

Chương III

TRUYỀN SỐ LIỆU

3.1 Khái quát

Thông tin sau khi mã hóa được biểu diễn trong dạng nhị phân và lưu trữ trong những thiết bị đầu cuối. Những thiết bị đầu cuối có thể là máy tính hoặc những terminal thông minh.

Sự trao đổi dữ liệu giữa A và B được thiết lập theo mạch như hình vẽ 3.1. Nó bao gồm thiết bị đầu cuối nguồn (ETTD), modem (ETCD) đường dây truyền và thiết bị đầu cuối thu.

Trong thiết bị đầu cuối xử lý dữ liệu chúng ta chia chúng làm 2 phần thực hiện chức năng khác nhau: phần xử lý thông tin và phần kiểm tra sự liên lạc. Phần đặc biệt được thực hiện ở đây là sự bảo vệ chống sai số và tạo ra các ký tự, phục vụ cho sự đối thoại giữa hai thiết bị đầu cuối. Bộ phận kiểm tra liên lạc có thể có hoặc không có bộ phận lọc.

Hình 3.1 Mạch truyền dữ liệu từ A - B.

Thiết bị đầu cuối mạch dữ liệu (ETCD) là thiết bị có nhiệm vụ đáp ứng tín hiệu điện từ thiết bị đầu cuối để truyền đi. Chức năng đó được thực hiện bằng cách tạo tín hiệu nhiều mức (để truyền trong khoảng cách không xa lắm) hoặc điều chế hoặc giải điều chế (modem). Thiết bị đầu cuối mạch dữ liệu còn có nhiệm vụ thiết lập và giải phóng mạch.

Sau đây chúng ta sẽ khảo sát kỹ hơn phần ETCD phát, ETCD thu.

3.1.1 Đường truyền

Một đường dây truyền cho phép nối về vật lý 2 địa điểm mà ở đó đặt các terminal. Chúng ta đã gặp nhiều loại đường dây như:

- dây kim loại dùng trong thành phố - đường dây song hành.
- dây đồng trục dùng cho các hệ multiplex.
- cáp quang.
- sóng vô tuyến điện.

Ta cần làm sao đó để việc vận chuyển số liệu là đơn giản. Chúng ta có 2 cách: truyền số liệu trên băng cơ bản và truyền thông qua điều chế (phần này sẽ nói kỹ ở sau).

3.1.2 ETCD phát

ETCD có nhiệm vụ biến đổi những thông báo dữ liệu thành tín hiệu tương thích mà đường dây truyền sử dụng. Sự biến đổi đó có thể phân tích thành 2 sự biến đổi như đã chỉ ra ở hình vẽ: mã hóa và điều chế.

a. Bộ mã hóa

Bộ này biến đổi dãy tín hiệu nhị phân $\{d_k\}$ thành dãy ký hiệu hữu hạn q mức :

$$(\dots, a_i, a_{j+1}, \dots) a_j \quad (q_1, \dots, q_q) \quad (3-1)$$

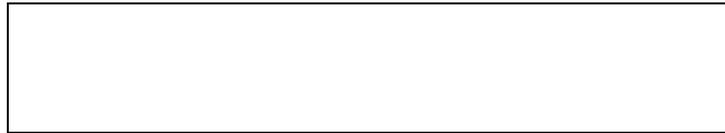
Số lượng cuối của d_k và a_i không giống nhau.

Hình 3.2a ETCD phát.

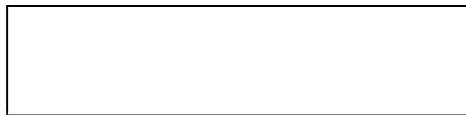
Giá trị q gọi là **hóa trị** và q luôn là số nguyên.

Chú ý rằng nhận biết a_i chưa đủ để nhận biết d_j . Để nhận biết nó cần phải biết mã khác của dãy.

Ví dụ:



Trong trường hợp thông báo dữ liệu đồng bộ thì biểu thức (3-1) được cung cấp qua mã hóa dưới dạng thông báo $a(t)$ cùng dạng thông báo dữ liệu.



Thông báo đó được gọi là thông báo ở băng cơ sở và mỗi D cho phép có được a_i .

Giá trị tức thời $iD + t_0$ được gọi là giá trị tức thời của thông báo băng cơ sở. Khoảng cách giữa 2 giá trị tức thời gọi là khoảng đặc trưng của thông báo băng cơ sở.

D có thể bằng T hoặc bằng bội số của T .

Trong trường hợp đó giá trị q được chọn sao cho $a(t)$ và $d(t)$ cùng có một lưu lượng thông tin, tức:

$$\text{[Empty Box]} \quad (\text{lưu lượng nhị phân của nguồn})$$

như hình vẽ chỉ ra:

$$q = 4, q_1 = -3, q_2 = -1, q_3 = 1, q_4 = 3 \text{ với dãy } T_0.$$

Hình 3.2b Mã hóa $q=4$.

Dãy (d_k) dữ liệu nhị phân không phải luôn luôn thỏa mãn cho sự truyền và nhận với phương thức trực tiếp. Do vậy trong trường hợp mọi sự truyền đồng bộ nếu d_k là sự truyền đạt thành từng gói (khối), sự vắng mặt tín hiệu giữa các khối tạo ra qua độ dài 1 của dãy đồng bộ nhị phân d_k , là trở ngại để thành lập tín hiệu đồng hồ khi thiết kế. Nếu sự đồng bộ dựa trên tín hiệu truyền ở băng cơ sở, sau khi gửi đơn đặt hàng, mặc dù nó chỉ tự sinh trong khi truyền tín hiệu đồng bộ nhưng không dễ dàng thiết lập.

Tồn tại dãy $\{d_k\}$ đủ để cho vận chuyển, d_k cũng phải thỏa mãn biểu thức sau :

ì d_k tồn tại độc lập.

í

$$\Pr\{d_k = 0\} = \Pr\{d_k = 1\} = 1/2$$

b. Bộ điều chế

Bộ điều chế biến đổi thông báo ở băng cơ sở $a(t)$ thành tín hiệu $s(t)$. Nhận biết $s(t)$ trong khoảng đặc trưng $[iD, (i+1)D]$, nó cho phép tìm ra ký hiệu a_i tương ứng với khoảng đó. Sự biến đổi đó gọi là điều chế. **Tín hiệu $s(t)$ gọi là tín hiệu dữ liệu hay tín hiệu phát (emis).** Về cơ bản nó có bằng cách biến đổi **tham số của một tín hiệu $\sin A \cos(2\pi g_c t - j)$.**

Bộ điều chế biến đổi thông báo $a(t)$ thành tín hiệu $s(t)$. Có 3 loại cơ bản:

- Nếu tín hiệu điều chế làm dịch chuyển góc pha ban đầu @ điều pha.
- Nếu làm biến đổi tần số @ điều tần.
- Nếu làm biến đổi biên độ @ điều biên.

Ba loại điều chế trên đây tương ứng sự dịch chuyển tần số. Sự truyền trong băng cơ sở không gây dịch chuyển tần số. Người ta dùng biểu thức sau:



g là hàm vuông góc RD

$g = RD$ đơn giản

3.1.3 ETCD thu

Có nhiều loại ETCD thu khác nhau phụ thuộc vào tín hiệu truyền đồng bộ hay không đồng bộ, có điều chế hay không điều chế.

a. ETCD thu cho tín hiệu không đồng bộ :

Một ETCD thu cho tín hiệu không đồng bộ vận chuyển bằng tần số tôn trọng những tin tức (thông báo) trong băng cơ sở, $a(t)$ xuất phát từ tín hiệu nhận được $X(t)$ và cung cấp $x'(t)$ vào terminal nhận. Sự biến đổi $X(t)$ @ $x'(t)$ là sự giải điều chế được giải thích ở phần sau. Nhớ rằng tín hiệu nhận được $x'(t)$ chỉ gần đúng với tín hiệu $x(t)$ do hiện tượng méo trong khi truyền.

b. ETCD thu cho tín hiệu đồng bộ :

ETCD thu cho tín hiệu đồng bộ với mục đích để cung cấp dãy $\{d_k\}$ khẳng định bởi $\{d_k\}$ từ tín hiệu $X(t)$ nhận được. Nó bao gồm 3 phần như đã chỉ ở hình vẽ.

- Bộ giải điều chế biến đổi $X(t)$ thành $x(t)$ quyết định bởi $a(t)$.
- Tín hiệu được biến đổi qua bộ thu ở băng cơ sở thành dãy ký hiệu $\{\dots, \hat{a}_j, \hat{a}_{j+1} \dots\}$ đánh giá như là dãy $\{a_j\}$ đã phát.
- Cuối cùng sự đánh giá $\{d_k\}$ của dãy dữ liệu nhị phân đã phát qua bộ giải mã. Từ \hat{a}_j qua tác dụng khẳng định lại nó.

Hình 3.3

- a. ETC cho tín hiệu không đồng bộ.
- b. ETC cho tín hiệu đồng bộ và truyền ở băng cơ sở.
- c. ETC cho tín hiệu đồng bộ và vận chuyển bằng phương pháp định tần số.

Trong trường hợp tín hiệu là đồng bộ ở băng cơ sở, giải điều chế được giảm bớt sự biến đổi đồng nhất.

3.1.4. Modem

Nó được dùng để nối thành một tập ETCD phát và thu. Nhóm này thực hiện chức năng riêng biệt là điều chế và giải điều chế. Vì vậy nó được gọi là MODEM, viết tắt của Modulation và Demodulation. Cần nhớ rằng nó không tương ứng với ETCD mã hóa và giải mã. Nó thực hiện một chức năng khác là phân đệm với các terminal.

3.1.5 Giao tiếp

Các phần tử trong mạch dữ liệu được ghép với nhau nhờ bộ phận giao tiếp, nó được chuẩn hóa theo tiêu chuẩn quốc gia hoặc quốc tế.

a. Giao tiếp ETTD/ETCD

Tín hiệu trao đổi giữa ETCD và ETTD thường như sau: thông báo dữ liệu, chuẩn thời gian, tín hiệu cảnh báo, tín hiệu cho phép thành lập hoặc ngắt sự liên lạc, những chức năng đặc trưng, điện, cơ của sự giao tiếp đều được tiêu chuẩn hóa bởi CCITT và ISO.

b. Giao tiếp ETCD/đường dây

Khác với sự giao tiếp ở trên thông qua nó chỉ trao đổi những tín hiệu dạng số (trong trường hợp đồng bộ). Sự giao tiếp giữa ETCD và đường dây truyền là tín hiệu analog trong trường hợp có điều chế. Những đặc trưng có 2 loại :

- Những đặc trưng bắt buộc qua đường dây như là băng thông, công suất tín hiệu, trở kháng ... nó đều được định nghĩa bởi CCITT và nhiều được cải thiện bởi sự điều chỉnh của địa phương.
- Những đặc trưng tương đối của tín hiệu vận chuyển như là: nguyên tắc điều chế, ma trận, sự chuẩn hóa các đặc trưng đó nhằm tương thích các modem của các nơi, sự thử nghiệm nó đảm bảo nhờ CCITT.

