring được khởi tạo bởi một node đặc biệt để chứa một số không đổi các bit, node đặc biệt này được gọi là bộ giám sát (monitor). Luồng bit chạy liên tục trên vòng từ DTE này đến DTE khác. Sau đó, khi mỗi bit được nhận bởi một DTE, giao tiếp DTE kiểm tra bit này và chuyển sang cho DTE kế trong ring và cứ thế. Monitor đảm bảo rằng luôn có một số bit không đổi chạy vòng trên ring, bất chấp số DTE tạo nên ring. Ring hoàn chỉnh được bố trí chứa một số không đổi các khe, mỗi

khe hình thành một tập các bit và có khả năng mang một frame thông tin có kích thước cố định.

Khởi đầu, tất cả các khe được đánh dấu rỗng khi bộ giám sát đặt bit đầy/rỗng tại đầu mỗi khe ở trạng thái rỗng. Khi một DTE muốn truyền một frame, nó đợi cho đến khi phát hiện được một khe rỗng. Sau đó DTE đánh dấu khe này là đầy và chèn nội dung frame vào khe cùng với địa chỉ DTE nguồn yêu cầu, địa chỉ DTE đích đặt tại đầu của frame và cả hai bit đáp ứng tại cuối của frame lúc này đều được set là 1. Khe chứa frame này sau đó chạy vòng trên ring vật lý từ DTE này đến DTE khác. Mỗi DTE trong ring kiểm tra địa chỉ đích tại đầu của các khe được đánh dấu là đầy, nếu phát hiện thấy trùng với địa chỉ của nó và giả sử nó sẵn sàng tiếp nhận, nội dung frame được đọc vào từ khe, cùng thời điểm này nó vẫn tiếp tục lặp lại các nội dung của frame khác (không sửa chữa) lên ring. Sau khi đọc nội dung frame, DTE đích sửa cắp bit ở cuối khe để chỉ ra rằng nội dung khe đã được đọc hoặc DTE đích đang bận hay không ở trạng thái hoạt động, các bit được đánh dấu tùy theo đó hay để nguyên.

Đối với DTE nguồn, sau khi khởi động truyền một frame, đợi cho đến khi frame chạy vòng trên ring bằng cách đếm số khe được lặp lại tại giao tiếp vòng. Khi tiếp nhận bit đầu tiên của khe dùng để truyền frame này, nó lại đánh dấu khe là rỗng và đợi đọc bit đáp ứng từ đuôi của khe này để xác định hành động sẽ tiến hành kế tiếp.



Hình 7.7 Các nguyên lý vòng phân khe:

(a) Các định nghĩa bit của mỗi khe (b) Cấu hình.

Bit giám sát được dùng bởi bộ giám sát để phát hiện xem một DTE có hỏng hay không để giải phóng một khe sau khi nó đã được truyền. Bit này được reset bởi DTE nguồn khi nó truyền một frame lên ring. Sau đó bộ giám sát set bit này cho mỗi khe đầy khi được lắp tại giao tiếp vòng của nó. Nếu monitor phát hiện bit giám sát này được set trong khi lắp một khe đầy, bộ giám sát giả sử rằng DTE nguồn đã không thành công trong việc đánh dấu khe này là rỗng và do đó reset bit (đầy/rỗng) này tại đầu của khe này.

Lưu ý rằng ứng với một phương pháp truy xuất môi trường ring phân khe, mỗi DTE chỉ có thể có một frame đang chuyển trên vòng tại một thời điểm. Nó cũng phải giải phóng khe được dùng để truyền frame trước khi cố truyền một frame khác. Theo cách này, thì việc truy xuất vào ring là chia sẻ và bình đẳng giữa các DTE kết nối vào vòng. Các khuyết điểm chính của một ring phân khe là :

- i. Cần một node giám sát đặc biệt để duy trì cấu trúc cơ bản của vòng.
- ii. Việc truyền mỗi frame hoàn chỉnh ở mức liên kết thông thường cần nhiều khe.

Đi nhiên, với token ring một khi DTE nhận một token điều khiển nó có thể truyền một frame hoàn chỉnh chứa nhiều byte thông tin tương đương một đơn vị truyền.

1.4. Hệ thống Ethernet và Fast Ethernet (CSMA/CD)

1.4.1. Điều khiển truy xuất IEEE 802.3

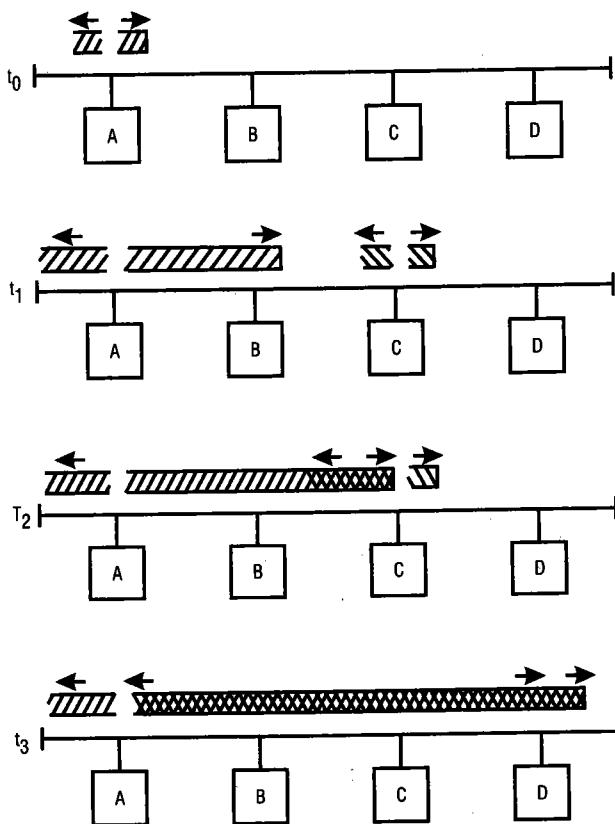
Như đã đề cập ở phần phương pháp điều khiển truy xuất môi trường CSMA/CD, CSMA/CD chỉ được dùng cho mạng bus. Vì các lý do lịch sử, một mạng bus CSMA/CD cũng còn gọi là mạng Ethernet. Về nguyên tắc của CSMA/CD có thể được tóm tắt lại như sau:

Bước 1. Nếu môi trường là nhàn rỗi thì truyền, ngược lại chuyển đến bước 2.

Bước 2. Nếu môi trường là bận, tiếp tục lắng nghe cho đến khi kênh rảnh, truyền ngay tức thì.

Bước 3. Nếu phát hiện dung độ trong quá trình truyền, truyền một tín hiệu nhồi (jam sequence) để đảm bảo tất cả các trạm khác biết được dung độ này và ngừng ngay công việc truyền của chúng.

Bước 4. Sau khi truyền tín hiệu nhồi, đợi trong một thời gian ngẫu nhiên, sau khoảng thời gian đợi này tiến hành truyền lại.

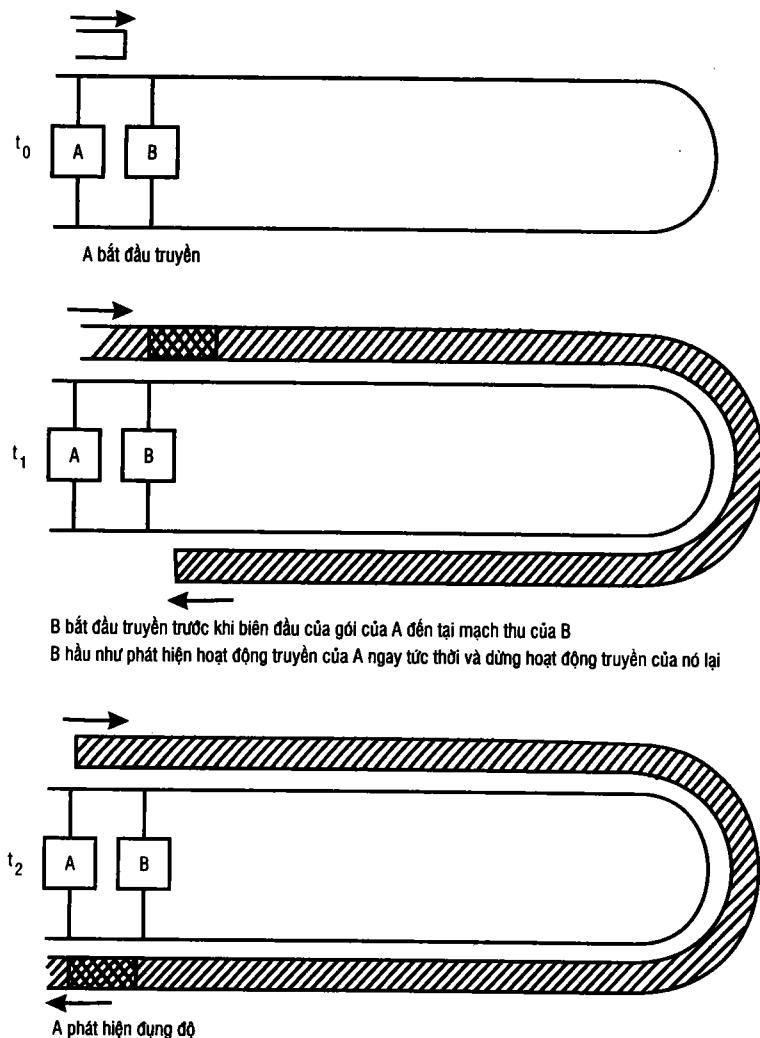


Hình 7.8

Hoạt động của CSMA/CD

Hình 7.8 mô tả kỹ thuật cho bus băng cơ bản. Tại thời điểm t_0 , trạm A bắt đầu truyền một frame đến D. Tại thời điểm t_1 , cả B và C đều sẵn sàng truyền. B cảm nhận có một hoạt động truyền trên mạng nên dừng lại. Tuy nhiên C không được cảnh báo về hoạt động truyền của A nên xúc tiến công việc truyền của mình. Khi hoạt động truyền của A đạt đến C, tại thời điểm t_2 , C phát hiện dung độ và dừng việc truyền của nó. Ảnh hưởng của dung độ lan truyền đến A, tại A nó phát hiện được dung độ vào thời điểm t_3 và dừng việc truyền lại. Đối với CSMA/CD lượng thời gian lãng phí được giảm xuống bằng với thời lượng cần để phát hiện một dung độ. Câu hỏi đặt ra ở đây là : mất bao lâu? Trước hết, xem xét trường hợp một bus băng cơ bản và hai trạm cách xa nhau nhất. Ví dụ trong hình 7.8, giả sử A bắt đầu truyền và chỉ ngay sau khi hoạt động truyền này đạt đến D, D sẵn sàng truyền. Vì D không được cảnh báo rằng có A đang truyền nên nó bắt đầu truyền. Một dung độ xảy ra ngay tức thì và được nhận biết bởi D. Tuy nhiên, dung độ này phải lan truyền theo mọi đường quay về A trước khi A được cảnh báo về dung độ này. Với lý do này, có thể kết luận

rằng thời gian cần thiết để phát hiện dung độ không thể lớn hơn hai lần thời gian trễ lan truyền (propagation delay) giữa hai trạm. Trong một bus băng cơ bản, thời gian trễ lại còn dài hơn. Hình 7.9 trình bày một hệ thống cáp đôi. Thời gian trễ này xấu nhất khi hai trạm gần và xa headend nhất. Trong trường hợp này, thời gian tối đa để phát hiện một dung độ là gấp bốn lần thời gian trễ lan truyền từ điểm cuối của cáp đến headend.



Hình 7.9 Định thời phát hiện dung độ băng rộng.



Một nguyên tắc quan trọng trong hầu hết các hệ thống CSMA/CD, và bao gồm chuẩn IEEE, là các frame phải đủ dài để dung độ được phát hiện trước khi kết thúc việc truyền. Nếu chiều dài frame ngắn hơn, không thể phát hiện ra dung độ và chất lượng của CSMA/CD kém như trường hợp giao thức CSMA (đa truy xuất cảm nhận sóng mang không phát hiện dung độ).

Mặc dù việc triển khai CSMA/CD về cơ bản giống nhau trong băng cơ bản và băng rộng, vẫn có vài điểm khác nhau. Một trong những khác nhau này là các phương tiện thực hiện cảm nhận sóng mang; trong các hệ thống băng cơ bản, điều này được thực hiện bằng cách phát hiện một chuỗi xung điện áp. Trong các hệ thống băng rộng, là phát hiện sóng mang RF.

Phát hiện dung độ cũng khác nhau trong cả hai hệ thống. Trong hệ thống băng cơ bản, một dung độ về căn bản tạo ra một chuỗi điện áp cao hơn điện áp được tạo ra bởi bộ truyền. Tùy vào tiêu chuẩn của IEEE mà bộ truyền sẽ phát hiện được một dung độ khi tín hiệu trên cáp tại điểm nối mạch phát vượt quá giá trị tối đa mà mạch phát có thể tạo ra. Vì tín hiệu bị suy giảm khi truyền, nên có khả năng xảy ra vấn đề: nếu hai trạm cách xa nhau đang truyền, mỗi trạm sẽ nhận một tín hiệu suy giảm từ trạm kia. Mức tín hiệu có thể nhỏ đến nỗi khi được nhập vào tín hiệu truyền tại điểm nối của mạch phát, *tín hiệu tổng hợp này không vượt quá mức ngưỡng của CD*. Vì lý do này, IEEE giới hạn chiều dài tối đa của cáp đến 200m hay 500m ứng với 10BASE2 và 10BASE5. Một lược đồ phát hiện dung độ đơn giản hơn có thể dùng với cấu hình dạng star dùng cáp xoắn đôi. Trong trường hợp này, việc phát hiện dung độ mang tính luận lý thay vì dựa vào sự cảm nhận độ lớn của điện áp. Trong bất cứ một thiết bị hub nào, nếu có tín hiệu tích cực trên nhiều hơn một cổng, coi như có dung độ xảy ra. Một tín hiệu đặc biệt được phát lên trên tất cả các ngõ ra. Tín hiệu này được các node dịch ra như là có một dung độ đang xảy ra.

1.4.2. Hệ thống CSMA/CD

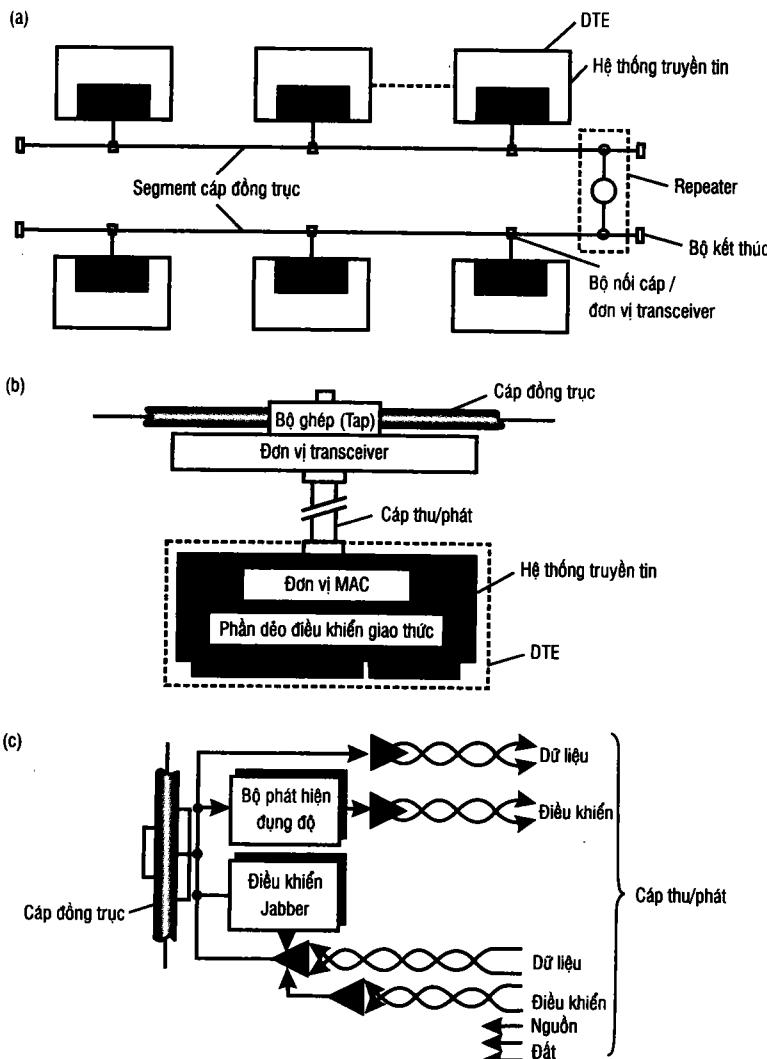
Mặc dù mạng CSMA/CD có thể dùng các môi trường khác nhau nhưng tất cả đều hoạt động theo cùng phương pháp MAC (media access control). Đối với cáp đồng trục dạng thin-wire và thick-wire, khác nhau chủ yếu là vị trí của mạch thu phát. Trong trường hợp thick-wire mạch này được đặt tại bộ nối rẽ cáp, do đó còn được gọi là *transceiver*. Đối với thin-wire cáp nối trực tiếp vào card giao tiếp trong DTE, do đó transceiver được tích hợp bên trong card giao tiếp.

Các thành phần khác nhau liên hệ với mạng thick-wire được trình bày trên hình 7.10. Một đầu nối rẽ (tap) được dùng để tạo một kết nối vật lý mà không phải cắt cáp. Đầu nối này gồm một cơ cấu lắp ghép, xuyên qua vỏ bảo vệ của cáp và tạo một kết nối với dây dẫn. Phần vỏ của cơ cấu này cũng nối với lưỡi ngoài của cáp, nhờ đó tạo được một kết nối về điện hoàn chỉnh.

Transceiver chứa các mạch điện cần thiết để thực hiện các chức năng sau đây:

- Truyền và nhận số liệu đi và đến từ cáp.

- Phát hiện dung độ trên môi trường cáp.
- Tạo nên một ngăn cách về điện giữa cáp đồng trục và các mạch điện giao tiếp cáp.
- Bảo vệ cáp tránh những tác động xấu bởi hỏng hóc trên transceiver hay DTE được nối vào.



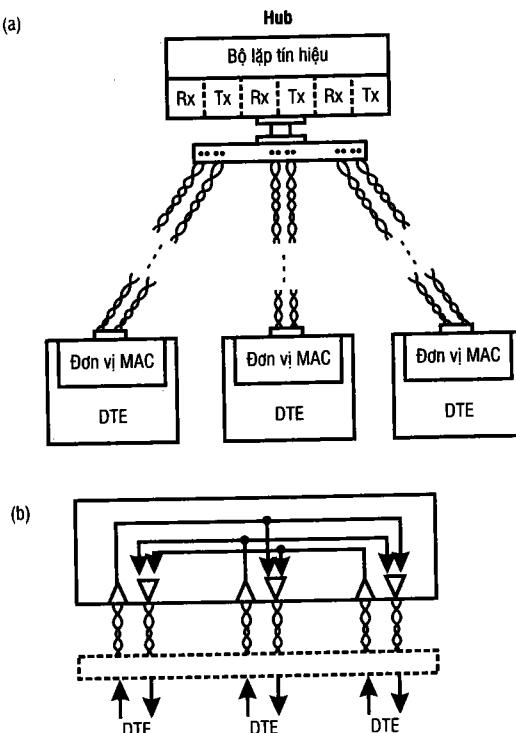
Hình 7.10 Các thành phần mạng thick-wire CSMA/CD bus:

(a) Sơ đồ cáp (b) Giao tiếp DTE (c) Lược đồ transceiver.

Chức năng sau cùng thường được gọi là *jabber control* vì không có mạch điện bảo vệ thích hợp, nếu một sai sót phát triển thành lỗi transceiver (hay DTE) có thể truyền số liệu ngẫu nhiên một cách liên tục lên môi trường cáp và do đó làm cản trở hay phá hủy tất cả các cuộc truyền khác. Jabber control có lập đường truyền số liệu đối với cáp nếu các giới hạn thời gian qui định nào đó bị xâm phạm. Ví dụ tất cả các frame truyền trên cáp có một chiều dài tối đa xác định. Nếu vượt quá, jabber control sẽ ngăn cản truyền thêm số liệu lên mạng.

Các đơn vị transceiver được kết nối vào DTE của nó nhờ một cáp bảo vệ chia bộ 5 dây: một cáp năng lượng cho transceiver từ DTE; hai cho số liệu (truyền và nhận), và hai cho mục đích điều khiển (một để transceiver phát tín hiệu dung độ đến DTE và một cho DTE khởi sự công lập đường truyền số liệu đối với cáp). Bốn cặp tín hiệu được điều khiển khác nhau, điều này có nghĩa là DTE có thể cách transceiver đến 50m và do đó cách điểm nối rẽ cáp đến 50m.

Với cáp xoắn đôi và cấu hình hub cũng như trường hợp cáp đồng trục dạng thin-wire, chức năng phát hiện dung độ được định vị trên card giao tiếp trong DTE. Chức năng của hub là nhận và truyền lại các tín hiệu một cách tin cậy. Một cấu hình hub và chức năng lắp lại tín hiệu được trình bày trên hình 7.11.



Hình 7.11 Các nguyên lý cấu hình hub: (a) Topo (b) Lược đồ lắp tín hiệu.

Như trên hình, có hai cặp cáp xoắn nối mỗi DTE đến hub, trong đó một cặp truyền và một cặp nhận. Để cho phép mạch phát hiện dung độ thực hiện chức năng tại các DTE, mạch lặp tín hiệu trên hub truyền lại tín hiệu nhận được từ một cặp ngõ vào lên tất cả các cặp ngõ ra khác. Nhiệm vụ chính của mạch repeater là đảm bảo rằng tín hiệu truyền lại khá mạnh ở trên các cặp ngõ ra không gây nhiễu đối với tín hiệu nhận trên cặp ngõ vào, tín hiệu nhận này vốn được coi là yếu do bị suy giảm khi truyền. Ảnh hưởng này được gọi là *nhiễu đầu cuối kẽ* hay gọi tắt là NEXT. Có một mạch tích hợp đặc biệt được gọi là *bộ khử xuyên âm thích nghi* được dùng để đảm bảo hoạt động tin cậy ở tốc độ 10Mbps trên chiều dài cáp 100m.

Bất chấp môi trường truyền dẫn được dùng, card điều khiển thông tin bên trong mỗi DTE gồm có các thành phần sau :

- i. Một đơn vị điều khiển truy xuất môi trường (MAC unit), nó có thể đáp ứng các chức năng như đóng và tách frame để truyền và nhận trên cáp, phát hiện lỗi là thực thi giải thuật MAC.
- ii. Một bộ nhớ RAM có port đối ngẫu cho phép đơn vị MAC truyền và nhận frame với tốc độ cao và để cho máy tính đọc và ghi nội dung thông tin của frame.

Hệ thống thông tin con hoàn chỉnh thông thường nằm độc lập trên một bản mạch in, bản mạch này được cắm vào khe bus của hệ thống máy chủ và được liên kết với các tuyến mức thấp nhằm cung cấp một tập xác định các dịch vụ truyền và nhận frame cho phần mềm host. Hầu hết các card thương mại cung cấp đa kết nối hỗ trợ cho các loại môi trường truyền dẫn khác nhau.

1.4.3. Cấu trúc frame và các tham số hoạt động

Hình 7.12 mô tả khuôn dạng của một frame theo giao thức IEEE 802.3; nó bao gồm các field (field là một phân vùng chứa một số bit trong frame) sau:

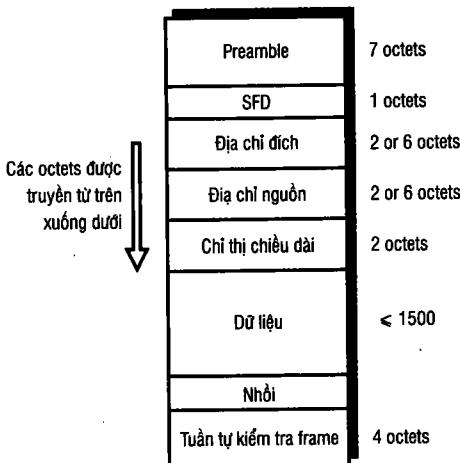
- Preamble : Một mảng gồm 7 octet, mỗi octet có dạng 10101010 được dùng bởi máy thu để thiết lập đồng bộ bit.
- SFD (start frame delimiter): là tuần tự 10101011, chỉ ra điểm bắt đầu thực sự của frame và cho phép máy thu định vị được bit đầu tiên của phần còn lại của frame.
- DA (destination address): chỉ định trạm (hay nhiều trạm) mà frame hướng đến. Nó có thể là một địa chỉ vật lý duy nhất, một địa chỉ nhóm hay một

địa chỉ quảng bá. Việc chọn chiều dài địa chỉ này là 16 hay 48 bit là một quyết định trong quá trình hiện thực, và phải thống nhất trên tất cả các trạm trên LAN.

- SA(source address) : chỉ định trạm đã truyền frame.
- Length : chỉ ra chiều dài field chứa số liệu _ LLC data.
- Data: vùng chứa số liệu được cung cấp bởi LLC.
- Pad: Các byte thêm vào để đảm bảo đủ chiều dài tối thiểu cho hoạt động phát hiện dung độ (CD).
- FCS (frame check sequence): Một vùng 32 bit chứa mã kiểm tra theo CRC , tính trên tất cả các field ngoại trừ Preamble, SFD và FCS.

Tất cả các frame được truyền trên cáp dùng mã đường dây Manchester.

(a)



(b)

Tốc độ	10 Mbps (Manchester encoded)
Khe thời gian	512 bit times
Bấy frame	9.6 μ s
Giới hạn truyền lại	16
Giới hạn quay lui	10
Kích thước nhồi	32 bit
Kích thước frame tối đa	1518 octets
Kích thước frame tối thiểu	512 bits

Hình 7.12 Các đặc tính của mạng bus CSMA/CD:

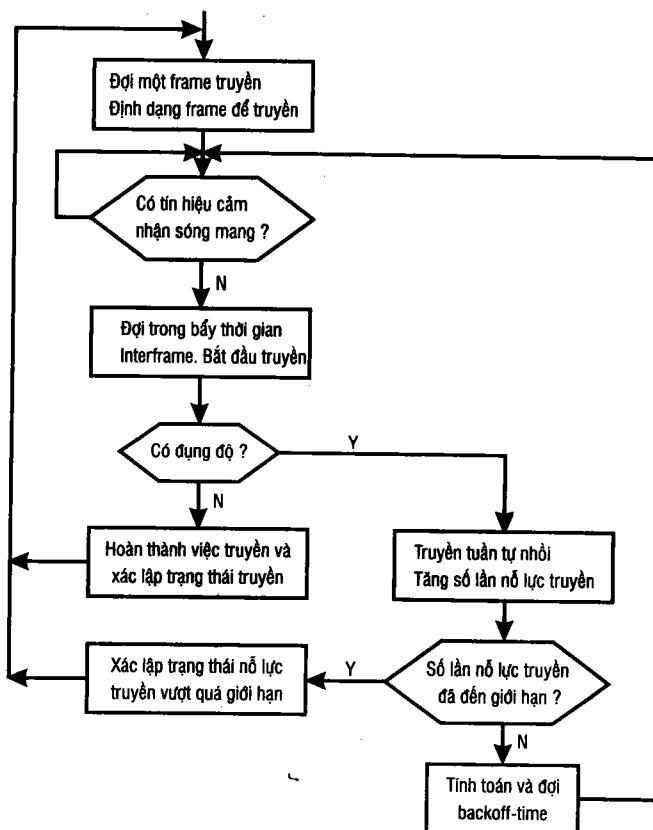
(a) Cấu trúc frame (b) Các tham số hoạt động.

1.4.4. Hoạt động truyền frame

Khi một frame được truyền, nội dung của frame được đóng gói bởi đơn vị MAC theo khuôn dạng ở trên. Để tránh đụng độ với các cuộc truyền khác trên môi trường, phần MAC của đơn vị MAC trước hết giám sát tín hiệu cảm nhận sóng mang, và nếu cần dừng ngay mọi hoạt động truyền. Sau một khoảng thời gian trì hoãn ngắn (được gọi là interframe gap) chờ cho frame đang chuyển đã đến DTE đích và được xử lý, lúc đó mới khởi động truyền frame.

Khi luồng bit được truyền, transceiver giám sát đồng thời tín hiệu nhận để xem có dung độ xảy ra hay không. Giả sử không có dung độ, frame hoàn chỉnh được truyền sau khi FCS đã được gửi, đơn vị MAC đợi frame mới đến từ cáp hay từ bộ xử lý điều khiển. Nếu có dung độ xảy ra, transceiver lập tức phát ra tín hiệu dung độ. Đến lượt đơn vị MAC nhận biết và tăng cường mức dung độ này bằng cách gửi tuần tự nhồi để cho tất cả các DTE liên quan đến dung độ này cũng nhận biết. Sau khi tuần tự nhồi được gửi đi, đơn vị MAC kết thúc việc truyền và lập lịch truyền lại sau một khoảng thời gian ngắn được chọn một cách ngẫu nhiên.

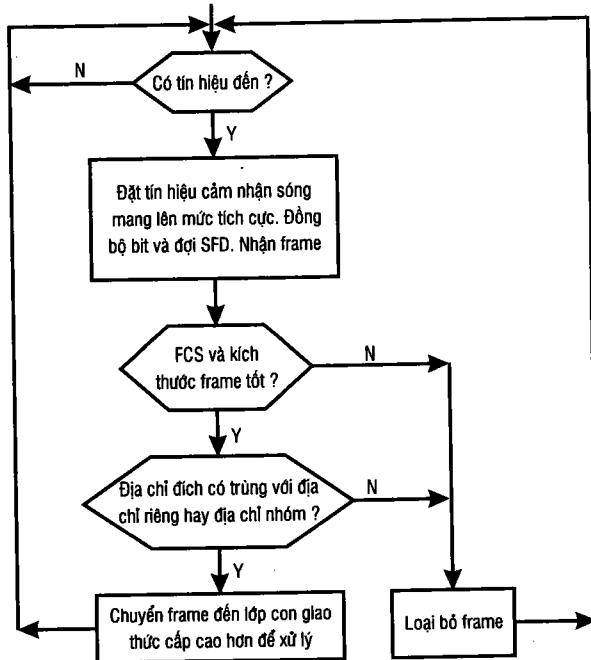
(a)



Hình 7.13 Lưu đồ hoạt động: (a) Truyền



(b)

**Hình 7.13 (tiếp theo) Lưu đồ hoạt động: (b) Nhận.**

Khi xảy ra đụng độ, nỗ lực truyền lại frame được xúc tiến đến một số lần nhất định được gọi là **attempt limit**. Hiện tượng lặp lại đụng độ chứng tỏ rằng môi trường truyền đang quá bận, đơn vị MAC thử điều chỉnh để nạp tải vào môi trường bằng cách gia tăng thời gian đợi giữa hai lần truyền lại. Lập lịch truyền lại được điều khiển bởi một quá trình được tóm tắt như sau: Khi việc truyền tuần tự nhồi (jam sequence) kết thúc, và giả sử attempt limit chưa đạt đến, đơn vị MAC trì hoãn một số nguyên lần khe thời gian ngẫu nhiên trước khi nỗ lực truyền lại frame. Như đã đề cập trong mục các phương pháp điều khiển truy xuất ở trên, một DTE có thể trải qua một dụng độ trong thời gian khởi động truyền, còn gọi là **cửa sổ đụng độ**, nó gấp hai lần khoảng thời gian để bit đầu tiên của frame lan truyền đến tất cả các phần của cáp. Vì thế khe thời gian là thời gian trì hoãn xấu nhất mà một DTE phải đợi trước khi nó nhận biết một đụng độ xảy ra. Khe thời gian được xác định:

$$\text{Khe thời gian} = 2 \times (\text{tri hoãn lan truyền trên đường dây}) + \text{lượng an toàn.}$$

Trong đó trì hoãn lan truyền trên đường dây là thời gian trì hoãn truyền xấu nhất để đi từ một trạm phát đến bất kỳ một trạm thu nào trên mạng cáp. Nó bao gồm cả thời gian trễ phải trải qua trên các repeater. Khe thời gian được làm tròn theo chiều tăng bằng với bội số của byte theo tốc độ bit được dùng. Số khe thời gian trì

hoãn trước lần truyền lại thứ N được chọn theo số ngẫu nhiên R phân bố đều trong dải $[0, 2^K]$, trong đó $K = \min(N, \text{giới hạn trì hoãn})$. Lưu đồ giải thuật tóm tắt hoạt động truyền được trình bày trên hình 7.13(a).

1.4.5. Tiếp nhận frame

Hình 7.13(b) tóm tắt một quá trình tiếp nhận một frame. Tại mỗi DTE tích cực kết nối vào mạng, trước hết đơn vị MAC phát hiện sự xuất hiện của tín hiệu đến từ transceiver và phát tín hiệu cảm nhận sóng mang để ngăn cản bất kỳ hoạt động truyền mới nào từ DTE này. Tuần tự mở đầu (preamble) đến được dùng để đồng bộ bit và sau đó luồng số liệu đã được mã hóa theo mã Manchester được dịch ngược lại thành dạng nhị phân thông thường. Luồng bit đến được xử lý ngay sau đó.

Trước hết các bit của preamble bị hủy bỏ cùng với SFD. Field địa chỉ đích được xử lý để xác định nơi nào nhận frame. Nếu đã xác định là của chính DTE liên quan, nội dung frame bao gồm địa chỉ đích, địa chỉ nguồn và số liệu được nạp vào bộ đệm để xử lý tiếp. Nội dung của FCS sẽ được so sánh với kết quả được tính toán bởi đơn vị MAC trong quá trình tiếp nhận frame, nếu bằng nhau, địa chỉ đầu tiên của bộ đệm chứa frame được chuyển cho lớp giao thức cấp cao hơn kế tiếp, dưới dạng một thực thể dịch vụ để tiếp tục xử lý. Các kiểm tra về tính hợp lệ cũng được thực hiện trên frame trước khi bắt đầu xử lý ở các công đoạn tiếp theo. Các kiểm tra này bao gồm đảm bảo frame chứa một số nguyên lân byte và nó không quá ngắn cũng không quá dài. Nếu một trong các kiểm tra này cho kết quả không hợp lệ thì frame sẽ bị hủy bỏ và một trạng thái lỗi được gửi lên mức cao hơn.

Khởi đầu, luồng bit truyền xuất phát từ một dụng độ được tiếp nhận bởi mọi DTE đang ở trạng thái tích cực theo phương thức như một frame hợp lệ. Sau khi các DTE liên quan đến dụng độ phát hiện được dụng độ này và truyền dây nhồi (jam sequence), chúng ngừng truyền. Các mảnh của các frame được tiếp nhận theo cơ chế này vì có kích thước nhỏ hơn kích thước giới hạn nhỏ nhất của frame nên chúng bị hủy bỏ bởi các DTE tiếp nhận. Chấp nhận kích thước frame tối đa có nghĩa là có thể xác định chiều dài của các bộ đệm được dùng cho truyền và nhận.

1.4.6. Các đặc tả IEEE 802.3 có băng thông 10Mbps

Để phân biệt trong số các ứng dụng thực tế khác nhau, tổ chức IEEE đã công bố ký hiệu ngắn theo qui tắc:

<Tốc độ dữ liệu theo Mbps><phương thức báo hiệu><chiều dài tối đa của một segment, tính theo đơn vị hàng trăm mét, hoặc loại cáp dùng>

Các kiểu có sẵn:

-10BASE 2

-10BASE 5

-10BASE-T



-10BROAD 36

-10BASE-F

1.4.6.1. Đặc tả môi trường 10BASE 5

10BASE 5 có nguồn gốc từ môi trường 802.3 và dựa chủ yếu vào Ethernet. 10BASE 5 chỉ định dùng cáp đồng trục có trở kháng 50Ω và dùng báo hiệu số Manchester. Chiều dài tối đa của một segment là 500m. Chiều dài của mạng có thể được mở rộng bằng cách dùng các repeater, nó trong suốt đối với lớp MAC; khi không đệm số liệu, chúng không cô lập segment này với segment khác. Vì thế nếu hai trạm trên hai segment khác nhau cùng nỗ lực truyền tại cùng một thời điểm, sẽ xảy ra đụng độ. Để tránh nối thành vòng, chỉ một đường dẫn của segment và repeater được phép giữa bất kỳ hai trạm nào. Tiêu chuẩn này cho phép có bốn repeater trên một đường dẫn giữa bất kỳ hai trạm nào, do đó chiều dài mở rộng tối đa chỉ đến 2,5km.

1.4.6.2. Đặc tả môi trường 10BASE 2

Để cung cấp một hệ thống có giá thành thấp cho LAN của các máy tính cá nhân, 10BASE2 được công bố. Cũng giống như 10BASE 5, đặc tả này cũng dùng cáp đồng trục trở kháng 50Ω và báo hiệu Manchester. Khác nhau chủ yếu là 10BASE 2 dùng cáp nhỏ hơn, nó hỗ trợ ít đầu nối hơn qua một cự ly ngắn hơn so với cáp 10BASE 5.

Vì có cùng tốc độ hoạt động, nên có thể kết hợp các segment 10BASE 2 và 10BASE 5 trên cùng một mạng, bằng cách dùng một repeater. Chỉ có một giới hạn là một segment 10BASE 2 không được bắt đầu cho hai segment 10BASE 5, vì một backbone phải là một tác nhân hạn chế nhiễu.

1.4.6.3. Đặc tả môi trường 10BASE-T

Chấp nhận bớt đi chiều dài mạng, có thể phát triển một LAN 10Mbps dùng cáp xoắn không được bảo vệ UTP (unshield twisted-pair). Loại dây này, thường đã được lắp đặt trước trong các cao ốc văn phòng như thường hợp cáp điện thoại, và có thể được dùng cho các LAN. Một tiếp cận như vậy được đặc tả trong 10BASE-T. Đặc tả này định nghĩa một topo dạng sao. Một hệ thống đơn giản bao gồm một số trạm được kết nối đến điểm trung tâm qua hai cặp cáp xoắn, điểm trung tâm được đề cập đến như một repeater đa port. Điểm trung tâm chấp nhận tín hiệu nhập trên bất kỳ một đường dây nào và lắp lại trên tất cả các đường dây khác.