3.2 Truyền ở băng tần cơ sở

3.2.1 Khái quát

Chúng ta đã định nghĩa những đường dây truyền thông tin ở băng cơ sở giống như phần tử mang vận chuyển không sinh ra sự dịch chuyển về tần số. Chúng ta thấy trong phần tiếp theo rằng điều kiện ban đầu đó cho phép xử dụng ETCD từng phần đơn giản để phát cũng như thu.

Trong trường hợp truyền đồng bộ, trước hết ở băng tần cơ sở, tiết kiệm hơn, ưu điểm hơn modem làm biến đổi tần số.

Ngược lại trong trường hợp truyền không đồng bộ (với tốc độ chậm hơn). Sự bố trí một modem đều cùng giá trị với sự bố trí ở băng tần cơ sở cho dù nó được dùng tương đối phổ biến. Trong trường hợp đó, người ta thường truyền với sự điều chế.

Chúng ta hạn chế nó trong trường hợp truyền đối với tín hiệu đồng bộ.

Những tín hiệu phát qua ETCD thu cùng một loại với tín hiệu thông báo ở băng cơ sở



và nó được gọi là tín hiệu ở băng tần cơ sở.

Trong thực tế tín hiệu ở băng tần cơ sở dạng nhị phân như trên không được trực tiếp đưa lên đường dây do thành phần một chiều quá lớn, nhất là dùng cho đường dây có khoảng cách xa không thích hợp. Do đó trên thực tế tồn tại loại mã cho tín hiệu nhị phân có phổ thích hợp trước khi đưa lên đường dây đó là tín hiệu nhị phân dạng 2 mức và 3 mức ở băng tần cơ sở.

3.2.2 Tín hiệu 2 mức ở băng tần cơ sở

a. Mã NRZ

Mã được dễ dàng thực hiện là mã nhị phân NRZ (Non Return to Zero). Mã NRZ thường được chia thành 2 loại: NRZ-L (Non Return to Zero-Level) và NRZ-I (Non Return to Zero-Inverted) và nó được xác định như sau:

Với NRZ-L

$$d_i = 0 \text{ cao } (a_i = +a),$$

$$d_i = 1 \text{ thấp } (a_i = -a)$$

Với NRZ-I thay đổi, chuyển mức khi bắt đầu bit 1. Hình vẽ chỉ cho ta qui luật biến đổi của nó.

Tín hiệu ở băng tần cơ sở $S_{NRZ}(t)$ tương ứng với dạng đã viết ở trên, với $D = T$ và $q = 2$ và với giá trị $q_1 = -a$ và $q_2 = +a$ tương ứng với 1 và 0. Phổ công suất của nó có thể tính được. Ta có:



Hình 3.4 Mã NRZ.

Phổ này được biểu diễn ở hình 3-7. Ở đó phần cơ bản tập trung bên cạnh tần số zero. Tín hiệu này sẽ rất nhỏ truyền qua đường dây, ta đã biết ở trên.

Mã nhị phân NRZ cũng rất ít đáp ứng truyền trong băng tần cơ sở do có thành phần một chiều lớn và công suất tập trung ở gần tần số zero. Một biện pháp là tăng tần số, có nghĩa là truyền thông

phụ, ta dùng mã 2Fa.

b. Mã 2 Fa (Mã Manchester)

Theo định nghĩa mã này có $T = 2D$ và mỗi ký hiệu di được chuyển thành một dãy của 2 ký hiệu a'_i , a''_i , với $a'_i, a''_i \in \{-a, +a\}$ qua sự biến đổi sau:

Nếu:

$$d_i = 0 \Leftrightarrow a'_i = +a, a''_i = -a$$

$$d_i = 1 \Leftrightarrow a'_i = -a, a''_i = +a$$

Hình 3.5a đã chỉ cho chúng ta nguyên tắc đơn giản để có mã 2Fa và cấu trúc mạch tạo mã và giải mã trên cơ sở $H(t), (+) (x)$ chia và đường dây trễ.

Ta có thể vẽ phổ công suất của nó với phương trình sau :



Trên đồ thị ta thấy rằng công suất truyền với tần số cơ bản = 0 và 87% được tập trung trong băng tần $(0, 2/T)$. Mã 2 fa cần phải có bộ lọc đường dây làm cho khó khăn trong việc truyền tín hiệu.

Để giảm nhược điểm đó, người ta dùng mã 2Fa vi phân.

Mã 2Fa vi phân được định nghĩa sau:

$$d_i = 0 \Leftrightarrow a'_i = a'_{i-1}, a''_i = -a'_i$$

$$d_i = 1 \Leftrightarrow a'_i = -a'_{i-1}, a''_i = -a'_i$$

Hình 3.5a Mã 2 Fa, 2 Fa vi phân và Miller.

Trong hình vẽ đã chỉ ra tính chất của mã cũng như mạch thực hiện mã và giải mã.

Người ta có thể dễ dàng thấy rằng phổ công suất của mã 2Fa vi phân cũng giống như mã 2Fa. Trong mạng cục bộ (LAN) mã 2 Fa có thể sử dụng đạt đến tốc độ 10Mbps nhưng nó không cho phép truyền ở khoảng cách xa. Để tăng lưu lượng có thể dùng mã Miller.

c. Mã Miller

Mã Miller là mã giảm sự vận chuyển của tín hiệu 2Fa. Hình vẽ cho ta nguyên lý và mạch tạo mã. Phổ công suất của nó được miêu tả trên đồ thị rõ ràng khác với mã 2Fa, tại điểm $f = 0$ công suất không bằng 0. Dãy tần của băng truyền mã Miller không như mã 2Fa. Trong băng tần cơ sở nó có thể truyền từ 5.000 đến 6.000 bit/giây.

Hình 3-5b cho ta mạch tạo mã và giải mã tín hiệu 2 mức.

Hình 3.5b Mạch tạo mã và giải mã.

3.2.3 Tín hiệu 3 mức ở băng tần cơ sở

Chúng ta nhận thấy rằng sự sử dụng tín hiệu 3 trạng thái cho phép phổ không có giá trị ở tần số 0 và giảm bớt được độ rộng của băng thông.

a. Mã lưỡng cực tiêu chuẩn 1 (lưỡng cực đơn giản) AMI

Trong trường hợp $T = D$ và q_1, q_2, q_3 , là 3 giá trị $-1, 0, +1$. Mã lưỡng cực được định nghĩa:

$$d_i = 0 \Leftrightarrow a_i = 0$$

$$= y-1 \text{ nếu } a_i \text{ của bit giá trị 1 cuối cùng đã xét là } +1$$

Hình vẽ chỉ ra $N=1$ cho ví dụ tín hiệu lưỡng cực và một cấu trúc có thể cho bộ mã.

Phổ của nó có thể tính toán được và biểu thị như hình vẽ. Nó bằng 0 khi $f = 0$ và $1/T$.

Một loại mã lưỡng cực khác là mã lưỡng cực tiêu chuẩn 2.

b. Code lưỡng cực tiêu chuẩn 2 (AMI2)

Trong trường hợp này người ta mã hóa theo luật mã lưỡng cực cho dãy $\{d_{2i}\}$ và $\{d_{2i+1}\}$. Dãy $\{d_{2i}\}$ cho dãy dữ liệu chẵn và dãy $\{d_{2i+1}\}$ cho dãy lẻ. Tín hiệu mã hóa biểu thị ở hình vẽ 3-6. Phổ của nó có giá trị 0 ở $f=0$ và cực đại tại $1/T$ và bội số của nó.

c. Mã lưỡng cực mật độ cao (BHD)

Những mã lưỡng cực trên đây không thích ứng với tín hiệu phát khi mà dữ liệu là một dãy số 0 liên tiếp. Để dễ dàng đồng bộ khi nhận tín hiệu và giảm bớt sai số, người ta dùng mã lưỡng cực mật độ cao. Khác với mã lưỡng cực trên là trong trường hợp có dãy tín hiệu với nhiều giá trị 0 liên tục thì thay thế bằng dãy đặc biệt $(-1; 0$ hoặc $+1)$. Bộ phận thu sẽ ghi dấu dãy đã thay thế và thay thế lại bằng dãy tín hiệu 0. Nếu dãy được thay thế cho $(n+1)$ bit thì người ta gọi mã đó là mã tiêu chuẩn n và ký hiệu là BHD_n . Về lý thuyết ta có thể có $BHD_1, BHD_2, BHD_3 \dots$

Trong thực tế người ta thường dùng $n = 3$ để thay thế dãy 4 bit giá trị 0 liên tiếp và gọi đó là mã HDB_3 . Mã mật độ cao HDB_3 được dùng nhiều ở châu Âu và Nhật. Trước tiên nó dựa trên cơ sở mã AMI cho các bit giá trị 1. Trong trường hợp gặp dãy 4 bit liên tiếp giá trị 0 nó thay thế bằng 1 hoặc 2 xung phụ. Việc thay thế đó sao cho cực tính của nó tránh tạo ra thành phần 1 chiều trong khi truyền. Quy luật thay thế phụ thuộc vào số lượng các bit giá trị 1 kể từ sau lần thay thế trước và cực tính của xung bit 1 cuối cùng. Quy luật đó được thể hiện ở bảng sau:

Cực tính của xung cuối	Số lượng bit 1 kể từ khi thay thế lần cuối	
	Lẻ	Chẵn hoặc 0
-	000-	+00+
+	000+	-00-

d. Mã B8ZS :

Như trên ta thấy nếu theo qui luật để thay thế dãy 4 bit giá trị 0 liên tiếp bảo đảm những yêu cầu:

- Không sinh ra thành phần 1 chiều khi truyền.
- Không có dây tín hiệu giá trị 0 quá dài.
- Không giảm tốc độ truyền.
- Có thể bảo đảm phát hiện sai.

Nhưng khi chỉ xét thuần túy các bit giá trị 1 thì luật mã hóa AMI không còn bảo đảm nữa. Để bảo đảm được những mục đích trên và đồng thời không phá vỡ luật AMI với các bit giá trị 1, người ta dùng mã B8ZS. Hình 3.6 chỉ cho ta qui luật tạo mã 3 mức.

Hình 3.6 Qui luật tạo mã 3 mức.

Hình 3.7 Phổ của các tín hiệu 2 và 3 mức.

Mã B8ZS được mã hóa như sau:

Ta mã hóa các bit 1 theo luật AMI.

Nếu gặp dãy 8 bit giá trị 0 liên tiếp ta thay bằng dãy tín hiệu phụ 000- +0+-

e. Chọn cách mã hóa

Người ta thường chọn mã theo yêu cầu về băng thông và ảnh hưởng nhiễu với 3 loại mã: mã lưỡng cực (Manchester), mã 2 Fa (AMI) và mã Miller.

Trong 3 trường hợp:

- Mã 2 pha chiếm băng thông 2 lần lớn hơn mã NRZ, do đó công suất của nhiễu trắng 2 lần lớn hơn.
- Mã lưỡng cực cùng băng thông với NRZ nhưng nó giữ 3 trạng thái do đó độ nhạy đối với nhiễu 2 lần lớn hơn.
- Mã Miller không tồn tại độ rộng của băng tốt nhất của NRZ và chỉ có 2 mức nên độ nhạy với nhiễu ít hơn.

3.2.4 ECTD thu ở băng cơ sở

ETCD thu dùng để nhận các tín hiệu ở băng cơ sở về cơ bản như đã chỉ ra ở hình vẽ.

Thiết bị thu có nhiệm vụ phải làm sao thu đúng những tín hiệu ở băng tần cơ sở đã được phát đi. Để làm được việc đó, điều bắt buộc ở nó phải có tín hiệu đồng bộ cùng tần số với tín hiệu đồng bộ bên phát. Đồng thời do tín hiệu truyền trên đường dây bị làm méo dạng và bị nhiễu nên trước khi giải mã phải được sửa dạng và khử nhiễu.

Hình 3.8 ETCD ở băng cơ sở.

a. Sửa méo do đường truyền:

Hình 3.9 Tín hiệu ở băng tần cơ sở bị méo sau khi truyền.

Hình vẽ cho ta thấy khi truyền xung (tín hiệu ở băng tần cơ sở) đường truyền sẽ làm cho nó bị méo dạng. Biên độ và tần số có sự dịch chuyển nhất định. Rõ ràng cần phải sửa để có thể giải mã được chính xác. Thông thường tín hiệu thu từ đường dây được đưa vào một bộ khuếch đại đẳng biên có độ nhạy đủ lớn, mà độ khuếch đại của nó có thể tự động điều chỉnh hoặc điều chỉnh bằng tay tùy

yêu cầu. Bộ giải mã trong thiết bị thu đã được trình bày ở hình 3.5b.

b. Về tín hiệu đồng bộ ở bộ thu:

Để bảo đảm chính xác khi thu thông tin bắt buộc ở bộ phận thu có tín hiệu đồng bộ có tần số hoàn toàn đúng như tín hiệu đồng bộ ở bộ phận phát.

Có 2 cách để có tín hiệu đồng bộ ở bộ phận thu:

Phát tín hiệu đồng bộ ở bộ phận phát.

Tạo tín hiệu đồng bộ từ tín hiệu nhận được ở bộ phận thu.

Phát tín hiệu đồng bộ

Tín hiệu đồng bộ được phát bằng cách cộng vào tín hiệu ở băng tần cơ sở một tín hiệu sin. Tần số có thể là $1/T$ với loại mã 2 pha và $1/2T$ trong trường hợp mã lưỡng cực tiêu chuẩn 2 (AMI_Z). Cũng có thể gửi tín hiệu như một vạch pha để đồng bộ với tín hiệu giải mã.

Tạo tín hiệu đồng bộ ở bộ thu

Phương pháp thường được sử dụng là so sánh pha 1 tín hiệu đồng bộ được sinh ra ở bộ thu với pha của tín hiệu nhận được và điều chỉnh pha của tín hiệu đồng bộ tự tạo ở bộ thu. (Hình 3-8b)

3.2.5 Chuẩn giao tiếp ở băng tần cơ sở

Trong mạng máy tính cục bộ (LAN) hay các thiết bị truyền số liệu đặt trong một số phòng gần nhau hay trong một nhà, để tiết kiệm, người ta thường dùng đường truyền nối tiếp ở băng cơ sở. Tín hiệu ở đầu ra mức TTL khuếch đại hoặc của UART được khuếch đại và đưa vào trực tiếp trên đường dây kim loại hoặc cáp đồng trục. Mạch kéo tải hoặc mạch thu có yêu cầu nhất định về điện áp hay dòng điện khác với mức TTL tiêu chuẩn.

Việc tiêu chuẩn hóa sự giao tiếp đó là một yêu cầu cần thiết cho phép chúng ta nối mạch. Sau đây chúng ta hãy xem các tiêu chuẩn ghép mạch, một số đặc tính và mạch chức năng của nó.

Hình 3.10 Vòng lặp 20mA.

a. Giao tiếp vòng 20 mA:

Mạch giao tiếp vòng lặp 20mA được sử dụng cho mạch truyền số liệu liên tiếp, 2 dây đơn giản. Với nguyên tắc một dòng chạy trong mạch là $I = 20\text{mA}$. Dòng được cung cấp do bộ phận phát, và bộ phận thu phát hiện. Như vậy yêu cầu mạch phát hiện cách ly với nguồn của bộ phận thu. Tùy theo linh kiện sử dụng mà có cách mắc khác. Hình vẽ cho ta cách mắc dòng 20mA cho trường hợp dùng rơle hoặc optocoupler.

Kiểu mắc dòng $I = 20\text{mA}$ được sử dụng đầu tiên làm mạch tiêu chuẩn cho hệ teletype. Do đóng và mở công tắc mà ta có hay không có dòng. Do cách mắc mạch không cân bằng nên khoảng cách truyền cũng bị hạn chế.

b. EIA RS232C

Chuẩn RS232C từ năm 1969 được chấp nhận chuyên dùng cho truyền số liệu và các đường nối

kiểm tra giữa terminal và modem. Nó được sử dụng trên đường dây có lưu lượng cực đại 20kbps và khoảng cách không quá 15m. Nó là loại giao tiếp không cân bằng có mạch kéo tải (driver).

Mạch giao tiếp RS232C sử dụng nguồn +15V, -15V. Đường dây dữ liệu dùng logic âm, có nghĩa là: logic 1 thì trên đầu ra có điện áp từ -5V đến -15V và logic 0 thì ngược lại +5V đến +15V.

Tuy nhiên trên đường dây kiểm tra người ta dùng logic dương :

Đúng : +5V $\frac{3}{4}$ +15V

Sai : -5V $\frac{3}{4}$ -15V

Hình 3.11 Chuẩn giao tiếp RS232-C.

Chuẩn RS232C dùng conector 25 chân Dtype. Theo quy định chuẩn RS232C dùng để nối giữa terminal và modem (hình 3.11). Trong trường hợp không dùng modem, mắc trực tiếp các terminal với nhau, ta có cách mắc như hình vẽ 3.12.

Hình 3.12 Giao tiếp RS232-C không qua modem.

Hình 3.13 Ứng dụng RS232-C cho mạng kéo tải và mạch thu.

Các hãng chế tạo cung cấp cho ta những IC sử dụng làm mạch driver cho phần phát và thu. Người ta dùng MC1488 và MC1489 và cách ghép mạch như hình vẽ 3-13.

Đầu vào của MC1488 có mức logic của TTL, ngược lại đầu ra của MC1489 cho ta mức logic của TTL.

c. Chuẩn giao tiếp RS449, RS423, RS422

Khi dùng RS232 gặp phải một số phần hạn chế.

- Lưu lượng thông tin nhỏ hơn hay bằng 20 kbps và khoảng cách truyền không được vượt quá 15m.
- Chuẩn RS232 là mạch không cân bằng, dễ bị nhiễu và điều cơ bản là điện áp sử dụng quá cao so với các IC của modem.

Năm 1977 - 1978, EIA cho phép sử dụng một chuẩn mới đó là chuẩn RS449. Chuẩn RS449 bao trùm cả chuẩn RS232.

RS449 chia làm 2 loại: RS422A (cân bằng) và RS423A (không cân bằng). Sự lựa chọn giữa cân bằng hay không cân bằng do lưu lượng thông tin trên đường dây quyết định. Mạch cân bằng có thể chấp nhận trên 20 kbps.

RS449 có tất cả 46 chân (37 chân nối thông thường và 9 chân tùy ý). Và được quy định như bảng cho.

RS422 là chuẩn giao tiếp hoàn toàn cân bằng dùng cặp thu và phát riêng biệt. Các đường nối nó thuộc loại 1 khi lưu lượng quá 20 kbps.

RS423 dùng cho loại mạch không cân bằng. Đường đất của thiết bị phát và thu phân biệt khác nhau và mức logic là $\pm 6V$.

Hình vẽ cho ta sự so sánh mạch khuếch đại chuẩn của RS423 và RS422. Cần chú ý thêm về khoảng cách truyền.

EIA

<i>Circuit</i>	<i>Pin</i>		<i>RS232</i>	<i>Circuit</i>
<i>Designation</i>	<i>Number</i>	<i>Name</i>	<i>Equivalent</i>	<i>Type</i>
SD	4,22	Send data	TDATA	I
RD	6,24	Receive data	RDATA	I
TR	12,30	Terminal ready	DTR	I
DM	11,29	Data mode	DSR	I
RS	7,25	Request to send	RTS	I
CS	9,27	Clear to send	CTS	I
RR	13,31	Receiver ready	CD	I
IC	15	Incoming call	RI	II
SG	19	Signal ground	SIG GND	I
RC	20	Receive common		II
SC	37	Send common		I
SHIELD	1	Shield	CHAS GND	
IS	28	Terminal in service		II
LL	10	Local loopback		II
RL	14	Remote loopback		II

TM	18	Test mode	II
SS	32	Select standby	II
SB	36	Standby indicator	II
NS	34	New signal	II
SF or SR	16	Select frequency or select rate	II
SI	2	Signal rate indicator	II

Bảng EIA RS449 SIGNALS

Hình 3.14 Chuẩn RS422A và RS423A.

3.2.6 Kết luận

Sự truyền ở băng cơ sở cho phép sử dụng thiết bị truyền đơn giản và ít tốn kém. Nhưng kỹ thuật đó bị hạn chế bởi không cho phép sự dịch chuyển tần số. Hơn nữa trên cơ sở như vậy tín hiệu đồng bộ được lặp lại qua sự nhận không có sự dịch chuyển tần số dẫn đến số lượng truyền không liên hệ với luật mã và số lượng sẽ nhận.

Nếu người ta muốn phát tín hiệu ở băng tần cơ sở trên đường dây có trôi tần số, khi thu cần phải trả lại sự dịch tần của tần số qua đường dây.

3.3. Truyền dữ liệu thông qua điều chế

Với tín hiệu điều chế chúng ta đưa sự truyền về việc vận chuyển tần số. Các loại điều pha và điều tần, điều biên cho phép chống những sai lệch khi truyền.

Những modem tương ứng khi truyền có điều chế nói chung về giá mắc hơn modem ở băng tần cơ sở.