Các trạm nối vào repeater đa port thông qua các liên kết điểm nối điểm. Mỗi liên kết bao gồm hai cặp cáp xoắn UTP. Vì tốc độ truyền cao và chất lượng truyền dẫn của cáp UTP xấu, nên chiều dài của một liên kết bị giới hạn đến 100m. Tùy theo, vẫn có thể dùng cáp quang, trong trường hợp này chiều dài tối đa là 500m.

1.4.6.4. Đặc tả môi trường 10BROAD36

Đặc tả 10BROAD36 là đặc tả 802.3 chỉ cho băng rộng. Môi trường được dùng là cáp đồng trục CATV 75Ω . Cho phép cấu hình cáp đôi hay cáp được phân chia. Chiều

dài tối đa của một segment đơn, tính từ headend là 1800m; như vậy chiều dài điểm nối điểm mở rộng đến 3600m.

Báo hiệu trên cáp là DPSK (differential phase shift keying). DPSK đã được trình bày trong chương 2. Cũng cần nói thêm rằng theo PSK thông thường, bit nhị phân 0 được đại diện bởi một sóng mang với một pha đặc biệt, và bit nhị phân 1 được đại diện bởi một sóng mang có pha ngược lại (khác nhau 180°). DPSK dùng các mã vi phân, trong đó một thay đổi pha xảy ra khi xuất hiện nhị phân 0, và không có thay đổi pha khi xuất hiện nhị phân 1. Ưu điểm của mã hóa vi phân là bộ thu dễ phát hiện ra sự thay đổi pha hơn là xác định chính xác pha.

Các đặc tính của quá trình điều chế được đặc tả sao cho tốc độ 10Mbps phù hợp với băng thông 14MHz.

1.4.6.5. Đặc tả môi trường 10BASE-F

Đặc tả 10BASE-F cho phép user đạt được các ưu thế về cự ly và các đặc tính truyền dẫn khi dùng cáp quang. Chuẩn này gồm 3 đặc tả chính:

- 10BASE-FP (passive): Một cấu hình passive-star (dạng sao thụ động) để liên kết các máy trạm và các repeater với 1km trên một segment.
- 10BASE-FL(link): Định nghĩa một liên kết điểm nối điểm có thể nối các máy trạm và các repeater cách xa nhau đến 2km.
- 10BASE-FB(backbone), định nghĩa một liên kết điểm nối điểm có thể kết nối các repeater cách nhau đến 2km.

Tất cả 3 đặc tả trên đều dùng một cặp cáp quang cho mỗi liên kết truyền, một dây cho mỗi hướng truyền. Trong tất cả các trường hợp này đều dùng lược đồ mã hóa đường dây Manchester. Mỗi phần tử tín hiệu Manchester được đổi thành phần tử tín hiệu quang, sự xuất hiện ánh sáng tương ứng với mức cao và không có ánh sáng tương ứng với mức thấp. Do đó luồng bit Manchester tốc độ 10Mbps thực sự cần 20Mbps trên cáp quang.

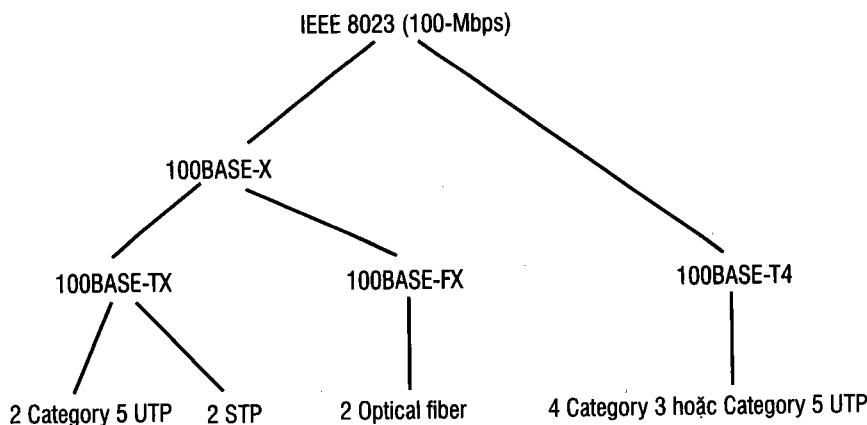
10BASE-FP định nghĩa một hệ thống dạng sao (star) thụ động có thể hỗ trợ đến 33 trạm kết nối vào trung tâm. 10BASE-FL và 10BASE-FB định nghĩa các kết nối điểm nối điểm được dùng để mở rộng chiều dài mạng. Khác nhau chủ yếu giữa hai loại là 10BASE-FB cho phép sử dụng truyền lại đồng bộ (synchronous retransmission). Với báo hiệu đồng bộ, tín hiệu cáp quang đi vào một repeater được định thời lại căn cứ vào một đồng hồ cục bộ và được truyền lại. Với báo hiệu bất đồng bộ chuẩn, được dùng với 10BASE-FL không có thao tác định thời lại, do vậy bất kỳ một sự gián đột rộng xung (thời gian) nào đều được giữ nguyên và lan truyền xuyên qua một dây nối tiếp các repeater. Kết quả là 10BASE-FB có thể được dùng để mắc nối tiếp đến 15 repeater để đạt được chiều dài mạng lớn hơn.



1.4.7. Các đặc tả IEEE 802.3 có băng thông 100Mbps

Fast Ethernet đề cập đến một tập các đặc tả được phát triển bởi IEEE 802.3 để cung cấp một giá thành rẻ hơn, là các LAN tương thích với Ethernet hoạt động với tốc độ 100Mbps. Thiết kế cho các chuẩn này là 100BASE-T. Tổ chức này cũng định nghĩa một số các tùy chọn được dùng với môi trường dẫn khác nhau.

Hình 7.14 trình bày một sơ đồ tóm tắt với các ký hiệu đặc tả môi trường được dùng. Tất cả các tùy chọn của 100BASE-T đều dùng giao thức và khuôn dạng của IEEE 802.3 MAC. 100BASE-X đề cập đến một tập các tùy chọn dùng các đặc tả môi trường vật lý xuất phát từ định nghĩa cho FDDI. Tất cả các lược đồ 100BASE-X dùng hai liên kết vật lý giữa hai node: một cho truyền và một cho nhận. 100BASE-TX dùng cáp xoắn có bảo vệ STP hay cáp UTP chất lượng cao. 100BASE-FX dùng cáp quang.



Hình 7.14 Các tùy chọn 100BASE-T của IEEE 802.3

Trong xây dựng mạng mỗi tùy chọn của 100BASE-X yêu cầu cài đặt một cáp mới. Trong số các trường hợp này, 100BASE-T4 định ra một kiểu có giá thành thấp có thể dùng cáp UTP loại 3 hay voice grade UTP cùng với cáp chất lượng cao UTP loại 5. Để đạt được tốc độ số liệu 100Mbps qua một cáp chất lượng thấp, 100BASE-T4 yêu cầu dùng 4 cặp dây xoắn giữa hai node, hoạt động truyền số liệu dùng 3 đôi dây trên một hướng tại một thời điểm.

Trong tất cả các tùy chọn của 100BASE-T, topo của chúng là tương tự như 10BASE-T, đó là dạng sao (star). Bảng 7.1 tóm tắt các tùy chọn của 100BASE-T.

Bảng 7.1 Các tùy chọn của 100BASE-T

	100BASE-TX	100BASE-FX	100BASE-T4
Môi trường truyền	2 đôi STP	2 đôi UTP loại 5	2 cáp quang
Kỹ thuật báo hiệu	4B5B, NRZI	4B5B, NRZI	8B6T, NRZ
Tốc độ dữ liệu	100Mbps	100Mbps	100Mbps
Chiều dài segment tối đa	100m	100m	100m
Mở rộng	200m	200m	400m
			200m

1.4.7.1. 100BASE-X

Trong tất cả các môi trường truyền được đặc tả dưới 100BASE-X, tốc độ số liệu 100Mbps đạt được bằng cách truyền qua một liên kết đơn (cáp xoắn đơn hay cáp quang đơn). Trong tất cả các môi trường này luôn yêu cầu một lược đồ mã hóa tín hiệu hiệu quả. Lược đồ mã hóa được chọn có nguồn gốc từ định nghĩa cho FDDI, thường được đề cập đến như là 4B5B-NRZI.

Thiết kế 100BASE-X gồm có hai đặc tả môi trường vật lý, một cho cáp xoắn được gọi là 100BASE-TX, và một cho cáp quang, được gọi là 100BASE-FX.

100BASE-TX dùng hai cặp cáp xoắn, một cặp cho truyền và một cặp cho nhận. Có thể dùng STP hay UTP loại 5. Lược đồ báo hiệu MTL-3 được dùng cho trường hợp này.

100BASE-FX dùng hai cáp quang, một cho truyền và một cho nhận. Với 100BASE-FX, cần dùng một thiết bị để chuyển đổi luồng mã hóa 4B/5B-NRZI thành các tín hiệu quang. Kỹ thuật điều chế được dùng ở đây là điều chế cường độ. Một bit nhị phân 1 được biểu diễn bởi một xung ánh sáng; một bit nhị phân 0 được biểu diễn bởi một xung ánh sáng cường độ thấp hay không có ánh sáng.

1.4.7.2. 100BASE-T4

100BASE-T4 được thiết kế để tạo ra tốc độ 100Mbps qua một cáp xoắn chất lượng thấp loại 3, do đó đạt được các ưu điểm về lắp đặt trên diện rộng của cáp loại 3 trong các tòa nhà. Đặc tả cũng chỉ ra rằng có thể tùy chọn cáp chất lượng cao loại 5. 100BASE-T4 không truyền một tín hiệu liên tục giữa các gói, điều này làm cho nó hữu dụng trong các ứng dụng dùng nguồn một chiều.

Trong 100BASE-T4 dùng cáp voie grade loại 3, đó không phải là lý do để đạt được 100Mbps trên một cặp cáp xoắn. Thay vì vậy, 100BASE-T4 chỉ ra rằng luồng số liệu truyền được chia thành 3 luồng số liệu riêng biệt, mỗi luồng có một tốc độ số liệu hiệu quả là 33Mbps. Có 4 cặp cáp xoắn được dùng. Số liệu truyền dùng 3 cặp và nhận dùng 3 cặp. Do đó trong 4 cặp cáp xoắn có hai cặp cần phải cấu hình để truyền theo hai hướng.

Giống với 100BASE-X, lược đồ mã hóa đơn giản NRZ không được dùng cho 100BASE-T4; vì nó sẽ phải cần một tốc độ báo hiệu 33Mbps trên mỗi cặp cáp xoắn và không cung cấp đồng bộ. Do đó lược đồ báo hiệu 3 trạng thái gọi là 8B6T được dùng.

1.5. Hệ thống Token ring và FDDI

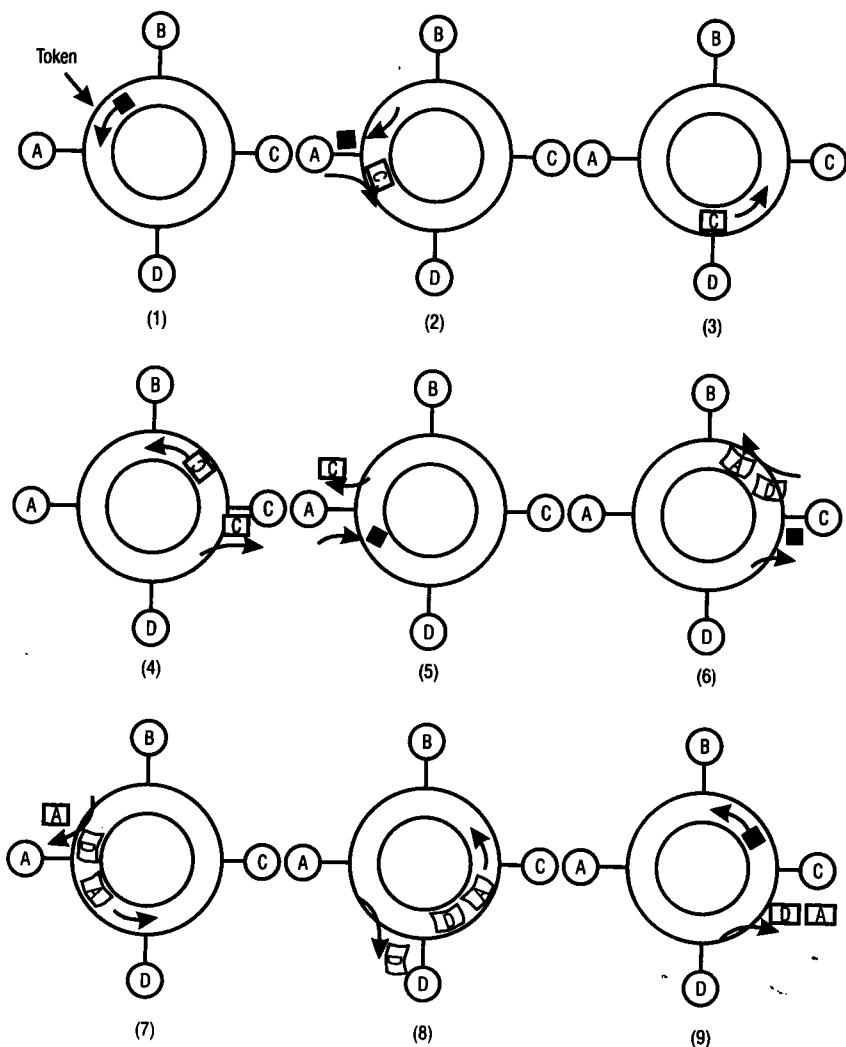
1.5.1. Điều khiển xuất môi trường trong IEEE 802.5

Như chúng ta đã biết kỹ thuật token ring nhờ vào hoạt động của một frame nhỏ, được gọi là token, token chạy vòng trên mạng khi tất cả các trạm đều rảnh rỗi. Các mạng token ring được dùng chủ yếu trong môi trường kỹ thuật và quản lý. Bất cứ khi nào một trạm muốn truyền phải đợi cho đến khi nó phát hiện một token chuyển qua nó. Trạm truyền sẽ bắt lấy token thông qua thao tác sửa một bit trong đó, chuyển nó từ một token thành một tuần tự bit đánh dấu đầu frame của một frame số liệu. Sau đó trạm này thêm và truyền phần còn lại của các field cần thiết để xây dựng một frame số liệu hoàn chỉnh.

Khi một trạm bắt lấy một token và bắt đầu truyền một frame số liệu, sẽ không còn token nào trên vòng nữa, vì vậy các trạm khác muốn truyền phải đợi. Frame trên vòng sẽ du ngoạn một vòng và được trạm truyền bắt lấy. Trạm truyền sẽ chèn một token mới vào vòng nếu cả hai điều kiện sau đây xảy ra:

- i. Trạm đã hoàn tất việc truyền tất cả các frame của nó.
- ii. Đầu của frame được truyền đã trở về (sau khi hoàn thành một vòng) trạm này.

Nếu chiều dài tính theo bit của ring ngắn hơn chiều dài của frame, thì điều kiện đầu tiên bao hàm cả điều kiện thứ hai; ngược lại trạm truyền sẽ giải phóng một token tự do sau khi truyền hoàn tất nhưng trước đó nó đã bắt đầu nhận frame quay về. Điều kiện thứ hai không cần khắt khe lắm, và nó được bỏ qua trong một số hoàn cảnh nào đó. Ưu điểm của việc tuân thủ điều kiện thứ hai là nó đảm bảo chỉ một frame số liệu tại một thời điểm trên vòng và chỉ một trạm có thể truyền tại một thời điểm, nhờ đó đơn giản được thủ tục khắc phục lỗi.



Hình 7.15 *Hoạt động của Token Ring.*

Một khi có một token mới được chèn vào ring, trạm kế tiếp nhận luồng số liệu để truyền sẽ bắt lấy token và truyền. Hình 7.15 mô tả kỹ thuật này. Trong ví dụ này, A gửi một gói đến C, C nhận gói này và sau đó gửi các gói của nó đến D.

Dễ thấy dưới điều kiện tải nhẹ, sẽ không hiệu quả đối với token ring vì một trạm phải đợi cho token đi được một vòng trước khi truyền. Tuy nhiên dưới điều kiện tải nặng, là lúc chúng phát huy được ưu thế, các chức năng ring trong *mô hình round-robin* đạt được hiệu quả khá cao. Mô hình round-robin là mô hình trong đó thứ tự truyền được điều khiển bằng *giải thuật lập lịch round-robin* (round-robin scheduling algorithm), nghĩa là tất cả các hoạt động truyền đều phải xảy ra theo một thứ tự



nghiêm ngặt. Để kiểm chứng hiệu quả của chức năng ring trong mô hình này, hãy xem cấu hình ở hình 7.15. Sau khi trạm A truyền, nó giải phóng token. Trạm đầu tiên có cơ hội truyền là D. Nếu D truyền, sau đó giải phóng token và C có cơ hội truyền kế tiếp, và cứ thế.

Ưu điểm mang tính nguyên lý của token ring là tính điều khiển linh hoạt qua thao tác truy xuất mà nó thực hiện. Trong lược đồ đơn giản này chỉ mô tả truy xuất có thành công hay không. Lược đồ này có thể được dùng để điều chỉnh truy xuất để cung cấp quyền ưu tiên và bảo đảm các dịch vụ về băng thông.

Điều bất tiện trong nguyên lý token ring là yêu cầu duy trì token. Mất token đồng nghĩa với sự cản trở các hoạt động tiếp theo. Sự gấp đôi token cũng dẫn đến tình trạng phá vỡ vòng. Một trạm phải được chọn làm giám sát (monitor) để đảm bảo tuyệt đối chỉ có một token trong ring và đảm bảo một token tự do chắc chắn được chèn vào nếu cần.

1.5.2. Giao tiếp ring

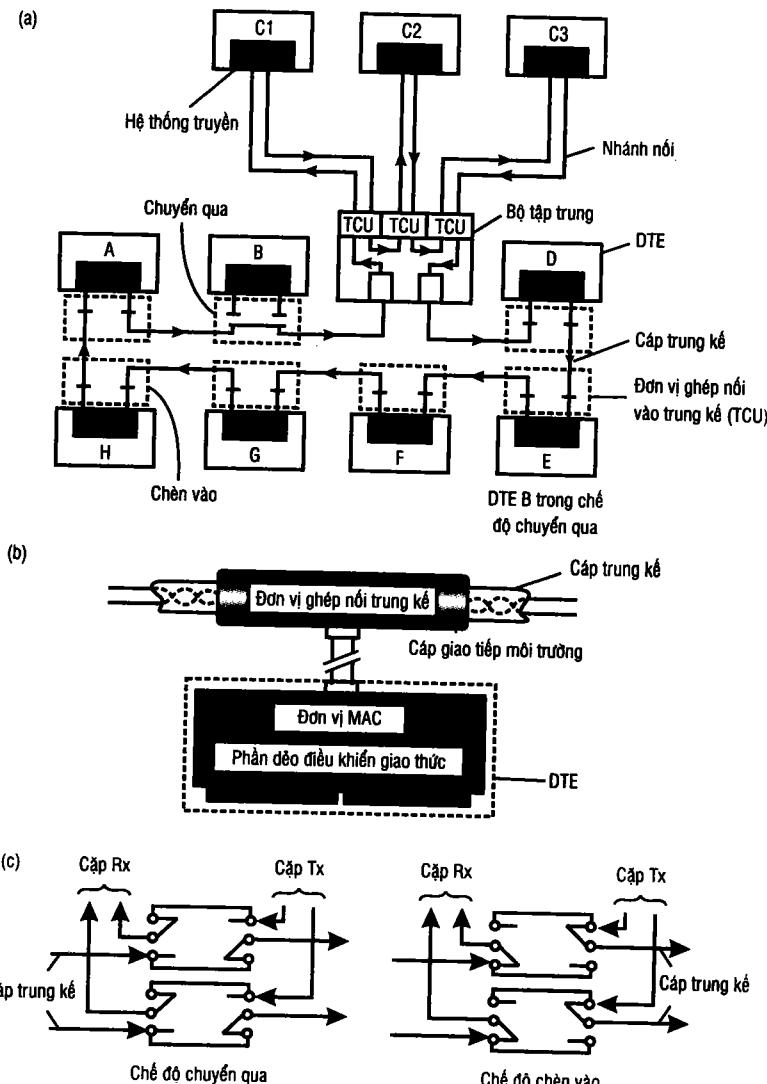
Một mạng token ring tiêu biểu được trình bày trên hình 7.16(a) và các thành phần cần thiết để kết nối một DTE vào môi trường cáp được mô tả trong phần (b) và (c). Như trong hình 7.16(b) một DTE có thể được kết nối một cách trực tiếp vào ring thông qua một bộ tập trung (concentrator). Thiết bị này được nối trực tiếp vào cáp trung kế chính và cung cấp một số điểm nối cho các DTE. Bộ tập trung thường được dùng làm ring đơn giản trong một tòa nhà. Thông thường nó được đặt tại vị trí mà cáp trung kế đi vào.

Đơn vị ghép trung kế TCU (trunk couple unit) hình thành nên một giao tiếp vật lý với môi trường cáp. Nó chứa một tập các rơ le và các mạch điện tử bổ sung để điều khiển và thu các tín hiệu đến và đi từ cáp. Các rơ le được sắp xếp sao cho bất cứ khi nào DTE ngừng hoạt động (tắt máy), TCU ở trong trạng thái trung chuyển và đường dẫn xuyên qua TCU tiếp tục được duy trì. Việc chèn một DTE vào vòng được điều khiển bởi đơn vị MAC nằm trên card điều khiển truyền bên trong DTE. Đơn vị MAC khởi động việc chèn DTE vào vòng bằng cách kích hoạt cả hai cặp rơ le của TCU. Như trình bày trong hình 7.16(c), khi chèn, với cách sắp xếp này làm cho tất cả các tín hiệu đến được định tuyến xuyên qua đơn vị MAC này. Các mạch truyền nhận trong đơn vị MAC này chỉ làm nhiệm vụ đơn giản là đọc và chuyển tiếp tín hiệu nhận được theo hướng truyền nếu DTE không phải là nguồn phát frame này, hoặc xóa tín hiệu nhận được từ vòng nếu nó đã khởi động truyền.

Việc dùng hai cặp rơ le trong phương pháp này có nghĩa là đơn vị MAC có thể phát hiện ra các lỗi hỏng mạch hay ngắn mạch trong các cặp dây tín hiệu truyền và nhận. Trong trạng thái cầu nối (bypass state) đơn vị MAC có thể kiểm soát các chức năng tự kiểm thử, vì bất cứ số liệu nào được gửi ra trên cặp dây truyền sẽ vòng lại trên cặp dây thu. DTE này được kết nối đến TCU bằng một cáp được bảo vệ chứa hai dây cáp xoắn: một cho truyền và một cho thu.

Đơn vị MAC đáp ứng các chức năng lưu tạo frame và tách frame, tạo FCS để phát hiện lỗi, và hiện thực giải thuật MAC. Nó cung cấp một đồng hồ chủ trên ring, đồng hồ này được dùng để mã hóa và giải mã số liệu khi DTE đóng vai là một monitor tích cực trên ring. Mỗi luồng bit chạy trên ring được mã hóa theo Manchester vi phân nhờ vào monitor tích cực trên vòng và tất cả các DTE khác trên ring sau đó

bắt nhịp vào tần số và pha của luồng bit này nhờ dùng mạch DPLL (digital phase lock-loop). Ngoài ra khi DTE là monitor tích cực, nó đảm bảo ring có một thời gian trễ tối thiểu (minimum latency time). Thời gian này được đo lường theo đơn vị bit cùng với tốc độ truyền số liệu trên ring, là khoảng thời gian cần để tín hiệu lan truyền đúng một vòng trên ring. Thời gian trễ này bao gồm trễ do truyền tín hiệu qua môi trường truyền của ring cùng với tổng thời gian trễ truyền xuyên qua mỗi đơn vị MAC. Để token điều khiển chạy vòng liên tục trên ring khi không có máy trạm nào cần dùng ring (là lúc tất cả các DTE chỉ ở chế độ lặp lại), ring phải có một thời gian trễ tối thiểu ít nhất bằng số bit trong token để đảm bảo token không bị hỏng.

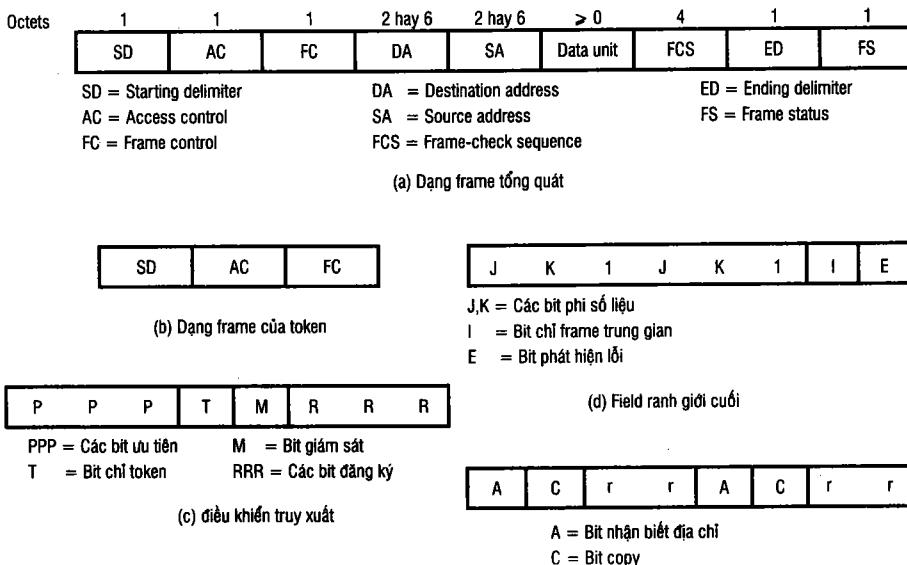


Hình 7.16 Các thành phần của mạng token ring:
(a) Cấu hình ring (b) Giao tiếp DTE (c) Sơ đồ TCU.



Token có chiều dài 24 bit, vậy khi DTE là monitor tích cực, MAC của nó cung cấp một bộ đếm 24 bit cố định, nó trở thành một bộ phận có hiệu quả của ring đảm bảo hoạt động chính xác dưới tất cả các điều kiện. Mặc dù tốc độ báo hiệu số liệu có nghĩa trên ring được điều khiển bởi một đồng hồ chủ trong monitor, nhưng việc dùng một mạch DLL trong mỗi đơn vị MAC có nghĩa là tốc độ báo hiệu thực sự có thể thay đổi một ít trên ring. Trường hợp thay đổi xấu nhất là khi số DTE ở trạng thái tích cực đạt tối đa (250), điều này tương đương với cộng hay trừ 3 bit. Ngoại trừ trường hợp thời gian trễ không đổi, khi thời lượng này giảm các bit sẽ bị hỏng, và các bit sẽ bị thêm vào nếu thời lượng này tăng. Để duy trì thời lượng này không đổi, một bộ đếm nhằm tăng cường độ linh hoạt có chiều dài 6 bit được thêm vào bộ đếm cố định 24 bit đã có. Kết quả là bộ đếm 30 bit được khởi động đến 27 bit. Nếu tín hiệu nhận được tại đơn vị MAC chủ là nhanh hơn bộ dao động chủ, thì bộ đếm được mở rộng một bit. Ngược lại, nếu tín hiệu nhận chậm hơn, bộ đếm giảm xuống một bit. Theo phương pháp này ring luôn có đủ số bit để đảm bảo token chạy vòng liên tục trên vòng khi tất cả các DTE ở trạng thái rảnh rỗi.

1.5.3. Khuôn dạng frame IEEE 802.5



Hình 7.17

Khuôn dạng frame IEEE 802.5

Hình 7.17 mô tả kiến trúc frame cho giao thức 802.5. Nó bao gồm các field sau đây:

- **Sarting delimiter (SD):** chỉ định đầu frame. SD bao gồm các mẫu báo hiệu có thể phân biệt với số liệu. Nó được mã hóa như sau: JK0JK000. Trong đó

J và K là các ký hiệu không phải số liệu. Dạng thực sự của một ký hiệu khác số liệu phụ thuộc vào tín hiệu được mã hóa trên môi trường.

- **Access control (AC):** có dạng PPPTMRRR, trong đó PPP và RRR là 3 bit ưu tiên và các biến dự trữ, M là bit giám sát; việc sử dụng nó sẽ được giải thích ngay dưới đây. T chỉ ra frame là token hay số liệu. Trong trường hợp frame là một token, chỉ còn lại field ED.
- **Frame control (FC):** chỉ cho biết đây có phải là frame số liệu LLC hay không. Nếu không phải, các bit trong field này điều khiển hoạt động của giao thức MAC trên ring.
- **Destination address (DA):** giống như ở 802.3.
- **Source address(SA):** giống như ở 802.3.
- **Data :** chứa các đơn vị số liệu của LLC.
- **Frame check sequence(FCS):** giống như ở 802.3.
- **End delimiter(ED):** chứa các bit phát hiện lỗi (E), nó được set nếu bất kỳ repeater nào phát hiện ra một lỗi, và bit trung gian (I) được dùng để chỉ rằng frame này không phải là frame cuối trong một cuộc truyền đa frame.
- **Frame status (FS):** chứa bit nhận biết địa chỉ (A) và bit copy frame (C), việc dùng các bit này được giải thích bên dưới. Vì bit A và bit C nằm ngoài phạm vi của FCS, nên chúng được gấp đôi để thực hiện một thao tác kiểm tra dự phòng nhằm phát hiện các cài đặt sai.

Có thể xem xét lại giải thuật token ring trong trường hợp có dùng quyền ưu tiên. Trong trường hợp này, các *bit ưu tiên* và *bit giữ chỗ* được cho là 0. Một trạm muốn truyền đợi cho đến khi token chuyển qua nó, tức là khi bit token trong frame chuyển qua có giá trị 0 trong field AC. Trạm bắt lấy token bằng cách đặt bit token thành 1. Các field SD và AC của token nhận được bây giờ đóng vai trò là hai field đầu tiên của frame gửi ra. Trạm sẽ truyền một hay nhiều frame, tiếp tục cho đến khi nguồn cung cấp frame của nó đã cạn hoặc thời gian được phép giữ token đã hết. Khi field AC của frame sau cùng đã quay trở về trạm, trạm đặt lại bit token thành 0 và thêm vào một field ED, kết quả là một token mới được chèn vào trong ring.

Các trạm trong chế độ nhận lăng nghe trên ring. Mỗi trạm có thể kiểm tra các frame chuyển qua nó để tìm lỗi và có thể đặt bit E thành 1 nếu phát hiện được một lỗi. Nếu một trạm nào đó phát hiện địa chỉ MAC trùng với địa chỉ MAC của nó, nó đặt bit A thành 1; cũng có thể copy frame này, và đặt bit C thành 1. Thao tác này cho phép trạm nguồn biết được ba kết quả khác nhau của một frame truyền là:

- Trạm đích không tồn tại hay không mở máy nếu A=0 và C=0.
- Trạm đích tồn tại nhưng frame không được copy A=1 và C=0.
- Frame được tiếp nhận A=1 và C=1.



1.5.4. Ưu tiên trên vòng token.

Chuẩn 802.5 bao gồm các đặc tả cho một cơ cấu ưu tiên tùy chọn. Có 8 mức ưu tiên được hỗ trợ bởi hai field 3 bit trong mỗi frame số liệu và token: một field ưu tiên và một field giữ chỗ. Để giải thích giải thuật này, cần định nghĩa các biến sau đây:

P_f = ưu tiên của frame được truyền bởi trạm.

P_s = ưu tiên phục vụ : ưu tiên của token hiện hành

P_r = giá trị của P_s chứa trong token sau cùng được trạm tiếp nhận.

R_s = giá trị đăng ký giữ chỗ trong token hiện hành.

R_k = giá trị đăng ký giữ chỗ cao nhất trong các frame nhận được bởi trạm này trong thời gian quay của token sau cùng.

Lược đồ này làm việc như sau:

- i. Một trạm muốn truyền phải đợi một token với $P_s \leq P_f$
- ii. Trong khi đợi, một trạm có thể đăng ký một token tương lai theo mức ưu tiên của nó (P_f). Nếu có một frame số liệu đi qua, và nếu field đăng ký giữ chỗ nhỏ hơn ưu tiên của nó ($R_s < P_f$), thì trạm này có thể đặt field giữ chỗ của frame này thành ưu tiên của nó ($R_s \leftarrow P_f$). Nếu một frame token chạy qua, và nếu ($R_s < P_f$ và $P_f < P_s$), thì trạm này đặt field giữ chỗ của frame này thành ưu tiên của nó ($R_s \leftarrow P_f$). Việc cài đặt này có tác dụng làm rõ ràng trước cho bất kỳ giá trị đăng ký ưu tiên mức thấp nào.
- iii. Khi một trạm bắt lấy một token, nó đặt bit token thành 1 để bắt đầu một frame số liệu, đặt field giữ chỗ của frame số liệu là 0, và giữ nguyên field ưu tiên (bằng với mức ưu tiên của token đến).
- iv. Tiếp theo, sau khi đã truyền một hay nhiều frame số liệu, trạm phát ra một token mới với field ưu tiên và field giữ chỗ được đặt như chỉ định trong bảng sau 7.2

Bảng 7.2 Các hoạt động được thực hiện bởi trạm giữ token để tiến hành lược đồ ưu tiên

CÁC ĐIỀU KIỆN	CÁC HÀNH ĐỘNG
Frame có sẵn AND $P_s \leq P_f$	Gửi frame
(Frame không có sẵn OR THT hết hạn) AND $P_s \geq \text{MAX}[R_r, P_f]$	Gửi token với $P_s \leftarrow P_f$ $P_s \leftarrow \text{MAX}[R_r, P_f]$
(Frame không có sẵn OR THT hết hạn) AND $P_s < \text{MAX}[R_r, P_f]$	Gửi token với $P_s \leftarrow \text{MAX}[R_r, P_f]$

AND $P_r > S_x$	$R_s \leftarrow 0$ Push $S_r \leftarrow P_r$ Push $S_x \leftarrow P_s$
(Frame không có sẵn OR THT hết hạn) AND $P_s < \text{MAX}[R_r, P_t]$ AND $P_r = S_x$	Gửi token với $P_s \leftarrow \text{MAX}[R_r, P_t]$ $R_s \leftarrow 0$ Pop S_r Push $S_x \leftarrow P_s$
(Frame không có sẵn OR (frame có sẵn AND $P_r < S_x$)) AND $P_r = S_x$ AND $R_r > S_r$	Gửi token với $P_s \leftarrow R_r$ $R_s \leftarrow 0$ Pop S_r Push $S_x \leftarrow P_s$
(Frame không có sẵn OR (frame có sẵn AND $P_r < S_x$)) AND $P_r = S_x$ AND $R_r \leq S_r$	Gửi token với $P_s \leftarrow R_r$ $R_s \leftarrow 0$ Pop S_r Pop S_x

Hiệu quả của các bước trên là sắp xếp các tranh chấp giữa các nhu cầu truyền và cho phép trạm đang đợi có mức ưu tiên cao nhất bắt lấy token_càng sớm càng tốt. Một tác dụng ngược cho thấy giải thuật này có ảnh hưởng xấu trong khi dùng lược đồ ưu tiên, đó là luôn có xu hướng đưa ưu tiên đến mức cao nhất và giữ nguyên ở đó. Để tránh xảy ra điều này, bắt buộc một trạm tăng mức ưu tiên (phát ra một token có ưu tiên cao hơn token mà nó nhận được) có trách nhiệm giảm mức ưu tiên về mức trước đó của nó. Do đó, một trạm tăng mức ưu tiên phải nhớ cả mức ưu tiên cũ và mới và phải giảm cấp ưu tiên của token vào thời điểm thích hợp. Mỗi trạm có trách nhiệm đảm bảo không có token chạy vòng một cách không xác định vì mức ưu tiên của nó quá cao. Bằng cách ghi nhớ mức ưu tiên của các hoạt động truyền trước đó, một trạm có thể phát hiện điều kiện này và giảm mức ưu tiên đến mức thấp hơn trước đó hay mức đã đăng ký.

Để thực hiện cơ cấu giảm cấp, mỗi trạm duy trì hai *ngăn xếp* (*stack*), một cho đăng ký và một cho ưu tiên.

S_x = stack được dùng để lưu giữ các giá trị mới của mức ưu tiên của token

S_r = stack được dùng để lưu giữ các giá trị ưu tiên cũ

Lý do dùng stack thay vì dùng các biến vô hướng là vì mức ưu tiên có thể được gia tăng một số lần bởi một hay nhiều trạm. Các giá trị tăng xếp nối tiếp phải được lấy ra theo thứ tự ngược.

Tóm lại, một trạm có một frame ưu tiên cao hơn frame hiện hành có thể đăng ký token kế tiếp cho mức ưu tiên của nó khi frame chuyển qua. Khi token kế được phát ra, nó sẽ có mức ưu tiên thấp hơn và không thể bắt lấy token, vì vậy nó chuyển đến trạm đã đăng ký hay một trạm trung gian cùng với số liệu gửi của mức ưu tiên bằng hay cao hơn mức ưu tiên đã đăng ký. Trạm nâng cấp mức ưu tiên này sẽ chịu trách nhiệm giảm cấp đến mức trước đó của nó khi tất cả các trạm có mức ưu tiên cao hơn đã hoàn tất việc truyền. Khi trạm này thấy một token với mức ưu tiên cao hơn sau khi đã truyền xong, nó giả sử rằng không có tài có mức ưu tiên cao hơn đang đợi, và nó giảm cấp ưu tiên của token trước khi chuyển nó đi tiếp.