3.3.1. Sự truyền qua điều biên

a. Điều chế biên độ với 2 biên (MDB)

Với loại điều chế này tín hiệu điều chế $S(t)$ có được từ sóng sin có dạng $A \cos(2\pi n_0 t - j)$ gọi là sóng mang. Biên độ A của nó được thay đổi bởi tín hiệu thông báo $m(t)$ tạo từ $a(t)$.

$$S(t) = A [K + m(t)] \cos(2\pi n_0 t - j)$$

Tín hiệu đó được biểu diễn ở hình vẽ 3.15 với $K \neq 0$ hoặc $K = 0$.

Phổ của tín hiệu $g(n)$ đã điều chế được tính:



Hình 3.15 Tín hiệu điều biên và phổ của nó.

Ta thấy rằng: sau khi điều chế MDB bên cạnh tần số mang n_c có phổ của tín hiệu điều chế và nó có độ rộng gấp đôi phổ tín hiệu điều chế. Giả thiết để cho thuận lợi, tần số điều chế g_c là tốt nhất với độ rộng B của thông báo $m(t)$.

$$g(n) = 0 \text{ nếu } |n| > B$$

$$B < n_c$$

Như vậy độ rộng của băng tín hiệu gấp đôi độ rộng thông báo. Đó là nhược điểm của MDB, ta dùng một loại điều chế khác, đó là điều chế đơn biên.

b. Điều chế đơn biên (BLU)

Người ta bỏ đi một bên của tín hiệu MDB. Để làm việc đó người ta đặt ở đầu ra bộ điều chế bộ lọc F mà băng thông do ta chọn $(n_c - B, n_c)$ trong trường hợp đó phổ của tín hiệu đơn biên phù hợp với phần bên phải của tín hiệu MDB. Như vậy độ rộng của băng tần bằng độ rộng của tín hiệu.

Trong trường hợp đường dây telephon: băng tần 300 - 3400 Hz. Người ta chọn tần số mang n_c bên cạnh 3000 Hz.

Bộ lọc trong bộ BLU thường yêu cầu có sườn thẳng dốc bên cạnh tần số mang n_c . Chúng ta đã thấy rằng làm việc với điều chế đơn biên thì bộ lọc tốt hơn nhiều so với lọc ở băng cơ sở, điều đó dẫn đến một độ nhạy bén trong thời gian lấy mẫu. Đó là nhược điểm của BLU. Để tránh có quá nhiều bộ lọc tại tần số n_c , yêu cầu trên thực tế những thông báo ở băng cơ sở chỉ gồm công suất có ý nghĩa bên cạnh tần số 0. Vì vậy trong thực tế người ta thường dùng sự tổng hợp giữa MDB và BLU. Đặc biệt với điều chế đơn biên còn dư (BLR) trong bộ lọc như vậy cho phép còn đi qua vùng tần số dưới n_c . Người ta cũng có thể dùng BLU bằng cách xếp đặt để cho các thông báo không có công suất bên cạnh tần số $n = 0$ nhờ loại mã ở băng tần gốc. Mã đó dẫn đến tăng số lượng mức cho 1 ký tự a_i . Tín hiệu BLU có thể miêu tả trong dạng :



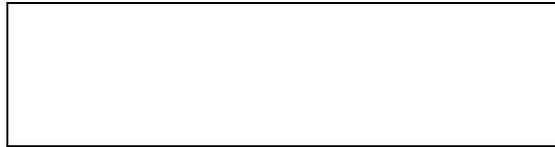
$m(t)$ được suy ra từ thông báo qua sự lọc Q , gọi là bình phương (phép tính diện tích) của độ lợi.



c. Điều biên 2 tần số mang MAQ (QAM)

Một cách điều chế biên độ khác cho phép ta đạt được nó thông qua MDB. Cũng như điều chế BLU, ở điều chế MAQ độ rộng băng thông sau điều chế có cùng độ rộng băng thông của thông tin cần truyền.

Ta có 2 tín hiệu MDB



với $m_1(t)$ và $m_2(t)$ có cùng phổ. Hoặc là người ta có thể thay thế thông báo $a(t)$ bằng 2 thông báo:



Như vậy các nấc thang kéo dài 2D (tốc độ điều chế giảm 2 lần), trong khi $a_1(t)$ và $a_2(t)$ có cùng phổ.

Thông báo $m_1(t)$ và $m_2(t)$ hợp thành thứ tự $a_1(t)$ và $a_2(t)$ qua cùng bộ lọc tạo dạng nó có cùng phổ.

$$g_1(n) = g_2(n)$$

phân bố trong băng $(0, B/2)$ 2 lần nhỏ hơn phổ của $m(t)$. Tín hiệu cần truyền là tổng của 2 tín hiệu.



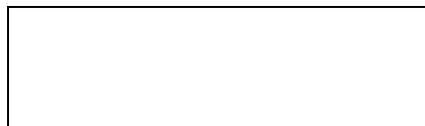
Bộ điều chế được giới thiệu như hình vẽ 3.16.

Hình 3.16 Bộ điều chế MAQ.

Phổ của tín hiệu bao trùm trong khoảng $[n_c - B/2, n_c + B/2]$ trong trường hợp đường dây điện thoại có băng thông (300 - 3400 Hz), người ta có thể cho qua thông báo $m(t)$ có tần số cực đại là 3100 Hz và dùng tần số mang là 1850 Hz.

Chú ý rằng điều chế MAQ có thể cũng coi là một điều chế đệm của biên độ $A(t)$ và pha $j(t)$ của một sóng mang, ta có thể viết biểu thức trên với dạng:

$$s(t) = A \{ A(t) \cos(2\pi \nu_c t - j) + j(t) + K \cos(2\pi \nu_c t - j) \}$$

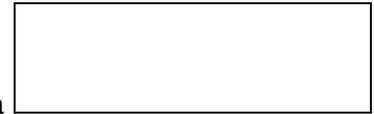
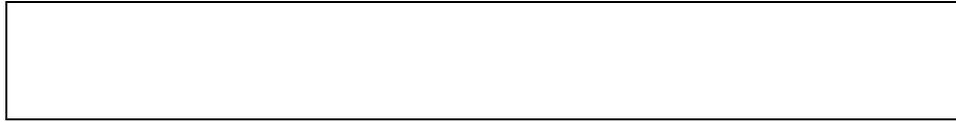


Hình 3.17 Đồ thị không gian.

Nếu không có bộ lọc tạo dạng, hàm $A(t)$ và $j(t)$ được định nghĩa ở trên đều là những hàm vuông góc như là các thông báo $a_1(t)$ và $a_2(t)$ với các nấc thang dài 2D. Mỗi hàm đó chỉ có số lượng hữu hạn các giá trị A_j và j_j tương ứng tất cả các mức có thể $q_1 \otimes q_2$ cho $a_1(t)$ và $a_2(t)$. Tập các giá trị

q^2 điểm của M_1 trên sơ đồ biểu diễn toàn diện $Ae^{-j\omega}$ gọi là đồ thị không gian của $s(t)$.

Trong trường hợp chỉ có 2 mức $q_1 = 1$ và $q_2 = -1$ đồ thị không gian được chỉ ra ở hình vẽ qua các điểm M_1, M_2, M_3, M_4 .



Chú ý: Đồ thị trên cũng chỉ ra sự điều pha. Với 4 giá trị có thể có của pha. Như vậy rõ ràng điều pha chỉ là trường hợp riêng của MAQ với điều chế biên độ cố định. Trong trường hợp đó người ta thích không định nghĩa tín hiệu MAQ dưới dạng ở trên. Xuất phát từ 2 tín hiệu khung corrélés ở băng gốc $a_1(t)$ và $a_2(t)$ mã thường dưới dạng sau. Tổng hợp cho mỗi điểm q mức q_j của thông báo ở băng cơ sở một điểm $M_j(A_j, \varphi_j)$ của đồ thị không gian. Hình b cho ta một ví dụ $q=16$. Đồ thị này đã được tiêu chuẩn của CCITT sử dụng cho sự truyền số liệu với tốc độ $D=9600$ b/s. Nó tương ứng sự điều chế 8 pha với 2 mức biên độ cho mỗi pha. Trong ví dụ đó người ta có thể nhận thấy rằng, những thông báo tổng hợp $a_1(t)$ và $a_2(t)$ được cho như biểu thức là bất buộc nhưng không tương hỗ. Điều đó rất quan trọng trong điều chế. Mỗi điểm của đồ thị phù hợp với tín hiệu truyền trong khoảng có ý nghĩa D và vận chuyển thông tin cho 4 phần tử nhị phân liên tiếp của thông báo dữ liệu nhị phân. Dựa vào cách tính lưu lượng ta có :



Độ nhanh của điều chế đã cho dẫn đến tín hiệu được điều chế có độ rộng phổ 4800 Hz, rộng hơn của telephone ($\gg 3000$ Hz). Điều đó giải thích vì sao khi điều chế cần có bộ lọc tương đương ở đầu ra bộ điều chế với phần tử thay đổi trong băng của kênh, hoặc trực tiếp với thông báo $a_1(t)$ và $a_2(t)$ cùng một phần cơ bản nhưng đưa về băng gốc.

3.3.2 Giải điều chế không liên tục hoặc giải điều chế bao

Ta có tín hiệu điều chế trong dạng:

$$S(t) = A(K+m(t) \cos(2\pi\nu_c t - j))$$

hay

hoặc là

thông qua đường dây truyền nó bị méo dạng nói chung và cộng thêm nhiễu. Các bộ phận thu cần giải điều chế để tìm lại được thông báo $a(t)$. Có 2 loại giải điều chế: giải điều chế liên tục và giải điều chế không liên tục.

Sự giải điều chế được gọi là có liên tục (mạch lạc) hay không tùy theo bộ thu có hay không có sóng

tham khảo liên tục $\cos(2\pi n_c t - j)$.

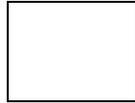
Chúng ta xét trước hết loại giải điều chế không liên tục dù nó ít hiệu quả và ít dùng trong khi truyền số liệu. Sự phát triển của nó không nhiều như giải điều chế liên tục.

Một bộ giải điều chế không liên tục chỉ có thể phân biệt pha của tín hiệu nhận. Nó không thể phân biệt 2 tín hiệu dạng bình phương. Vì vậy mạch giải điều chế không cần 2 sóng mang dạng bình phương và loại này thích hợp là cho MDB.

Chúng ta chỉ để ý đến trường hợp một kênh không có méo. Ta muốn chỉ ra trường hợp mà tất cả các q_i đều được cho, bộ giải điều chế tối ưu có nghĩa là nó giảm tối thiểu khả năng sai cho từng ký tự a_i , gồm có lọc đáp ứng tín hiệu duy nhất, không có tham khảo về pha.



Một bộ kiểm tra phát hiện bao, một bộ lấy mẫu tại thời điểm $(i+1)D$ và một lưới giới hạn. Bộ lọc



cho n_c có dãy thông là \dots . Bộ kiểm tra (dò) bao được thực hiện nhờ bộ phận cho ra 0 và 1 và bộ lọc lý tưởng ở băng $[0, B]$ tương ứng với độ lợi



cuuduongthancong.com

Hình 3.18 Bộ giải mã bao (MDB).

Bộ lấy mẫu trong thời điểm $(i+1)D$ tồn tại sự tham khảo đồng hồ (clock) được dùng để phát các tín hiệu. Sự xem xét đồng hồ là vấn đề chung cho tất cả các loại giải điều chế và điều chế. Ta đã đưa ra về nguyên tắc ở phần 3.2.4B cho sự truyền trong băng gốc, cho tín hiệu giải điều chế cũng như vậy. Các lưới hạn chế có thể tính được nhờ những thay đổi mức của q và mức độ K của phần tử mang.

Bộ thu này không có sự hồi phục tín hiệu tham khảo liên tục của phần tử mang không giải quyết sự nhân. Công nghệ thực hiện nó thật dễ dàng và những bộ giải điều chế thực tế đều đã đạt được mỹ mãn.

3.3.3 Giải điều chế liên tục

cuuduongthancong.com

a. Phục hồi sóng mang

Sự giải điều chế liên tục tồn tại sự phục hồi ở bộ thu sóng mang phục hồi tần số và pha của sóng mang còn lại của tín hiệu nhận được. Ba cách làm thường áp dụng đạt đến kết quả đó:

1. Hồi phục trực tiếp từ tín hiệu nhận được: có thể thực hiện trong trường hợp tín hiệu được điều biên mà tần số mang bỏ đi (với $K=0$) bằng cách nhân tín hiệu nhận được với chính nó.

Mặc dù giá trị trung bình của $a^2(t)$ sít sao với tần số $2n_c$ khôi phục sóng $\cos(2\pi n_c t - j)$ với sự thay đổi pha là p .

Phương pháp đó cũng được áp dụng cho điều chế BLR. Thực tại, trong băng tần số phần này và phần khác của n_c , tín hiệu BLR tương đương với tín hiệu ở điều chế 2 biên.

Có thể bắt đầu lọc tín hiệu trong băng tần số trước khi nhân chúng với nó.

2. Phụ thuộc dao động tại chỗ.

Trong trường hợp sự điều chế MAQ sự khôi phục trực tiếp sóng mang không được thực hiện cả khi 2 thành phần giao thoa. Có thể lấy ra tín hiệu nhận được nhờ sự phụ thuộc vào pha của tín hiệu dao động tại chỗ. Nguyên lý của phương pháp này bao gồm giải điều chế tín hiệu nhận được theo phương pháp 2 tần số mang bình phương qua một dao động tại chỗ có tần số n_c bằng tần số của tần số sóng mang thực.

Từ tín hiệu nhận được khi giải điều chế sự nhận biết D cho phép tạo giản đồ không gian.

Bộ giao động tại chỗ nhận được sự bắt đầu quay với tốc độ góc là $2p(n'_c - n_c)$. Ta giảm góc chuyển động đó bằng cách biến đổi tần số và pha của dao động gốc bằng cách phản hồi.

Giản đồ không gian chỉ có thể biểu diễn trong trường hợp với tần số . Sự phát tần số đó chính là điều kiện để phục hồi tần số mang. Người ta có thể thực hiện sự phát đó từ tín hiệu nhận được $s(t)$ bằng cách tách qua bộ lọc thành phần của $s(t)$ có tần số

và từ đó ta tìm được tần số .

3. Truyền tần số mang: trong trường hợp đơn biên, 2 phương pháp trên không được dùng. Quá trình thường được dùng là thêm vào tín hiệu đơn biên sóng mang với tần số n_c . Sóng đó trùng hợp với một điểm 0 của phổ tín hiệu BLU. Ở bộ phận thu một bộ lọc thẳng đứng sẽ nhận ra sóng mang.

Trong trường hợp truyền tần số mang, cũng giống như trường hợp phục hồi trực tiếp, tần số $1/D$ có thể có được tín hiệu giải điều chế bằng phương pháp dao động tại chỗ.

b. Nguyên lý giải điều chế liên tục

Nếu K là bộ lọc qua đường dây vận chuyển, $K(t)$ là tín hiệu ra của nó, $K(n)$ là độ lợi và $[w_1, w_2]$ là băng thông của nó (300 Hz - 3400 Hz) cho đường điện thoại.

Ở đầu vào của bộ thu, tín hiệu nhận được có giá trị là hàm của tín hiệu dữ liệu $s(t)$.

$$X(t) = K(s(t)) + B(t) \text{ tín hiệu bị méo}$$

$$S(t) = K(s(t))$$

Phổ của nó trong khoảng $[w_1, w_2]$.

$B(t)$ là phổ trắng.

Hình 3.19 Bộ lọc đường dây và bộ lọc giải điều chế.

Ở đầu vào bộ thu người ta đặt bộ lọc C là bộ lọc lý tưởng. Ta có $X(t) \otimes Y(t)$

$$Y(t) = S(t) + B(t)$$

Ta có thể viết

$$Y(t) = x_c(t) \cdot \cos(2\pi n_c t - j) - x_s(t) \sin(2\pi n_c t - j)$$

trong đó $x_c(t)$ và $x_s(t)$ nhận được qua điều chế liên tục nhờ 2 sóng mang $2\cos(2\pi n_c t - j)$ và $-2\sin(2\pi n_c t - j)$ với bộ lọc lý tưởng b ở trong băng thông $[0, (w_2 - n_c, n_c - w_2)]$ như hình vẽ 3.19. Hình vẽ cũng cho ta vùng tần số cho mỗi bộ lọc.

Sau khi có $Y(t)$ ta tạo ra $x_c(t)$ và $x_s(t)$ dễ dàng nó là 2 thành phần của phức $x(t) = x_c(t) + jx_s(t)$

3.4 Sự truyền qua điều pha

3.4.1 Khái niệm

Trong trường hợp điều pha tín hiệu phát được biểu diễn

$$s(t) = A \cos(2\pi n_c t + a(t) - j)$$

$a(t)$ là thông báo ở băng cơ sở. Có 2 loại nguyên tắc để mã hóa được sử dụng trong điều pha. Loại thứ nhất giá trị a_j đủ để xác định d_j tương ứng. Loại mã hóa đó chỉ có thể dùng nếu như sự giải điều chế là liên tục.

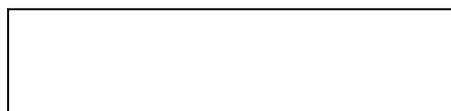
Trong loại thứ hai, để mã hóa, gọi là vi phân, tín hiệu khác nhau giữa a_j và a_{j-1} là những tín hiệu mang thông tin. Ta có:

$$a_j = a_{j-1} + a'_j$$

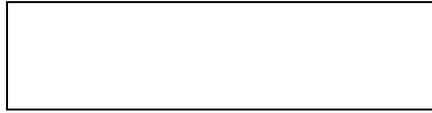
Ở đây a'_j liên hệ với d_j cũng như trong mã hóa trực tiếp. Người ta nói rằng sự điều pha là lấy vi phân. Sự giải điều chế có thể liên tục hoặc không liên tục.

Giá trị của ký tự a_j , a'_j đều có pha được chọn q giá trị (q_1, \dots, q_q) thường $q = 2^n$. Người ta phục hồi từng giá trị q_k qua điểm $P_k = Ae^{jF}$ của đồ thị tổng hợp.

Trong trường hợp chỉ có 2 giá trị ($q=2$) $q_2 = q_1 + p$ loại điều pha thực hiện điều chế biên độ với 2 mức đối nhau và nó không cần bộ lọc. Sự giải điều chế tín hiệu như vậy trong trường hợp sự thu nhận liên tục đã chỉ ra ở phần trên trong khi q là tối ưu 2 giá trị cần chọn là:



hoặc



Hình 3.20a Tín hiệu điều pha.

Hình vẽ 3.20a chỉ cho ta giản đồ thời gian của tín hiệu điều pha.

Ta đã có:



Với



Với $q=4$ ta có giản đồ thời gian như hình 3.20b

Hình 3.20b Đồ thị không gian một tín hiệu điều pha (4 pha).

Dạng này chỉ ra sự tồn tại MAQ như thông báo $m_1(t)$ và $m_2(t)$. Thông thường giản đồ về pha đối xứng qua OX và OY.

Ngoài ra trong đồ thị dùng đã có pha $2a_j$ là:

$$E \sin(2a_j) = 0$$

Do vậy $m_1(t)$ và $m_2(t)$ là không tương hỗ cũng như trong trường hợp MAQ.

3.4.2 Sự thu nhận liên tục

Cũng như MAQ dãy những số $Y_i = u_i + jv_i$ là sự tổng hợp đầy đủ cho sự chính xác của dãy các pha a_i .

3.4.3 Sự thu nhận vi phân liên tục

Rõ ràng rằng **bằng cách điều pha** người ta không thể dùng đường truyền không liên tục được vì **trong đó pha của sóng mang biến đổi ngẫu nhiên** từ vùng D này đến vùng khác. Chính do kênh đó làm hư thông tin. Cần phải bảo đảm **pha của tần số mang không thay đổi tối thiểu là $2D$** . Vì vậy ngay cả nếu pha của sự tham khảo là không biết **nó sẽ lấy ra một** thông tin thông qua sự biến đổi về pha **của sóng nhận được giữa 2 tín hiệu truyền** liên tiếp. Trong trường hợp đó người ta chọn **sự điều chế vi phân** ở chỗ mà ở đó có sự nhảy pha **của tín hiệu mang thông tin**.

Bộ thu nhận vi sai tối ưu bằng cách không có sự giao thoa có thể tính toán được. Chúng ta không đưa ra cách tính và chỉ đề cập một loại bộ thu vi sai như hình vẽ :

Nguyên lý và chức năng bộ thu có thể miêu tả đơn giản khi không có nhiễu trong khoảng $[iD, (i+1)D]$ hàm $Z_1(t)$ và $Z_2(t)$ có thể biểu diễn theo dạng:

Nếu $2p n_{cD} = 0 \pmod{2p}$

Hình 3.21 Bộ nhận vi sai không liên tục.

Bộ lọc thông thấp ở đầu ra với mục đích là làm mất phần cuối của tần số $2p n_c$.