Hình 7.18 là một ví dụ, các sự kiện sau xảy ra:

- i. A đang truyền một frame số liệu đến cho B với một mức ưu tiên 0. Khi frame đã hoàn thành một mạch của vòng và trở lại A, A sẽ phát ra một frame token. Tuy nhiên, khi frame số liệu chuyển đến D, D thực hiện một đăng ký với mức ưu tiên 3 bằng cách đặt field đăng ký giá trị 3.
- ii. A phát một token với field ưu tiên đặt là 3.
- iii. Nếu cả B và C không có số liệu có mức ưu tiên 3 hay lớn hơn cần phải gửi, chúng không thể bắt lấy token. Token này chạy vòng đến D, D nắm bắt token này và phát ra một frame số liệu.
- iv. Sau khi frame số liệu của D trở về D, D phát ra một token mới với cùng mức ưu tiên với token mà nó đã nhận: mức ưu tiên 3.
- v. A thấy một token có mức ưu tiên mà nó đã dùng để phát ra một token sau cùng: do đó nó bắt lấy token này ngay cả khi nó không có số liệu gửi.
- vi. A phát ra một token có mức ưu tiên trước đó: mức 0

Lưu ý rằng, sau khi A đã phát ra một token mức ưu tiên 3, bất kỳ một trạm nào có số liệu có mức ưu tiên 3 hay lớn hơn đều có thể bắt lấy token này. Giả sử vào thời điểm này trạm C có số liệu mức ưu tiên 4 cần truyền. C sẽ bắt lấy token này, truyền frame số liệu và phát lại một token mức ưu tiên 3, token này sau đó được D bắt lấy.

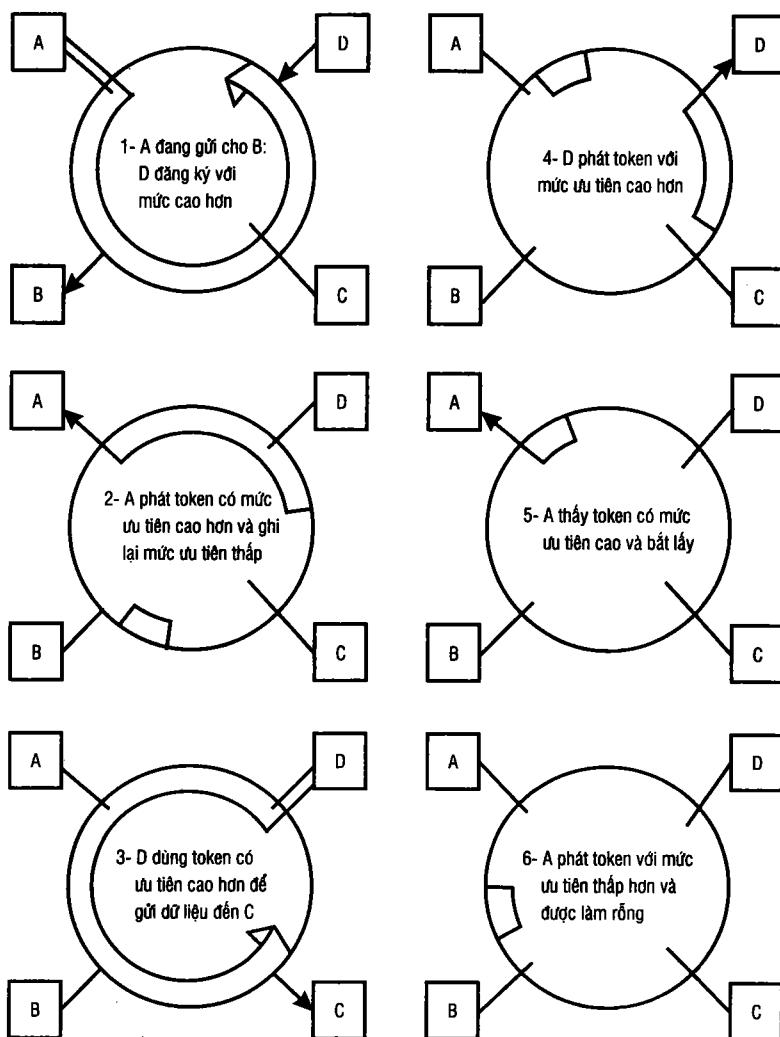
Vào thời điểm mà một token mức ưu tiên 3 đến A, tất cả các trạm có số liệu cần gửi có mức ưu tiên 3 hay lớn hơn sẽ có cơ hội. Do vậy đây là thời điểm thích hợp để A giảm cấp ưu tiên của token này.

Sự giải phóng token sớm

Khi một trạm phát ra một frame, nếu chiều dài của vòng nhỏ hơn frame, thì điểm đầu của frame được truyền trả về trạm truyền trước khi nó hoàn thành việc truyền; trong trường hợp này, trạm này có thể phát ra một token sớm hơn thời điểm nó kết thúc truyền frame số liệu. Nếu frame ngắn hơn chiều dài ring, thì sau khi một

trạm đã hoàn tất việc truyền một frame, nó phải đợi cho đến khi điểm đầu của frame trở về trước khi phát ra một token, trong trường hợp này, một phần công suất của ring không được dùng.

Để tăng hiệu quả sử dụng ring, một tùy chọn phát token sớm được thêm vào chuẩn 802.5. Việc phát token sớm cho phép một trạm đang truyền giải phóng một token sớm hơn thời điểm nó hoàn tất truyền frame, bắt chước đầu của frame đã trở về trạm truyền hay chưa. Mức ưu tiên được dùng cho một token được giải phóng trước khi nhận được header của frame truyền trước đó là mức ưu tiên của frame mới vừa nhận.



Hình 7.18 Lược đồ ưu tiên token ring



Một ảnh hưởng của việc giải phóng token sớm là thời gian trễ truy xuất của tài ưu tiên có thể gia tăng khi ring mang tải nặng với các frame ngắn. Vì một trạm phải phát một token trước khi nó có thể đọc các bit đăng ký của frame mới được truyền, trạm sẽ không đáp ứng được các đăng ký ưu tiên này. Do đó, cơ cấu ưu tiên bị hạn chế một phần.

1.5.5. Đặc tả lớp vật lý IEEE 802.5

Chuẩn 802.5 chỉ định dùng cáp STP với các tốc độ 4 và 16 Mbps dùng mã hoá Manchester vi phân. Một đặc tả trước đây cho hệ thống tốc độ 1 Mbps đã bị bỏ qua trong các tài liệu gần đây của chuẩn này.

Một nét mới thêm vào chuẩn này là dùng cáp UTP với tốc độ 4 Mbps.

Bảng 7.3 Các tùy chọn môi trường lớp vật lý của IEEE 802.5

Môi trường truyền	STP	UTP
Tốc độ dữ liệu (Mbps)	4 hay 6	4
Kỹ thuật báo hiệu	Manchester vi phân	Manchester vi phân
Số lượng repeater tối đa	250	72
Khoảng cách tối đa giữa các repeater	Không đặc tả	Không đặc tả

1.5.6. Điều khiển trung xuất FDDI

FDDI là một lược đồ token ring, tương tự các đặc tả IEEE 802.5, nó được thiết kế cho các ứng dụng trên cả LAN và MAN.

Tuy vậy, cũng có vài khác biệt được thiết kế để phù hợp với tốc độ cao (100Mbps) của FDDI.

1.5.6.1. Cấu trúc frame MAC

Hình 7.19 mô tả khuôn dạng của frame cho giao thức FDDI. Chuẩn này định nghĩa nội dung của khuôn dạng này qua các ký hiệu, mỗi ký hiệu số liệu tương ứng với 4 bit số liệu. Sở dĩ dùng các ký hiệu này là vì tại lớp vật lý số liệu được mã hóa trong các nhóm 4 bit. Tuy nhiên, các thực thể MAC thực tế phải liên hệ đến các bit riêng, vì vậy các thảo luận sau đây lúc thì tham chiếu đến các ký hiệu 4 bit và đôi khi lại là các bit. Một frame khác một token và gồm có các field sau:

- Preamble:** Đồng bộ frame với mỗi đồng hồ trạm. Nguồn của frame dùng một field 16 ký hiệu (64 bit) nhàn rỗi, các trạm lặp tiếp sau có thể thay đổi chiều dài của field này để phù hợp với các yêu cầu đồng bộ. Ký hiệu nhàn rỗi là một mẫu không phải dữ liệu. Dạng thực sự của ký hiệu không phải số liệu tùy vào mã hoá tín hiệu trên môi trường.
- Starting delimiter (SD):** chỉ định đầu của frame. Nó được mã hoá như JK, trong đó J và K là các ký hiệu khác số liệu.
- Frame control (FC):** có khuôn dạng bit CLFFZZZZ, trong đó C chỉ rằng đây là frame đồng bộ hay bất đồng bộ (được giải thích bên dưới); L chỉ định dùng địa chỉ 16 bit hay 48 bit, FF cho biết đây là frame LLC, MAC hay frame đăng ký. Trong một frame điều khiển, 4 bit còn lại chỉ ra loại frame điều khiển.
- Destination address (DA):** chỉ định trạm mà frame hướng đến. Nó có thể là một địa chỉ vật lý duy nhất, một địa chỉ nhóm hay một địa chỉ quảng bá. Ring có thể chứa một hỗn hợp chiều dài địa chỉ 16 bit hay 48 bit.

- Source address (SA):** chỉ ra trạm gửi frame này.
- Information:** chứa một đơn vị số liệu LLC hay thông tin liên quan đến hoạt động điều khiển.
- Frame check sequence (FCS):** một mã CRC - 32 bit được tính trên các field FC, DA, SA và Information.
- Ending delimiter (ED):** chứa một ký hiệu khác số liệu (T) đánh dấu cuối frame, ngoại trừ field FS.
- Frame status (FS):** chứa các bộ chỉ thị lỗi phát hiện được (E), địa chỉ xác định được (A) và frame được copy (F).

Bits	64	8	8	16 hay 48	16 hay 48	≥ 0	32	4	1
	Preamble	SD	FC	DA	SA	Info	FCS	ED	FS

(a) Dạng frame tổng quát

Preamble	SD	FC	FS
----------	----	----	----

(b) Dạng frame của token

SD = Start-frame delimiter

FC = Frame control

DA = Destination address

SA = Source address

FSC = Frame-check sequence

Info

ED = Ending delimiter

FS = Frame status

Hình 7.19 Khuôn dạng frame FDDI

Mỗi bộ chỉ thị được biểu diễn bởi một ký hiệu, nếu là R chỉ thị cho "reret" hay "false" và nếu là S chỉ thị cho "set" hay "true".

Một token frame bao gồm các field sau:

- Preamble:** giống như trên.
- Starting delimiter :** giống như trên.
- Frame control (FC):** có khuôn dạng bit 10000000 hay 11000000 để chỉ đây là một token.
- Ending delimiter (ED):** chứa một cặp ký hiệu khác số liệu (T) kết thúc một token frame.

So sánh với frame 802.5 thấy rằng cả hai hoàn toàn giống nhau: Frame FDDI bao gồm một preamble để đồng bộ, đây là yếu tố rất cần khi tốc độ số liệu cao. Cả hai đều chỉ 16 và 48 bit cho phép trong cùng một mạng FDDI. Điều này linh hoạt hơn lược đồ được dùng trên tất cả các chuẩn 802. Cuối cùng, có vài khác biệt trong bit điều khiển. Ví dụ, FDDI không bao gồm các bit ưu tiên và bit đăng ký giữ chỗ; phân phôi truy xuất được kiểm soát theo một phương pháp khác, như mô tả dưới đây.

1.5.6.2. Giao thức MAC

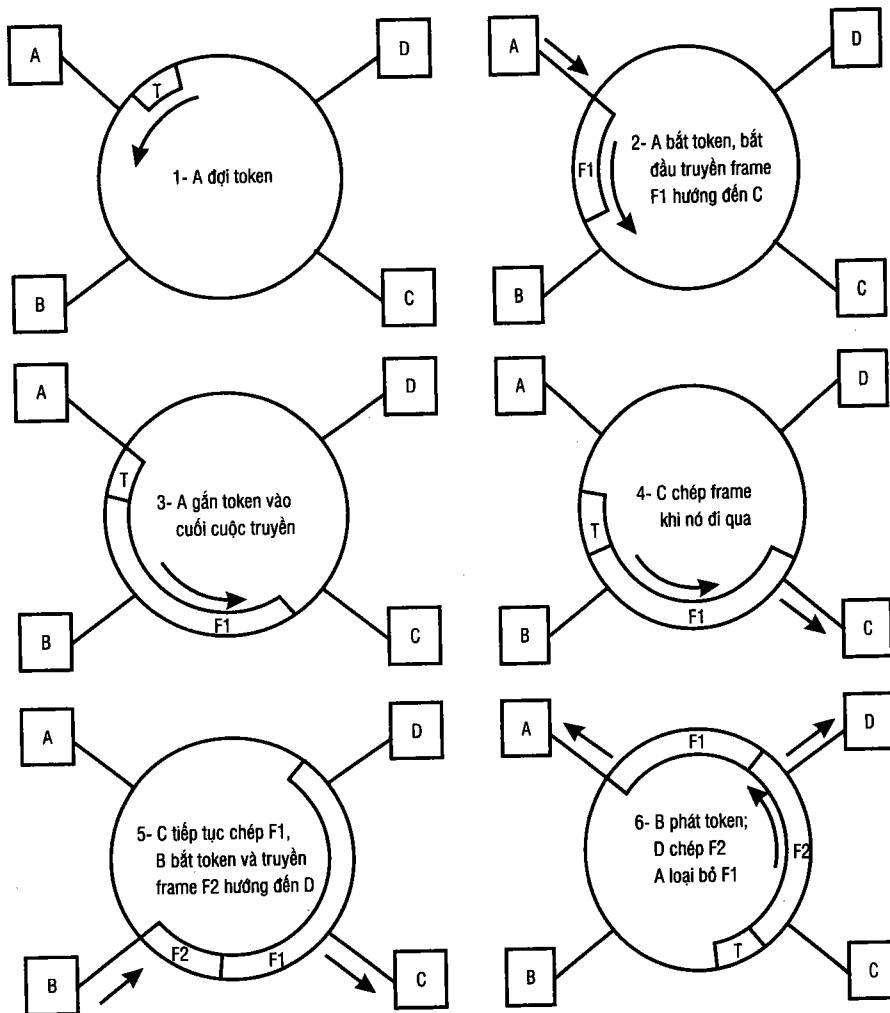
Về cơ bản giao thức MAC của FDDI giống với IEEE 802.5. Chỉ có hai khác biệt chính:

- Trong FDDI, một trạm đang đợi một token bắt láy token bằng cách hủy bỏ thao tác chuyển tiếp frame ngay khi frame được nhận ra là một token. Sau khi token được tiếp nhận hoàn toàn, trạm bắt đầu truyền một hay nhiều frame số liệu. Kỹ thuật



802.5 đảo một bit để chuyển một token thành phần đầu của một frame số liệu được xem như không thực tế đối với FDDI vì tốc độ số liệu của FDDI quá cao.

- ii. Trong FDDI, một trạm đang truyền các frame số liệu giải phóng một token ngay khi nó hoàn tất việc truyền frame, ngay cả khi nó chưa bắt đầu nhận các frame quay về. Kỹ thuật này giống như tùy chọn giải phóng token sớm của 802.5. Một lần nữa lý do của chọn lựa này cũng vì tốc độ số liệu cao, sẽ hoàn toàn không hiệu quả nếu yêu cầu trạm đợi các frame của nó quay về giống như trong hoạt động bình thường của 802.5.



Hình 7.20 Ví dụ hoạt động của FDDI token ring

Hình 7.20 trình bày ví dụ hoạt động của ring. Sau khi A đã bắt token này, nó truyền frame F1 và truyền một token mới ngay tức thì. F1 được chỉ định đến trạm C, trạm C copy khi frame chạy qua. Khi frame này quay về A, A hủy bỏ frame. Trong lúc đó B bắt token được phát ra bởi A và truyền một frame F2 kèm theo sau là một token. Hoạt động này có thể lặp lại một số lần bất kỳ, vì vậy tại bất kỳ thời điểm nào đều có thể có nhiều frame chạy trên vòng. Mỗi trạm có trách nhiệm hủy bỏ các frame của nó căn cứ vào field địa chỉ nguồn.

Mỗi trạm có thể kiểm tra các bit chuyển qua để phát hiện các lỗi và có thể cài bộ chỉ thị E nếu phát hiện được một lỗi. Nếu một trạm phát hiện thấy địa chỉ của nó, trạm sẽ cài bộ chỉ thị A, nó cũng có thể copy frame này, khi đó tiếp tục cài bộ chỉ thị C, điều này cho phép trạm nguồn khi hủy bỏ một frame truyền trước đó, có thể phân biệt trong số ba điều kiện:

- Trạm không tồn tại hay không mở máy.
- Trạm mở máy nhưng frame không được copy
- Frame được copy

Trước khi một frame bị hủy bỏ, các bộ chỉ thị trạng thái (E, A, C) trong field FS có thể được kiểm tra để xác định kết quả truyền. Tuy nhiên, nếu có lỗi hay không nhận được điều kiện thu, thực thể giao thức MAC không nỗ lực truyền lại frame, nhưng sẽ thông báo điều kiện này cho LLC. Đây là trách nhiệm của LLC hay vài giao thức lớp cao hơn tiến hành hoạt động sửa chữa.

1.5.6.3. Phân phối dung lượng

Lược đồ ưu tiên được dùng trong 802.5 sẽ không làm việc được trong FDDI, vì một trạm thường phát ra một token trước khi frame truyền của nó quay về. Từ đó, việc dùng một field đăng ký là không hiệu quả. Hơn nữa, chuẩn FDDI hướng đến thực hiện điều khiển trên một mạng có dung lượng lớn hơn 802.5 để phù hợp các yêu cầu cho một mạng LAN tốc độ cao. Đặc biệt, lược đồ phân phối dung lượng FDDI cố gắng phù hợp với luồng số liệu hỗn hợp và bùng nổ tải.

Để phù hợp với yêu cầu này, FDDI định nghĩa hai loại tải đồng bộ và bất đồng bộ. Mỗi trạm được phân phối một phần khả năng (phần khả năng này có thể bằng 0), cụ thể là một khoảng thời gian dùng mỗi trường; các frame của trạm truyền trong thời gian này được xem như là các frame đồng bộ. Bất kỳ khả năng nào không được phân phối hay được phân phối nhưng không được dùng đều sẵn sàng để truyền thêm các frame khác, các frame truyền thêm này được xem như là các frame bất đồng bộ.

Lược đồ này làm việc như sau: thời gian quay token đến đích TTRT (Target token - rotation time) được định nghĩa; mỗi trạm lưu cùng giá trị của TTRT. Vài trạm hay tất cả các trạm có thể được cung cấp một phân phối đồng bộ (SA_i), có thể thay đổi trong các trạm. Phân phối này phải là:

$$D_{\text{Max}} + F_{\text{Max}} + \text{Token Time} + \Sigma S_{A_i} \leq \text{TTRT}$$

trong đó

S_{A_i} = phân phối đồng bộ cho trạm i

D_{MAX} = thời gian lan truyền trên một vòng

F_{MAX} = thời gian cần để truyền một frame dài nhất (45000 octet)

Token Time = thời gian cần để truyền một token

Việc gán các giá trị cho S_{A_i} nhờ vào một giao thức quản lý trạm thông qua trao đổi các frame quản lý mạng. Giao thức này đảm bảo thỏa mãn công thức trên. Khoi đầu, mỗi trạm có một phân phối bằng 0 và nó phải yêu cầu một thay đổi trong phân phối. Hỗ trợ phân phối đồng bộ là tùy chọn; một trạm không hỗ trợ phân phối đồng bộ có thể chỉ truyền tải bất đồng bộ.

Tất cả các trạm có cùng giá trị TTRT và được gán một giá trị SA, một cách riêng biệt. Ngoài ra, có vài biến cần cho hoạt động của giải thuật phân phối dung lượng được duy trì tại mỗi trạm:

Token - rotation timer (TRT)

Token - holding timer (THT)

Late counter (LC)

Mỗi trạm được khởi động với TRT bằng giá trị của TTRT và LC bằng 0. Khi bộ định thời (timer) được phép hoạt động, TRT bắt đầu đếm xuống. Nếu nhận được một token trước khi TRT hết hạn thì TRT được cài đặt lại bằng TTRT. Nếu TRT đếm đến 0 trước khi nhận một token, thì LC được gia tăng 1 và TRT được cài đặt lại bằng TTRT và bắt đầu đếm xuống lần nữa. Nếu TRT hết hạn lần thứ 2 trước khi nhận một token, LC tăng lên 1 đơn vị nữa thành 2, token này bị coi là đã mất và một quá trình ‘thỉnh cầu’ (claim) được khởi động (được miêu tả bên dưới). Do đó, LC ghi số lần mà TRT hết hạn do token được tiếp nhận trễ tại trạm này. Token được xem là đến sớm nếu TRT chưa hết hạn mà trạm đã nhận được token này, nghĩa là LC = 0.

Khi một trạm nhận token, các hành động tiếp theo của nó sẽ tùy thuộc vào token đến sớm hay trễ. Nếu token đến sớm, trạm này lưu thời lượng còn lại từ TRT trong THT, cài đặt lại TRT và cho phép TRT hoạt động.

$\text{THT} \leftarrow \text{TRT}$

$\text{TRT} \leftarrow \text{TTRT}$

enable TRT

Sau đó trạm có thể truyền theo các luật sau đây:

- Có thể truyền các frame đồng bộ trong một thời gian S_{A_i}

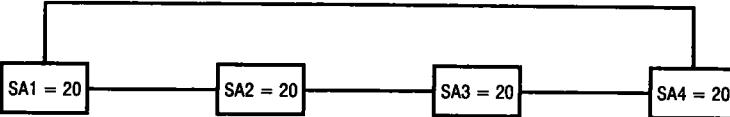
- ii. Sau khi truyền các frame đồng bộ, hay nếu không có frame đồng bộ để truyền, THT được phép hoạt động. Trạm này có thể bắt đầu truyền các frame bất đồng bộ khi THT > 0

Nếu một trạm nhận một token đến trễ, thì LC được cài zero và TRT tiếp tục chạy. Trạm có thể truyền các frame đồng bộ trong thời gian SA_i. Trạm này không truyền bất kỳ frame bất đồng bộ nào.

Lược đồ này được thiết kế để đảm bảo rằng thời gian giữa các lần quan sát liên tiếp của token tuân theo TTTRT hay nhỏ hơn. Với thời gian này, luôn có một phần khả năng phục vụ cho trước cho tải đồng bộ và bất kỳ khả năng nào vượt quá phần khả năng cho trước là sẵn sàng cho tải bất đồng bộ. Do tính ngẫu nhiên của tải nên thời gian chạy vòng thực sự của token có thể vượt quá TTTRT, như mô tả dưới đây.

Hình 7.21 đưa ra một ví dụ đơn giản của một ring có 4 trạm. Giả sử:

- Tải gồm các frame có chiều dài cố định
- TTTRT = 100 lần thời gian một frame
- SA_i = 20 lần thời gian một frame cho mỗi trạm
- Mỗi trạm luôn được chuẩn bị để gửi trong toàn bộ phân phối đồng bộ của nó cũng như nhiều frame bất đồng bộ có thể.
- Tổng thời gian phân tích overherd trong quá trình của một vòng token hoàn chỉnh bằng 4 lần thời gian một frame (một thời gian frame trên một trạm).



Thời gian đến	TRT	Sync	Async	Thời gian đến	TRT	Sync	Async	Thời gian đến	TRT	Sync	Async	Thời gian đến	TRT	Sync	Async
0	100	0	0	1	100	0	0	2	100	0	0	3	100	0	0
4	96	20	■	121	80*	20	0	142	60*	20	0	163	40*	20	0
184	20*	20	0	205	96*	20	0	226	76*	20	0	247	56*	20	0
268	36*	20	0	289	12	20	■	322	80*	20	0	343	60*	20	0
364	40*	20	0	385	4	20	0	410	92*	20	0	431	72*	20	0
452	52*	20	0	473	16	20	■	510	92*	20	0	531	72*	20	0
552	52*	20	0	573	0	20	0	594	8	20	■	623	80*	20	0
644	60*	20	0	665	8	20	■	694	0	20	0	715	88*	20	0
736	68*	20	0	757	8	20	■	786	8	20	■	815	88*	20	0
836	68*	20	0	857	0	20	0	878	8	20	■	907	96*	20	0
928	76*	20	0	949	8	20	■	978	0	20	0	999	4	20	■
1024	80*	20	0	1045	4	20	■	1070	8	20	■	1099	0	20	0

*LC = 1 khác LC = 0

Hình 7.21 Hoạt động của lược đồ phân phối khả năng phục vụ của FDDI.



Một hàng trong hình 7.21 tương ứng với một vòng của token. Trong mỗi trạm, thời gian đến của token được trình bày, kèm theo là giá trị của TRT tại thời điểm đến kèm theo cả số frame đồng bộ và bắt đồng bộ được truyền trong thời gian trạm nắm giữ token này.

Ví dụ này bắt đầu sau một khoảng thời gian không có frame số liệu nào được truyền. Vì vậy, token quay vòng nhanh nhất (4 lần thời gian frame). Do đó, khi trạm 1 nhận token tại thời điểm 4, đơn vị đo lường thời gian 1 vòng (TRT của nó = 96). Do đó, nó không chỉ có thể gửi 20 frame đồng bộ mà còn gửi được đến 96 frame bắt đồng bộ; ghi nhớ rằng THT bị cấm cho đến thời điểm, sau khi trạm này đã gửi các frame đồng bộ. Trạm 2 trải qua một vòng thời gian là 120 (20 frame + 96 frame + 4 frame) nhưng được quyền truyền 20 frame đồng bộ của nó. Lưu ý rằng nếu mỗi trạm tiếp tục truyền số frame đồng bộ tối đa được phép của chúng thì thời gian vòng lên đến 180 (vào thời điểm 184) nhưng sớm ổn định tại thời điểm xấp xỉ 100. Với tổng thời gian hữu dụng cho đồng bộ là 80 và một khoảng thời gian cần cho phân tích overheard là 4, thì trung bình có 16 thời gian frame dành cho truyền bắt đồng bộ. Chú ý rằng nếu tất cả các trạm luôn có đầy tải bắt đồng bộ phải mang, thì cơ hội truyền các frame bắt đồng bộ được phân phối đều cho chúng.

1.5.6.4. Đặc tả lớp vật lý FDDI

Chuẩn FDDI đặc tả một toto dạng ring hoạt động tốc độ 100Mbps. Hai môi trường được tóm tắt trong bảng 7.4, trong đó môi trường cáp quang dùng mã hoá 4B/5B - NRZI. Hai môi trường cáp xoắn cũng được đặc tả: cáp xoắn đôi UTP loại 5 trở kháng 100Ω và cáp xoắn đôi STP trở kháng 150Ω . Trong cả hai môi trường cáp xoắn, đều dùng mã hoá MLT- 3.

Bảng 7.4 Môi trường lớp vật lý của FDDI

Môi trường truyền	Cáp quang	Cáp xoắn đôi
Tốc độ dữ liệu (Mbps)	100	100
Kỹ thuật báo hiệu	4B/5B/NRZI	MLT - 3
Số lượng repeater tối đa	100	100
Khoảng cách tối đa giữa hai repeater	2 km	100m

1.6. Hệ thống mạng Token bus

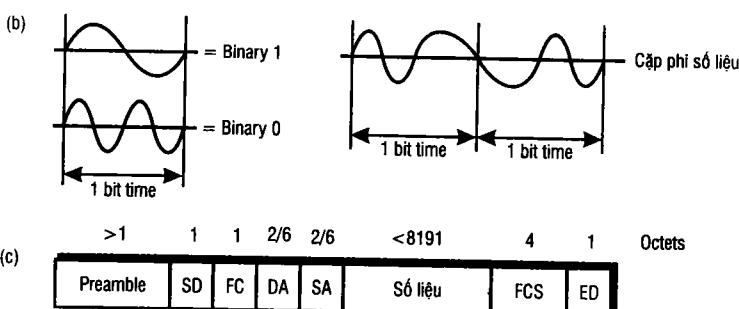
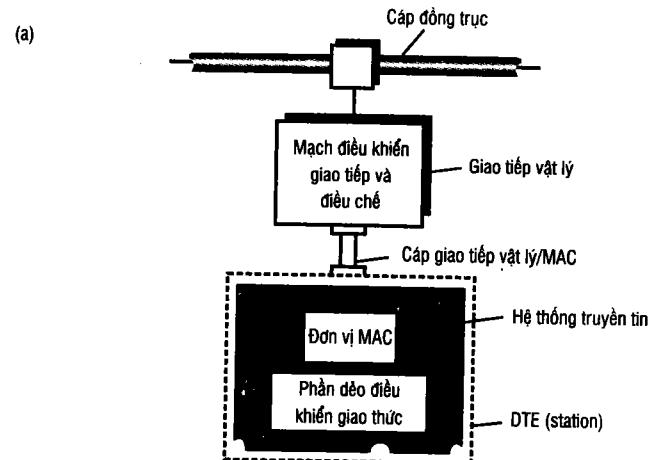
1.6.1. Khái quát

Là lược đồ được đặc tả trong IEEE 802.4, do bản chất xác định tự nhiên của phương pháp điều khiển truy xuất môi trường dùng token và khả năng ưu tiên hoá việc truyền frame, các mạng token bus được dùng trong công nghệ chế tạo (cho chế tạo tự động) và các lĩnh vực liên quan khác. Dưới điều kiện bình thường (không lỗi)

hoạt động của token bus tương tự như mạng token ring. Tuy nhiên, do khác biệt giữa hai phương pháp truy xuất môi trường (quảng bá trong bus và tuần tự trong ring), nên các thủ tục được dùng trong điều khiển quản lý vòng luận lý, như khởi động và đánh mất token có khác nhau. Để tránh trình bày lặp lại, ở đây chỉ tập trung chủ yếu trên các thủ tục quản lý liên quan đến mạng token bus.

Các khía cạnh hoạt động và các thành phần liên quan đến mạng token bus được trình bày trên hình 7.22. Mạng token bus thường dùng cáp đồng trục và hoạt động trong chế độ băng rộng hoặc chế độ băng cơ bản đã được sửa đổi hay gọi là băng sóng mang (carrierband). Mạch điều chế và điều khiển giao tiếp được miêu tả trên hình 7.22(a) chúng thực hiện các chức năng sau:

- Mã hóa số liệu truyền (điều chế)
- Giải mã số liệu thu (giải điều chế)
- Tạo xung đồng hồ



Hình 7.22 Các nguyên lý mạng token bus:

(a) Lược đồ giao tiếp DTE (b) Mã hóa carrierband (c) Dạng frame.



Có giao tiếp chuẩn giữa module giao tiếp vật lý PIM (Physical Interface Module) và DTE nối vào. Trong vài trường hợp, PIM được tích hợp vào card truyền số liệu trong DTE.

Nguyên lý hoạt động của chế độ carrierband được trình bày trên hình 7.22(b). Mặc dù chế độ carrierband giống như baseband (băng cơ bản) ở chỗ mỗi phiên truyền chiếm toàn bộ băng thông cáp, nhưng trong carrierband tất cả số liệu phải được điều chế trước khi truyền, dùng điều chế phase-coherent FSK (điều chế FSK thuần nhất pha). Một bit nhị phân 1 được truyền như một chu kỳ của một tín hiệu hình sin có tần số bằng tốc độ bit, thông thường giữa 1Mbps và 5 Mbps, trong khi một bit nhị phân 0 được truyền như là 2 chu kỳ của một tín hiệu có tần số gấp hai lần tốc độ bit. Cũng lưu ý rằng không có thay đổi pha tại các ranh giới của thời bit do đó dùng thuật ngữ "*phase - coherent*".

Như đã biết một tín hiệu băng cơ bản được cấu thành từ một số vô hạn các thành phần tần số. Ngược lại, một dạng sóng carrierband chỉ có 2 thành phần tần số. Do đó, có thể dùng một bộ lọc tại đầu thu chỉ cho qua hai thành phần tần số này, khoá kín một cách hiệu quả đối với hầu hết các tín hiệu nhiễu của hệ thống. Điều mà không thể thực hiện được đối với băng cơ bản vì bộ lọc cũng sẽ ảnh hưởng xấu đến tín hiệu số liệu.

Dạng frame được dùng cho token bus được trình bày trên hình 7.22 (c). Nó gần như đồng dạng với frame được dùng trong mạng token ring. Tuy nhiên, J và K là hai bit khác số liệu được dùng trong các field SD và ED của token ring để đạt được sự trong suốt dữ liệu, chúng được thay thế trong chế độ carrierband bởi hai ký hiệu khác số liệu đặc biệt.

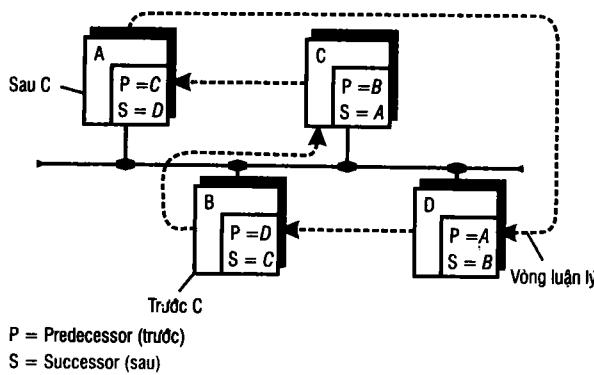
1.6.2. Hoạt động cơ bản:

Hình 7.23 miêu tả hoạt động cơ bản của một mạng token bus. Có một token điều khiển đơn và chỉ có trạm sở hữu token mới có thể truyền frame. Tất cả các DTE có thể khởi động truyền frame được liên kết với nhau thành một *vòng luật lý* (logical ring). Token được dịch chuyển vật lý bằng cách dùng bus như đang chạy trên vòng luật lý. Do đó, khi tiếp nhận token từ trạm trước (predecessor) trên ring, một DTE có thể truyền một số bất kỳ các frame đang đợi nhỏ hơn một số tối đa qui định. Sau đó, chuyển token cho trạm đứng sau (successor) trên ring.

Trước khi mô tả các thủ tục quản lý ring khác nhau, xem xét lại hai thuộc tính cơ bản của các mạng bus. Thứ nhất là với một mạng bus tất cả các DTE được kết nối một cách trực tiếp vào môi trường truyền. Từ đó, khi một DTE phát tán một frame lên môi trường, frame được nhận thấy bởi tất cả các DTE đang mở máy trên mạng. Thứ hai, có một thời lượng tối đa để một DTE đợi một đáp ứng tương ứng với một frame truyền, trước khi nó có thể xem frame truyền bị hỏng hay trạm đích không hoạt động. Thời lượng này được gọi là "*thời khe*" (slot time), chú ý rằng khái niệm này

không giống với phân khe thời gian được dùng trong một CSMA/CD bus. Thời khe có thể được xác định như sau:

Thời khe = 2 x (thời gian trễ trên đường truyền dẫn) + thời gian trễ xử lý



Hình 7.23 Nguyên lý hoạt động của mạng token bus

Trong đó trễ trên đường truyền là thời gian trễ truyền trong trường hợp xấu nhất từ bất kỳ trạm nào đến bất kỳ trạm thu nào trên mạng và trễ xử lý là thời gian lớn nhất để đơn vị MAC trong DTE xử lý một frame nhận và phát ra một đáp ứng thích hợp. Một giá số an toàn được thêm vào và giá trị thời khe được biểu diễn theo thời bit được làm tròn lên thành bội số của octet.

Dưới điều kiện hoạt động bình thường token được chuyển từ một DTE trong vòng luân lý đến DTE khác dùng một frame token ngắn. Do đó mỗi DTE chỉ cần biết địa chỉ của DTE kế (successor) trong vòng luân lý. Nếu một DTE không chấp nhận token, thì DTE gửi dùng một loạt các thủ tục phục hồi để tìm một DTE mới đóng vai là trạm kế sau của nó trên ring (new successor), các thủ tục này sẽ xúc tiến mạnh mẽ hơn nếu DTE không tiếp nhận được một đáp ứng từ DTE láng giềng. Các thủ tục khác có liên quan đến khởi tạo ring và duy trì hoạt động của ring khi các DTE mới gia nhập hay DTE rời khỏi ring. Mặc dù có thể ưu tiên hoá token này, như token ring, nhưng khởi đầu chúng ta chỉ xem xét một ring đơn không ưu tiên. Các loại frame MAC được dùng với các thủ tục quản lý ring khác nhau được trình bày trong bảng 7.5.

Bảng 7.5 Các loại frame và chức năng

Loại frame	Chức năng
Claim token	Được dùng để khởi tạo vòng luân lý

Solicit successor	Được dùng trong thủ tục phục hồi khi một trạm rời khỏi vòng và thủ tục cho phép một trạm gia nhập vòng.
Who-follows-me	Được dùng trong thủ tục cho phép một trạm xác định địa chỉ trạm là successor của nó trên vòng.
Resolve contention	Được dùng trong thủ tục cho phép một trạm mới gia nhập vòng
Set successor	Cho phép một trạm mới gia nhập vòng thông báo với trạm đứng trước nó biết nó đã gia nhập vòng.
Token	Frame token điều khiển

1.6.3. Chuyển token

Khi nhận một frame token hợp lệ, một DTE có thể truyền bất kỳ các frame nào đang đợi. Sau đó nó chuyển token đến DTE đứng sau nó. Sau khi gửi token này DTE lắng nghe bất kỳ động thái nào tiếp theo sau trên bus để chắc chắn rằng DTE đứng sau nó đang hoạt động và đã nhận token này. Nếu nó nghe thấy một frame hợp lệ đang được truyền, nó coi như mọi việc đều tốt đẹp và DTE đứng sau đã nhận token một cách chính xác. Nếu không nghe thấy một frame hợp lệ đang được truyền sau khoảng thời gian của "thời khe", nó phải tiến hành hoạt động sửa chữa.

Nếu sau khi gửi token, DTE nghe một khối nhiều hay frame với FCS sai, nó tiếp tục lắng nghe trong hơn 4 thời khe. Nếu không nghe thấy gì khác hơn, DTE giả sử rằng bản thân token đã bị hỏng trong quá trình truyền và truyền lại token này. Mặt khác, nếu nghe thấy một frame hợp lệ trong thời gian 4 thời khe, DTE này lại xem như DTE đứng sau đã có token. Nếu một khối nhiều thứ hai xảy ra trong thời gian này, DTE coi nó như một frame hợp lệ đang được truyền bởi DTE đứng sau đó và coi như token này đã được chuyển.

Nếu sau khi lặp lại hoạt động chuyển token và các thủ tục giám sát, DTE đứng sau không đáp ứng cho frame token thứ hai, DTE coi như DTE đứng sau nó đã hỏng và tiến hành thiết lập một DTE mới đóng vai trò đứng sau nó. Trước hết, DTE này phát tán một frame Who-follows-me với địa chỉ của successor hiện hành của nó trong field số liệu của frame. Khi tiếp nhận loại frame này, mỗi DTE so sánh địa chỉ trong field số liệu của frame với địa chỉ của DTE đứng trước nó. Địa chỉ DTE có predecesrror giống như successor chứa trong frame này sẽ đáp ứng bằng cách gửi địa chỉ của nó trong một frame set-successor. Vì vậy, DTE giữ token này thiết lập một



successor mới và trong khi làm việc này vô hình chung đã tạo một cầu vượt qua DTE hỏng này.

Nếu DTE truyền không nhận một đáp ứng của frame who-follows-me, nó lặp lại frame này lần thứ hai. Nếu vẫn không có đáp ứng, nó hành động mạnh hơn bằng cách gửi đi một frame solicit-successor chứa địa chỉ của nó trong field DA. Điều này có tác dụng yêu cầu bất kỳ DTE nào trong mạng đều có thể đáp ứng cho nó. Nếu có một số bất kỳ DTE đang ở trạng thái mở máy nghe thấy frame này, chúng đáp ứng và ring luận lý sẽ được thiết lập trở lại bằng cách dùng một thủ tục được gọi là "*cửa sổ đáp ứng*". Một khác, nếu không có đáp ứng thì DTE coi như có một biến cố đã xảy ra, ví dụ tất cả các DTE đã hỏng, môi trường đã bị gián đoạn hay phần thu của DTE này đã bị hỏng (vì thế chẳng nghe thấy gì). Dưới các điều kiện này, DTE này trở nên im lặng nhưng vẫn tiếp tục lắng nghe hoạt động truyền của các DTE khác.

1.6.4. Cửa sổ đáp ứng

Thủ tục này xúc tiến trong những khoảng thời gian ngẫu nhiên để cho phép các DTE mới gia nhập vòng luận lý đang hoạt động. Cửa sổ đáp ứng là khoảng thời gian mà một DTE cần phải đợi một đáp ứng sau khi truyền một frame và do đó giống như thời khe của mạng. Mỗi frame solicit-successor được truyền bởi một DTE chỉ ra một SA và một DA; frame này được đáp ứng bởi một DTE muốn gia nhập vòng và có địa chỉ nằm giữa hai địa chỉ đã được chỉ định này. Mỗi DTE gửi một frame solicit-successor tại những thời điểm ngẫu nhiên bất cứ khi nào nó sở hữu token.

Khi một DTS gửi một frame solicit-successor nó được coi là đã mở một cửa sổ đáp ứng từ thời điểm đó, sau khi truyền frame này, DTE đợi một đáp ứng trong thời gian của cửa sổ đáp ứng. Nếu một DTE có địa chỉ nằm trong dải chỉ định trong frame solicit-successor đang đợi gia nhập mạng, nó sẽ đáp ứng bằng cách gửi một yêu cầu đến trạm truyền frame này để trở thành một successor mới của nó trong vòng luận lý. Nếu trạm truyền nghe đáp ứng này, nó gọi một frame set-successor, cho phép DTE mới gia nhập vòng bằng cách biến nó thành successor mới của nó và đến lượt lại chuyển token cho nó. Rõ ràng, dải địa chỉ được chỉ định có thể chứa nhiều DTE đang đợi để gia nhập vòng, trong trường hợp này chúng cũng đáp ứng các frame nhưng đến sau, các frame đáp ứng sau được gửi bởi mỗi DTE trong số này sẽ bị hủy bỏ. Với trường hợp này, DTE phải cố gắng nhận dạng một trạm đáp ứng bằng cách đưa vào một thủ tục làm việc như sau:

Chắc chắn rằng có nhiều DTE trong dải địa chỉ được chỉ định đang đợi gia nhập vòng, DTE bắt đầu tiến hành lần lượt với chúng bằng cách gửi một frame resolve-contention. Thủ tục này tiếp tục cho đến khi DTE nhận một phuc đáp xác thực. Bất kỳ DTE nào đáp ứng frame solicit-successor ban đầu nhưng liền sau đó không nhận token, mỗi DTE đó chọn một giá trị trong dải 0 đến 3 và lắng nghe bất kỳ hoạt động nào tiếp theo trên bus trong thời gian bằng số thời khe này. Nếu một DTE nghe thấy một hoạt động truyền trong thời gian đã chọn, nó tạm hoãn yêu cầu và đợi cơ hội khác để gia nhập ring, nghĩa là chờ cửa sổ đáp ứng kế tiếp được mở. Mặt khác, nếu

không nghe thấy một hoạt động truyền nào trong khoảng thời gian đã chọn, nó tiếp tục đợi nhận frame resolve-contention.

1.6.5. Khởi động

Thủ tục khởi động được xây dựng dựa trên thủ tục cửa sổ đáp ứng. Mỗi DTE trong mạng giám sát tất cả các hoạt động truyền trên bus và bắt đầu khi nào nó nghe có một hoạt động truyền thì cài lại một bộ định thời về giá trị ban đầu, bộ định thời này được gọi là *inactivity timer* (bộ định thời trong trạng thái không tích cực). Nếu DTE mất token trong hoạt động bình thường, bộ định thời này hết hạn và DTE bước vào cung đoạn khởi động, tại thời điểm này nó gửi một frame claim-token. Như trên, một số DTE có thể cố gắng gửi một frame claim-token một cách đồng thời. Vì vậy, thủ tục sau đây đảm bảo rằng chỉ có một token được tạo ra:

Mỗi trạm có khả năng khởi động gửi một frame claim-token với chiều dài field thông tin bằng một số nguyên lần thời khe. Số nguyên này có thể là 0, 2, 4 hoặc 6, sự chọn lựa căn cứ vào hai bit đầu tiên trong địa chỉ mạng của DTE. Sau khi truyền frame claim-token, DTE này đợi thêm một thời khe trước khi lắng nghe môi trường truyền. Nếu nghe thấy một hoạt động truyền, nó biết rằng DTE khác đã gửi một frame claim-token lâu hơn và do đó DTE từ bỏ việc tranh chấp trở thành chủ đầu tiên của token. Nếu không nghe thấy hoạt động truyền vào, DTE lặp lại quá trình trên dùng hai bit kế tiếp từ field địa chỉ của nó. Lần nữa, không phát hiện hoạt động truyền nào, nó dùng cặp bit kế tiếp và cứ thế cho đến khi nó đã dùng tất cả các bit địa chỉ. Nếu môi trường vẫn hoàn toàn im lặng, DTE này trở thành chủ sở hữu đầu tiên của token. Chủ duy nhất của token này tiếp tục quá trình khởi động bằng cách dùng thủ tục cửa sổ đ답 ứng để cho phép các DTE đang đợi gia nhập vòng.

Mặc dù một DTE có thể tự rời khỏi vòng luân lý tại bất kỳ thời điểm nào bằng một thao tác đơn giản là không đáp ứng khi token được chuyển đến nó, vẫn có một phương pháp rời bỏ vòng là DTE đợi cho đến khi nó nhận token và sau đó gửi một frame set-successor đến trạm đứng trước nó với địa chỉ DTE này sau đó gửi token đến successor của nó như bình thường theo cách hiểu nó không phải là thành phần của ring luân lý.

1.6.6. Hoạt động ưu tiên

Cũng tương tự như mạng token ring, một cơ cấu ưu tiên có thể được thực hiện đối với mạng token bus. Tuy nhiên, phương pháp truy xuất được dùng với token bus khác biệt bởi bốn mức ưu tiên, được gọi là các lớp truy xuất, gọi là 0, 2, 4 và 6, với 6 là mức ưu tiên cao nhất. Như đã đề cập ở phía trước, các mạng token bus được dùng chủ yếu trong các lĩnh vực ứng dụng như điều khiển tự động. Phạm vi sử dụng các lớp truy xuất như sau:

- Lớp 6: Các thông điệp khẩn liên quan đến điều kiện cảnh báo nguy cấp và liên hệ với các chức năng điều khiển.

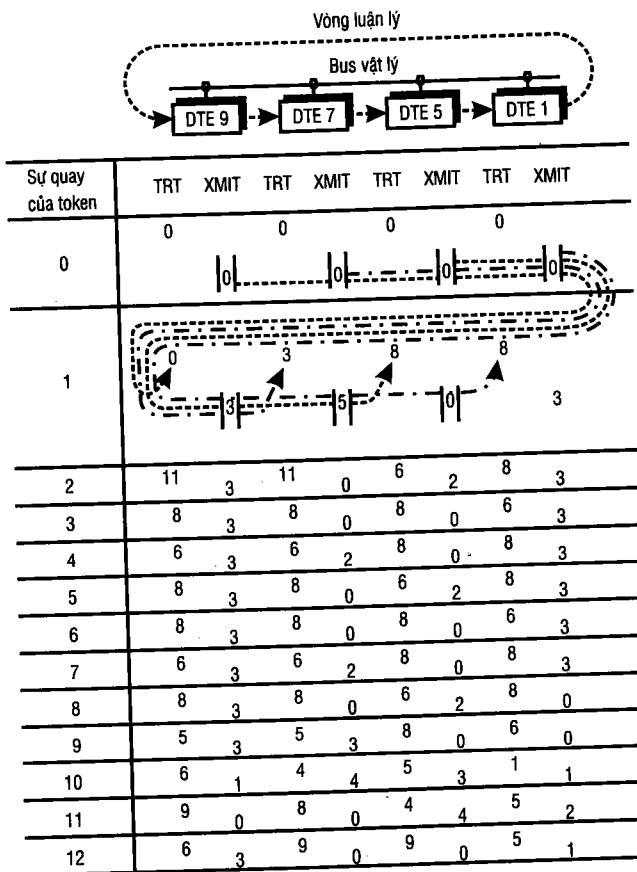
- Lớp 4: Các thông điệp liên quan đến các hoạt động điều khiển bình thường và các chức năng quản lý vòng.
- Lớp 2: Các thông điệp liên quan đến thủ tục tập hợp số liệu thông thường.
- Lớp 0: Các thông điệp liên hệ đến chương trình tải xuống (download) và truyền tập tin tổng quát, là các thông điệp dài có mức ưu tiên thấp.

Mỗi DTE có hai bộ định thời điều khiển việc truyền frame: token - hold - timer (THT) và high - priority token hold timer (HP - THT). Bộ định thời HP - THT điều khiển việc truyền các frame có ưu tiên cao đảm bảo rằng băng thông của ring được chia sẻ cho tất cả các DTE. Do đó, khi một DTE nhận một token, trước hết nó gửi một số frame bất kỳ có ưu tiên cao đang đợi trong thời gian tối đa được xác định trong HP - THT. Giả sử rằng DTE đang dùng cơ cấu ưu tiên này và THT chưa hết hạn, DTE bắt đầu truyền một số frame bất kỳ có ưu tiên thấp đang đợi truyền dùng giải thuật điều khiển sau:

Mỗi DTE trong ring luận lý giữ một timer chỉ thời gian hết hạn vì nhận token. Nó được giữ trong một biến gọi là token rotation time (TRT). Khi DTE kế tiếp nhận token, trước hết truyền giá trị hiện hành TRT sang THT và đặt lại giá trị của TRT thành zero. Sau đó nó truyền một số frame có ưu tiên cao đang đợi, tăng TRT và tính toán hiệu số chênh lệch giữa thời gian cố định được gọi là TTTR (target token rotation time) và THT hiện hành của nó. Nếu giá trị sai lệch này là dương thì DTE có thể gửi một số frame có ưu tiên thấp cho đến khi đạt đến TTTR; nếu giá trị sai lệch này là zero hay âm thì DTE không thể gửi bất kỳ frame có ưu tiên thấp nào. Mỗi DTE dùng cơ cấu ưu tiên có thể truyền một số frame bất kỳ từ lớp truy xuất cao đến thấp cho đến khi đạt đến TTTR.

Để diễn tả hoạt động của cơ cấu này, xem xét ví dụ được trình bày trong hình 7.24. Để đơn giản, ví dụ này giả sử chỉ có hai lớp truy xuất. Cũng giả sử các frame được truyền có chiều dài cố định vì vậy các thời lượng khác được tham chiếu theo tỷ lệ với số frame. Trong ví dụ này giả sử rằng DTE 9 và 1 chỉ gửi các frame có mức ưu tiên cao mỗi khi chúng nhận một token trong khi các DTE 7 và 5 gửi các frame ưu tiên thấp bất cứ khi nào có thể. Lưu ý rằng vòng luận lý được xây dựng sao cho địa chỉ vật lý của các DTE có thứ tự giảm. TTTR của các frame ưu tiên thấp được cố định giá trị bằng 8 frame. Các giá trị dưới cột trái của DTE được gắn nhãn "TRT" là thời gian quay của token được đo bởi DTE trong lần quay trước của token. Giá trị dưới cột phải được gắn nhãn là "XMIT" là số frame được truyền bởi DTE mỗi khi nhận token. Mỗi hàng biểu diễn một vòng quay của token.

Giả sử rằng tất cả các hoạt động truyền bắt đầu diễn ra sau khoảng thời gian ở trạng thái không tích cực (inactive) và sau khi token đang quay nhanh nhất. Do đó TRT trong DTE 9 được ghi là 0 để bắt đầu. Điều này xem thời gian trễ do chuyển token và lan truyền có thể bỏ qua so với thời gian truyền một frame thông thường. Cũng giả sử rằng thời gian giữ token có ưu tiên cao là thời gian mà một DTE có thể gửi ba frame có mức ưu tiên cao sau khi tiếp nhận token.



Hình 7.24 Ví dụ một ring được ưu tiên hóa

Trong lần quay đầu tiên của token, DTE 9 nhận token này và gửi tối đa 3 frame ưu tiên cao trước khi chuyển token. Khi DTE 7 nhận token này từ DTE 9 thì TRT của nó sẽ tăng lên 3 vì có 3 frame đã được truyền từ thời điểm sau cùng nhận token này. Điều này có nghĩa là DTE 7 có thể truyền 5 frame có mức ưu tiên thấp (TTRT trừ TRT $\leftrightarrow 8 - 3 = 5$) trước khi chuyển token. Khi nhận token, TRT được giữ bởi DTE 5 lúc này sẽ là 8, bằng tổng số frame được truyền kể từ lần cuối nhận token. Do đó, nó không thể truyền bất cứ frame ưu tiên thấp nào nữa trong lần chuyển token này. DTE 1 sau đó truyền 3 frame có mức ưu tiên cao không bị ràng buộc bởi sự tính toán trên TRT của nó.

Trong lần quay thứ hai của token, cả DTE9 và DTE1 gửi 3 frame ưu tiên cao không bị ảnh hưởng bởi sự tính toán TRT của chúng, nhưng vào thời gian này DTE7 bị khoá không thể truyền bất kỳ frame ưu tiên mức thấp nào (vì giá trị tính toán TRT

vượt quá 8 khi nhận token) và DTR5 có thể truyền hai frame có mức ưu tiên thấp (TTRT - TRT = 2).

Trong vòng quay thứ 3 của token, DTE9 và DTE1 mỗi DTE lần nữa gởi 3 frame ưu tiên cao nhưng vào thời điểm này cả DTE7 và DTE5 đều bị khoá không thể gởi các frame ưu tiên thấp vì cả hai giá trị tính toán TRT của chúng đã đạt giá trị giới hạn (8).

Trong vòng quay thứ tư của token, trở lại tình huống tương tự vòng quay thứ 2, nhưng lưu ý rằng lúc này các giá trị TRT cho phép DTE7 gởi 2 frame có mức ưu tiên thấp thay vì DTE5, DTE5 không thể gởi bất kỳ frame nào vào thời điểm này. Tương tự, trong vòng quay thứ 5, DTE5 có thể truyền 2 frame mức ưu tiên thấp trong khi DTE7 bị cấm truyền. Chu kỳ này sau đó lặp lại, có thể dễ dàng suy ra rằng qua bất kỳ 3 vòng quay nào, DTE9 và DTE1 dùng 82% băng thông sẵn có và DTE7 và DTE5 chia sẻ 18% còn lại.

Trong vòng quay lần 8, giả sử rằng DTE1 hết frame có ưu tiên cao để truyền và do đó DTE7 và DTE5 có thể truyền nhiều frame có ưu tiên thấp đang đợi. Tương tự, trong lần quay thứ 10, DTE 9 không có frame ưu tiên cao để truyền và cứ thế.

Mặc dù đây chỉ là một ví dụ, nhưng nó giải thích làm thế nào cơ cấu ưu tiên cho phép các frame ưu tiên cao được truyền trong mối quan hệ không bị ràng buộc và truyền các frame ưu tiên thấp theo một phương thức hợp lý bất cứ khi nào dung lượng mạng vẫn còn.

1.7. Hệ thống 100 VG - AnyLAN

Giống như 100 BASE-T, 100VG-AnyLAN hướng đến tốc độ 100Mbps mở rộng từ 10Mbps Ethernet và hỗ trợ các kiểu frame IEEE 802.3. Nó cũng tạo khả năng tương thích với các frame IEEE802.5 token ring. 100 VG - AnyLAN dùng một lược đồ MAC mới gọi là 'ưu tiên theo nhu cầu' (deman priority) để xác định thứ tự mà các node cùng chia sẻ mạng. Vì đặc tả này không dùng CSMA/CD, nên nó được chuẩn hoá dưới một nhóm công tác mới là IEEE 802.12 thay vì được phép thuộc về nhóm công tác IEEE 802.3.

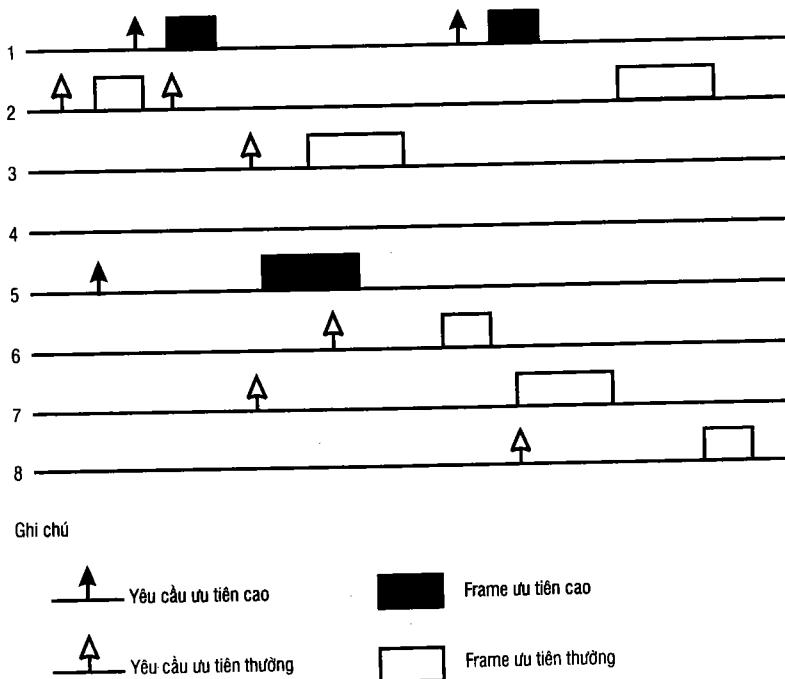
1.7.1. Topo

Topo trong một mạng 100 VG-AnyLAN là hình sao phân cấp. Topo đơn giản nhất gồm có một hub trung tâm và một số thiết bị nối vào. Nhiều tổ chức phức tạp hơn cũng được dùng trong đó có một hub gốc với một hay nhiều hub mức 3 và cứ thế tiến đến một độ sâu tùy ý.

1.7.2. Điều khiển xuất môi trường

Giải thuật MAC cho 802.12 là một lược đồ theo kiểu round-robin với hai mức ưu tiên. Trước hết xem xét mạng chỉ có một hub sau đó sẽ bàn đến trường hợp tổng quát.

Khi một trạm muốn truyền một frame, trước hết nó phát ra một yêu cầu đến hub trung tâm và sau đó đợi cấp phép từ hub để truyền. Mỗi trạm phải gán cho mỗi yêu cầu một mức ưu tiên thường hay cao.



Hình 7.25 Ví dụ tuần tự frame trong một mạng đơn hub.

Hub trung tâm tiếp tục quét tất cả các port của nó để phát hiện một yêu cầu theo kiểu round-robin. Do đó, một hub có n-port trước hết tìm kiếm yêu cầu trên port 1, sau đó đến port 2 và tiếp tục đến port thứ n. Quá trình quét sau đó bắt đầu trở lại tại port 1. Hub này duy trì hai con trỏ: một con trỏ ưu tiên cao và một con trỏ ưu tiên thường. Trong thời gian một chu kỳ hoàn chỉnh, hub phục vụ mỗi yêu cầu ưu tiên cao theo thứ tự mà các yêu cầu được phát hiện. Nếu tại bất kỳ thời điểm nào không còn các yêu cầu mức cao, hub sẽ phục vụ bất kỳ các yêu cầu có mức ưu tiên thường mà nó bắt gặp.

Hình 7.25 đưa ra một ví dụ. Tuần tự các sự kiện ở đó như sau:

- Hub trỏ cả hai con trỏ đến port 1 và bắt đầu quét. Yêu cầu bắt gặp đầu tiên là một yêu cầu mức ưu tiên thấp từ port 2. Hub phục vụ yêu cầu này và cập nhật con trỏ mức ưu tiên thấp chỉ đến port 3.

- ii. Port 2 truyền một frame ưu tiên thấp. Hub nhận frame này và truyền lại nó. Trong khoảng thời gian này, hai yêu cầu mức ưu tiên cao được tạo ra.
- iii. Một khi frame từ port 2 được truyền, hub bắt đầu phục vụ các yêu cầu ưu tiên cao theo thứ tự round-robin, bắt đầu với port 1 và sau là port 5. Con trỏ ưu tiên cao được trỏ đến port 6.
- iv. Sau khi frame ưu tiên mức cao từ port 5 hoàn tất, không có yêu cầu mức ưu tiên cao vượt trội và hub chuyển đến các yêu cầu mức ưu tiên thường. Có 4 yêu cầu đến từ khi frame ưu tiên thấp sau cùng đã được truyền: bốn yêu cầu này đến từ port 2, 7, 3 và 6. Vì con trỏ ưu tiên thường được chỉ đến port 3, nên các yêu cầu này sẽ được phục vụ theo thứ tự 3, 6, 7 và 2 nếu không có yêu cầu nào khác chèn ngang.
- v. Các frame từ port 3, 6 và 7 được truyền lần lượt. Trong khi truyền frame của 7, một yêu cầu ưu tiên cao đến từ port 8. Hub chỉ con trỏ ưu tiên thường đến port 8.
- vi. Vì các yêu cầu ưu tiên cao giành quyền phục vụ trước nên yêu cầu từ port 1 được chuyển đến lần truy xuất kế.
- vii. Sau khi frame từ port 1 đã được truyền hub còn lại hai yêu cầu có mức ưu tiên thường trội hơn. Yêu cầu từ port 2 đợi lâu nhất, tuy nhiên, port 8 lại là kế tiếp theo thứ tự thỏa mãn round - robin và do đó các yêu cầu của nó được phục vụ, tiếp theo mới là của port 2.

Mạng phân cấp

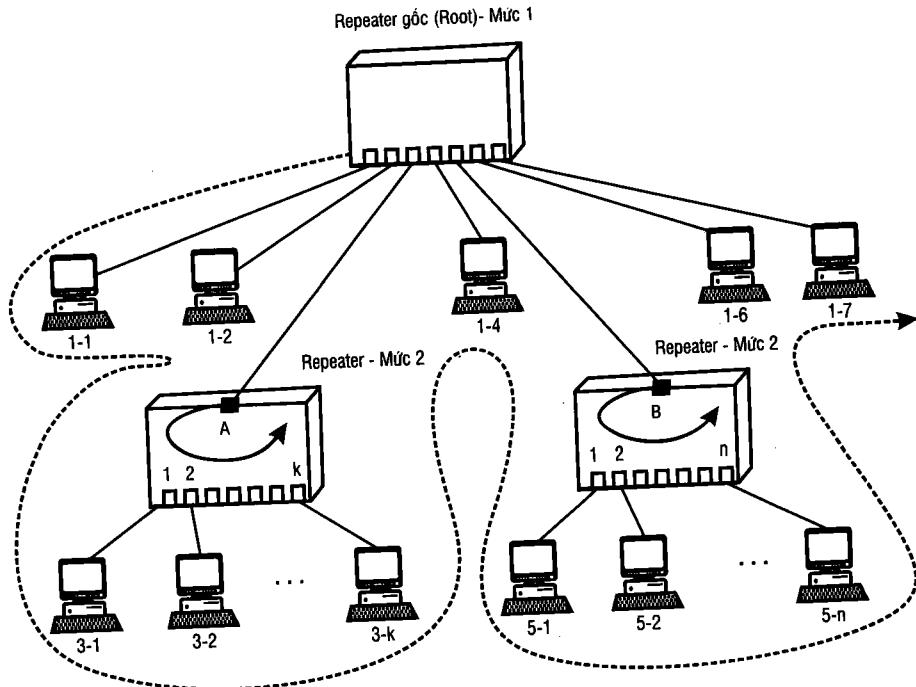
Trong một mạng phân cấp, tất cả các port của hệ thống đều cuối trên tất cả các hub được coi như một tập các port nhằm thỏa mãn các mục đích của giải thuật round-robin. Các hub được cấu hình để phối hợp trong quá trình quét các port theo thứ tự thích hợp. Ở phương pháp khác, tập các hub được xem như một hub luận lý.

Hình 7.26 chỉ ra thứ tự port trong mạng phân cấp. Thứ tự này được tạo ra bằng cách duyệt cây tương trưng cho mạng này trong đó các nhánh dưới mỗi node trong cây được xếp theo thứ tự tăng từ trái sang phải. Với tiêu chuẩn này, thứ tự port được tạo ra bởi duyệt cây là những gì được xem như đi theo một trật tự cho trước (preorden traversal), thứ tự được định nghĩa một cách đệ quy như sau:

- i. Đến gốc
- ii. Đi đến các cây con từ trái qua phải

Phương pháp này cũng được gọi là depth - first search (tìm kiếm theo chiều sâu). Bây giờ sẽ tiến hành xem xét cơ chế truy xuất môi trường và hoạt động truyền frame trên một mạng phân cấp. Trước hết xem xét vai trò của hub gốc. Hub này thực hiện các giải thuật round-robin ưu tiên cao và ưu tiên thường cho tất cả các thiết bị kết nối trực tiếp. Do đó, nếu có một hay nhiều yêu cầu có mức ưu tiên cao còn tồn đọng, hub này phục vụ các yêu cầu này theo kiểu round-robin. Nếu không còn yêu cầu mức ưu tiên cao thì hub phục vụ bất kỳ yêu cầu có mức ưu tiên thường nào theo kiểu round-robin. Khi một yêu cầu được phục vụ bởi hub gốc, hệ thống phát ra yêu

cầu này có thể truyền một frame ngay tức thì. Khi một yêu cầu được phục vụ bởi hub gốc mà hệ thống yêu cầu này nối trực tiếp vào hub mức 2, thì điều khiển chuyển đến hub mức 2, hub mức 2 này sau đó tiến hành thực thi các giải thuật round-robin của nó.



Hình 7.26 Định thứ tự port trong mạng IEEE 802.12 có 2 mức

Bất kỳ một hệ thống đầu cuối nào sẵn sàng truyền đều gửi một tín hiệu yêu cầu đến hub mà nó nối vào. Nếu hệ thống đầu cuối được nối trực tiếp vào hub gốc, thì yêu cầu được truyền trực tiếp đến hub gốc. Nếu hệ thống đầu cuối nối đến hub thấp hơn, thì yêu cầu được truyền trực tiếp đến hub này. Nếu hiện thời hub này không có điều khiển của giải thuật round-robin, thì nó sẽ chuyển yêu cầu đến hub mức cao hơn kế tiếp. Thực tế, tất cả các yêu cầu mà không được phục vụ tại một mức thấp sẽ được chuyển lên hub gốc.

Lược đồ được mô tả trên đây áp đặt một nguyên lý của round-robin vào tất cả các trạm nối vào mạng, nhưng cần hai cải tiến quan trọng. Trước hết cần một cơ cấu ngăn chặn trước. Cơ cấu này sẽ được giải thích tốt nhất qua ví dụ. Xem xét tuần tự các sự kiện sau đây:

- i. Giả sử hub gốc (R) trong hình 7.26 đang trong trạng thái điều khiển và không còn yêu cầu mức ưu tiên cao nào ở bất cứ nơi nào trong mạng. Tuy nhiên, các trạm 5-1, 5-2 và 5-3 đều phát ra yêu cầu mức ưu tiên thường khiếu cho hub B phát ra một yêu cầu mức ưu tiên thường đến R.
- ii. R sẽ phục vụ yêu cầu này, chuyển điều khiển cho B
- iii. Sau đó B tiến hành tiếp đón yêu cầu vượt trội tại thời điểm này
- iv. Trong khi B đang tiếp đón yêu cầu ưu tiên thường đầu tiên của nó, trạm 1-6 phát ra một yêu cầu ưu tiên cao.
- v. Trong khi đáp ứng yêu cầu từ 1-6, R phát ra một tín hiệu chặn trước đến B bảo với B rằng từ bỏ điều khiển sau khi hoàn thành hoạt động truyền hiện hành.
- vi. R phục vụ yêu cầu của 1-6 và sau đó tiếp tục giải thuật round-robin.

Cải tiến thứ hai là một cơ cấu ngăn chặn tình trạng một hub không phải gốc nhưng giữ điều khiển trong thời gian vô định. Để thấy rõ vấn đề này giả sử rằng B trong hình 7.26 có một yêu cầu mức cao còn lại từ 5-1. Sau khi nhận điều khiển từ R, B phục vụ yêu cầu 5-1. Trong khi đó, các trạm thứ cấp khác của B phát ra các yêu cầu ưu tiên cao. B có thể tiếp tục trong kiểu round-robin để tiếp đón tất cả các yêu cầu này. Nếu các yêu cầu thêm vào đến từ các trạm thứ cấp của B trong khi diễn ra các hoạt động truyền này, thì B sẽ có thể tiếp tục phục vụ các yêu cầu này một cách không xác định ngay cả khi còn tồn đọng các yêu cầu mức ưu tiên cao ở vị trí khác trong mạng. Để ngăn chặn trường hợp này, một hub thứ cấp chỉ có thể giữ điều khiển trong một chu kỳ tín hiệu round-robin xuyên qua tất cả các port.

Giải thuật IEEE 802.12 hoàn toàn hiệu quả. Khi nhiều trạm cung cấp lượng tải nặng, giao thức này phản ứng giống như một giao thức token ring với truy xuất mạng xoay quanh trong số tất cả các trạm yêu cầu mức ưu tiên cao, kể sau là các trạm yêu cầu mức ưu tiên thấp khi không có các yêu cầu mức ưu tiên cao vượt trội. Đối với tải thấp, giao thức hoạt động theo kiểu giống như CSMA/CD. Một trạm có yêu cầu là duy nhất vào một thời điểm nào đó sẽ truy xuất vào môi trường hầu như ngay tức thì.

1.7.3. Đặc tả lớp vật lý của 100VG -AnyLAN

Phiên bản hiện hành của IEEE 802.12 chỉ định dùng cáp xoắn UTP 4 đôi, loại 3, 4 hay 5. Các phiên bản tương lai cũng sẽ hỗ trợ cáp UTP, loại 5 có 2 đôi, cáp STP và cáp quang. Trong tất cả các trường hợp tốc độ số liệu đều là 100 Mbps.

Mã hóa tín hiệu

Một đối tượng mục tiêu quan trọng của 100VG-AnyLAN là có thể đạt tốc độ 100Mbps qua cự ly ngắn dùng cáp loại 3. Ưu điểm của nó là trong nhiều tòa nhà hiện hữu thường có sẵn một dạng cáp tương đương. Do đó, nếu loại cáp này có thể được tận dụng thì giá thành lắp đặt sẽ giảm xuống tối thiểu.



Với kỹ thuật hiện nay, tốc độ dữ liệu 100 Mbps qua một hay 2 đôi cáp loại 3 là không thực tế. Để đạt được mục tiêu này 100VG-AnyLAN chỉ định một lược đồ mã hoá liên quan đến việc dùng 4 đôi để truyền số liệu trong chế độ bán song công. Do đó, để đạt đến tốc độ 100Mbps, chỉ cần tốc độ 25 Mbps cho mỗi kênh. Một lược đồ mã hoá gọi là 5B6B được dùng.

Số liệu từ lớp MAC có thể biểu diễn như một luồng bit. Các bit từ luồng này được tách thành nhóm 5 tại một thời điểm để hình thành một luồng gồm những nhóm 5 sau đó được chuyển xuống 4 kênh truyền theo dạng round-robin. Kế đến, mỗi nhóm 5 chuyển qua một giải thuật xáo trộn đơn giản (scrambling) để tăng số lần chuyển đổi giữa 0 và 1 nhằm cải thiện phổ tín hiệu. Đến đây có thể truyền số liệu một cách đơn giản dùng NRZ. Tuy nhiên, ngay cả với giải thuật xáo trộn vẫn còn các bước tiếp theo của mã hoá 5B6B được dùng để đảm bảo đồng bộ và duy trì việc giảm thiểu thành phần mờ chiêu (DC).

Vì frame MAC được phân chia vào 4 kênh nên đầu và cuối của frame MAC phải được phân định ranh giới trên mỗi kênh và đây là chức năng của các *bộ kiến tạo ranh giới* (delimiter generator). Cuối cùng, mỗi kênh sẽ dùng mã hóa NRZ để truyền.

1.8. ATM LAN

Trong tài liệu làm tiền đề cho các mạng được hợp tác soạn thảo bởi Apple Bellcore, Sun và Xerox phân chia các mạng LAN thành 3 thế hệ:

- **Thế hệ thứ nhất:** Tiêu biểu là CSMA/CD LAN và Token ring LAN. Thế hệ thứ nhất cung cấp kết nối terminal-to-host và hỗ trợ các kiến trúc client/server với tốc độ vừa phải.
- **Thế hệ thứ hai:** Tiêu biểu là FDDI, thế hệ thứ hai đáp ứng nhu cầu cho các LAN đường trục (back bone) và hỗ trợ cho các máy trạm có tốc độ cao.
- **Thế hệ thứ ba:** Tiêu biểu là các ATM LAN, thế hệ thứ ba được thiết kế để cung cấp khả năng phối hợp thông lượng và đảm bảo chuyển tải theo thời gian thực, đáp ứng nhu cầu cho các ứng dụng đa phương tiện.

Các yêu cầu thông thường đối với một LAN thế hệ thứ ba gồm có:

- i. Hỗ trợ nhiều lớp dịch vụ tin cậy, ví dụ dịch vụ video trực tuyến có thể yêu cầu một cầu nối tin cậy có tốc độ 2Mbps để chất lượng dịch vụ có thể chấp nhận được, trong khi một chương trình chuyển tập tin chỉ cần dùng một lớp dịch vụ cẩn bản.

- ii. Cung cấp thông lượng dài rộng, có khả năng mở rộng dung lượng trên từng host (để cho phép các ứng dụng cần lượng dữ liệu xuất/nhập lớn trên một host) và cả trên dung lượng phổi hợp (để cho phép cài đặt mở rộng từ vài host đến vài trăm host tốc độ cao)
- iii. Làm phương tiện liên kết mạng giữa kỹ thuật LAN và WAN.

ATM rất lý tưởng cho việc đáp ứng các yêu cầu ở trên nhờ vào các đường dẫn ảo và các kênh ảo, rất dễ tích hợp các lớp đa dịch vụ. Theo kiểu kết nối cố định hay chuyển mạch, ATM rất dễ mở rộng bằng cách thêm nhiều node chuyển mạch và dùng tốc độ cao hơn (hay thấp hơn) cho các thiết bị kết nối vào. Sau cùng, với việc tăng cường sử dụng phương pháp vận chuyển bằng tế bào (cell) trong xây dựng mạng diện rộng, thì việc dùng ATM trong một mạng đầu cuối cho phép xóa dần ranh giới giữa LAN và WAN.

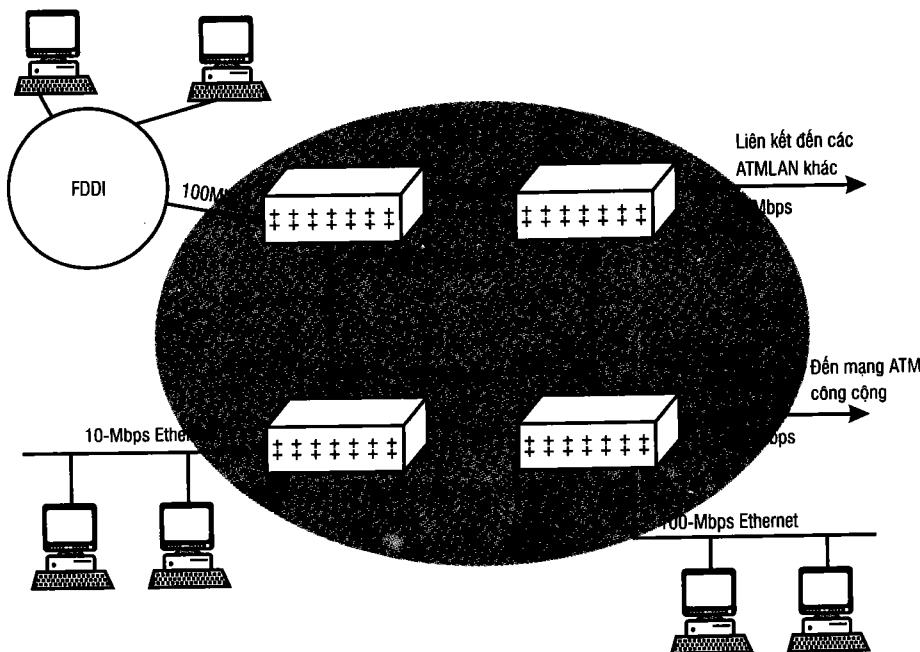
Thuật ngữ ATM LAN được dùng bởi các nhà chế tạo và các nhà nghiên cứu cho các dạng cấu hình khác nhau. Tối thiểu, một ATM LAN cũng bao hàm việc dùng ATM như một giao thức chuyển số liệu ở đâu đó trong vùng cục bộ. Các loại ATM LAN gồm có:

- **Gateway to ATM LAN** : là một chuyển mạch ATM đóng vai trò như một router và bộ tập trung tải để liên kết một mạng đầu cuối phức tạp vào ATM WAN
- **Backbone ATM Switch**: là một chuyển mạch ATM đơn hay một chuyển mạch ATM cục bộ liên kết các LAN khác nhau.
- **Workgroup ATM**: là các máy trạm đa phương tiện chất lượng cao và các hệ thống đầu cuối khác được kết nối trực tiếp vào một chuyển mạch ATM.