Ta nhận được mẫu $Z_1(t)$ và $Z_2(t)$ và pha a_i là lấy lại bằng cách áp dụng sự ghép theo nguyên tắc quyết định ở trên. Những bộ nhận đó dùng cho bộ điều chế có 4 mức, để cho loại có nhiều mức hơn người ta dùng bộ điều chế loại khác.

cuu duong than cong . com

3.5 Sự truyền qua điều tần

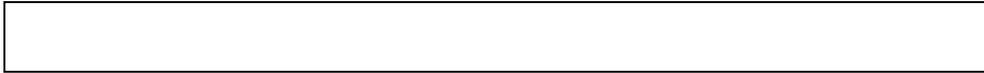
3.5.1 Định nghĩa, phổ

a. Định nghĩa

Tín hiệu $s(t)$ được gọi là điều tần do thông báo $a(t)$ nếu:

Ở đây A và n_c là hằng số biểu diễn biên độ và tần số của sóng tham khảo $A \cos 2p n_c t$. Trong thực tế để sinh ra tín hiệu điều tần người ta dùng (nói chung) một bộ dao động tần số được thay đổi qua $a(t)$. Người ta có thể giả định rằng pha của dao động là hằng số. Do đó $s(t)$ là hàm liên tục. Đôi khi người ta dùng 2 bộ phát tần số n_1, n_2 khác được miêu tả ở hình vẽ cho trường hợp nhị phân.

Dựa vào giá trị ký tự được phát là 1 hoặc 0 tín hiệu phát sẽ có tần số n_2 hoặc n_1 . Những tín hiệu sin đó do các bộ phát độc lập. Tín hiệu $s(t)$ được phát là gián đoạn về pha, nguyên tố mà từ nó (pha) ta có thể xác định phổ. Sự tạo ra băng thông hạn chế cho sự truyền dẫn đến giới hạn phổ của $s(t)$. Vì vậy thông thường người ta dùng loại điều tần (MF) với pha liên tục. Tín hiệu điều tần có dạng đặc biệt nếu thông báo $a(t)$ được số hóa. Có thể biểu diễn:



với j_i là phần kết thúc của pha biến đổi với i , bảo đảm nó liên tục.

Cho dù thông báo $a(t)$ là duy nhất, có nghĩa là trong khi

$$\max |a(t)| = 1$$

Sự trôi, ví dụ theo thời gian chia cho 2π , pha của $s(t)$ lúc đó được gọi là tần số tức thời. Nó được tính

$$n_{inst}(t) = n_c + w_1 a(t)$$

Phần dư $w_1 a(t)$ biểu thị sự thâm nhập của tần số tức thời. Ví dụ: Modems được tạo ra theo tiêu chuẩn của CCITT sử dụng điều chế tần số bậc 2. Với 2 tần số tức thời

$$n_1 = n_c - w_1 = 1300 \text{ Hz và } n_2 = n_c + w_1 = 2100 \text{ Hz}$$

Sự lựa chọn đó tương ứng với tốc độ truyền 1200 baud.

Hình 3.22 Tín hiệu điều tần.

b. Phổ

Phổ của tín hiệu được điều tần không biểu thị đơn giản là hàm phổ của thông báo $a(t)$ mà nó tồn tại nhiều thành phần khác.

Hình 3.23 Phổ của tín hiệu điều tần do thông báo nhị phân.

Hình vẽ đã chỉ ra tín hiệu điều tần $s(t)$ có được do thông báo nhị phân làm thay đổi thông số: $m = 2w_1D$ gọi là độ sâu điều chế.

Rõ ràng rằng công suất của $s(t)$ phân chia trên trục tần số từ $-W$ đến $+W$. Nếu định nghĩa rằng độ rộng của phổ từ 90% công suất $s(t)$ thì phổ thẳng đứng ở giá trị m xấp xỉ 0,64.

Ví dụ: Sự lựa chọn này trong trường hợp modem đã dùng trên



3.5.2 Nhận tín hiệu điều tần

Trong chương trình người ta giả thiết rằng đường truyền không làm méo dạng tín hiệu phát. Tín hiệu đầu vào của bộ thu là:



$B(t)$ là nhiễu chính, trắng.

Chúng ta không đưa ra cấu trúc của bộ thu tối ưu (sai số bé nhất) bởi vì nó ít đưa vào làm việc với

những modem hiện có.

Thông thường người ta dùng 3 loại bộ thu sau:

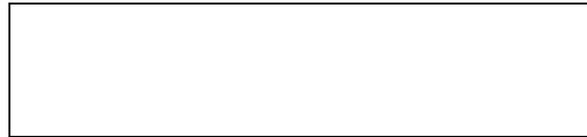
a. Bộ thu hạn biên và tách sóng điều tần (discriminateur)

Hình 3.24 Bộ thu discriminateur.

Bộ thu này ứng dụng công nghệ thật là đơn giản. Cấu trúc của nó được miêu tả như ở hình vẽ.

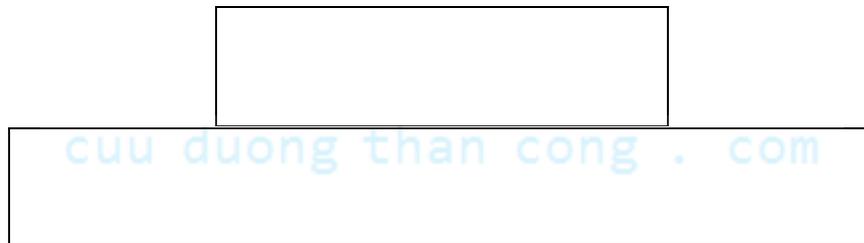
Bộ lọc đầu vào e giảm nhiễu ngoài băng của tín hiệu, không gây méo tín hiệu, độ làm chậm có thể bỏ qua.

Độ lợi của nó



Bằng cách chọn w là $1/2$ độ rộng băng thông sao cho tín hiệu đi qua không bị méo dạng.

Ở đầu ra tín hiệu có thể biểu diễn là một hàm của pha tín hiệu điều tần nhờ các thành phần bậc 2.



Hình 3.25 Đồ thị không gian, các thành phần.

Thành phần bậc 2 chính $a(t)$ và $b(t)$ tương ứng sự phân tích cơ bản tần số nhiễu ở đầu ra bộ lọc băng thông hẹp. Sự phân tích này không được dùng khi mã 2 tín hiệu mang có pha ngẫu nhiên $j(t)$ (không phải là hằng số theo thời gian dù tần số cơ bản).

Từ công thức $Y(t)$ ở trên ta có thể biểu diễn biên độ và pha của $Y(t)$.

$$Y(t) = A(t)\cos(2\pi$$

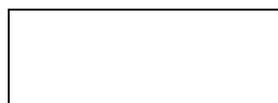
$$n_c t + j(t) + q(t))$$

Với

cuu duong than cong . com



Thường tỉ số tín hiệu trên nhiễu (S/N) đủ lớn ta có



Bộ hạn chế có tín hiệu đầu vào là $Y(t)$ và đầu ra là $Z(t)$ có dạng:

$$Z(t) = \text{sgn}[\cos z(t)]$$

$$Z(t) = 2p n_c t + j(t) + q(t)$$

Hàm $\cos Z(t)$ trong vùng băng hẹp quanh n và ký hiệu của nó $Z(t)$ là điều hòa với tần số $3n_c, 5n_c$. Tín hiệu điều hòa này bị hạn chế bởi bộ lọc vùng \hat{z} , nó cho phép qua không gây méo tín hiệu có tần số cạnh n_c và cho ra $U(t)$ có dạng:

Bộ tách sóng tần số là thiết bị để nhớ sự trôi về pha của sóng nhận được. Tín hiệu ở đầu ra của nó là :

Từ các biểu thức trên:

cuu duong than cong . com

Người ta dùng bộ lọc \hat{s} không làm méo thông báo (sau khi bị chặn) hạn chế nhiễu. Như biểu thức trên đã chỉ rõ. Nhiễu ra phụ thuộc w_1 , A có nghĩa là cùng một công suất của nhiễu ở đầu ra bộ thu có thể nhận được bằng cách giảm công suất phát ($A^2/2$) với điều kiện tăng băng thông.

Công suất của nhiễu có thể giảm tùy ý khi người ta tăng w_1 . Điều đó không có khả năng khi người ta dùng điều biên. Vì vậy MF có thể ưu việt hơn điều biên khi độ rộng băng tần đủ lớn.

* Sự phân tích kỹ hơn về nhiễu ở đầu ra được trình bày trong các giáo trình về kỹ thuật Vô tuyến điện.

b. Thiết bị thu bằng cách đếm

Ý tưởng của phương pháp này đơn giản là: tần số tức thời của một sóng có thể đếm số lần qua điểm 0 của nó trong một đơn vị thời gian.

Cấu trúc của nó được chỉ ra ở hình vẽ.

Hình 3.26 Giải điều tần bằng phương pháp đếm.

Người ta có thể coi đây là một bộ thu hạn biên và tách sóng tần số.

Hình 3.27 Tín hiệu tại các điểm của bộ thu bằng phương pháp đếm.

Các giá trị $a(t), Y(t) \dots W(t), a(t)$ tại từng điểm trên sơ đồ được chỉ ra như ở hình vẽ 3.27. Thông báo $a(t)$ có các giá trị +1, -1 mà không có nhiễu. Bộ thu bằng cách đếm được coi như là bộ thu hạn biên

và tách sóng điều tần.

Hình 3.28 Bộ thu vi phân.

c. Bộ thu vi phân

Nguyên lý của bộ thu như hình vẽ:



Người ta dựa vào nguyên lý nhân tín hiệu thu được với chính nó làm trễ

Nếu ta bỏ đi phần nhiễu thì $Y(t)$ sẽ là:



Hàm $Z(t)$ được tạo ra từ $Y(t)$ và $y(t - t)$ ® ta được



Đưa $Z(t)$ qua bộ lọc thông thấp được $\hat{a}(t)$:



và với điều kiện  ta có:



và với $w_1 \ll n_c$ ta có được $a(t) = 1 \cdot a_i$ với 

d. Kết luận

Với điều tần cho phép ta sử dụng modem với kỹ thuật đơn giản và băng thông cho phép của đường truyền trong khoảng 300 – 3300Hz. Đồng thời độ khuếch đại trong điều tần cũng hạn chế không như trong điều biên. Đó cũng là lý do tại sao điều tần không được dùng trong truyền số liệu với tốc độ nhỏ và băng thông lớn hơn 3000Hz khi phát tín hiệu.

3.6 Truyền nối tiếp không đồng bộ

3.6.1 Nguyên tắc

Hình 3.29 Truyền không đồng bộ nối tiếp.

Để truyền số liệu cho đến hiện nay người ta dùng phương pháp truyền nối tiếp. Do đặc tính của cách truyền, người ta chia ra 2 cách truyền: đồng bộ và không đồng bộ.

Hình vẽ chỉ cho ta nguyên tắc một mạch truyền không đồng bộ, trong đó bộ ghi dịch là thành phần chính của phần cứng.

Khi Load input cao: (1) ® mã số của một ký tự thông qua 8 đường vào, data được nhận vào bộ ghi dịch.

Khi Load input thấp: (0) ® các bit của ký tự lần lượt dịch ra truyền trên đường dây.

Bộ ghi dịch phát bao gồm cả mạch tạo tín hiệu Start, Stop.