Trên đây là 3 cấu hình thuần nhất. Trong thực tế một hỗn hợp của hai hay cả 3 loại cũng có thể được dùng để tạo ra một ATM LAN.

Hình 7.27 trình bày ví dụ một backbone ATM LAN bao gồm các liên kết đến thế giới bên ngoài. Trong ví dụ này, mạng cục bộ ATM gồm có bốn chuyển mạch được nối với nhau bằng các liên kết điểm nối điểm tốc độ cao hoạt động với các tốc độ chuyển 155 Mbps hay 622 Mbps. Về phía người dùng, có 3 LAN khác nhau, mỗi LAN có một liên kết nối trực tiếp đến một trong các chuyển mạch ATM. Tốc độ số liệu từ một chuyển mạch ATM đến một LAN kết nối vào nó là theo tốc độ của LAN này. Ví dụ, kết nối đến mạng FDDI là 100 Mbps, do đó chuyển mạch phải bao gồm vài bộ đệm và có khả năng chuyển đổi tốc độ để ánh xạ tốc độ số liệu từ LAN được

nối vào sang tốc độ số liệu của ATM. Chuyển mạch ATM cũng phải thực hiện vài dạng chuyển giao thức từ giao thức MAC được dùng trên LAN đến luồng cell ATM được dùng trên mạng ATM. Một tiếp cận đơn giản là mỗi chuyển mạch ATM đóng vai trò một bridge (cầu) hay router cho LAN nối vào nó.



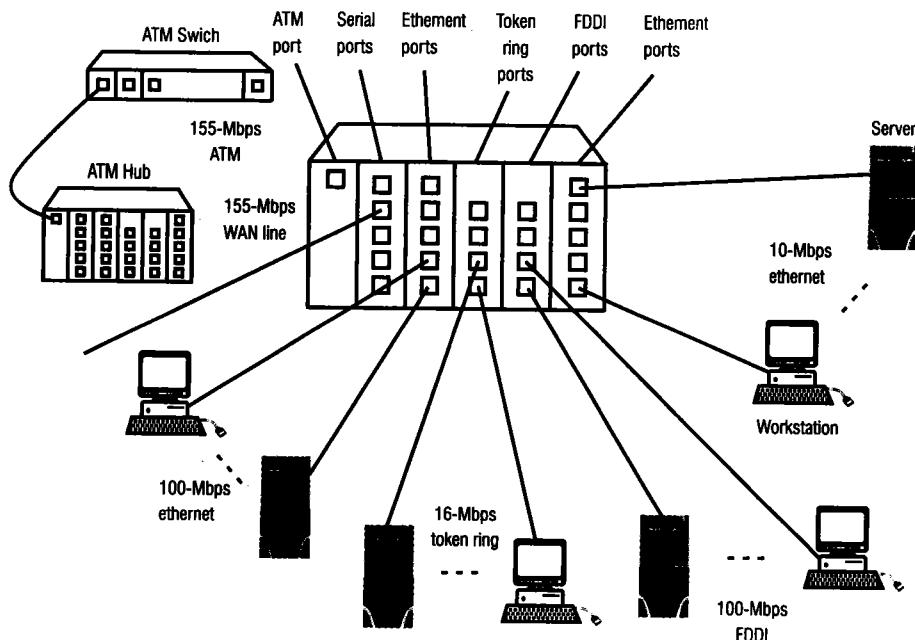
Hình 7.27 Ví dụ một cấu hình ATM LAN.

Một cấu hình ATM LAN như trình bày trên hình 7.27 cung cấp một phương pháp mềm dẻo để chèn một backbone tốc độ cao và một môi trường cục bộ. Khi nhu cầu trên các điểm tăng, vấn đề chỉ đơn giản là tăng khả năng (dung lượng) của backbone bằng cách thêm nhiều chuyển mạch, tăng thông lượng của mỗi chuyển mạch và tăng tốc độ số liệu của các trung kế giữa các chuyển mạch. Với chiến lược này, tải trên các LAN riêng biệt có thể gia tăng, và có thể phát triển số lượng các LAN.

Tuy nhiên, backbone ATM LAN đơn giản này không hướng đến tất cả nhu cầu trong truyền tin cục bộ. Đặc biệt, trong cấu hình backbone đơn giản này, các hệ thống đầu cuối (máy trạm, máy chủ,...) duy trì kết nối vào các LAN vốn có một số giới hạn về tốc độ quy định bởi môi trường chia sẻ của LAN này.

Một tiếp cận ưu việt và mạnh mẽ hơn là dùng kỹ thuật ATM trong một hub. Hình 7.28 đề nghị một số khả năng có thể có từ tiếp cận này. Mỗi ATM hub gồm có

một số các port hoạt động với các tốc độ khác nhau và dùng các giao thức khác nhau. Thông thường, một hub như vậy gồm có một số các rack-mounted module, mỗi module chứa các port có một tốc độ và giao thức cho trước.



Hình 7.28 Cấu hình ATM LAN hub

Khác nhau cơ bản giữa ATM hub trình bày trên hình 7.28 và ATM node mô tả trên hình 7.27 là phương pháp kiểm soát, điều khiển các hệ thống đầu cuối riêng rẽ. Ghi chú rằng trong ATM hub, mỗi hệ thống đầu cuối có một liên kết cố định thường xuyên đến hub. Mỗi hệ thống đầu cuối bao gồm phần cứng và phần mềm truyền số liệu để giao tiếp với một loại LAN đặc biệt, nhưng trong môi trường hợp LAN chỉ chứa hai loại thiết bị: hệ thống đầu cuối và hub. Ví dụ, mỗi thiết bị kết nối vào một port Ethernet 10 Mbps hoạt động với giao thức CSMA/CD tốc độ 10 Mbps. Tuy nhiên, vì mỗi hệ thống đầu cuối sở hữu một đường nối thường xuyên nên kết quả là mỗi hệ thống có một Ethernet 10 Mbps thường trực. Do đó, mỗi hệ thống đầu cuối có thể hoạt động với tốc độ gần đạt đến tốc độ tối đa 10 Mbps.

Việc dùng cấu hình như vậy có ưu điểm là các cài đặt LAN và phần cứng LAN đã có trước đây - được gọi là các LAN kế thừa - có thể được sử dụng trở lại trong bối cảnh kỹ thuật ATM xuất hiện. Nhược điểm chính của nó là khi chấp nhận một môi trường hỗn hợp giao thức như vậy thì cần phải hiện thực một số các chuyển đổi giao thức. Một tiếp cận khác đơn giản hơn, nhưng tiếp cận này yêu cầu tất cả các hệ thống



đầu cuối được trang bị khả năng ATM và do đó mạng trở thành một ATM LAN thuần nhất.

Có một vấn đề mà trên đây chưa được đề cập đến, đó là phải thực hiện *khả năng liên kết hoạt động* giữa các hệ thống đầu cuối trên các mạng LAN liên kết khác nhau. Mỗi hệ thống đầu cuối nối trực tiếp vào một trong các LAN kế thừa thực thi lớp MAC phù hợp với loại LAN này. Các hệ thống đầu cuối nối trực tiếp vào một mạng ATM thực thi các giao thức LLC và ATM. Kết quả có 3 khía cạnh tương thích cần xem xét:

- i. Tương tác giữa một hệ thống đầu cuối trên một mạng ATM và một hệ thống đầu cuối trên một LAN kế thừa.
- ii. Tương tác giữa một hệ thống đầu cuối trên một LAN kế thừa và một hệ thống đầu cuối trên một LAN kế thừa khác cùng loại (ví dụ hai mạng IEEE 802.3)
- iii. Tương tác giữa một hệ thống đầu cuối trên một LAN kế thừa và một hệ thống đầu cuối trên LAN kế thừa khác khác loại (ví dụ một mạng IEEE 802.3 và một mạng 802.5).

1.9. Fibre Channel

1.9.1. Khái quát

Khi tốc độ và dung lượng bộ nhớ của các máy tính cá nhân, các máy trạm, các server tăng lên, và khi các ứng dụng trở nên phức tạp hơn yêu cầu độ tin cậy cao hơn với đồ họa và video, thì yêu cầu về tốc độ phân phối số liệu của các bộ xử lý cũng tăng theo. Các yêu cầu này ảnh hưởng đến hai phương pháp thông tin số liệu với bộ xử lý: kênh xuất nhập I/O và truyền số liệu trên mạng.

Một kênh xuất nhập I/O là một liên kết thông tin trực tiếp điểm nối điểm hay đa điểm, chủ yếu dựa vào phần cứng và được thiết kế cho tốc độ cao qua các cự ly rất ngắn. Kênh I/O truyền số liệu giữa một bộ đệm tại thiết bị nguồn và một bộ đệm tại thiết bị đích, chỉ di chuyển các nội dung của user từ một thiết bị này đến thiết bị khác, không quan tâm đến khuôn dạng và ý nghĩa của số liệu. Tính luận lý liên hệ với kênh thường cung cấp khả năng điều khiển tối ưu để quản lý truyền, phối hợp với phần cứng phát hiện lỗi. Các kênh I/O thường quản lý các hoạt động truyền giữa các bộ xử lý và các thiết bị ngoại vi, như đĩa, thiết bị đồ họa, CD-ROM và các thiết bị I/O video.

Mạng là một tập các điểm truy xuất liên kết với một kiến trúc giao thức mềm cho phép hiện thực công việc truyền số liệu. Mạng luôn cho phép nhiều dạng truyền số liệu khác nhau, dùng phần mềm để hiện thực các giao thức mạng và cung cấp phương tiện điều khiển luồng (flow control), phát hiện lỗi và khắc phục lỗi. Như đã thảo luận ở trên, mạng thường quản lý việc trao đổi số liệu giữa các hệ thống đầu cuối trong cự ly ngắn, trung bình hay diện rộng.

Fibre Channel được thiết kế để phối hợp các đặc tính tốt nhất trong cả hai kỹ thuật thông tin số liệu nói trên. Đó là sự đơn giản và tốc độ cao của kênh I/O cùng với đặc tính linh hoạt và liên kết của truyền số liệu trên mạng nhờ vào các giao thức. Sự phối hợp này cho phép các nhà thiết kế tổ hợp các kết nối ngoại vi truyền thống, liên kết host với host, nối vòng các nhóm vi xử lý, và các ứng dụng đa phương tiện trong một giao tiếp đa giao thức. Các dịch vụ theo hướng kênh được kết hợp trong kiến trúc giao thức Fibre Channel gồm có :

- Kiểu dữ liệu phù hợp cho công việc định tuyến tải vào các bộ đệm giao tiếp đặc biệt.
- Mức liên kết hình thành nên mối liên hệ với các hoạt động xuất nhập (I/O) riêng rẽ.
- Các đặc tả giao tiếp của giao thức cho phép hỗ trợ các kiến trúc kênh I/O đã có trước đây, như là SCSI (Small Computer System Interface).

Các loại dịch vụ theo hướng mạng được kết hợp vào trong kiến trúc giao thức Fibre Channel gồm có :

- Ghép kênh đầy đủ các tải từ nhiều nguồn phát khác nhau
- Kết nối ngang hàng (peer-to-peer) giữa bất kỳ một cặp port nào trên một mạng Fibre Channel.
- Các khả năng liên kết mạng đối với các kỹ thuật kết nối khác nhau.

Tùy thuộc vào nhu cầu của ứng dụng, tiếp cận kênh hay tiếp cận mạng có thể được dùng cho bất kỳ công tác truyền số liệu nào. Fiber Channel Association là một tập đoàn công nghệ cải tiến Fibre Channel, đã liệt kê các yêu cầu thiết yếu mà Fibre Channel có khuynh hướng thỏa mãn:

- Các liên kết song công hoàn toàn với hai sợi cáp trên một liên kết.
- Hoạt động với tốc độ từ 100 Mbps đến 800 Mbps trên một liên kết đơn (200 Mbps đến 1600 Mbps trên một liên kết 2 hướng).
- Cự ly đạt đến 10km.
- Các bộ nối nhỏ.
- Các ứng dụng dung lượng cao không nhạy cảm đối với cự ly.
- Các kênh nối có qui mô lớn hơn các kênh đa điểm đang có sẵn.
- Mức độ phổ dụng lớn
- Hỗ trợ đa mức phẩm chất theo giá cả, từ hệ thống rất nhỏ đến các máy tính lớn.
- Khả năng thực thi nhiều tập lệnh của nhiều giao tiếp có sẵn trong kênh hiện hữu và các giao thức mạng.

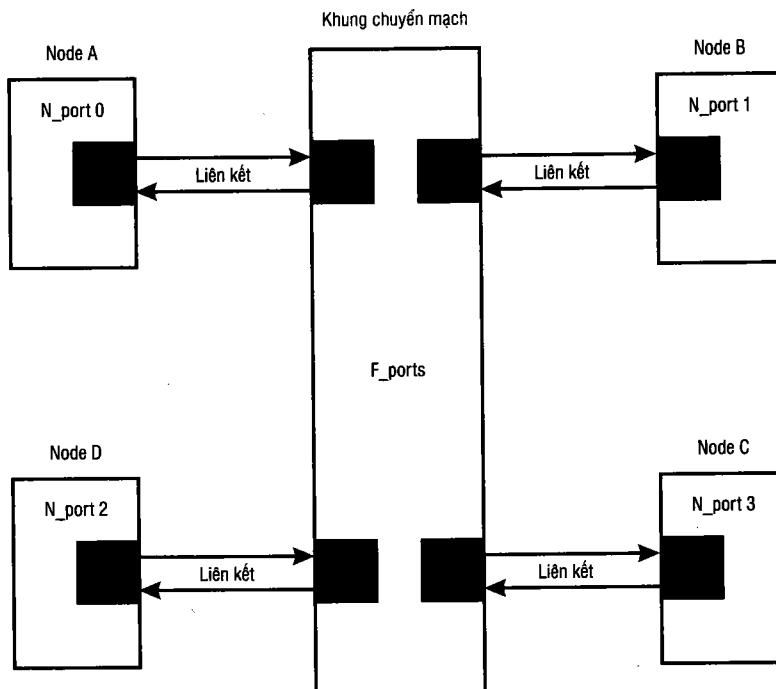


Giải pháp này là phát triển một cơ cấu vận chuyển chung đơn giản dựa vào các liên kết điểm nối điểm và một mạng chuyển mạch. Dưới cơ sở hạ tầng này có hỗ trợ một lược đồ mã hóa và đóng bộ đơn giản, từ đó chúng lại hỗ trợ một số các kênh và giao thức mạng khác nhau.

1.9.2. Các phần tử của fibre channel

Các phần tử chủ yếu của mạng Fibre Channel là các hệ thống đầu cuối, được gọi là các **node** và bản thân mạng này, nó bao gồm một hay nhiều phần tử chuyển mạch. Tập hợp các phần tử chuyển mạch được gọi là một **fabric**. Các phần tử này được nối với nhau bởi các liên kết điểm nối điểm giữa các port trên các node riêng rẽ và các chuyển mạch. Công tác truyền thông tin bao gồm truyền dẫn các frame xuyên qua các liên kết điểm nối điểm.

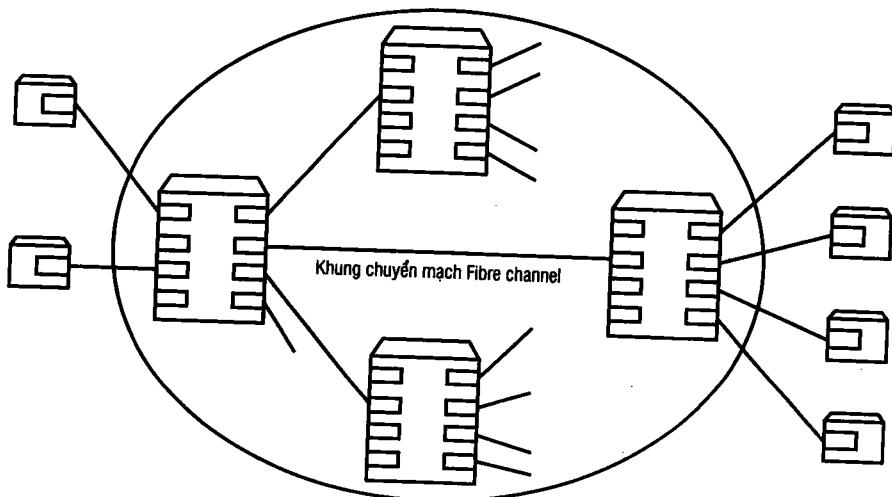
Hình 7.29 mô tả các phần tử cơ bản này. Mỗi node bao gồm 3 hay nhiều port, port này được gọi là **N-port**, để liên kết với các phần tử khác. Tương tự mỗi phần tử fabric bao gồm một hay nhiều port, được gọi là **F-port**. Kết nối liên tiếp được thực hiện bởi các liên kết hai chiều giữa các port. Bất kỳ node nào đều có thể thông tin với bất kỳ một node khác được kết nối vào cùng một fabric, nhờ vào các dịch vụ của fabric. Tất cả các hoạt động định tuyến cho các frame giữa các N-port đều được thực hiện bởi fabric này. Các frame có thể được đệm trong fabric, điều này cho phép các node khác nhau của fabric kết nối vào nó với các cấp tốc độ khác nhau.



Hình 7.29 Các loại port của Fibre Channel.

Một fabric có thể được xây dựng thành một fabric đơn, như mô tả trên hình 7.29 hay như một mạng của các phần tử fabric, như mô tả trên hình 7.30. Trong cả hai trường hợp này, fabric chịu trách nhiệm đệm và định tuyến các frame giữa các node nguồn và đích.

Mạng Fibre Channel khác nhau hoàn toàn với các mạng LAN khác. Mạng này rất giống với một mạng chuyển mạch kênh hay mạng chuyển mạch gói, ngược lại với các mạng LAN chia sẻ môi trường thông thường. Do đó, mạng Fibre Chanel không cần để ý đến các vấn đề liên quan điều khiển truy xuất môi trường (MAC). Vì chỉ dựa vào một mạng chuyển mạch, nên mạng Fibre Channel dễ dàng minh họa trong các thuật ngữ N-port, tốc độ số liệu, và cự ly bao phủ. Tiếp cận này tạo ra mức độ linh hoạt cao, Fibre Channel có thể hội nhập nhanh chóng với môi trường truyền dẫn mới và các cấp tốc độ khác nhau bằng cách thêm các chuyển mạch mới và các F-port vào fabric có sẵn. Do đó, các khoản đầu tư trước không bị mất trong khi nâng cấp mạng với kỹ thuật và thiết bị mới. Hơn nữa, như chúng ta sẽ thấy, kiến trúc giao thức phân lớp hội nhập các giao tiếp I/O đã có với các giao thức lập mạng, bảo toàn các khoản đầu tư trước đây.



Hình 7.30 Mạng Fibre Channel.

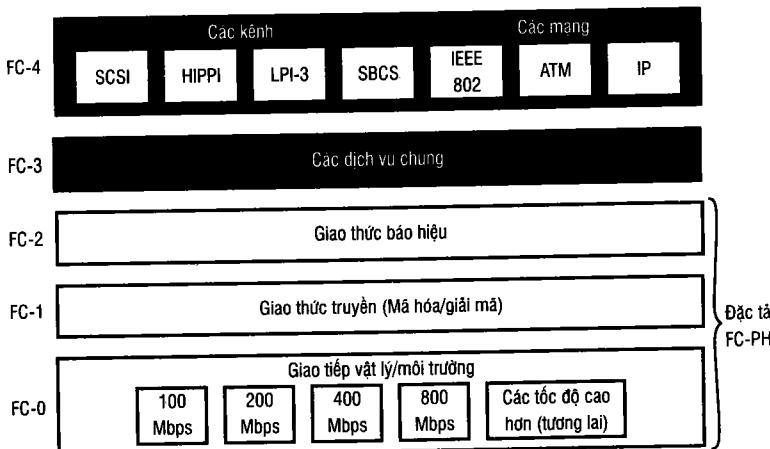
1.9.3. Kiến trúc giao thức Fibre Channel

Chuẩn Fibre Channel được tổ chức thành 5 mức. Các mức này được mô tả trên hình 7.31, cùng với các định nghĩa ngắn gọn trong bảng 7.6 đi kèm. Mỗi mức định nghĩa một chức năng hay một tập các chức năng liên quan. Chuẩn này không có chủ ý chỉ ra một tương ứng nào giữa các mức và các triển khai thực tế, với một giao tiếp



đặc biệt giữa các mức kế nhau. Thay vì vậy, chuẩn này đề cập đến các mức như là một *tài liệu cơ bản* được dùng để nhóm các chức năng liên quan lại với nhau.

Các mức từ FC-0 đến FC-2 của phân cấp Fibre Channel hiện tại được định nghĩa trong một chuẩn được đề cập đến như là *Giao tiếp Vật lý* và *Báo hiệu Fibre Channel* FC-PH. Hiện tại không có chuẩn cụ thể cho FC-3. Tại mức FC-4, các chuẩn riêng được tạo ra để ánh xạ một kênh khác và các giao thức khác xuống các mức thấp hơn.



Hình 7.31 Các mức Fibre Channel.

Bảng 7.6 Các mức của Fibre Channel

FC-0 môi trường vật lý
*Cáp quang với các bộ phát LED hay LD trên cự ly truyền dài.
*Cáp đồng trục bằng đồng cho các tốc độ cao nhất trên các cự ly ngắn.
*Cáp STP cho các tốc độ thấp hơn qua các cự ly ngắn.
FC-1 đồng bộ byte và mã hóa
*Lược đồ mã hóa và giải mã 8B/10B cung cấp khả năng cân bằng, dễ hiện thực và cung cấp khả năng phát hiện lỗi hiệu quả.
*Ký tự mã đặc biệt duy trì đồng bộ bit và đồng bộ byte
FC-2 cơ cấu chuyển tải thực
*Giao thức đồng bộ frame và điều khiển luồng giữa các N-port
*Ba lớp dịch vụ giữa các port
FC-3 lớp các dịch vụ chung



*Các dịch vụ liên quan port

*Các dịch vụ xuyên qua hai hay nhiều port trong một node

FC-4 các giao thức lớp trên

*Hỗ trợ kênh khác và các giao thức khác.

Chúng ta sẽ kiểm tra sơ lược các mức này lần lượt trong phần còn lại của mục này.

1.9.4. Môi trường và giao tiếp vật lý

Mức FC-0 của Fibre Channel cho phép một số môi trường vật lý cũng như tốc độ số liệu khác nhau; đây là một trong các điểm mạnh của đặc tả này. Hiện tại, dài tốc độ nằm trong khoảng từ 100 Mbps đến 800Mbps trên một sợi. Môi trường vật lý là sợi quang, cáp đồng trục và cáp STP. Tùy vào tốc độ số liệu và môi trường liên quan, cự ly tối đa cho các liên kết điểm nối điểm riêng rẽ nằm trong khoảng từ 50m đến 10km.

1.9.5. Giao thức truyền

Mức FC-1 là mức giao thức truyền định nghĩa một kỹ thuật mã hóa số liệu được dùng để truyền và đồng bộ xuyên qua các liên kết điểm nối điểm. Lược đồ mã hóa được dùng là 8B/10B, trong đó mỗi 8 bit số liệu từ mức FC-2 được chuyển đổi thành 10 bit để truyền.

1.9.6. Giao thức đóng bộ frame

Mức FC-2 được đề cập đến như một giao thức đóng bộ frame, liên quan đến việc truyền số liệu giữa N-port dưới dạng các frame. Trong số các khái niệm được định nghĩa tại mức này là :

- Node và N-port cùng với các danh định của chúng
- Các cấu hình
- Các lớp dịch vụ được cung cấp bởi fabric
- Phân đoạn số liệu thành các frame và tái hợp trở lại
- Nhóm các frame thành các thực thể luận lý, thường được gọi như vậy trong tuần tự thiết lập và trao đổi.
- Hoạt động tuần tự, điều khiển luồng và kiểm soát lỗi.

1.9.7. Các dịch vụ chung

FC-3 cung cấp một tập các dịch vụ chung qua nhiều N-port của một node. Các chức năng được định nghĩa cho đến thời điểm này trong các tài liệu gồm có:

- **Stripping.** Cho phép dùng nhiều N-port dưới dạng song song để truyền một đơn vị thông tin đơn xuyên qua nhiều liên kết một cách đồng thời;



điều này đạt được sự tích hợp thông lượng cao hơn. Sử dụng nhiều trong các ứng dụng cần truyền nhiều tập số liệu lớn theo thời gian thực, như trong các ứng dụng video.

- **Hunt Groups.** Một Hunt Group là một tập các N-port liên kết nhau tại một node đơn. Tập này được gán một phiên hiệu danh định cho phép bất kỳ một frame nào được gửi đến phiên hiệu này được định tuyến đến bất kỳ N-port nào có sẵn trong tập này. Điều này làm giảm thời gian trễ nhờ giảm các trường hợp đợi khi một N-port đang bận.
- **Multicast.** Phân phối truyền dẫn đến nhiều đích. Điều này gồm cả truyền đến tất cả các N-port trên fabric (quảng bá) hay đến một tập các N-port trên một fabric.

1.9.8. Ánh xạ

FC-4 định nghĩa ánh xạ giữa các giao thức mạng và kênh khác sang FC-PH. Các giao tiếp kênh xuất nhập (I/O) bao gồm :

- **Giao tiếp SCSI.** Là một giao thức tốc độ cao được dùng rộng rãi nhất và được hiện thực trên các máy tính cá nhân, các máy trạm và các server. SCSI được dùng để hỗ trợ các thiết bị có tốc độ và dung lượng cao, như là đĩa và các thiết bị video hay đồ họa.
- **Giao tiếp HIPPI (High-Performance Parallel Interface).** Là một chuẩn về kênh tốc độ cao được dùng chủ yếu cho các môi trường siêu máy tính hay mainframe. HIPPI và các mở rộng được mô tả như là mục tiêu tổng quát cho giải pháp LAN tốc độ cao, nhưng HIPPI đã bị thay thế bởi Fibre Channel.

Các giao tiếp mạng bao gồm :

- **IEEE 802.** Các frame MAC của IEEE 802.3 ánh xạ vào các frame Fibre Channel.
- **Asynchronous Transfer Mode**
- **Internet Protocol (IP)**

Các giao thức ánh xạ FC-4 cho phép dùng các khả năng của FC-PH để truyền các thông tin giao thức mức cao hơn. Mỗi đặc tả FC-4 định nghĩa khuôn dạng và các thủ tục cho giao thức thông tin mức cao ULP (upper-layer protocol).

1.9.9. Môi trường vật lý và cấu hình

Một trong những điểm mạnh của chuẩn Fibre Channel là cung cấp một dải các tùy chọn về môi trường vật lý, về tốc độ số liệu trên môi trường này và cấu hình của mạng.

Bảng 7.7 Khoảng cách ứng với môi trường truyền và tốc độ

	800Mbps	400Mbps	200Mbps	100Mbps
Sợi đơn mode	10km	10km	10km	-
Sợi đa mode 50µm	0,5km	1km	2km	10km
Sợi đa mode 62,5µm	175m	350m	1500m	1500m
Cáp đồng trục video	25m	50m	75m	100m
Cáp đồng trục	10m	15m	25m	35m
Cáp xoắn STP	-	-	50m	100m

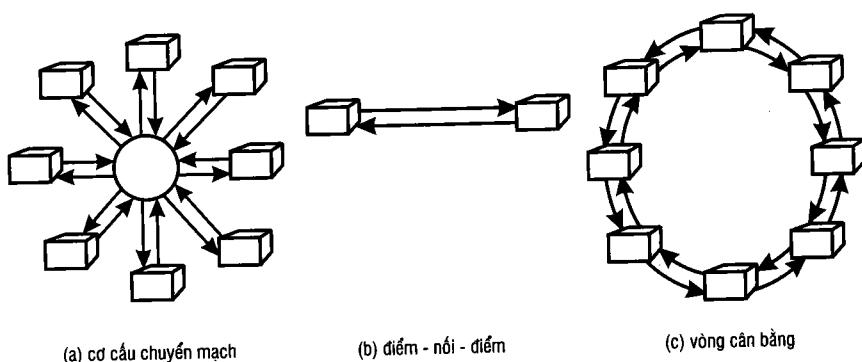
Bảng 7.7 tóm tắt các tùy chọn cho môi trường vật lý và tốc độ số liệu sẵn có trong chuẩn Fibre Channel.

Mỗi hàng mô tả cự ly tối đa của liên kết điểm nối điểm, được chỉ định theo môi trường vật lý và tốc độ cho trước. Các môi trường này có thể là hỗn hợp trong một cấu hình toàn cục. Ví dụ, một liên kết quang đơn mode có thể được dùng để kết nối các chuyển mạch trong các tòa nhà khác nhau, với các liên kết quang đa mode được dùng cho phân tán đọc bên trong, và các liên kết cáp đồng trục và cáp STP dẫn đến các máy trạm.

Cấu hình tổng quát nhất được hỗ trợ bởi Fibre Channel được xem như một cấu hình fabric hay chuyển mạch. Đây là một cấu hình cân bằng gồm có ít nhất một chuyển mạch được kết nối đến một số các N-port như trình bày trên hình 7.32(a). Cấu hình fabric cũng bao gồm một số các chuyển mạch hình thành nên một mạng chuyển mạch, trong đó một vài hay tất cả các chuyển mạch này đều hỗ trợ các node đầu cuối.

Định tuyến trong cấu hình fabric là trong suốt đến các node. Mỗi port trong cấu hình có một địa chỉ duy nhất. Khi số liệu từ một node được truyền đến fabric, tại kết cuối của chuyển mạch dùng địa chỉ port đích này trong frame số liệu đến để xác định vị trí của port đích. Sau đó chuyển mạch phân phối các frame vào node khác cùng nối vào chuyển mạch này hay truyền frame đến chuyển mạch kế để bắt đầu định tuyến các frame đến đích ở xa.

Cấu hình fabric cung cấp một dải dung lượng rộng. Khi thêm các port vào, dung lượng tích hợp của mạng gia tăng, nhờ đó tối thiểu được tắt nghẽn và tranh chấp, gia tăng thông lượng. Cấu hình fabric độc lập với giao thức và không nhạy cảm đối với cự ly. Kỹ thuật của chính chuyển mạch và của các liên kết nối các node vào chuyển mạch có thể bị thay đổi nhưng không ảnh hưởng đến toàn bộ cấu hình. Ưu điểm khác của kỹ thuật fabric là gánh nặng trên mỗi node được giảm đáng kể. Một node của Fibre Channel chỉ chịu trách nhiệm quản lý một liên kết điểm nối điểm đơn giản giữa nó và fabric này; fabric chịu trách nhiệm định tuyến giữa các N-port và phát hiện lỗi.



Hình 7.32 Các cấu hình Fibre Channel cơ bản.

Thêm vào kỹ thuật fabric này, chuẩn Fibre Channel còn định nghĩa hai kỹ thuật khác. Với kỹ thuật **điểm nối điểm** (hình 7.32 (b)), chỉ có hai N-port, và các port này được nối một cách trực tiếp, không có các chuyển mạch fabric xen giữa. Trong trường hợp này không cần phải định tuyến.

Cấu hình sau cùng là cấu hình vòng cân bằng (hình 7.32(c)) đơn giản, giá thành thấp cho phép kết nối đến 126 node trong vòng. Các port trên một vòng cân bằng phải có các chức năng của cả các N-port và F-port; các port này được gọi là các **NL-port**. Vòng cân bằng này hoạt động theo phương thức gần tương đương với các giao thức token ring. Mỗi port thấy tất cả các frame và chuyển tiếp hay bỏ qua các frame nếu địa chỉ đích không chỉ đến chính nó. Có một giao thức thu nhận token để điều khiển truy xuất vào vòng.

Các kỹ thuật vòng cân bằng và fabric có thể kết hợp với nhau trong một cấu hình để tối ưu giá thành. Trong trường hợp này, một trong các node trên vòng phải là một **fabric-loop node** (FL-node) để tham gia vào việc định tuyến với các chuyển mạch khác trong cấu hình fabric này.

Loại cấu hình này không cần được cấu hình thường xuyên bởi một người quản lý mạng. Thay vì vậy loại cấu hình này được xác lập trong quá trình khởi động liên kết.

2. CÁC LAN KHÔNG DÂY

2.1. Khái quát

Các loại LAN mà chúng ta vừa thảo luận ở trên, hầu hết đều dùng cáp xoắn đôi hay cáp đồng trực để làm môi trường vật lý truyền. Giá thành chủ yếu liên quan đến các LAN này chính là chi phí lắp đặt các đường cáp vật lý. Hơn thế nữa, nếu kiến trúc của sơ đồ kết nối các máy tính thay đổi thì chi phí để thực hiện tương đương với chi phí lắp đặt từ đầu khi thay đổi kế hoạch nối dây. Đây là một trong các lý do vì sao các mạng LAN không dây được phát triển. Các LAN không

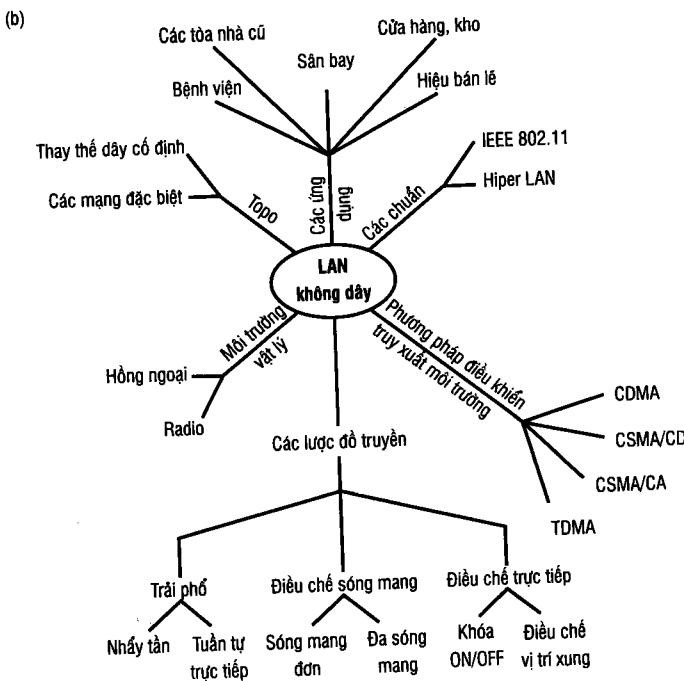
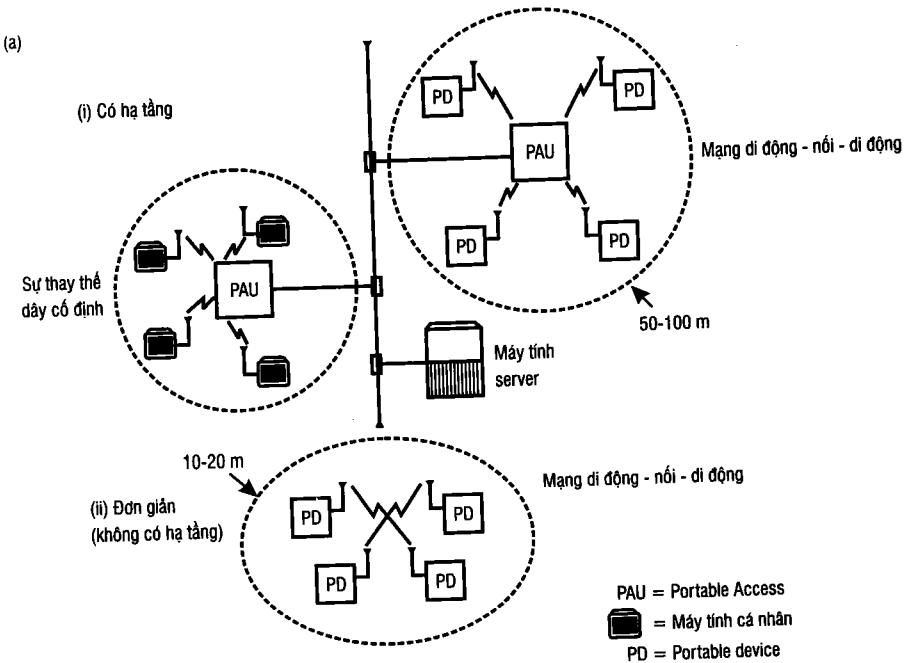
dây là các LAN không dùng các dây nối vật lý làm môi trường vật lý truyền dẫn chính.

Một lý do thứ hai là sự xuất hiện các thiết bị đầu cuối và máy tính xách tay. Khi kỹ thuật ngày càng trở nên tiên tiến thì các thiết bị như vậy nhanh chóng so sánh được về sức mạnh với các máy tính cố định. Mặc dù lý do chính để dùng các thiết bị này là tính di động, chúng thường phải thông tin liên lạc với các máy tính khác. Các máy tính khác có thể là máy tính xách tay (di động) hoặc phổ biến hơn là các máy tính được nối vào các mạng LAN nối dây. Ví dụ như các thiết bị đầu cuối trong siêu thị thông tin với máy tính lưu trữ ở xa để cập nhật các record trong cơ sở dữ liệu của kho hàng, hoặc trong một bệnh viện, một y tá với một máy tính xách tay có thể truy xuất vào hồ sơ của bệnh nhân được giữ trong cơ sở dữ liệu tại một mainframe.

Một tập các chuẩn LAN không dây đã được phát triển bởi tổ chức IEEE gọi là IEEE 802.11. Thuật ngữ và vài thuộc tính đặc biệt của 802.11 là duy nhất đối với chuẩn này và không bị ảnh hưởng trong tất cả các sản phẩm thương mại. Tuy nhiên, việc làm quen với chuẩn này là rất hữu ích vì các đặc tính của nó tương ứng cho các năng lực mạng được yêu cầu đối với LAN không dây.

Một sơ đồ minh họa hai ứng dụng của các LAN không dây được trình bày trên hình 7.33(a). Như chúng ta có thể thấy, trong ứng dụng thứ nhất, để truy xuất vào máy tính server đang được nối vào một LAN nối dây, cần dùng một thiết bị trung gian được gọi là đơn vị truy xuất di động PAU (Portable Access Unit). Thông thường vùng phủ của PAU là từ 50 đến 100m và trong một dự án lắp đặt lớn có nhiều đơn vị như vậy phân bố xung quanh một điểm. Tập hợp các đơn vị này cung cấp khả năng truy xuất vào LAN nối dây và do đó là truy xuất vào các máy tính server cho các máy cầm tay, máy tính xách tay hay máy tính cố định, mỗi thiết bị đầu cuối này có thể tọa lại tại bất cứ nơi nào xung quanh điểm này. Loại ứng dụng này được gọi là *LAN không dây có hạ tầng cơ sở*.

Trong ứng dụng thứ hai, một tập các máy tính di động có thể thông tin với nhau hình thành một LAN đơn giản hay *LAN không dây không có hạ tầng cơ sở*. Ví dụ điều này có thể trong một phòng hội thảo, hay tại một sân bay. Vì các mạng như vậy được tạo ra theo yêu cầu, nên chúng thường được xem như các LAN không dây đơn giản. Cũng như các LAN nối dây, có một số điểm cần phải xem xét khi mô tả hoạt động của các LAN không dây và chúng được tóm tắt trên hình 7.33(b). Trong thực tế, có nhiều mối liên quan giữa các chủ điểm đã được xác định, nhưng vì mục đích mô tả nên mỗi chủ điểm sẽ được xem xét một cách độc lập.



Hình 7.33 Các LAN không dây:
(a) Các topo ứng dụng (b) Các khía cạnh kỹ thuật.



2.2. Đường truyền không dây

Có hai loại đường truyền được dùng cho các LAN không dây là sóng trong dải tần số radio và các tín hiệu hồng ngoại tuyến. Chúng ta sẽ xem xét các đặc trưng của từng loại riêng biệt, cho dù các kỹ thuật được dùng cho cả hai loại này là hoàn toàn tương tự nhau.

2.2.1. Đường truyền bằng sóng radio

Các sóng radio được dùng rộng rãi trong nhiều ứng dụng. Bao gồm phát thanh và truyền hình đại chúng, và các mạng điện thoại di động. Vì sóng radio có thể xuyên qua dễ dàng các chướng ngại vật như các bức tường, nên các phương pháp điều khiển chất chẽ được áp dụng khi dùng phổ của sóng radio. Dải ứng dụng rộng cũng có nghĩa là băng thông của radio khan hiếm. Đối với một ứng dụng đặc biệt, một băng tần xác định phải được phân phối một cách chính thức. Trong quá khứ điều này đã được thực hiện cơ bản trên một quốc gia, nhưng với đà gia tăng ứng dụng thì các sáp xếp mang tính quốc tế đang được ký kết, qua đó để riêng các băng tần đã chọn cho các ứng dụng liên quan đến quốc tế.

Các nhu cầu giới hạn phát sóng radio vào một băng tần nào đó và trong các máy thu liên quan chỉ chọn các tín hiệu trong băng tần này làm cho các mạch điện liên quan đến các hệ thống truyền tin radio phức tạp hơn nhiều so với những gì được dùng trong các hệ thống thông tin bằng hồng ngoại. Tuy nhiên, việc sử dụng rộng rãi sóng radio, đặc biệt là trong số lượng lớn sản phẩm dân dụng khiến cho giá thành thiết kế các hệ thống radio ở mức chấp nhận được.

Tổn thất đường truyền (path loss)

Tất cả các máy thu radio đều được thiết kế để hoạt động với một tỉ số SNR qui định, nghĩa là tỉ số giữa năng lượng tín hiệu thu được trên năng lượng của nhiễu tại máy thu không được xuống thấp hơn một giá trị cho trước. Nhìn chung, độ phức tạp của máy thu tăng thì SNR giảm. Tuy nhiên, với giá thành hạ của các máy tính xách tay cũng có nghĩa là giá cả chấp nhận được của các đơn vị giao tiếp mạng radio phải có thể so sánh tương xứng với giá thành của các máy tính xách tay. Do đó, điều này cũng đồng nghĩa với tỉ số SNR của máy thu phải được thiết kế ở mức cao nếu có thể.

Trong thực tế, SNR tùy thuộc vào một số các tham số liên quan và mỗi tham số này phải được xem xét trong mối liên hệ với thiết kế của máy thu radio. Như chúng ta đã mô tả trong chương 2, nhiễu tại máy thu tùy thuộc vào nhiệt độ môi trường làm gia tăng nhiễu nhiệt và băng thông của tín hiệu thu; băng thông hay nhiệt độ càng lớn thì nhiễu càng lớn. Do đó trong một ứng dụng xác định, hệ số nhiễu tại máy thu về cơ bản là cố định.

Năng lượng tín hiệu tại máy thu không chỉ phụ thuộc vào năng lượng tín hiệu đã phát đi mà còn phụ thuộc vào khoảng cách giữa máy thu và máy phát. Trong không gian tự do, năng lượng của một tín hiệu radio suy thoái tỉ lệ nghịch với bình phương khoảng cách tính từ nguồn. Ngoài ra trong môi trường bị bao phủ bởi văn phòng công sở, sự suy thoái còn tăng hơn nữa, trước hết là do sự che chắn



của các chương ngai vật như tiện nghi sinh hoạt và con người, kể đến là do sự can nhiễu của các tín hiệu phản xạ từ các chương ngai vật này. Chúng phối hợp với nhau gây ra những gì được gọi là tổn thất đường truyền của kênh sóng vô tuyến.

Do đó để cho máy thu hoạt động với một SNR có thể chấp nhận được, nó phải hoạt động trong hệ thống có mức năng lượng máy phát càng cao càng tốt và (hoặc) với một vùng phủ có giới hạn. Trong thực tế, với các máy tính xách tay, năng lượng của tín hiệu được phát bị giới hạn bởi sự tiêu thụ tại đơn vị giao tiếp mạng radio, điều này làm tăng một lượng tài đối với nguồn của máy tính. Đó cũng là lý do vì sao vùng phủ của một LAN đơn giản không hạ tầng cơ sở (ad hoc) lại ngắn hơn so với LAN có hạ tầng cơ sở.

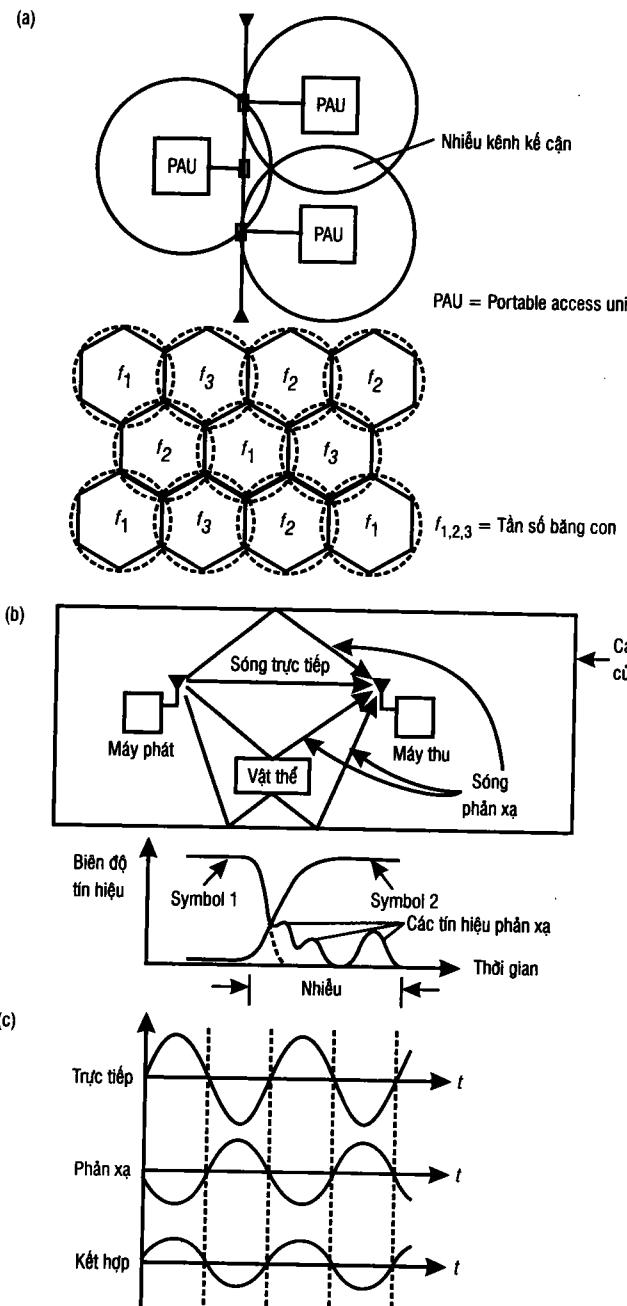
Nhiều xuyên kenh

Vì sóng radio lan truyền xuyên qua hầu hết các chương ngai vật với mức suy giảm vừa phải, điều này có thể tạo ra sự tiếp nhận nhiễu từ các máy phát khác cũng đang hoạt động trong cùng băng tần và được đặt trong phòng kế cận của cùng một tòa nhà hay trong tòa nhà khác. Do đó, với các LAN đơn giản, vì nhiều LAN như vậy có thể được thiết lập trong các phòng gần nhau, nên các kỹ thuật phải theo là cho phép vài user trong cùng một băng tần cùng tồn tại.

Trong một mạng LAN không dây có hạ tầng cơ sở, vì topo đã biết và tổng diện tích vùng phủ của mạng không dây là lớn hơn nhiều _tương tự như một LAN nối dây_ thì băng thông có sẵn có thể được chia thành một số băng con sao cho vùng phủ của các băng kề nhau dùng một tần số khác nhau. Lược đồ tổng quát được trình bày trên hình 7.34(a). Lược đồ này được gọi là mẫu lặp 3 cell (three-cell repeat pattern) cho dù có thể có các mẫu lớn hơn. Số lượng băng thông có sẵn trong mỗi cell được chọn để cung cấp một mức độ phục vụ chấp nhận được cho số lượng user hoạt động được qui định trước trong vùng này. Điều này tạo ra một hiệu suất sử dụng băng thông tốt hơn và do đảm bảo tất cả các cell kề nhau, mỗi cell đều dùng một tần số khác nhau nên mức nhiễu xuyên kenh được giảm tối đa.

Đa đường (multipath)

Cũng giống như các tín hiệu quang, các tín hiệu radio chịu ảnh hưởng bởi đa đường, nghĩa là tại bất cứ thời điểm nào máy thu đều nhận nhiều tín hiệu xuất phát từ cùng một máy phát, mỗi tín hiệu được dẫn theo một con đường khác nhau giữa máy phát và máy thu. Điều này được gọi là sự phân tán đa đường và khiến cho các tín hiệu liên quan đến mẫu/bit trước xuyên nhiễu các tín hiệu liên quan đến mẫu/bit kế tiếp. Điều này được gọi là nhiễu xuyên mẫu và được trình bày trên hình 7.34(b). Rõ ràng tốc độ bit càng cao, khoảng thời bit càng ngắn thì mức xuyên nhiễu mẫu càng lớn.



Hình 7.34 Sự suy thoái tín hiệu radio: (a) Nhiều xuyên kẽm và ví dụ về phân phối tần số (b) Nhiều xuyên mâu (c) Rayleigh fading.

Ngoài ra, còn có một suy giảm được gọi là **fading** gây ra bởi sự thay đổi chiều dài đường đi của các tín hiệu thu khác nhau. Điều này làm gia tăng các khoảng cách pha tương quan giữa chúng, có thể tạo ra các tín hiệu phản xạ khác nhau làm suy giảm đáng kể tín hiệu trên tuyến trực tiếp, và trong một giới hạn nào đó có thể khử lẫn nhau. Hiện tượng này được gọi là **Rayleigh fading** và được trình bày trên hình 7.34(c). Trong thực tế, biên độ của sóng phản xạ là một thành phần của sóng trực tiếp, mức độ suy giảm được xác định bởi vật liệu phản xạ. Một giải pháp cho vấn đề này là khai thác đặc tính bước sóng rất ngắn của các tín hiệu tần số radio khiến cho nó rất nhạy cảm với các thay đổi nhỏ về vị trí của anten. Để khắc phục ánh hưởng này, hai anten thường được dùng với khoảng cách vật lý giữa chúng bằng $1/4$ bước sóng. Các tín hiệu thu từ cả hai anten được kết hợp lại thành một tín hiệu thu thống nhất. Kỹ thuật này được gọi là **phân tập không gian** (*space diversity*).

Một giải pháp khác là dùng kỹ thuật được gọi là **cân bằng** (*eqlualization*). Các ánh suy giảm và trễ của tín hiệu trực tiếp thường như các tín hiệu phản xạ đã đường bị loại trừ khỏi tín hiệu thu thực sự. Vì các tín hiệu phản xạ sẽ thay đổi theo các vị trí khác nhau của máy phát và máy thu, nên quá trình này phải thích nghi. Do đó mạch điện được dùng ở đây được gọi là **bộ cân bằng thích nghi** (*adaptive equalizer*). Rõ ràng, việc sử dụng các mạch như vậy sẽ làm tăng giá thành của máy thu.

2.2.2. Đường truyền bằng sóng hồng ngoại

Các bộ thu phát hồng ngoại đã được sử dụng từ nhiều năm nay trong các ứng dụng khác nhau. Các ứng dụng gồm có các hệ thống truyền dẫn bằng sợi quang và các ứng dụng điều khiển từ xa khác nhau ví dụ như hệ thống điều khiển từ xa cho tivi, máy CD.v.v. Sóng hồng ngoại có tần số cao hơn rất nhiều so với tần số sóng radio_lớn hơn 10^{14} Hz và các thiết bị thường được phân loại theo chiều dài bước sóng của tín hiệu hồng ngoại được thu phát thay vì dùng tần số. Chiều dài bước sóng hồng ngoại được đo lường theo nm ($1\text{nm} = 10^{-9}\text{m}$) và là khoảng cách mà ánh sáng truyền trong thời gian bằng chu kỳ của tín hiệu. Nghĩa là:

$$\text{Chiều dài bước sóng}, \lambda = c/f$$

Trong đó c là tốc độ lan truyền của ánh sáng ($c = 3 \cdot 10^8\text{m/s}$) và f là tần số của tín hiệu tính bằng Hz.

Hai thiết bị hồng ngoại được dùng rộng rãi nhất có bước sóng lần lượt là 800nm và 1300nm.

Một ưu điểm của hồng ngoại so với sóng radio là không có qui định nào trong việc dùng nó. Hồng ngoại có một bước sóng tương tự như sóng ánh sáng thấy được và do đó có biểu hiện như nhau: ví dụ nó bị phản xạ từ các bề mặt nhẵn bóng, nó sẽ xuyêp qua được thủy tinh nhưng không xuyêp qua được các bức tường hay các vật thể mờ đục khác. Do đó sóng hồng ngoại bị giới hạn trong một căn phòng, từ đó làm giảm mức nhiễu xuyêp kênh trong các ứng dụng mạng LAN không dây.

Một điểm khác cũng cần xem xét khi sử dụng hồng ngoại làm môi trường truyền đó là nhiễu gây ra bởi ánh sáng của môi trường xung quanh. Ánh sáng mặt trời, ánh sáng đèn điện tròn và các nguồn sáng huỳnh quang, tất cả đều chứa một mức đáng kể tia hồng ngoại. Lượng ánh sáng hồng ngoại này được thu bởi bộ thu quang cùng với lượng hồng ngoại tuyển từ nguồn phát chính. Điều này có nghĩa là năng lượng nhiễu có thể cao, dẫn đến nhu cầu năng lượng phát tín hiệu phải cao để đạt được một tỉ số SNR chấp nhận được. Trong thực tế, tổn thất đường truyền đối với hồng ngoại có thể cao. Ngoài ra các bộ phát hồng ngoại có hiệu suất thấp khi biến đổi năng lượng từ điện sang quang. Tổng hợp lại có thể dẫn đến nhu cầu năng lượng khá cao đối với nguồn cung cấp. Để giảm mức nhiễu, trong thực tế thường chuyển hỗn hợp tín hiệu thu được qua một bộ lọc quang băng gốc (optical bandpass filter), bộ lọc này làm suy giảm các tín hiệu hồng ngoại nằm ngoài băng tần gốc của tín hiệu đã được truyền.

Các thiết bị

Trong các ứng dụng LAN không dây, chế độ hoạt động là điều chế cường độ của tín hiệu hồng ngoại ở ngõ ra của bộ phát quang bằng một tín hiệu điện. Các dấu hiệu thay đổi cường độ trong tín hiệu hồng ngoại thu được tại bộ thu quang được chuyển trở lại thành các tín hiệu điện tương ứng. Chế độ hoạt động này được gọi là điều chế cường độ với tách trực tiếp IMDD (intensity modulation with direct detection) và như chúng ta sẽ thấy trong mục 3.3, có các phương pháp điều chế khác nhau bao gồm cả điều chế băng tần cơ bản (baseband).

Có hai loại thiết bị phát ra hồng ngoại: là các laser diode và diode phát quang (LED_light emitting diode). Laser diode được dùng rộng rãi trong các hệ thống truyền dẫn băng sợi quang. Chúng tạo ra nguồn sáng liên tục có băng tần rất hẹp (thường nằm khoảng giữa 1 và 5 nm), và khi ánh sáng được nhốt trong một không gian hẹp sẽ thu được mật độ năng lượng cao. Trong các ứng dụng LAN không dây, vì ánh sáng không bị ràng buộc lan truyền trong phạm vi một sợi quang, nên nguồn sáng laser phải được khuếch tán nếu không sẽ gây nguy hiểm cho mắt. Ngược lại, diode phát quang (LED) tạo ra nguồn sáng gồm một dải tần, thường nằm giữa 25 đến 100mm, và với năng lượng ngõ ra thấp nên hoàn toàn vô hại. Băng thông điều chế có sẵn đối với LED giới hạn khoảng 20MHz, chấp nhận một hạn chế về tốc độ bit tối đa có thể dùng nhỏ hơn 10Mbps. Bởi giá thành thấp nên chúng ta thường sử dụng LED trong các ứng dụng có tốc độ thấp hơn tốc độ giới hạn này.

Đối với các tốc độ lớn hơn 10Mbps chúng ta phải dùng laser. Băng thông điều chế có sẵn với laser là hàng trăm MHz. Liên hệ với LED là băng tần rộng điều này có nghĩa là tại đầu thu phải có bộ lọc băng gốc rộng để tách tất cả tín hiệu đã truyền. Tuy nhiên, điều này làm tăng tín hiệu nhiễu tại đầu thu, đặc biệt với tốc độ bit cao, khiến cho việc thiết kế máy thu khó khăn hơn.

Các topo

Các liên kết hồng ngoại được dùng theo một trong hai chế độ: điểm-nối-diểm và phát tán. Trong chế độ điểm-nối-diểm, thiết bị phát quang hướng trực tiếp đến

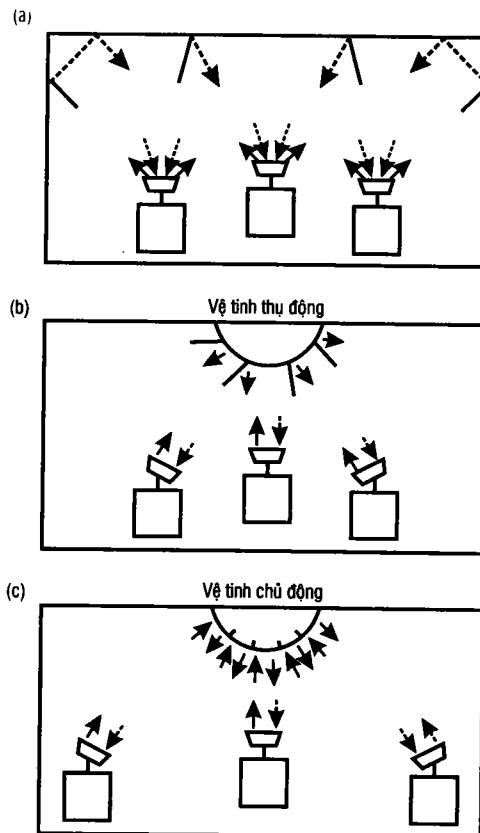


bộ thu quang trong thực tế thường là photodiode do đó các bộ phát quang công suất thấp hơn nhiều và các bộ thu quang có độ nhạy kém hơn có thể được dùng. Chế độ hoạt động này khá thích hợp cho việc cung cấp một liên kết không dây giữa hai thiết bị, ví dụ như cho phép một máy tính xách tay tải về các tập tin từ một máy tính khác.

Trong các ứng dụng LAN không dây yêu cầu chế độ hoạt động broadcast. Để thực hiện điều này, ngõ ra của nguồn phát hồng ngoại là phát tán để ánh sáng được trãi rộng ra theo một góc nào đó. Đây là chế độ được phát tán: có 3 chế độ hoạt động được trình bày trên hình 7.35. trong chế độ cơ bản_phản (a) có một bộ phát quang góc rộng và một bộ thu quang liên kết với mỗi máy tính. Tín hiệu hồng ngoại phát ra bởi bất kỳ bộ phát quang nào đều được thu bởi tất cả các bộ thu quang sau nhiều lần phản xạ trong phòng. Hiệu quả của chế độ hoạt động này là nhiều nguyên bản của cùng một tín hiệu nguồn đến tại mỗi bộ thu quang theo từng thời điểm khác nhau được xác định bởi đường truyền vật lý của mỗi tín hiệu. Như đã nói ngay từ đầu, đây là sự tản sáu đa đường và ảnh hưởng của nó là thời gian trễ trãi rộng vì các xung biểu diễn từng bit độc lập trong luồng bit truyền bị giãn ra. Cũng giống như sóng radio, biên độ của các tín hiệu phản xạ khác nhau thay đổi so với tín hiệu trực xạ tùy vào đường truyền của chúng và sự suy giảm mà chúng phải chịu. Trong một căn phòng thông thường, các tín hiệu có ý nghĩa có thể thu được với thời gian trễ trãi rộng khoảng 100nm. Chế độ hoạt động này chỉ phù hợp với tốc độ đến 1Mbps, vì với tốc độ cao hơn ảnh hưởng của nhiễu xuyên mầu sẽ tăng lên đáng kể.

Với hồng ngoại (và radio), bên cạnh dùng equalization, chúng ta còn có thể giảm ảnh hưởng của sự giãn xung bằng cách dùng nhiều bộ phát và thu định hướng_các anten định hướng dùng với radio_như trình bày trên hình 7.35(b). Với tiếp cận này, tất cả các bộ phát quang và thu quang đều được hướng đến một điểm có vòm phản xạ đặt trên trần nhà như một vệ tinh. Để tối đa năng lượng của tín hiệu thu và tối thiểu sự phản xạ, tín hiệu nguồn được hội tụ thành một chùm sáng tương đối hẹp. Hình dạng của vòm phản xạ vệ tinh được chọn sao cho đảm bảo tất cả các tín hiệu truyền đều được nhận bởi tất cả các thiết bị thu quang. Nhằm giảm ảnh hưởng của đa tuyến, độ mở của thiết bị thu quang được làm nhỏ sao cho chúng chỉ nhận tín hiệu trực xạ từ vệ tinh (vòm phản xạ).

Vệ tinh trong lược đồ vừa mô tả chỉ đóng vai trò của vật phản xạ ánh sáng. Do đó, để có một mức năng lượng tín hiệu chấp nhận được tại thiết bị thu quang, năng lượng tín hiệu bức xạ ra phải tương đối cao. Đối với các thiết bị di động, lấy năng lượng từ nguồn pin thì điều này là bất lợi và do đó một thủ thuật tinh vi hơn là dùng một vệ tinh tích cực (active satellite) như trên hình 7.35(c). Với lược đồ này, một dây các bộ thu quang_photodiode được phân bố xung quanh vòm cùng với một tập các bộ bức xạ hồng ngoại. Tất cả các tín hiệu nhận được bởi một hay nhiều tập thu quang đều được lặp lại bởi các bộ phát quang sau đó. Điều này có nghĩa là năng lượng của tín hiệu bức xạ bởi mỗi thiết bị di động có thể thấp hơn nhiều vì nó chỉ cần đủ để hình thành một tuyến dẫn đến vệ tinh.



Hình 7.35 Các chế độ hoạt động của hồng ngoại:
(a) Điểm-nối-điểm (b) Vệ tinh thụ động (c) Vệ tinh tích cực.

2.3. Các lược đồ truyền

Các đặc tính lan truyền khác nhau của hồng ngoại và sóng radio là nguyên nhân dẫn đến các lược đồ truyền khác nhau. Chúng ta sẽ bàn đến các lược đồ được dùng với mỗi loại môi trường một cách riêng biệt.

2.3.1. Lược đồ truyền sóng radio

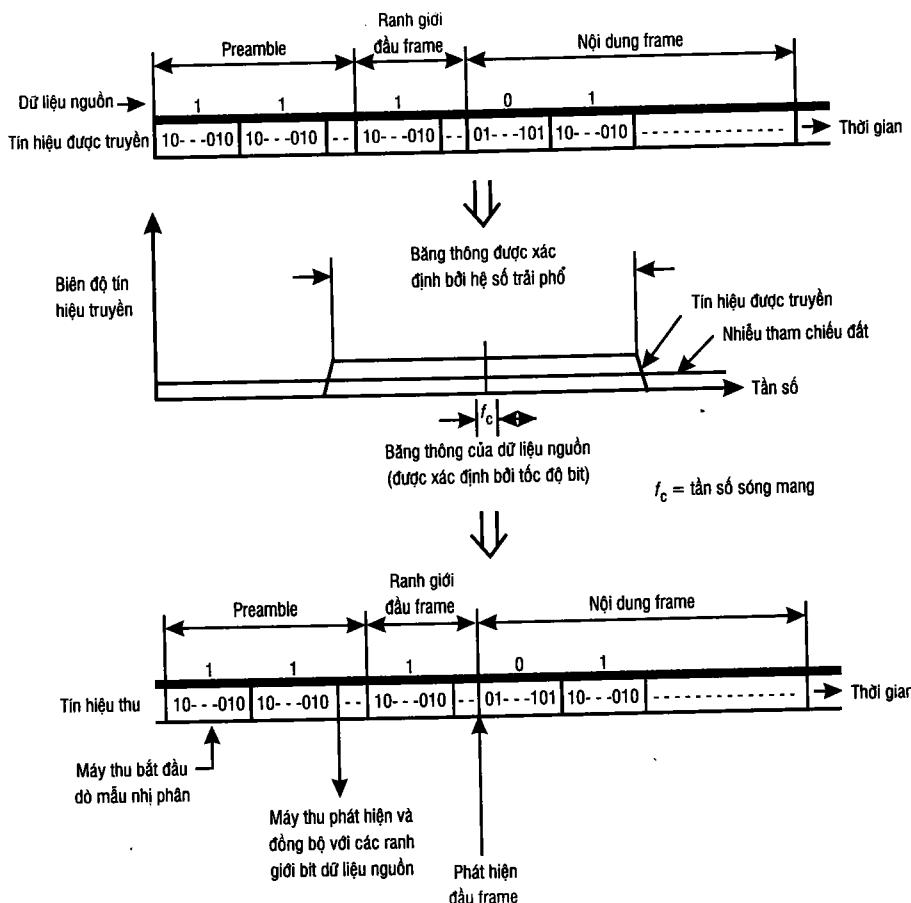
Có bốn lược đồ truyền được dùng với các mạng LAN không dây dùng sóng radio: trai phổ tuần tự trực tiếp (direct sequence spread spectrum), trai phổ nhảy tần (frequency-hopping spread spectrum), điều chế sóng mang đơn (single-carrier modulation), điều chế đa sóng mang (multi-subcarrier modulation).

(1) Trái phổ tuần tự trực tiếp

So với hầu hết các ứng dụng dùng phổ radio thì các LAN không dây là tương đối mới mẻ. Mặc dù phổ radio tự do có sẵn, nhưng nhìn chung đó là các tần số tương đối cao trong dải hàng chục GHz. Làm việc tại các tần số như vậy các trang



thiết bị được yêu cầu là mới và có giá thành tương đối cao. Đây là nhược điểm dễ thấy trong các ứng dụng LAN không dây khi so với giá thành rất thấp của các card giao tiếp mạng trong các mạng LAN nối dây hiện nay. Hơn nữa, khi giá thành của các máy tính xách tay giảm xuống thì giá thành của một giao tiếp LAN không dây cũng giảm theo. Vì các lý do này mà chuẩn LAN không dây radio đầu tiên sử dụng một băng tần đã từng được dùng với các trang thiết bị dễ tìm. Đây là một trong các băng tần được dành riêng cho các ứng dụng khoa học công nghệ và y khoa và thường được gọi là các băng tần ISM (industrial, scientific, and medial bands). Các ví dụ về ứng dụng các băng tần này gồm thiết bị tiệt trùng dùng tần số radio có công suất cao và các lò vi ba. Các hội viên vô tuyến nghiệp dư cũng dùng các băng tần này, thường có công suất phát cao. Để cùng tồn tại với các ứng dụng này, điều quan trọng là lược đồ truyền dẫn được chọn phải có mức khử nhiễu đồng kênh cao. Đối với các ứng dụng LAN không dây, điều này đạt được nhờ sử dụng một kỹ thuật gọi là trải phổ. Có hai hình thức trải phổ là tuần tự trực tiếp và nhảy tần. Chúng ta sẽ lần lượt đề cập đến hai dạng này.



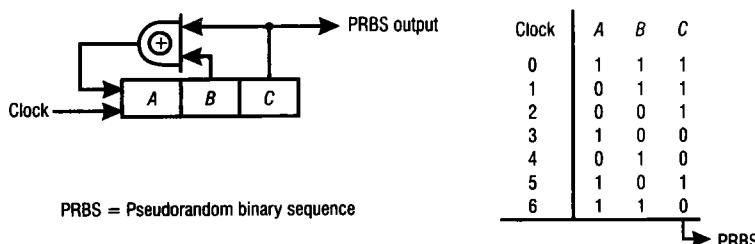
Hình 7.36 Nguyên lý hoạt động của trải phổ tuần tự trực tiếp

Nguyên lý hoạt động của trại phổ tuần tự trực tiếp được trình bày theo lược đồ hình thức trên hình 7.36. Số liệu nguồn được truyền trước hết được xor (cộng modulo) với một *tuần tự nhị phân giả ngẫu nhiên*, nghĩa là các bit tạo ra tuần tự mã giả này là ngẫu nhiên nhưng tuần tự này lớn hơn nhiều so với tốc độ của số liệu nguồn. Do đó khi tín hiệu đã xor được điều chế và truyền đi, nó chiếm và được gọi là *trái ra*_một băng tần tương ứng rộng hơn băng thông số liệu nguồn đã phát, làm cho tín hiệu xuất hiện như tạp âm đối với các user khác của cùng một băng tần.

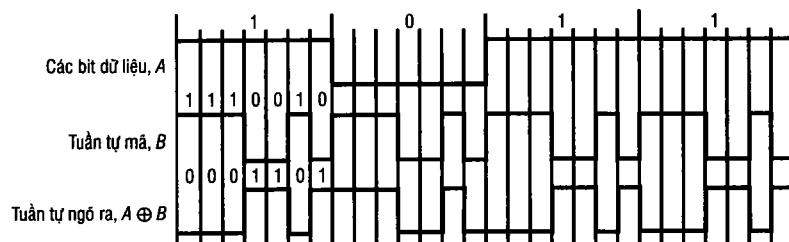
Tất cả các thành viên khác của cùng một LAN không dây đều biết tuần tự nhị phân giả ngẫu nhiên này đang được dùng. Tất cả các frame số liệu đang được truyền đều được đặt trước một tuần tự mở đầu kèm theo một mẫu đánh dấu đầu của frame. Do đó sau khi giải điều chế tín hiệu đã truyền, trước hết tất cả các máy thu phải tìm tuần tự mở đầu này thường là một chuỗi các bit 1_và khi đã tìm thấy tuần tự này máy thu bắt đầu dịch luồng bit theo các ranh giới bit chính xác. Đợi cho đến khi thu được mẫu đánh dấu đầu frame, sau đó sẽ xử lý thu nội dung của frame. Nơi nhận số liệu được xác định bởi địa chỉ đích đặt tại vùng header của frame theo cách thông thường.

Rõ ràng, vì tất cả các trạm thuộc về cùng một LAN không dây cùng chiếm hữu cùng băng tần được phân phối và dùng cùng tuần tự nhị phân giả ngẫu nhiên, nên hoạt động truyền của chúng sẽ quấy rối lẫn nhau. Vì vậy một phương pháp điều khiển truy xuất môi trường thích hợp phải được dùng nhằm đảm bảo chỉ một hoạt động truyền xảy ra tại một thời điểm.

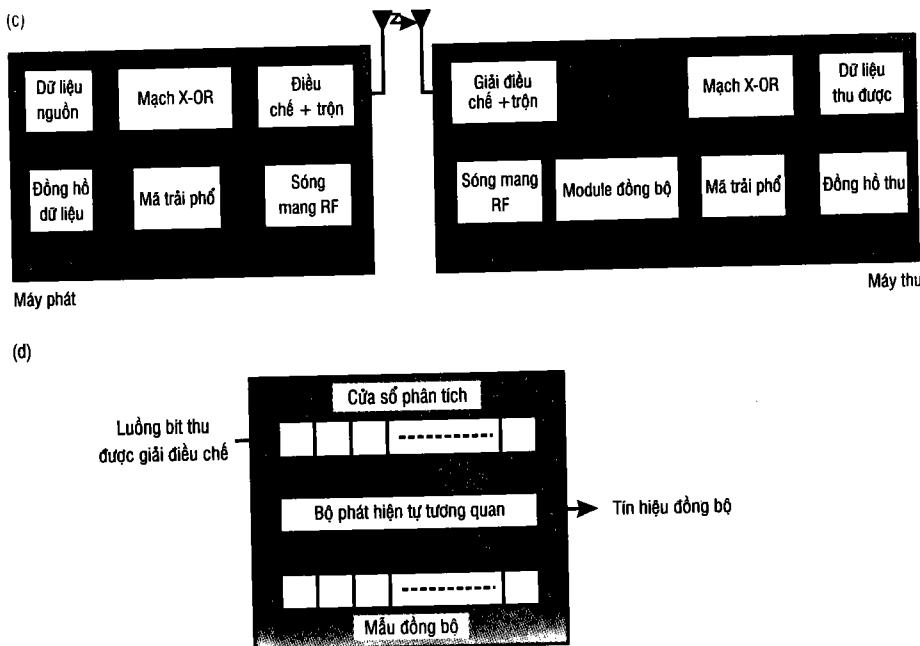
(a)



(b)



Hình 7.37 Các nguyên lý của trại phổ tuần tự trực tiếp: (a) Bộ sinh tuần tự giả ngẫu nhiên (b) Sự phát sinh tuần tự giả ngẫu nhiên



Hình 7.37 (tiếp theo) Các nguyên lý của trai phổ tuần tự trực tiếp:
(c) Lược đồ phát và thu (d) Sơ đồ module đồng bộ..

Trong thực tế, việc phát sinh một tuần tự nhị phân giả ngẫu nhiên là tương đối dễ vì nó có thể được tạo ra theo phương pháp kỹ thuật số chỉ dùng một số các thanh ghi dịch và một số cổng xor được nối trong một vòng hồi tiếp. Nguyên lý được trình bày trên hình 7.37(a). Trong ví dụ này dùng một thanh ghi dịch 3 bit và một cổng xor và điều này tạo ra 7 mảnh nhị phân giả 3 bit cũng được gọi là các trạng thái _ trước khi nó lặp lại. Lưu ý rằng trạng thái 000 không xuất hiện vì nội dung của thanh ghi dịch sẽ không thay đổi sau mỗi xung đồng hồ nhịp liên tiếp. Tổng quát, có tối đa ($2^n - 1$) trạng thái cho một thanh ghi dịch n bit và nếu tổ hợp hồi tiếp tạo ra tất cả ($2^n - 1$) trạng thái thì gọi là thanh ghi dịch có chiều dài tối đa. Ngõ ra từ phần tử có ý nghĩa lớn nhất của thanh ghi dịch sẽ là các tuần tự nhị phân giả ngẫu nhiên, trong trường hợp này nó bằng với mẫu nhị phân 7 bit 1110010.

Tuần tự giả ngẫu nhiên được dùng bằng cách thực hiện xor giữa tuần tự này với mỗi bit số liệu nhị phân được truyền. Ví dụ, mẫu tuần tự giả ngẫu nhiên 7 bit được lấy từ hình 7.37(a), thì phần (b) mô tả mảnh bit được truyền tương ứng với tập bốn bit số liệu. Lưu ý rằng ứng với mỗi bit số liệu có 7 bit được truyền. Và với bất kỳ vị trí bit số liệu nào, tuần tự bit được truyền cho một nhị phân 0 chỉ đơn giản là đảo ngược tuần tự bit được truyền cho nhị phân 1. Tuần tự nhị phân giả ngẫu nhiên này cũng được gọi là tuần tự trai phổ, mỗi bit trong tuần tự được gọi là chip,

tốc độ bit truyền sau cùng chính là *tốc độ chip*, và số bit trong tuần tự được gọi là *hệ số trải phổ* (spreading factor).