Ở bộ phận thu:

Mạch thu phát hiện từng ký tự do phát hiện các bit start từ vùng 0. Khi phát hiện nó, mạch kiểm tra sẽ dịch các bit liên tiếp từ đường dây vào bộ ghi. Sau khi dịch 11 bit (cho mã ASCII có P + 1 start + 2 stop), người ta có thể đọc ký tự đó song song từ đầu ra bộ ghi.

Tùy theo quy ước dùng các bit 1 là chẵn hay lẻ mà người ta thêm 1 hay 0 vào nó sau khi đếm số bit 1 trên 1 ký tự. Thông thường trong truyền không đồng bộ, người ta dùng kiểm tra chẵn, một trong những cách tạo tín hiệu kiểm tra như mạch ở hình 3.30.

Vấn đề xung đồng bộ

Do trên đường truyền không có xung clock nên bản thân bên nhận phải có clkR riêng để có thể đảm bảo thu nhận đúng thông thường $clkT = clkR$. Như vậy phải có mạch hiệu chỉnh clkR cho phù hợp.

Hiện nay do kỹ thuật chế tạo phát triển nhiều mạch LSI trên thị trường đáp ứng các chuẩn truyền không đồng bộ như UART, ACIA ... (Universal Asynchronous Receiver Transmitter, Asynchronous Communication Interface Adapter).

Hình 3.30 Mạch tạo tín hiệu chẵn lẻ.

3.7 Truyền đồng bộ nối tiếp

Sử dụng truyền không đồng bộ đơn giản và rẻ tiền. Thường truyền không đồng bộ được dùng để truyền với những thông báo ngắn. Để truyền những files dữ liệu dài, phương pháp truyền đồng bộ có nhiều hiệu quả hơn. Thông báo được truyền thành khối mà những bit start, stop cho các ký tự bỏ đi. Các ký tự trong khối được truyền liên tiếp nhau, kiểm tra sai được thực hiện theo từng khối.

Để giải quyết đồng bộ giữa phần thu và phát người ta có thể truyền clock đồng thời với dữ liệu, hoặc bên bộ phận thu có mạch tạo clock đồng bộ dựa vào dữ liệu truyền.

Để đơn giản, bộ phát truyền một hay nhiều ký tự nhận dạng sync hoặc cờ (flags) khi bắt đầu truyền một khối.

Nhiều mạng đồng bộ sử dụng một trong các protocol do hãng IBM đưa ra như BISYNC (BINARY Synchronous Communication) hoặc SDLC (Synchronous Data-Link Control) hoặc HDLC. Người ta chia protocol truyền đồng bộ thành 2 loại: định hướng theo ký tự hoặc định hướng theo bit. Chúng ta sẽ khảo sát ở phần sau.

3.7.1 Giao tiếp giữa DTE - DCE trong truyền đồng bộ

Khi tốc độ mạng truyền khoảng 1200 b/s thì người ta dùng truyền đồng bộ, modem đồng bộ được sử dụng, nó sẽ tạo clock thu tự động đồng bộ với dòng dữ liệu.

RS232			RS449		
Symbol	Pin	Name	Symbol	Pins	Name
TCLK	15	Transmit clock (from DCE)	ST	6 & 23	Send timing
RCLK	17	Receive clock (from DCE)	RT	8 & 26	Receive timing
ETCLK	24	External transmit clock	TT	17 & 35	Terminal timing (from DTE)

Chuẩn giao tiếp RS232 và RS449 được sử dụng. Các chân của nó được quy định như bảng.

Khi sử dụng modem đồng bộ, thời gian thu thường được cung cấp do modem. Clock phát có thể từ modem hay từ terminal. Sự cung cấp tín hiệu clock có thể xảy ra như hình vẽ 3.31

Hình 3.31 Nối đồng bộ RS 449.

3.7.2 Truyền đồng bộ định hướng theo ký tự Protocol Bisync.

Những protocol thông tin thường chia theo lớp Bisync là protocol ở lớp 2. Lớp 1 miêu tả đặc tính vật lý và điện (như RS449 chẳng hạn). Lớp 2 miêu tả quá trình kiểm tra sự dịch chuyển thông tin.

Tiêu chuẩn Bisync (BSC) là protocol kiểm tra theo ký tự. Từ lâu quá trình kiểm tra theo ký tự đã được dùng để kiểm tra đường dữ liệu.

Thông báo dạng Bisync được truyền theo khối, để nhận dạng sự bắt đầu khối, mỗi khối có một hay nhiều tín hiệu đồng bộ (syn).

Mã được dùng trong Bisync là mã ASCII. Các ký tự điều khiển được dùng là:

syn ký tự đồng bộ \$16

SOH ký tự bắt đầu header \$01

STX ký tự bắt đầu cho text \$02

EXT ký tự kết thúc cho text \$03

Thông báo Bisync có dạng:

syn	syn	SOH	header	STX	text	ETX	BCC
-----	-----	-----	--------	-----	------	-----	-----

Nội dung của thông báo nằm ở phần text, phần header để ghi địa chỉ người nhận và ACK/NCK khi cần thiết. Ký tự BCC (block check character) là byte ký tự kiểm tra khối. Byte này là phần không thể thiếu của khối. BCC thông thường chỉ là ký tự xét tổng kiểm tra chẵn lẻ.

Ví dụ: cần truyền chữ TEST thì khối thông tin như sau:

STX	T	E	S	T	ETX	BCC	
0	0	1	1	0	1	1	b ₁
1	0	0	1	0	1	1	
0	1	1	0	1	0	1	
0	0	0	0	0	0	0	
0	1	0	1	1	0	1	
0	0	0	0	0	0	0	
0	1	1	1	1	0	0	b ₇
1	1	1	0	1	0	0	b ₈

Dòng dữ liệu trên đường dây sẽ là:

STX	T	E	S	T	ETX	BCC
01000001	...					

Trong ví dụ trên người ta dùng kiểm tra chẵn và BCC kiểm tra từ STX đến ETX. Thông thường BCC được kiểm tra từ SOH đến ETX.

Khi nhận toàn khối, bộ phận thu sẽ tính BCC và so sánh với BCC đã nhận sau đó nó thông báo ACK hay NAK. Bộ phận phát sẽ phát lại (nếu sai) cho đến khi khối đó được nhận đúng. Như vậy Bisync là protocol half-duplex, kiểm tra theo byte.

3.7.3 Truyền đồng bộ định hướng theo bit: cuuduongthancong.com

3.7.3.1 Protocol – SDLC:

Bisync (BSC) là loại protocol sử dụng byte để kiểm tra, nó là protocol truyền half duplex. Bên thu, sau khi nhận một khối thông tin nếu tốt cần thông báo để bên phát phát khối tiếp theo.

Năm 1970, hãng IBM cho ra protocol cho việc truyền đồng bộ (SDLC).

SDLC là protocol kiểm tra theo bit, nó ít phụ thuộc vào các ký tự. Như vậy sự chấp nhận của bộ thu có thể chờ đến khi thông báo kết thúc hoặc thiết bị thu sẵn sàng phát.

Với một số ký tự dự trữ cho kiểm tra đường dây, SDLC dễ dàng làm cho dãy số liệu truyền trong suốt.

Thông tin trong SDLC được tổ chức theo khung (frame).

Flag	Address	Control	Use data	FCS	Flag
------	---------	---------	----------	-----	------

- **Vùng Flag:** dùng để truyền tín hiệu đồng bộ và báo cho sự bắt đầu của khung truyền. Giá trị của nó được quy định 01111110.
- **Vùng Address:** Thường có độ dài 1 byte. Nó dùng để ghi địa chỉ của trạm thứ 2 (ở đó nhận hoặc phát thông tin). Địa chỉ FF dùng cho tất cả các terminals trong mạng.
- **Vùng Control:** có độ dài 8 bit, để chỉ ra loại của khung thời gian: u (unnumbered), S (Supervisory), I (Information):
 - Khung U: dùng để kiểm tra trạm thu như thiết lập, mất liên lạc, thử trạm ...
 - Khung S: dùng đếm khung (phát và thu), trả lời ACK/NAK, thông báo bận ...
 - Khung I: truyền đạt thông báo, thông tin.

Bảng cho dưới đây chỉ cho ta mẫu các bit trong vùng kiểm tra của SDLC. Ở đây ta thấy nếu bit cuối cùng có giá trị 0, dạng I; 01 dạng S; 11 dạng U. Mỗi một trạm dùng SDLC có 3 trạng thái phương thức: thiết lập, trả lời, giải phóng. Mỗi một trạm gồm có: N_s : số của khung phát. Trạm phát đếm số khung cần phát N_s trong vùng kiểm tra và truyền. Ở trạm thu nếu mỗi lần thu không có lỗi thì N_r được giảm 1 hoặc tăng 1 và truyền lại ACK với chỉ số N_r tiếp theo.

Name	Type	Bit Pattern	Comments
I	I	N_r P/F N_s 0	
REJ	S	N_r P/F 1001	No I field
RNR	S	N_r P/F 0101	No I field
RR	S	N_r P/F 0001	No I field
TEST	U	111 P/F 0011	
UP	U	001 P 0011	No I field
XID	U	101 P/F 1111	
RD	U	010 F 0011	No I field
CFGR	U	110 P/F 0111	

BCN	U	111 F 1111	No I field
FRMR	U	100 F 0111	
UA	U	011 F 0011	No I field
DISC	U	010 P 0011	No I field
DM	U	000 F 1111	No I field
SNRM	U	100 P 0011	No I field
			Resets Nr and Ns
SIM	U	000 P 0111	No I field
			Resets Nr and Ns
RIM	U	000 F 0111	No I field
UI	U	000 P/F 0011	

- *Vùng Use data*: có thể có độ dài tùy ý nhưng trong khi sử dụng nó tùy thuộc vào số byte của SDLC quy định.
- *Vùng FCS*: có độ dài 16 bit,...

3.7.3.2 Protocol – HDLC:

HDLC hoàn toàn giống như SDLC, SDLC do ISO (International Standard Organization) đưa ra sử dụng khi truyền đồng bộ. HDLC do CCITT đề xuất sử dụng X-25 trong mạng chuyển mạch gói.

Có 3 điểm khác biệt giữa HDLC và SDLC. HDLC cho phép ta mở rộng địa chỉ và vùng kiểm tra, và ký tự báo dừng của SDLC là 8 bit một, còn của HDLC là 7 bit một.

3.7.4 Protocol HDLC mở rộng

Protocol bit định hướng được thiết kế để thỏa mãn sự thay đổi rộng những yêu cầu của đường nối dữ liệu bao gồm:

- Điểm - điểm và nhiều điểm.
- Hai chiều gián đoạn (half-duplex), 2 chiều toàn phần (duplex).
- Sơ cấp - thứ cấp (host-terminal) và tương đương (computer-computer).
- Đường nối với a lớn hoặc nhỏ.