Hệ số trải phổ xác định hiệu suất của một hệ thống trãi phổ. Thông thường, nó được biểu diễn bằng đơn vị decibel (dB) và được xem như *độ lợi gia công* (processing gain), độ lợi này bằng logarithm của hệ số trãi phổ. Ví dụ một hệ thống trãi phổ với hệ số trãi phổ là 10:1 có độ lợi gia công là 10dB, 100:1 thì độ lợi là 20dB, và cứ thế. Trong tỉ số SNR_cũng được biểu diễn bằng dB_sẽ trừ bớt độ lợi này. Do đó đối với một hệ thống không trãi phổ giả sử yêu cầu SNR bằng 10dB_nghĩa là, năng lượng tín hiệu phải gấp 10 lần năng lượng tín hiệu nhiễu để thỏa mãn hoạt động_ nếu dùng trãi phổ với độ lợi gia công là 10dB thì hệ thống sẽ hoạt động thỏa mái ngay cả khi năng lượng tín hiệu bằng với năng lượng nhiễu.

Sơ đồ của máy phát và thu sóng radio trãi phổ tuần tự trực tiếp đơn giản được trình bày trên hình 7.37(c). Sau khi mỗi bit số liệu đã được xor với tuần tự nhị phân giả ngẫu nhiên, tín hiệu nhị phân có tốc độ cao được truyền bằng cách điều chế lên một tín hiệu sóng mang. Tần số của tín hiệu đã điều chế tăng lên_dùng mạch trộn (mixer circuit)_ sao cho tín hiệu truyền nằm trong băng tần đã được xác định. Các lược đồ điều chế thông thường được dùng là BPSK và QPSK, các nguyên lý điều chế này đã được trình bày trong chương 2.

Từ đó chúng ta có thể suy ra rằng máy thu phải hoạt động theo cơ cấu đồng bộ với tín hiệu thu để hoạt động xor được tiến hành trên các ranh giới bit chính xác. Để đạt được điều này, một mẫu nhị phân biết trước được truyền ngay khi bắt đầu mỗi frame_tuần tự mở đầu_và máy thu dùng mẫu này để đạt được sự đồng bộ bit và đồng bộ ký tự. Sơ đồ khối của module đồng bộ được trình bày trên hình 7.37(d).

Sự đồng bộ đồng hồ nhịp (tốc độ phát chip) đạt được bằng cách dùng một trong các phương pháp chuẩn được trình bày trong chương 3. Để đạt được sự đồng bộ ký tự (theo tốc độ số liệu), mỗi frame truyền được đặt phía trước một mẫu mở đầu bao gồm một chuỗi bit số liệu 1. Khi thu mẫu mở đầu này, nó được chuyển qua một thanh ghi n-bit_trong đó n là số bit trong tuần tự trãi_và được so sánh từng chip một với tuần tự đã biết tương ứng với bit số liệu 1. Nếu hai bit tại một vị trí chip đặc biệt là giống nhau, thì một sự *tán thành* (A) xảy ra, trong khi đó nếu không giống nhau thì một sự *bất đồng* (B) xảy ra. Tham số xác định sai biệt giữa hai mẫu được tính bằng cách lấy số lượng A trừ cho số lượng B và tham số này được gọi là *hàm tự tương quan* (autocorrelation function). Rõ ràng, khi mẫu ký tự đã biết được định vị thì hàm tự tương quan sẽ đạt giá trị đỉnh bằng với số chip trong tuần tự trãi phổ. Sau đó máy thu ở trong trạng thái đồng bộ ký tự. Độ tin cậy của xử lý đồng bộ được xác định bởi sự tự tương quan giữa tuần tự trãi phổ đã chọn và phiên bản được dịch chuyển của nó. Điều này trở nên quan trọng hơn khi các lỗi chip xuất hiện trong tuần tự trãi phổ thu được (được giải điều chế), ví dụ như do nhiễu quá mức. Điều này sẽ được miêu tả trong ví dụ dưới đây.

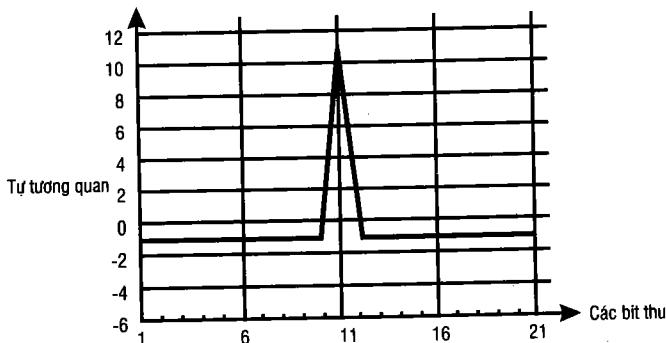
Ví dụ: Một tuần tự trãi tiêu biểu được dùng trong các hệ thống trãi phổ tuần tự trực tiếp là tuần tự nhị phân có 11 bit gồm 10110111000. Là một ví dụ của tuần

tự Barker. Hãy xác định và vẽ hàm tự tương quan cho tuần tự này trong hai biên ± 10 bit của nó.

(a)

		Cửa sổ phân tích	
Luồng bit thu	$\leftarrow 10110111000$	10110111000	Thời gian
		10110111000	
		<u>AAAAAAA</u> A - D = +11	
(i) + 1 bit		01101110001	
		10110111000	
		<u>DDADDAADAAD</u> A - D = -1	
(ii) - 1 bit		01011011100	
		10110111000	
		<u>DDDADDAADAA</u> A - D = -1	

(b)

**Hình 7.38 Ví dụ về đồng bộ:****(a) Ví dụ tự tương quan (b) Đồ thị hàm tự tương quan.**

Lời giải được tóm tắt trên hình 7.38. Trong phần (a) tuần tự bit (chip) thu được, được trình bày với cửa sổ phân tích trùng khớp với tuần tự tráï_ký tự đồng bộ_cùng với tuần tự thêm vào từ hai biên của nó. Rõ ràng, tại vị trí này, tất cả các vị trí bit sẽ là tần thành và giá trị của hàm tự tương quan sẽ là +11. Hai ví dụ khác tương ứng với tình huống cộng hay trừ 1 bit của tuần tự tráï và như chúng ta có thể thấy, nó tạo ra giá trị tự tương quan bằng -1. Sự thật, với tuần tự giả ngẫu nhiên này tất cả các vị trí bit ở cả hai biên của nó đều tạo ra giá trị tự tương quan là -1. Do đó đồ thị hàm tự tương quan được trình bày trên hình 7.38(b) và như chúng ta có thể thấy, ngõ ra của module đồng bộ chỉ dương khi nào nhận được ký tự đồng bộ.

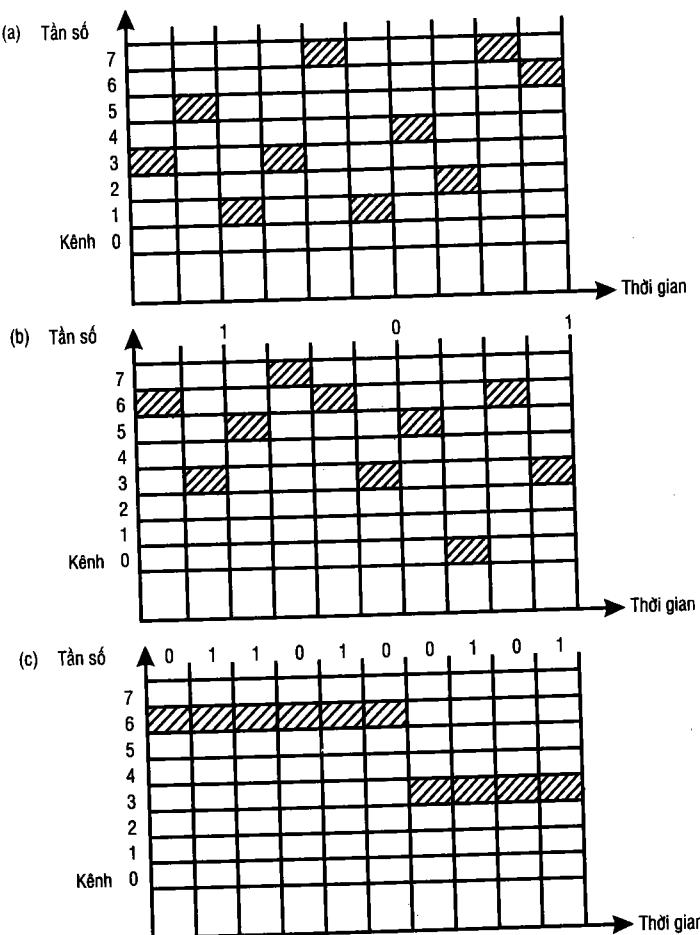
(2) Trải phổ nhảy tần

Nguyên lý hoạt động của trải phổ nhảy tần được trình bày trên hình 7.39(a). Băng tần được phân phối sẽ được chia thành một số các băng tần con thấp hơn được gọi là các kênh. Mỗi kênh đều có băng thông bằng nhau và được xác định bởi tốc độ bit và phương pháp điều chế được dùng. Máy phát dùng mỗi kênh trong một khoảng thời gian ngắn trước khi nhảy đến một kênh khác. Khi đang dùng một kênh, tần số sóng mang trung tâm của kênh được điều chế với các bit đang được truyền tại các thời điểm đó. Mẫu hữu dụng của kênh này là giả ngẫu nhiên và được gọi là *tuần tự nhảy (hopping sequence)*, thời gian trải qua trên mỗi kênh được xem như *khoảng thời gian của một chip (chip period)*, và tốc độ nhảy xem như *tốc độ phát chip*.

Có hai chế độ hoạt động tương ứng với trải phổ nhảy tần và được xác định bởi tỉ số giữa tốc độ phát chip so với tốc độ số liệu gốc (nguồn). Chúng được trình bày trên hình 7.39(b) và (c). Khi tốc độ chip lớn hơn tốc độ số liệu thì chế độ hoạt động tương ứng được gọi là *nhảy tần nhanh*, trong khi nếu tốc độ chip thấp hơn tốc độ số liệu thì gọi là *nhảy tần chậm*. Trong cả hai trường hợp, có một tần số sóng mang được dùng tại trung tâm của mỗi kênh.

Một ưu điểm của nhảy tần so với tuần tự trực tiếp là khả năng tránh dùng các kênh đã chọn (băng hẹp) trong toàn bộ băng tần đã được phân phối. Điều này đặc biệt hữu ích đối với các băng ISM bởi khả năng xuất hiện của một hay nhiều nguồn nhiễu băng hẹp công suất lớn trong vùng phủ của LAN. Như đã được thảo luận ở phía trước, mặc dù với tuần tự trực tiếp tín hiệu nhiễu được trải qua băng tần đã được gán, nhưng với các nguồn có công suất cao thì điều này vẫn gây ra mức nhiễu đáng kể và trong một chừng mực nào đó có thể khiến cho các băng tần nào đó không thể sử dụng được. Tuy nhiên, với nhảy tần, nếu một nguồn nhiễu hoạt động tại một tần số nào đó sẽ được xem như đang hiện diện và chiếm tần này, thì có thể loại trừ việc sử dụng tần số này khỏi tuần tự nhảy tần.

Kỹ thuật này đặc biệt hữu dụng đối với nhảy tần chậm vì với nhảy tần nhanh, có nhiều nhảy tần trên một bit số liệu và do đó chỉ một chip sẽ bị ảnh hưởng. Một quyết định vượt trội được dùng sau đó để xác định bit số liệu phù hợp nhất được truyền, 0 hay 1. Tuy nhiên, các hệ thống nhảy tần nhanh có giá thành cao hơn nhiều so với các hệ thống nhảy tần thấp. Cả máy phát và máy thu cũng phải đồng bộ_nghĩa là cũng phải cùng nhảy_các hệ thống nhảy tần chậm dễ đồng bộ hơn. Do đó các hệ thống nhảy tần chậm cung cấp một giải pháp có giá thành hạ cho các LAN không dây.

**Hình 7.39 Trái phổ nhảy tần:**

(a) Nguyên lý hoạt động (b) Nhảy tần số nhanh (c) Nhảy tần số chậm.

Điều chế sóng mang đơn

Với tiếp cận này, một sóng mang được đặt tại trung tâm của băng tần đã gán được điều chế với số liệu truyền sử dụng một mạch điều chế thích hợp. Về nguyên lý, chỉ đơn giản là sự mở rộng của các lược đồ điều chế đã được mô tả trong chương 2 để truyền số liệu qua một mạng điện thoại chuyển mạch analog, ngoại trừ trong các mạng LAN không dây tốc độ bit yêu cầu và do đó là băng thông lớn hơn nhiều.

Nhớ lại rằng có nhiều lược đồ điều chế khác nhau liên quan đến biên độ, tần số và pha hoặc kết hợp giữa chúng. Tuy nhiên, băng thông cao được yêu cầu với

các LAN không dây làm giảm đi tính chống đối các lược đồ điều chế có liên quan đến sự thay đổi biên độ vì các bộ khuếch đại công suất là tuyến tính qua các băng thông rộng cả về giá thành cao và tiêu hao đáng kể công suất. Thông thường, các lược đồ điều chế đều dựa trên sự thay đổi pha của sóng mang có biên độ cố định, ví dụ như điều chế QPSK hay các dẫn xuất của nó. Ngoài ra, như chúng ta đã nói đến từ đầu, đối với các tốc độ bit vượt quá 1 đến 2Mbps, thì hiện tượng tán sắc đa đường làm tăng nhiễu xuyên mẫu (intersymbol interference) và do đó phải sử dụng các mạch cân bằng phức tạp hơn.

Điều chế đa sóng mang

Nguyên lý hoạt động của tiếp cận này trước hết chia tín hiệu nhị phân tốc độ cao thành một số các luồng có tốc độ bit thấp hơn. Sau đó mỗi luồng bit tốc độ thấp được dùng để điều chế lên một sóng mang con riêng_lấy từ băng tần đã được phân phối_theo phương pháp như lược đồ sóng mang đơn. Tuy nhiên, trong trường hợp này, vì tốc độ bit trên một sóng mang tương đối thấp, nên mức nhiễu xuyên mẫu được giảm nhiều, và không cần đến mạch cân bằng. Mặc dù fading có thể xảy ra, rất có khả năng chỉ một (hay một nhóm nhỏ) trong số các sóng mang con bị ảnh hưởng. Các kỹ thuật sửa lỗi hướng tới_ví dụ như các mã tích chập_có thể được dùng để cải thiện tham số BER của kênh. Trong thực tế, các sóng mang con chính là bộ số nguyên lần của sóng mang con đầu tiên $f_1 - 2f_1, 3f_1, \dots$ và do đó lược đồ này cũng còn được gọi là *ghép kênh phân tần trực giao OFDM* (orthogonal frequency division multiplexing).

Trước khi truyền, các sóng mang con được điều chế riêng được tổng hợp lại thành một tín hiệu duy nhất dùng thuật toán biến đổi Fourier nhanh _FFT (Fast Fourier Transform). Điều này tạo một tín hiệu ngoã ra biến thiên trong miền thời gian có băng thông tương tự như băng thông được yêu cầu đối với một lược đồ đơn sóng mang. Tuy nhiên, trong trường hợp này, tại máy thu tín hiệu sẽ được đổi trở lại thành dạng đa sóng mang của nó dùng thuật toán FFT ngược (Bạn đọc có thể tìm hiểu FFT thuận và ngược kỹ hơn trong các tài liệu xử lý số tín hiệu). Các luồng bit tốc độ thấp đã được giải điều chế được kết hợp lại thành một luồng bit ngoã ra có tốc độ cao.

Việc cân đối giữa hai lược đồ điều chế dựa vào giá thành (và nhu cầu công suất) để có được năng lực xử lý thực hiện hoạt động cân bằng (equalization) so với khả năng xử lý cần thiết để thực hiện các thuật toán FFT.

2.3.2. Lược đồ hồng ngoại (infrared)

Có một số phương pháp được dùng để truyền số liệu thông qua tín hiệu hồng ngoại, bao gồm điều chế trực tiếp và điều chế sóng mang.

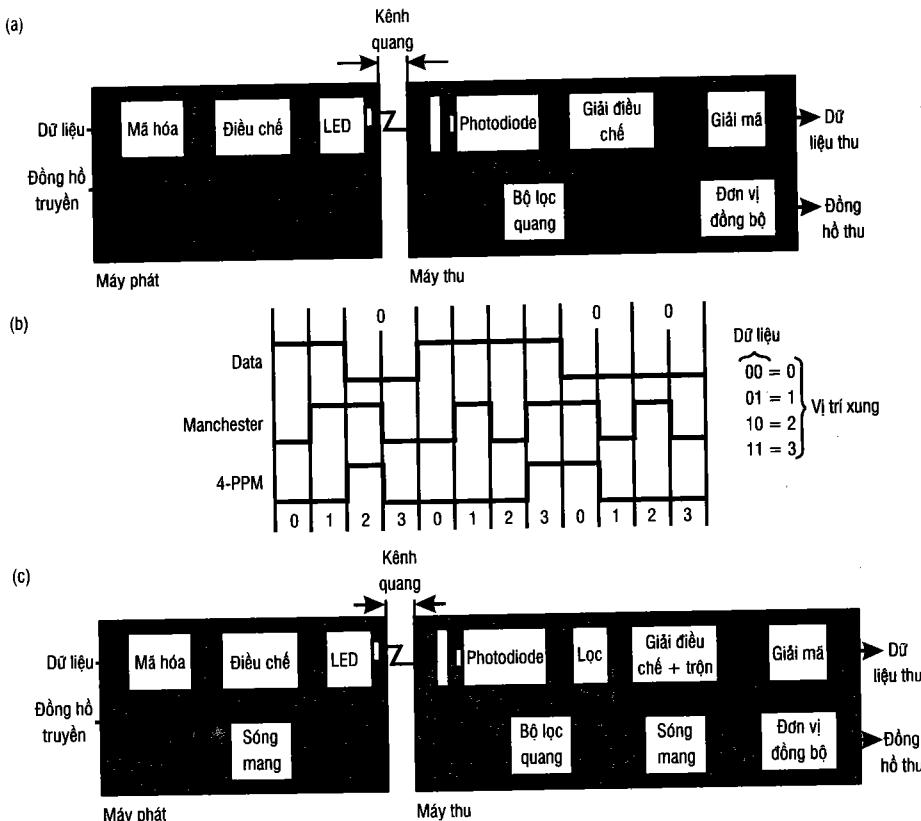
Điều chế trực tiếp

Không giống như sóng radio phải hoạt động trong một băng tần nào đó, sóng hồng ngoại bị ràng buộc một cách tự nhiên vào trong một căn phòng, do đó có thể điều chế tín hiệu hồng ngoại gốc một cách trực tiếp, một nhị phân 1 cho phép bộ phát mở và nhị phân 0 lại đóng bộ phát. Loại điều chế này được gọi là *on-off*



keying (OOK) và được dùng khá rộng rãi trong các hệ thống truyền dẫn sợi quang. Đây là loại điều chế đơn giản nhất và việc thi công mạch điện tử tương đối dễ. Số đồ khồi của lược đồ này được trình bày trên hình 7.40(a).

Cũng giống như truyền dẫn bằng cơ bản qua các liên kết cố định, để máy thu đạt được sự đồng bộ bit thì luồng bit truyền từ gốc phải được mã hóa tại máy phát trước khi được điều chế, dùng một trong các phương pháp mã hóa tín hiệu đồng hồ chuẩn đã được trình bày trong chương 3. Thông thường, dùng mã hóa Manchester hay NRZI kết hợp với chèn bit 0 và một DPLL.



Hình 7.40 Các lược đồ điều chế và mã hóa sóng hồng ngoại: (a) Sơ đồ điều chế trực tiếp (b) Điều chế vị trí xung (c) Sơ đồ điều chế sóng mang.

Ngoài ra, còn có một kỹ thuật khác được gọi là *điều chế vị trí xung* PPM (position-pulse modulation) được dùng trong các hệ thống quang nhằm giảm sự đồi hồi về công suất đối với các LED phát tia hồng ngoại. Nguyên lý hoạt động của PPM được trình bày trên hình 7.40(b). Với lược đồ này luồng bit được truyền trước hết được chia thành các tập gồm các mẫu có n-bit. Trong mỗi mẫu, một xung được

gửi vào một trong 2^n vị trí khe thời gian. Trong ví dụ này, $n = 2$ và do đó một xung được gửi vào một trong 4 khe thời gian. Các hệ thống có bốn bit trên một mẫu _do đó có 16 vị trí xung có thể_ cũng được dùng trong các hệ thống tốc độ cao từ 1-2Mbps. Đó là kích thước mẫu tối đa có thể dùng với các thiết bị hiện hành. Ngoài ra, đối với các tốc độ bit cao hơn thì thường dùng các mạch cân bằng để giảm ánh hưởng của tán sắc đa đường.

Như đã nói đến trong mục 3.2, một bộ lọc quang được dùng để giảm xuyên nhiễu gây ra bởi ánh sáng mặt trời và ánh sáng nhân tạo. Ánh hưởng của bất cứ nhiễu nào còn lại là tạo ra mức tín hiệu đóng photodiode và trong một chừng mực nào đó có thể khiến cho mạch thu quang bỏ sót tín hiệu thu. Đó là một yếu tố khác nữa góp phần tạo nên hạn chế về tốc độ của OOK, chỉ vào khoảng 2Mbps.

Điều chế sóng mang

Để đạt được tốc độ bit cao hơn, chúng ta cần phải dùng các kỹ thuật điều chế sóng mang tương tự như trong các hệ thống radio. Một sơ đồ được trình bày trên hình 7.40(c). Các phương pháp điều chế như FSK và PSK đã được mô tả trong chương 2. Vì với các lược đồ như vậy số liệu nhị phân được truyền dùng một tín hiệu sóng mang được điều chế _theo tần số hay pha_ nên tại máy thu có thể chuyển tín hiệu ngõ ra từ bộ thu hồng ngoại qua một bộ lọc điện tử tăng cường trước khi giải điều chế. Như chỉ ra trong chương 2, các bộ lọc như vậy chỉ cho qua các băng tần giới hạn _chứa số liệu gốc_xung quanh sóng mang. Hiệu quả của việc lọc này còn ở chỗ lọc bỏ các tín hiệu nhiễu còn lại nhờ đó cải thiện hiệu suất qua một hệ thống điều chế trực tiếp. Có thể đạt được tốc độ từ 2 đến 4Mbps một cách dễ dàng.

Với các tốc độ bit cao hơn, nhiễu xuyên mẫu bởi tán sắc đa đường gây ra trở thành một yếu tố hạn chế và do đó các kỹ thuật nhằm khắc phục vấn đề này phải được dùng. Một trong những tiếp cận giải quyết vấn đề này là dùng điều chế đa sóng mang. Với lược đồ này băng thông có sẵn được chia thành một số các băng tần con và mỗi băng tần được dùng để truyền một phần của luồng bit. Ví dụ, nếu dùng hai băng tần con thì mỗi băng tần sẽ luôn phiên truyền các bit trong luồng. Điều này có nghĩa là mỗi băng tần con chỉ cần truyền một nửa tốc độ bit với kết quả là mỗi thời bit được gia tăng gấp đôi so với thời bit gốc. Điều này khiến cho tín hiệu ít nhạy cảm với nhiễu xuyên mẫu và tốc độ bit có thể đạt được lên đến 10Mbps. Rõ ràng việc tăng chu kỳ tín hiệu này sẽ làm tăng tính phức tạp trong các mạch phát và thu.

2.4. Các phương pháp điều khiển truy xuất môi trường

Cá radio và hồng ngoại đều hoạt động với một môi trường quảng bá (broadcast), nghĩa là tất cả các hoạt động truyền đều được nhận bởi tất cả các máy thu nằm trong phạm vi phủ của máy phát. Như vậy, tương tự chúng ta phải cần đến một phương pháp điều khiển truy xuất môi trường (MAC) chia sẻ như trong trường hợp của LAN nối dây_CSMA/CD, token điều khiển, ... nhằm đảm bảo rằng chỉ một máy phát đang sử dụng môi trường. Các lược đồ được dùng chủ yếu là CDMA, CSMA/CD, CSMA/CA, TDMA và FDMA.

2.4.1. CDMA (*Code-Division Multiple Access*)

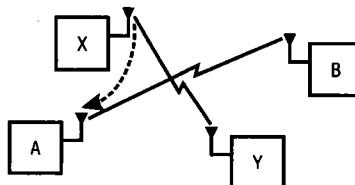
CDMA được gọi là đa truy nhập phân chia mã, đặc biệt được dùng với các hệ thống radio trai phô. Như đã mô tả trong mục 3.3.1, cả hai phương pháp trai phô tuần tự trực tiếp và nhảy tần đều dùng một tuần tự giả ngẫu nhiên duy nhất làm cơ sở cho các chế độ hoạt động của chúng. Do đó, trong các hệ thống như vậy, một tuần tự giả ngẫu nhiên khác nhau có thể được phân phối cho mỗi node và tất cả các node đều biết tập hoàn chỉnh các tuần tự này. Để thông tin với node khác, máy phát chỉ cần chọn và dùng tuần tự giả ngẫu nhiên của nơi muốn truyền số liệu đến (node đối tác). Bằng cách này, nhiều hoạt động truyền giữa một cặp node có thể diễn ra một cách đồng thời.

Trong thực tế, như trên hình 7.41 đã trình bày, điều này chỉ có thể đối với các hệ thống trai phô nhảy tần vì đối với trai phô trực tiếp, sẽ vấp phải một hiện tượng được gọi là *hiệu ứng gần-xa* (near-far effect). Hiện tượng này xảy ra khi một máy phát thứ hai ví dụ node X trong hình đang hoạt động về mặt vật lý thì nó gần với máy thu dự định hơn_node A_so với các node thành viên khác_node B_, mặc dù các hoạt động truyền từ node X đều bị triệt bởi quá trình giải trai phô (despreading process) tại node A, nhưng vì nó gần hơn nên tín hiệu nhiễu từ nó có thể có năng lượng lớn hơn tín hiệu yêu cầu từ node B, nên khiến cho máy thu tại A đánh mất hoạt động truyền đến từ B. Hiện tượng này cũng còn được gọi là *hiệu ứng đầu cuối ẩn* (hidden terminal effect).

Ngược lại, với trai phô nhảy tần vì hai máy phát hoạt động với các kênh có tần số trao đổi không ngừng, nên xác suất để cả hai hoạt động cùng một kênh là rất thấp. Điều này có thể giảm hơn nữa bằng cách hoạch định cẩn thận các tuần tự nhảy. Tuy nhiên, bất tiện của cả hai lược đồ là phải cho tất cả các node biết tuần tự giả ngẫu nhiên của tất cả các node khác, đây là điều khó đối với các nhà quản lý mạng LAN không dây.

2.4.2. CSMA/CD

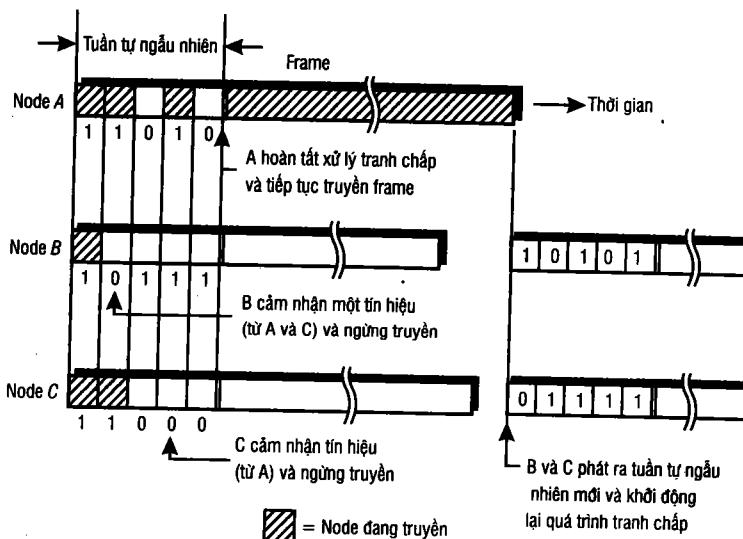
Trong phần đầu của chương đã mô tả cách thức dùng CSMA/CD một cách rộng rãi trong các mạng LAN nối dây như là một phương pháp điều khiển truy xuất môi trường dùng chung. Trong các mạng LAN không dây, CSMA cũng cho phép một node tạm dừng khi có một node khác đang sử dụng môi trường_radio hay hồng ngoại. Tuy nhiên, với sóng radio và hồng ngoại thì không thể truyền và nhận một cách đồng thời và do đó sự phát hiện dung độ ở dạng cơ bản là không thể dùng được ở đây. Tuy vậy, một chức năng phát hiện dung độ khác đã được đưa ra để dùng với LAN không dây và được gọi là *sự phát hiện dung độ*.



Hình 7.41 Hiệu ứng gần-xa trong CDMA.

Với lược đồ này, khi một node đã có một frame để truyền, trước hết nó phát ra một tuần tự nhị phân giả ngẫu nhiên ngắn được gọi là *comb* và gắn vào phía trước của mẫu mở đầu frame. Sau đó node này tiến hành hoạt động cảm nhận sóng mang theo lối thông thường và giả sử môi trường hoàn toàn tĩnh lặng, nó xúc tiến truyền tuần tự *comb* này. Đối với một bit nhị phân 1 trong tuần tự, node truyền một tín hiệu trong một khoảng thời gian ngắn nhưng đối với một bit nhị phân 0 thì node lại chuyển sang chế độ thu. Nếu một node phát hiện hoạt động truyền tín hiệu trong thời gian nó đang ở trong chế độ thu, thì nó sẽ ngưng tranh chấp kênh và đợi cho đến khi node đang truyền khác đã truyền xong. Nguyên lý hoạt động của lược đồ này được trình bày trên hình 7.42.

Trong ví dụ này, ba node_A, B và C đang tranh chấp kênh và các mã giả ngẫu nhiên được phát ra bởi mỗi node như đã trình bày. Vì bit đầu tiên trong tuần tự này là 1 cho tất cả các node, nên không có node nào đang lắng nghe và do đó hoạt động truyền không được phát hiện. Trong khoảng thời gian comb thứ hai, các node A và C vẫn đang truyền nhưng node B ở trong chế độ thu và do đó sẽ phát hiện một tín hiệu và ngừng tranh chấp kênh tại thời điểm này. Trong khoảng thời gian thứ ba, vì node B lúc này đang ở trong trạng thái không hoạt động và cả node A và node C đều ở trong chế độ thu, A và C sẽ không phát hiện một tín hiệu nào, trong khoảng thời gian thứ tư node A đang truyền và node C trong chế độ thu và do đó node C sẽ phát hiện một tín hiệu và ngừng tranh chấp. Sau đó node A được phép truy xuất môi trường, sau khi hoàn thành phần xử lý tranh chấp một cách thành công, tiến hành truyền frame đang đợi của nó.



Hình 7.42 CSMA/CD comb.

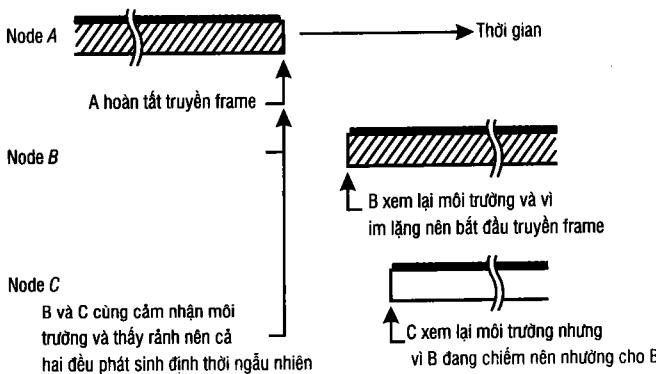
Hiệu quả của lược đồ này được xác định bởi số bit trong tuần tự giả ngẫu nhiên và do đó là *comb*, vì nếu hai node phát ra cùng tuần tự thì một đụng độ sẽ xảy ra. Trong thực tế, số lượng node tranh chấp tại cùng một thời điểm là tương đối thấp, do đó chiều dài của *comb* có thể tương đối ngắn. Cũng vì có giới hạn tối



đa về tốc độ mà các máy thu radio hay hồng ngoại chuyển đổi giữa các chế độ phát và thu thường là một micro giây ($1 \mu\text{s}$)_nên một comb có chiều dài ngắn hơn giảm được khoảng thời gian tranh chấp.

2.4.3. CSMA/CA

Một biến thể của CSMA/CD được gọi là đa truy xuất cảm nhận sóng mang có tránh đụng độ CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) cũng được dùng để điều khiển khả năng truy cập môi trường. Nguyên lý hoạt động của nó được trình bày trên hình 7.43.



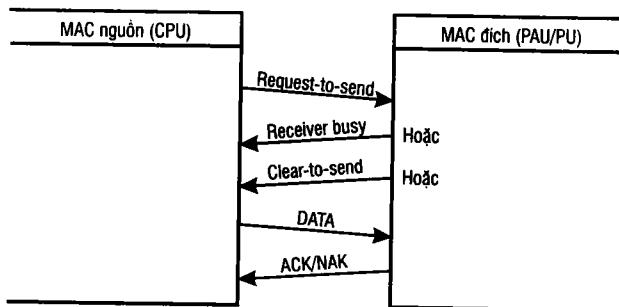
Hình 7.43 Giao thức CSMA/CA

Như chúng ta có thể thấy, thay vì khởi động truyền ngay một frame khi môi trường trở nên yên tĩnh, trước hết node phải đợi thêm một khoảng thời gian ngắn ngẫu nhiên, và chỉ khi môi trường vẫn còn yên tĩnh sau khoảng thời gian này thì nó mới bắt đầu truyền. Bằng cách này, nếu có các node khác cũng đang đợi thì node có khoảng thời gian ngắn nhất sẽ đạt được quyền truy nhập trước tiên và các node còn lại sẽ tạm dừng. Hiệu quả của lược đồ này lại phụ thuộc vào số bit trong tuần tự giả ngẫu nhiên hay lượng thời gian gia tăng trong khoảng thời gian tránh đụng độ lớn nhất.

Một vấn đề khác cũng phải được lưu tâm khi dùng radio (hay hồng ngoại) bởi không có gì chắc chắn rằng máy đang được hướng đến là đang liên lạc radio với node nguồn. Do đó mặc dù CSMA/CA_hay CSMA/CD đảm bảo một node đạt được truy nhập vào môi trường, nhưng máy đích của frame có thể chẳng bao giờ nhận nó bởi nó không liên lạc radio với node nguồn. Do đó, một thủ tục bắt tay qua lại trên phương pháp MAC cơ bản được kết hợp vào trong giao thức MAC này. Vì có chủ định rằng sẽ được dùng với các phương pháp MAC khác nhau nên nó được thiết kế như một giao thức MAC không dây cơ bản được phân tán DFW MAC (Distributed Foundation Wireless MAC). Thủ tục bắt tay bốn bước (4-way handshake procedure) được trình bày trên hình 7.44. Nó có thể được dùng trong cả hai ứng dụng LAN không dây có hạ tầng và không có hạ tầng (ad hoc).

Bất cứ khi nào một đơn vị di động cần gửi một frame, trước hết nó truyền một thông điệp điều khiển ngắn có tên là RTS (request-to-send)_đến PAU hoặc đến đơn vị di động khác_bằng cách dùng một trong các phương pháp MAC vừa mô

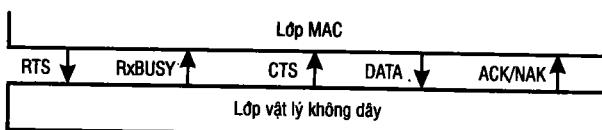
tả CSMA/CD hay CSMA/CA. Thông điệp điều khiển RTS chứa địa chỉ MAC của cả nguồn và đích và máy đích khi tiếp nhận thông điệp này nếu chấp nhận yêu cầu và sẵn sàng tiếp nhận frame, sẽ truyền một thông điệp phúc đáp CTS (clear-to-send) cũng chứa cặp địa chỉ như thông điệp yêu cầu, nhưng theo thứ tự ngược lại. Mặt khác, nếu đích không sẵn sàng nhận một frame, nó sẽ gửi phúc đáp RxBUSY (receiver busy). Nếu phúc đáp là tích cực thì đơn vị di động yêu cầu sẽ truyền frame số liệu và nếu frame này được nhận tốt thì đích sẽ gửi một thông báo nhận tốt (ACK). Tuy nhiên, nếu frame này bị hỏng thì máy thu sẽ gửi một thông báo từ chối (NAK) để máy nguồn cố gắng truyền lại. Thủ tục này sẽ lặp lại theo một số lần xác định. Nhớ lại rằng tất cả các thông điệp điều khiển đều được truyền bằng một phương pháp MAC đặc biệt đã chọn.



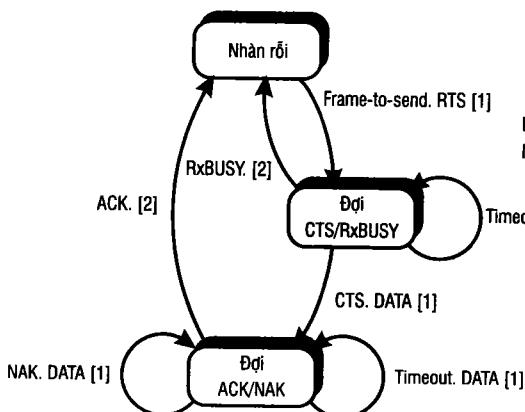
MAC = Medium access control layer

PU = Portable unit

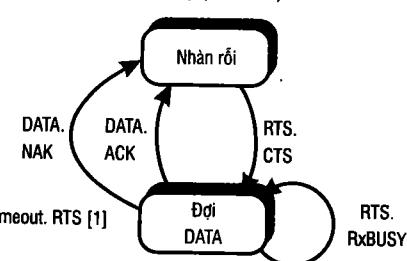
PAU = Portable access unit



Sơ cấp (MAC nguồn)



Thứ cấp (MAC đích)



+/- = Không hỏng/hỏng
[1] = Khởi động định thời
[2] = Dừng bộ định thời

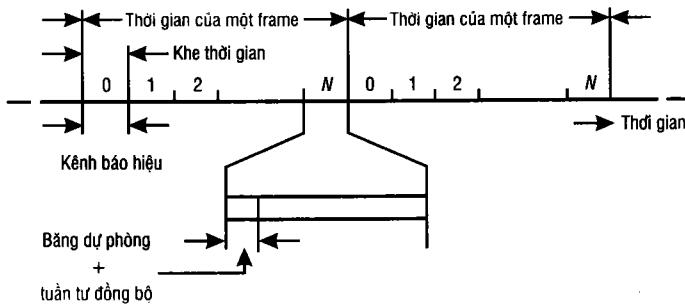
Hình 7.44 Thủ tục bắt tay bốn bước trong giao thức DFW MAC.



Sơ đồ chuyển trạng thái liên quan đến giao thức cũng được trình bày trên hình 7.44. Từ chương 5 chúng ta đã biết, bên cạnh việc định nghĩa hoạt động của giao thức dưới điều kiện không lỗi, cũng phải xác định hoạt động của nó khi các frame hay các thông điệp điều khiển bị hỏng. Hình này trình bày trường hợp thứ hai, một bộ định thời được dùng để khởi động truyền lại các thông điệp bị hỏng. Thông thường, một số giới hạn các biến cố xác định sẽ được dùng trong mỗi trường hợp.

2.4.4. TDMA

Nguyên lý hoạt động của đa truy xuất phân chia thời gian TDMA (Time-Division Multiple Access) trong ngữ cảnh của các mạng LAN không dây được trình bày trên hình 7.45. Theo phương pháp này, mỗi máy phát (node) có một khe thời gian nhất định, một khe thời gian đến, máy phát truyền với tất cả băng thông trong khoảng thời gian của khe này. Thông thường khoảng thời gian của mỗi khe là ngắn và được chọn sao cho xác suất xảy ra lỗi là rất thấp. Khoảng thời gian của frame được xác định bởi khoảng thời gian của mỗi khe và số khe hỗ trợ.



Hình 7.45 TDMA

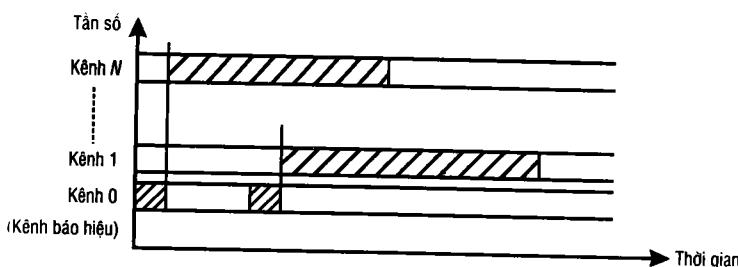
Thông thường TDMA được dùng khi có một trạm đàm trách tất cả các hoạt động truyền xảy ra. Ví dụ, trong một ứng dụng thay cho mạng nối dây cố định, một PAU_xem hình 7.33_hoạt động như một trạm cơ bản và nó thiết lập kiến trúc khe thời gian. Mỗi đầu cuối di động trong một vùng phủ của một trạm cơ bản được phân phối một khe thời gian nhất định hay thông dụng hơn là có một khe thời gian riêng (báo hiệu) được cung cấp nhằm cho phép mỗi thiết bị di động gửi yêu cầu cấp khe thời gian vào trạm cơ bản bất cứ khi nào nó có số liệu muốn truyền. Các hoạt động truyền từ trạm cơ bản đến các thiết bị di động diễn ra theo chế độ quảng bá (broadcast mode) bằng cách dùng một khe thời gian đặc biệt_với địa chỉ của đích được đặt ngay đầu của frame được truyền hoặc hoạt động truyền diễn ra trên một khe thời gian xác định được thiết lập bằng cách dùng kênh báo hiệu. Chế độ hoạt động này cũng được gọi là Aloha phán khe và gán theo yêu cầu (slotted Aloha with demand assignment). Còn có một chế độ hoạt động khác, trong

đó việc sử dụng mỗi khe có thể được điều khiển bởi một khe con làm nhiệm vụ báo hiệu riêng bên trong.

Như trình bày trên hình 7.45, có một băng bảo đảm (guard band) và một tuần tự đồng bộ (sync sequence) tại đầu của mỗi khe thời gian. Băng bảo đảm cho phép các khoảng thời gian trê lan truyền khác nhau giữa các đầu cuối di động phân tán và trạm cơ bản, trong khi tuần tự đồng bộ cho phép máy thu_di động hay trạm cố định_bắt nhịp được với máy phát trước khi tiếp nhận nội dung của khe.

2.4.5. FDMA

Nguyên lý hoạt động của đa truy xuất phân chia tần số FDMA (Frequency-Division Multiple Access) được trình bày trên hình 7.46. FDMA được dùng chủ yếu trong các hệ thống radio và giống như TDMA, nó cần một trạm cơ bản để điều khiển hoạt động của nó. Khi dùng FDMA thì tổng băng thông được phân phối sẽ được chia thành một số các băng tần con hay kênh giống như nguyên lý của trại phổ nhảy tần. Tuy nhiên, với FDMA, một khung đã được gán kênh tần số đặc biệt được dùng trong toàn bộ thời gian của hoạt động truyền frame. Thông thường, các kênh tần số được gán theo yêu cầu bằng cách dùng một kênh báo hiệu riêng.



Hình 7.46 FDMA

Nhìn chung, trạm cơ bản trong một hệ thống FDMA phức tạp hơn nhiều so với trạm cơ bản được dùng trong hệ thống TDMA, do đó hệ thống TDMA được dùng rộng rãi hơn. Các lược đồ phối hợp cũng được dùng, trong đó dùng FDMA để có được nhiều kênh và mỗi kênh lại được dùng với TDMA.

2.4.6. Chức năng bổ sung

Chúng ta sẽ thấy trong mục 4 khi thảo luận về các giao thức LAN, mặc dù có nhiều giao thức MAC khác nhau đang được dùng_cả cho LAN nối dây và LAN không dây_nhưng tất cả đều có một điểm chung là lớp MAC cung cấp một tập các hàm thực thể dịch vụ chuẩn cho lớp ở trên. Trong trường hợp của các LAN không dây, điều này có nghĩa là lớp MAC, bên cạnh việc thực hiện chức năng MAC còn thực hiện thêm một số chức năng khác nữa. Các chức năng này bao gồm phân mảnh, điều khiển luồng và kiểm soát đa tốc độ.



Sự phân mảnh là cần thiết bởi chỉ số BER khá cao đối với radio và hồng ngoại. Kích thước frame lớn dùng trong các LAN nội dây cố định là có thể bởi BER của đường truyền_cáp xoắn đôi, cáp đồng trực, cáp quang_ thường rất thấp, các giá trị trong khoảng 10^{-9} đến 10^{-11} . Ngược lại, ảnh hưởng của tán sắc và các nhiễu khác đối với sóng radio và hồng ngoại có thể làm tăng chỉ số này khá lớn, các giá trị tiêu biểu nằm trong khoảng 10^{-3} đến 10^{-5} . Điều này cũng có nghĩa là phải dùng các frame có kích thước nhỏ khi truyền qua các môi trường này. Nếu lớp MAC cung cấp một dịch vụ tương tự đã cung cấp trong mạng nội dây cố định, thì lớp MAC phải phân đoạn mỗi frame được nạp vào thành nhiều frame con nhỏ hơn để truyền qua môi trường không dây. Tương tự, khi tiếp nhận mỗi frame, nó phải được tái thiết lập trở lại dạng ban đầu trước khi phân phối cho các lớp cao hơn.

Có thể phối hợp với một lược đồ kiểm soát lỗi vào trong giao thức DFW MAC. Tuy nhiên, nếu chức năng kiểm soát lỗi không được thực thi bởi lớp MAC thì bất kỳ frame nào mà có lỗi đều bị loại bỏ. Tuy vậy, để lớp MAC thực hiện thêm chức năng điều khiển luồng luôn là điều cần thiết. Khi mỗi mảnh được chuyển cho lớp vật lý, lớp MAC phải đợi cho đến khi mảnh này đã được truyền xong mới nạp mảnh kế tiếp. Thông thường, luồng frame giữa hai lớp được điều khiển bởi các đường điều khiển đã được trình bày trên hình 7.44 phía trước.

Chức năng kiểm soát đa tốc độ cũng cần thiết bởi thông thường lớp vật lý có thể hoạt động theo một số tốc độ xác định. Ví dụ, đối với trai phổ tuân tự trực tiếp thì có thể dùng các tốc độ 1 Mbps và 2Mbps trong khi đó đối với hồng ngoại thì tốc độ này có thể là 1,2,4 hay 10Mbps. Thông thường, tốc độ hoạt động được chỉ định bởi lớp MAC trong một tham số liên kết với các hàm thực thể dịch vụ lớp vật lý. Tốc độ được xác định bởi chất lượng dịch vụ đang được cung cấp bởi lớp vật lý, ví dụ nếu tỉ lệ frame bị hỏng cao thì tốc độ bit được chọn sẽ thấp đi, ngược lại nếu không có lỗi thì tốc độ bit có thể tăng lên. Dĩ nhiên, lớp vật lý ở đầu thu phải hoạt động cùng một tốc độ _và cùng phương pháp điều chế_ như lớp vật lý ở máy phát và do đó điều này phải được thống nhất bởi cả hai đối tác thông tin nếu như được trao đổi. Thông thường, điều này được kiến tạo bằng cách trao đổi các tham số điều khiển thêm vào trong các frame RTS và CTS. Hoạt động trao đổi này được thực hiện với tốc độ thấp nhất và chỉ khi máy thu đáp ứng một thông báo tích cực thì máy phát mới tăng lên một tốc độ cao hơn. Thường thì tốc độ hoạt động hiện hành với nhiều đích khác nhau được lưu giữ trong một bảng để tránh đàm phán lại tốc độ trước mỗi lần truyền.

2.5. Các chuẩn

Hiện có một số loại LAN không dây đang được triển khai. Tuy nhiên, tất cả chúng đều được phát triển từ một công ty nào đó và chúng khác nhau đáng kể theo từng công ty. Nhu cầu chuẩn hóa quốc tế ngày nay được công nhận rộng khắp và hiện nay có hai tiêu chuẩn cho các LAN không dây. Tại Mỹ, chuẩn đang được phát triển dưới sự bảo trợ của IEEE và được gọi là IEEE 802.11. Còn ở Châu Âu, chuẩn đang được phát triển bởi Viện Tiêu Chuẩn Viễn Thông Châu Âu gọi tắt là ETSI và được gọi là HiperLAN. Cá hai chuẩn dùng nhiều đặc tính đã được mô tả trong các phần trước.

Cũng giống như các LAN nối dây, không chỉ có một chuẩn duy nhất, ví dụ IEEE 802.11 cho phép một loạt các tiêu chuẩn lớp vật lý khác nhau dựa vào hai loại đường truyền. Chúng bao gồm:

- 1 và 2 Mbps dùng sóng radio trai phổ nhảy tần.
- 1 và 2Mbps dùng sóng radio trai phổ tuần tự trực tiếp.
- 1 và 2Mbps dùng sóng hồng ngoại điều chế trực tiếp.
- 4Mbps dùng sóng hồng ngoại điều chế đơn sóng mang.
- 10Mbps dùng sóng hồng ngoại điều chế đa sóng mang.

Chuẩn HiperLAN có xu hướng áp dụng cho cả hai dạng ứng dụng LAN không dây có hạ tầng và không có hạ tầng (ad hoc). Một vài tham số hoạt động vẫn còn đang trong tiến trình hoàn chỉnh, tuy nhiên đặc tả hiện nay là:

- Tốc độ bit của user là 10-20Mbps
- Phạm vi hoạt động 50m
- Đường truyền là sóng radio
- Điều chế đơn sóng mang dùng phiên bản có sửa đổi của QPSK được gọi là offset QPSK và một mạch cân bằng (equalizer).
- Phương pháp điều khiển truy xuất môi trường là CSMA/CD hoặc CSMA/CA.

Để phục vụ cho các loại điều chế và đường truyền khác nhau, lớp vật lý bao gồm hai lớp con: lớp con hội tụ lớp vật lý PLC (Physical Layer Convergence) và lớp con phụ thuộc đường truyền vật lý PMD (Physical Medium Dependent). Lớp con PMD của các loại đường truyền và điều chế khác nhau là khác nhau và các dịch vụ mà nó cung cấp được xác định bởi các loại đường truyền và phương pháp điều chế tương ứng. Lớp con PLC thực hiện các chức năng hội tụ cần thiết để ánh xạ các dịch vụ chuẩn được cung cấp tại giao tiếp lớp vật lý với các dịch vụ được cung cấp bởi lớp con PMD đặc biệt đang được dùng.

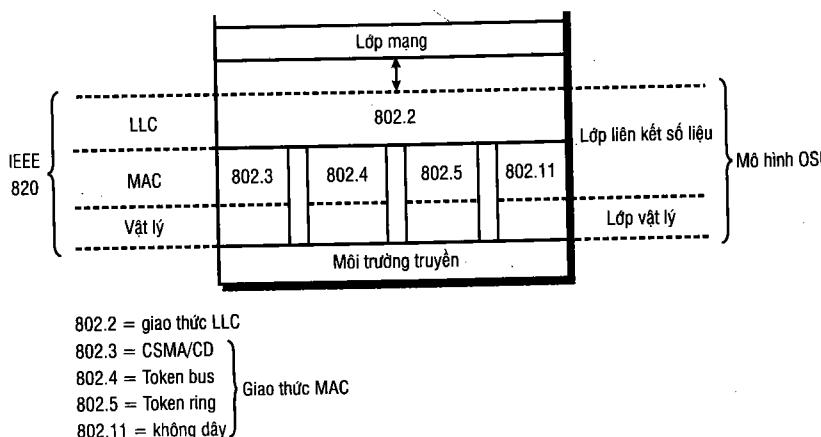
3. CÁC GIAO THỨC

Các tiêu chuẩn giao thức khác nhau cho các LAN liên quan đến các lớp vật lý và lớp liên kết số liệu trong ngữ cảnh của mô hình tham chiếu của ISO, là những gì được định nghĩa trong IEEE 802. Chuẩn này định nghĩa một họ của các giao thức, mỗi giao thức liên quan đến một phương pháp MAC đặc biệt. Các chuẩn IEEE khác nhau và các mối liên hệ mật thiết của chúng với mô hình tham chiếu của ISO được trình bày trên hình 7.47.

Có ba chuẩn MAC cùng với các đặc tả đường truyền vật lý liên kết với chúng là nội dung trong các tài liệu tiêu chuẩn của IEEE sau:

- IEEE 802.3: CSMA/CD bus
- IEEE 802.4: Token bus
- IEEE 802.5: Token ring
- IEEE 802.11: Wireless

Các tiêu chuẩn ISO liên quan đều tương tự ngoại trừ thêm một số 8 vào đầu của mỗi tên: ví dụ ISO 8802.3, .v.v.



Hình 7.47 Tập giao thức IEEE 802.

Các mô tả được trình bày cho đến lúc này đều liên quan đến các lớp vật lý và MAC của bốn chuẩn này. Mặc dù hoạt động bên trong các chuẩn là khác nhau, nhưng tất cả chúng đều đưa ra một tập dịch vụ chuẩn đối với lớp điều khiển liên kết luận lý (LLC), chúng có khuynh hướng được dùng để tiếp hợp với bất cứ chuẩn MAC nào ở bên dưới. Nhìn chung, như đã đề cập đến trong mục 2, các lớp vật lý cũng như MAC khác nhau thường được hiện thực dưới dạng phần mềm trong các mạch tích hợp có chức năng đặc biệt. Do đó, trong phần này chúng ta chỉ tập trung vào lớp LLC và lớp mạng và định nghĩa một cách đơn giản giao tiếp giữa các lớp LLC và MAC. Lưu ý rằng đối với một LAN thì các lớp MAC, LLC và lớp mạng là các giao thức ngang hàng (đầu cuối đến đầu cuối) vì không có các node chuyển mạch trung gian giống như trong các mạng số liệu công cộng (có tổng đài chuyển mạch gói).

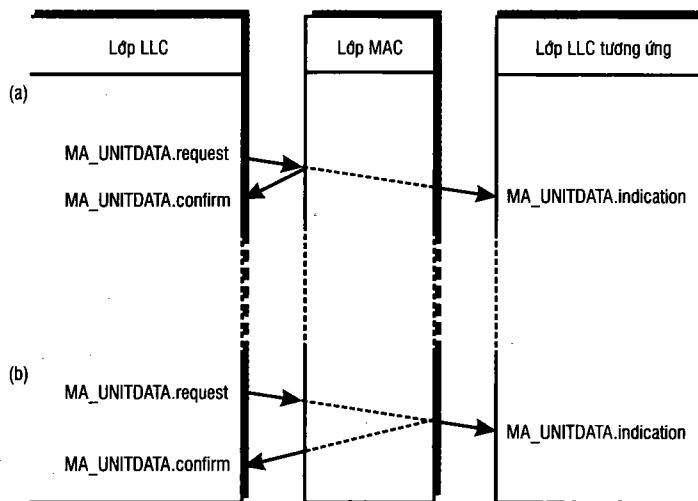
Như chúng ta có thể thấy trong hình 7.47, trong ngữ cảnh của mô hình tham chiếu của ISO, các lớp LLC và MAC hợp với nhau thành chức năng của lớp liên kết số liệu (lớp 2). Trong ngữ cảnh này các lớp MAC và LLC được xem là các lớp con. Ở chương trước chúng ta đã đề cập đến các chức năng của lớp liên kết số liệu bao gồm định dạng frame (dánh dấu bắt đầu và kết thúc một frame) và phát hiện lỗi. Ngoài ra để cung cấp dịch vụ tin cậy còn có điều khiển luồng, kiểm soát lỗi và quản lý liên kết. Do đó lớp MAC thực hiện định dạng frame và phát hiện lỗi cùng với hoạt động điều khiển truy xuất môi trường trong khi đó lớp LLC thực hiện các chức năng còn lại.

3.1. Các dịch vụ lớp MAC

Bất chấp chế độ hoạt động của lớp con MAC bên dưới là gì đi nữa_CSMA/CD, token ring, token bus, không dây đều có một tập các dịch vụ user chuẩn được định nghĩa cung cấp cho lớp LLC giúp lớp này chuyển các đơn vị số liệu giao thức của nó (LLC PDU) đi đến lớp tương ứng. Các hàm thực thể dịch vụ user được hỗ trợ bao gồm:

- MA_UNITDATA.request
- MA_UNITDATA.indication
- MA_UNITDATA.confirm

Một sơ đồ tuần tự theo thời gian minh họa việc sử dụng các hàm này được trình bày trên hình 7.48. Trong một LAN dùng CSMA/CD, hàm xác nhận chỉ ra rằng yêu cầu đã được *truyền* thành công (hay thất bại) trong khi đó đối với một LAN dùng token thì lại chỉ ra rằng yêu cầu đã được *phân phối* thành công (hay thất bại).



Hình 7.48 Các hàm thực thể dịch vụ user MAC:

(a) CSMA/CD (b) Token ring/Bus.

Mỗi hàm thực thể dịch vụ đều có các tham số liên hệ. Bao gồm trong hàm thực thể MA_UNITDATA.request là địa chỉ đích được yêu cầu (địa chỉ này có thể là địa chỉ của cá nhân, nhóm hay quảng bá), một đơn vị số liệu dịch vụ (chứa số liệu được truyền đó là LLC PDU), và một lớp dịch vụ được yêu cầu liên hệ với PDU này. Thành phần sau cùng được dùng với các mạng token ring và token bus, ví dụ khi dùng một giao thức MAC có ưu tiên.

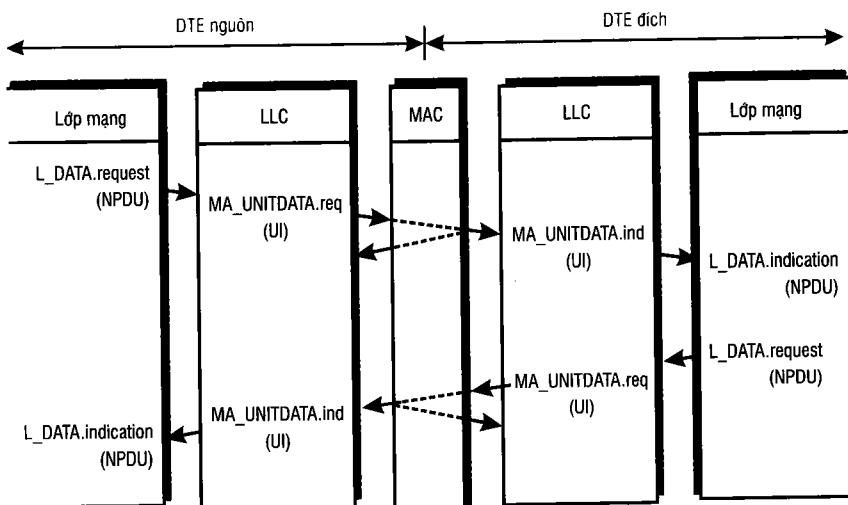
Hàm thực thể dịch vụ MA_UNITDATA.confirm bao gồm một tham số chỉ ra sự thành công hay thất bại của hàm thực thể MA_UNITDATA.request. Tuy nhiên, hình 7.48 cho thấy hàm xác nhận không được phát ra như là kết quả của một đáp ứng từ lớp con LLC ở xa, thay vì vậy là của thực thể MAC cục bộ. Nếu tham số này bảo là thành công, thì điều này chỉ đơn giản là thực thể giao thức MAC đã



thành công trong việc truyền đơn vị số liệu dịch vụ vào môi trường mạng. Nếu không thành công, tham số này chỉ ra tại làm sao hoạt động truyền thất bại. Ví dụ nếu mạng là CSMA/CD bus thì ‘độ quá mức’ là một trong các tham số chỉ ra sự thất bại.

3.2. Lớp LLC

Các dịch vụ user và hoạt động của lớp con LLC đã được thảo luận trong chương 6 khi mô tả về các giao thức điều khiển liên kết số liệu. Nhớ rằng giao thức LLC dựa trên giao thức HDLC, và hai loại dịch vụ user và giao thức liên quan được hỗ trợ là : không tạo cầu nối (connectionless) và có tạo cầu nối (connection-oriented). Tuy nhiên trong hầu hết các triển khai LAN, đặc biệt là trong môi trường kỹ thuật và văn phòng thì chỉ có giao thức không tạo cầu nối truyền số liệu mà không có báo nhận thường được dùng. Do đó chỉ có hàm thực thể dịch vụ user L_DATA.request, và bởi là giao thức tổng lực (best-try) nên tất cả các số liệu đều được chuyển dưới dạng các frame thông tin không đánh số (UI). Giao tác giữa các lớp con LLC và MAC được trình bày trên hình 7.49.

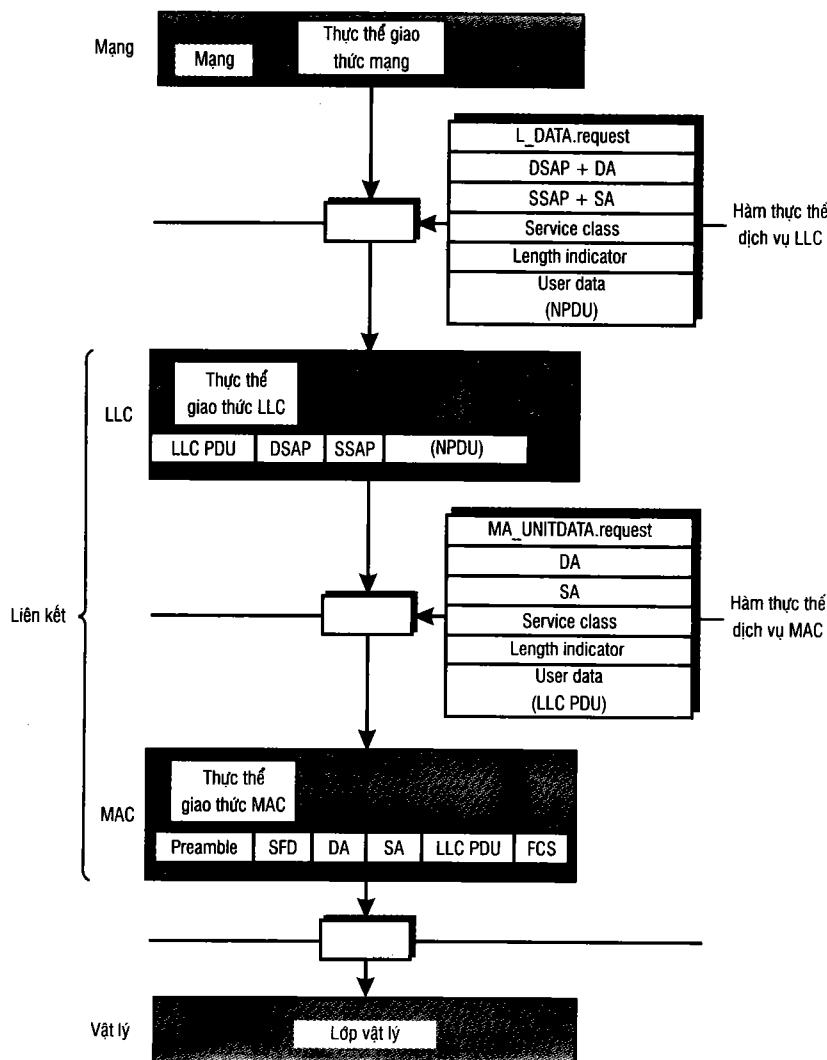


Hình 7.49 Các giao tác giữa MAC và LLC.

Hàm thực thể L_DATA.request luôn có các tham số đi kèm. Các tham số này là một đặc tả của các địa chỉ nguồn và đích và số liệu của user. Số liệu của user chính là đơn vị số liệu giao thức lớp mạng NPDU (network protocol data unit). Các địa chỉ nguồn và đích đều gắn liền với địa chỉ MAC của DTE và một địa chỉ liên kết lớp hay gọi là điểm truy xuất dịch vụ (SAP: service access point)_LLC SAP. Địa chỉ này được dùng cho định tuyến trên giao tiếp lớp bên trong một DTE.

Một minh họa chi tiết hơn về các giao tác giữa các lớp LLC và MAC được trình bày trên hình 7.50. Lớp LLC đọc các địa chỉ điểm truy xuất dịch vụ LLC nguồn và đích (DSAP và SSAP) từ hai tham số địa chỉ liên kết với hàm thực thể dịch vụ L_DATA.request và chèn chúng vào đầu của LLC PDU. Sau đó nó gắn đơn

vị số liệu giao thức lớp mạng (NPDU) vào để hình thành một LLC PDU hoàn chỉnh và chuyển LLC PDU này cho lớp MAC như là tham số số liệu user của một hàm thực thể MA_UNITDATA.request của lớp MAC. Các tham số khác liên quan đến hàm thực thể này bao gồm các địa chỉ MAC nguồn và đích (DA và SA), lớp dịch vụ cần, và số lượng octet trong field số liệu của user. Thông thường, lớp dịch vụ được dùng bởi thực thể giao thức lớp MAC để xác định mức ưu tiên liên hệ với frame nếu đang dùng mạng token ring.



Hình 7.50 Các tham số và các hàm thực thể liên kết lớp.

Khi tiếp nhận yêu cầu này, thực thể giao thức MAC tạo ra một frame để truyền lên liên kết. Trong trường hợp mạng CSMA/CD bus, nó tạo ra một frame chứa các field mở đầu và SFD, các field DA và SA, một I-field và một field FCS thích hợp. Frame hoàn chỉnh này được truyền nối tiếp (bit) vào môi trường cáp bằng cách dùng một phương pháp MAC phù hợp.

Tại DTE đích một thủ tục tương tự được nối tiếp, chỉ khác là các field tương ứng trong mỗi PDU được đọc ra và được biên dịch bởi mỗi lớp. Sau đó field số liệu user trong mỗi PDU được chuyển lên cho lớp kế trên cùng với các tham số địa chỉ thích hợp.

3.3. Lớp mạng

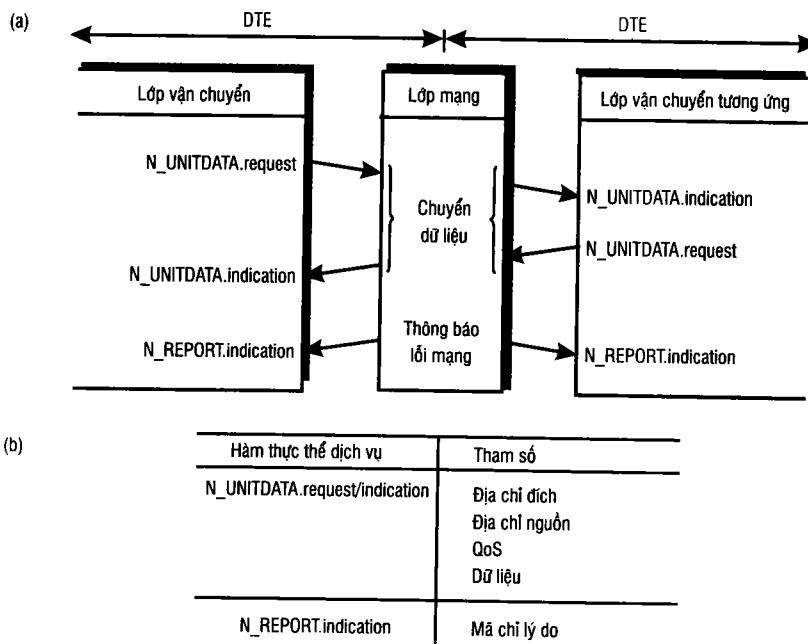
Vai trò chủ yếu của lớp mạng là định tuyến các thông điệp liên quan đến các lớp giao thức cao hơn trong ngữ cảnh của mô hình tham chiếu của ISO_xuyên qua một hay nhiều mạng nối liền một nhóm các DTE với nhau. Tương tự như lớp liên kết số liệu, lớp mạng có thể hoạt động trong chế độ không tạo cầu nối hoặc trong chế độ có tạo cầu nối. Trong trường hợp của các mạng LAN, các frame được hướng đến và được định tuyến giữa các DTE kết nối vào cùng một LAN dùng các địa chỉ MAC. Hơn thế nữa, vì các LAN dùng đường truyền tốc độ cao có BER rất thấp, nên thời gian trễ từ DTE đến DTE liên hệ với mỗi thông điệp và xác suất thông điệp bị hỏng đều rất thấp. Chính vì vậy, một dịch vụ lớp mạng không tạo cầu nối và giao thức liên quan thường được dùng khi tất cả các DTE được nối vào cùng một LAN. Bất kỳ thủ tục kiểm soát lỗi và điều khiển luồng cần thiết nào đều được để lại cho giao thức của lớp vận chuyển ở trên.

Bởi thiếu chức năng trong các LAN nên lớp mạng thường được gọi là *lớp i hay null*. Các hàm thực thể dịch vụ user liên quan đến dịch vụ mạng và các tham số của nó được trình bày trong hình 7.51.

Dịch vụ chuyển thông điệp cơ bản là N_UNITDATA (request hay indication) là một dịch vụ tổng lực (best-try service). Các tham số DA và SA liên hệ với hàm này là gắn liền với địa chỉ MAC của DTE (nguồn và đích) và địa chỉ mở rộng LLC SAP. Một địa chỉ điểm truy xuất dịch vụ mạng (NSAP) cũng được dùng. Vai trò của nó tương tự như LLC SAP và cho phép các thông điệp được định tuyến xuyên qua các lớp giao thức khác nhau đến các AP khác nhau trong cùng một DTE. Ví dụ một DTE làm chức năng server trong mạng hỗ trợ nhiều ứng dụng như truyền thư điện tử và truyền tập tin.

Nhìn chung, tham số QoS (chất lượng dịch vụ) bao gồm các field cho phép chỉ định thời gian trễ, mức ưu tiên, và các tham số mạng khác. Trong trường hợp có một LAN, chỉ có field ưu tiên là có ý nghĩa nào đó. Sau cùng, tham số số liệu user chỉ ra số liệu của bản tin được truyền.

Hàm thực thể N_REPORT.indication được dùng bởi nhà cung cấp mạng_lớp con LLC và MAC để thông báo bất kỳ điều kiện gây lỗi nào có thể xảy ra liên quan đến yêu cầu truyền. Trong trường hợp của các LAN, ví dụ như 'dụng độ quá mức' nếu dùng CSMA/CD.



Hình 7.51 Các dịch vụ lớp mạng:

(a) Sơ đồ tuần tự theo thời gian (b) Các tham số dịch vụ.

Do đó, chúng ta có thể kết luận rằng giao thức liên quan rất ít đến lớp mạng. Nó liên quan đến việc tạo ra một NPDU từ các tham số trong hàm N_UNITDATA.request và chuyển NPDU này cho lớp con LLC dưới dạng là tham số số liệu của user của một L_DATA.request. Tương tự, khi nhận một NPDU từ lớp LLC, chứa trong tham số số liệu của user trong một L_DATA.indication, giao thức này lấy các địa chỉ mạng nguồn và đích từ NPDU và chuyển chúng cùng với phần số liệu user còn lại đến lớp vận chuyển (lớp 4) thông qua hàm thực thể N_UNITDATA.indication.

Tóm lại, nếu mạng gồm một số các mạng liên kết lại với nhau thay vì chỉ một mạng LAN đơn, thì giao thức lớp mạng sẽ phức tạp hơn. Mạng toàn cục được gọi là *liên mạng* hay *internet* và các mạng thành phần được xem như các *mạng con*.



SÁCH ĐÃ XUẤT BẢN

LẬP TRÌNH :

1. Bài tập ngôn ngữ C từ A đến Z
2. Giáo trình lý thuyết và bài tập ngôn ngữ C [tập 1 & 2]
3. Lập trình Windows (Bằng Visual C++)
4. Thiết kế đồ họa định hướng đối tượng với C++
5. Giáo trình lý thuyết và bài tập Pascal [tập 1 & 2]
6. Giáo trình lý thuyết và bài tập Foxpro [tập 1]
7. Sử dụng và khai thác Visual Foxpro 6.0
8. Access 2000 lập trình ứng dụng cơ sở dữ liệu [tập 1 & 2]
9. Giáo trình lý thuyết và bài tập Oracle
10. Giáo trình lý thuyết và bài tập Java
11. Java lập trình mạng
12. Giáo trình lý thuyết và bài tập visual J++6
13. Tự học lập trình cơ sở dữ liệu với VB6 trong 21 ngày [tập 1 & 2]
14. Visual Basic 6.0 – Lập trình cơ sở dữ liệu
15. Các kỹ xảo lập trình với VB6 và Delphi
16. Giáo trình lý thuyết và bài tập Delphi
17. Hợp ngữ và lập trình ứng dụng [tập 1 & 2]
18. Đồ họa vi tính [tập 1 & 2]
19. Giáo trình trí tuệ nhân tạo – Mạng Nơron phương pháp và ứng dụng
20. Giáo trình trí tuệ nhân tạo – Cấu trúc dữ liệu + Thuật giải di truyền = Lập trình tiến hóa
21. Tự học lập trình bằng ví dụ với VISUAL C++ MFC
22. Lập trình ứng dụng chuyên nghiệp SQL Server 2000 (tập 1)

INTERNET & VIỄN THÔNG

23. Internet Explorer toàn tập
24. Internet working với TCP/IP [tập 1&2]
25. Thực hành thiết kế trang web với Frontpage 2000
26. Frontpage 2000 toàn tập
27. Thiết kế hoạt hình cho web với Macro Media Flash

28. Hướng dẫn thiết kế trang web tương tác bằng Java Script
29. Sử dụng E-mail và tin học văn phòng trên mạng với Outlook 2000
30. Modem truyền số liệu
31. Cơ sở kỹ thuật chuyển mạch và tổng đài [tập 1 & 2]
32. XML – Nền tảng và ứng dụng
33. Xây dựng ứng dụng web với JSP, servlet, Javabeans.
34. ASP 3.0 / ASP.NET
35. Thiết kế trang WEB động với DHTML.
36. Lập trình ứng dụng WEB với JSP servlet
37. Kỹ thuật truyền số liệu

THIẾT KẾ ĐỒ HỌA

38. Vẽ minh họa với Corel Draw 9.0
39. Vẽ minh họa với Corel Draw 10.0 [tập 1 & 2]
40. Autocad 2000 [tập 1 & 2]
41. Thiết kế 3 chiều với 3D Studio Max 3.0
42. Adobe Photoshop 5.5 và Imageready 2.0
43. Adobe Photoshop 6.0 và Imageready 3.0
44. Adobe Illustration 8.0
45. Adobe Illustration với các kỹ thuật thiết kế nâng cao
46. Adobe Indesign
47. Dàn trang với Quarkxpress

HỆ ĐIỀU HÀNH VÀ MẠNG

48. Vân hành và khai thác Windows 98
49. Làm chủ Windows 2000 Server [tập 1 & 2]
50. Giáo trình mạng Novell Netware 5.0
51. Giáo trình cấu trúc máy tính
52. Vi mạch và mạch tạo sóng
53. Họ vi xử lý 8051
54. Tìm hiểu cấu trúc và hướng dẫn sửa chữa, bảo trì máy PC [tập 1 & 2]
55. Giáo trình hệ thống mạng máy tính CCNA Semester I

VĂN PHÒNG

56. Microsoft Word 2000
57. Đồ họa và Multimedia trong văn phòng với MS Power Point 2000

KỸ THUẬT TRUYỀN SỐ LIỆU

NHÀ XUẤT BẢN LAO ĐỘNG – XÃ HỘI

41B Lý Thái Tổ – Hà Nội

Tel: 8.241706 – Fax: 9.348283

* * *

Chịu trách nhiệm xuất bản :

NGUYỄN ĐÌNH THIÊM

Chịu trách nhiệm nội dung:

NGUYỄN BÁ NGỌC

Biên tập :

LÊ THANH DUY

Sửa bản in :

NGỌC AN

Trình bày bìa :

NGUYỄN VIỆT DŨNG

Tổng phát hành tại **NHÀ SÁCH MINH KHAI**

249 Nguyễn Thị Minh Khai – Quận 1

TP. Hồ Chí Minh

ĐT:(08) 8331124 – (08) 9250590 – Fax:(08) 8331124

E-mail(CLBbandoc) : mk.pub @cinet.vnnews.com

E-mail(thuongmai) : mk.book @cinet. vnnews.com

Web site : www.minhkhai.com.vn

In 1.000 cuốn khổ 16 x 24 (cm) tại Xí nghiệp In MACHINCO – 21 Bùi Thị Xuân , Q.1.
Giấy CNDKKH xuất bản : 148/XB-QLXB của Cục Xuất Bản cấp ngày 06/02/2002.
In xong và nộp lưu chiểu tháng 03/2002.