Thêm vào đó những protocols này đều quan tâm thỏa mãn những mục đích sau:

- Mã độc lập: Nguồn xử dụng sẽ chấp nhận dùng một số loại mã hoặc bit mẫu trong dữ liệu để truyền.
- Phù hợp: Sự tạo dạng sẽ cung cấp cho sự biến đổi của loại đường nối và một số những yêu cầu tạo ra.
- Hiệu quả cao: Sự tạo dạng sẽ giảm thiểu bit vượt qua và cho phép kiểm tra dòng và sai số có hiệu quả.
- Sự linh động cao: Protocol sẽ có kết quả trong việc tìm sai và bao trùm được cả quá trình.

Chìa khóa để thỏa mãn những yêu cầu và mục tiêu đó là dựa vào vị trí tín hiệu và mã của vùng kiểm tra. Một cấu trúc có vị trí ý nghĩa khi nó chia vùng đó với mỗi một vị trí được cung cấp cố định trong frame. Vùng mã kiểm tra với một tổ hợp khác nhau của các bit có ý nghĩa rộng và mỗi một giá trị biểu thị của từng vị trí trong vùng được sử dụng. Chúng ta sẽ xét cụ thể ở phần sau và ta sẽ so sánh với phương pháp ít hiệu quả hơn trong phần định hướng ký tự.

Một số các protocol dùng phương pháp bit định hướng kiểm tra như sau:

- HDLC được phát triển do IOS (IOS3309).
- ADCCP (Advanced Data Communication Control Procedures).
- ANSI 3000: Do American National Standards Institute 3000.
- LAP-B (Link Access Procedure, Balanced).
- X-25: Do CCITT phát triển sử dụng trong mạng chuyển mạch gói.
- SDLC do IBM phát triển.

Trên thực tế không có sự khác nhau giữa HDLC và ADCCP. LAP-B là một phần phụ của HDLC.

Ta sẽ khảo sát chủ yếu HDLC.

Để thỏa mãn các yêu cầu nêu ở trên. HDLC được định nghĩa 3 loại của trạm, 2 cấu hình nối và 3 kiểu truyền dữ liệu.

Ba loại trạm là:

- *Trạm sơ cấp:* có nhiệm vụ kiểm tra các thao tác trên đường nối. Frames do sơ cấp cung cấp là các frames điều khiển.
- *Trạm thứ cấp:* được hoạt động dưới sự điều khiển của sơ cấp. Frames do trạm thứ cấp phát ra là: trả lời. Sơ cấp sẽ cho đường nối cách biệt giữa các thứ cấp.
- *Trạm tổ hợp:* sự tổ hợp giữa sơ và thứ cấp, có thể phát ra cả điều kiện và trả lời.

Hai loại cấu hình đường nối là:

- *Cấu hình không đối xứng:* dùng trong điểm - điểm và nhiều điểm. Cấu hình này bao gồm 1 sơ cấp và 1 hay nhiều thứ cấp, sử dụng full duplex hoặc half duplex.
- *Cấu hình đối xứng:* chỉ sử dụng cho điểm - điểm. Cấu hình này gồm có 2 tổ hợp sử dụng full duplex hoặc half duplex.

Hình 3.32 Đường nối HDLC.

Ba kiểu truyền dữ liệu là:

- *Kiểu trả lời bình thường NRM* (Normal response mode): Đây là kiểu truyền dùng cho cấu hình không đối xứng. Trạm sơ cấp có thể gửi dữ liệu đến thứ cấp nhưng thứ cấp chỉ có thể gửi khi sơ cấp yêu cầu.

NRM được dùng trong đường dây nhiều chỗ dừng, nhiều terminal được nối vào máy tính. Máy tính yêu cầu sự đưa vào của từng terminal. NRM cũng thường dùng trong đường nối điểm - điểm. Thành phần tham gia là một terminal hay một thiết bị ngoài nối với máy tính.

- *Kiểu không đồng bộ đối xứng ABM* (Asynchronous balanced mode): Đây là kiểu truyền dùng cho cấu hình đối xứng: Cả 2 trạm đều có thể thiết lập truyền dữ liệu không cần sự cho phép của trạm kia.

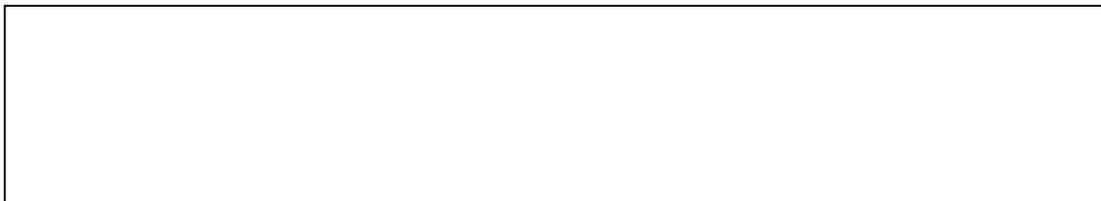
ABM được sử dụng có hiệu quả trong đường nối điểm - điểm full duplex. Bởi vì nó không cần yêu cầu phát.

- *Kiểu trả lời không đồng bộ ARM* (Asynchronous response mode): Đây là kiểu truyền dùng cho cấu hình không đối xứng. Trong cách này thứ cấp có thể thiết lập truyền dữ liệu không cần sự chấp nhận của sơ cấp (gửi 1 trả lời không chờ điều khiển). Trạm thứ cấp chờ sự trả lời của đường truyền bao gồm khởi động, kiểm tra sai và ngắt mạch.

ARM ít khi được sử dụng, nó được sử dụng trong sự yêu cầu của Hub (hub polling) và một số hoàn cảnh khác mà ở đó thứ cấp cần thiết lập sự truyền.

Cấu trúc Frame HDLC:

HDLC được sử dụng trong truyền đồng bộ. Tất cả thông tin truyền đều trong frame và một frame đơn giản sắp xếp cho tất cả loại dữ liệu và sự trao đổi điều khiển. Frame HDLC có cấu trúc như sau:



Frame có các vùng sau:

- Flag: 8 bits.
- Add: Một bytes. Trong HDLC mở rộng vùng Add có nhiều bytes.
- Control: 8,16 bits. Trong HDLC mở rộng vùng C có 16 bit.
- Information: thay đổi.
- FCS: (Frame check sequence) 16 hoặc 32 bits tùy yêu cầu sử dụng CRC.

Vùng Flag, Add., Control, trước dữ liệu được coi là header. Vùng FCS, Flag sau dữ liệu được coi là phần kết thúc.

Vùng Flag:

Vùng flag giới hạn frame về 2 phía với giá trị cho sẵn 01111110. Một flag đơn có thể coi là phần kết thúc cho một frame và mở đầu cho một frame khác. Các trạm tích cực mắc đến đường dây nối đều xử dụng flag để đồng bộ khi bắt đầu frame. Trong khi thu một frame trạm tiếp tục tìm flag tiếp theo để biết điểm kết thúc dữ liệu. Tuy nhiên, do trong HDLC frame chấp nhận 1 dãy bit, điều đó không thể tránh khỏi có khi có giá trị 01111110 ở chỗ nào đó trong frame. Điều đó sẽ phá vỡ sự đồng bộ của frame làm cho bên nhận hiểu tại điểm đo frame đã kết thúc. Để tránh điều đó, người ta sử dụng một quá trình gọi là bit - stuffing.

Hình 3.33 Bit stuffing.

Bit stuffing là cách làm mà trong đó bộ phận truyền sẽ thêm những bit 0 sau mỗi cụm 5 bit một liên tiếp trong frame (ngoài vùng flag). Khi thu, sau khi phát hiện được flag bắt đầu frame, bộ phận thu hiển thị dòng các bit. Khi dãy 5 bit 1 xuất hiện thì bit thứ 6 sẽ được kiểm tra nếu bit đó là 0 sẽ bỏ đi. Nếu bit đó là 1 và bit thứ 7 là 0 thì nó được chấp nhận là flag. Nếu bit thứ 6 và bit thứ 7 đều là 1, thì đó là tín hiệu dừng của bộ phận phát (01111111).

Với sự ứng dụng bit - stuffing, các bit có thể thêm vào trong frame giúp ta tránh nhầm lẫn sự kết thúc frame không đúng.

Hình vẽ cho chúng ta ví dụ về bit stuffing. Những "bẫy" của bit - stuffing được chỉ ra ở hình vẽ. Flag được dùng để kết thúc và mở đầu một frame, nếu một bit 1 trong flag sai dẫn đến tạo 2 frames thành 1 frame, ngược lại một bit 1 sai trong frame có thể tạo 1 frame thành 2 frame.

Vùng địa chỉ:

Vùng địa chỉ để nhận dạng trạm thứ cấp mà nó gửi hoặc nhận frame. Vùng này không cần thiết trong trường hợp điểm - điểm, nhưng nó vẫn được dùng để có dạng thống nhất. Vùng địa chỉ thông thường có 8 bit, nhưng trong trường hợp cần thiết nó có thể mở rộng thành 1 số lần 7 bit. Bit đầu tiên của byte là 1 hay là 0 phù hợp nó có phải là bytes cuối của vùng địa chỉ hay không? Người ta qui định khi bit đầu tiên của byte có giá trị 0 thì byte đó chưa phải là byte cuối cùng của vùng địa chỉ. Byte cuối cùng của vùng địa chỉ có bit đầu tiên mang giá trị 1.

Nếu byte địa chỉ có giá trị 11111111. Có nghĩa là sơ cấp sẽ gửi frame đó cho tất cả các thứ cấp trong mạng.

Hình 3.34 Cấu trúc vùng địa chỉ.

Vùng kiểm tra:

HDLC định nghĩa 3 loại frame mỗi loại có dạng sắp xếp khác nhau như hình vẽ 3.35

- *Information frame (I frame)*: mang dữ liệu cần truyền cho trạm, ta hiểu là dữ liệu được dùng, thêm vào đó kiểm tra vùng và kiểm tra sai xử dụng ARQ.

Cơ sở thao tác của HDLC là sự trao đổi thông tin với những frame mang dữ liệu. Mỗi một I frame gồm có số thứ tự của frame phát cũng như sự phụ thuộc vị trí của ACK. ACK là số thứ tự của frame tiếp theo sẽ được chấp nhận. Cửa sổ cực đại được chấp nhận là 7 hoặc 127.

	1	2	3	4	5	6	7	8
--	---	---	---	---	---	---	---	---

Dạng I	0	N(S)				P/F	N(R)									
Dạng S	1	S				P/F	N(R)									
Dạng U	1	M				P/F	M									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Dạng I	0	N(S)						P/F	N(R)							
Dạng S	1	0	S		0	0	0	0	P/F	N(R)						
Dạng U	1	1	M	M	U	M	M	M	P/F	X						X

Hình 3.35 Cấu trúc vùng C và giá trị các bit vùng C.

N (S) : chỉ ra bộ đếm dãy phát.

N (R) : chỉ ra bộ đếm dãy nhận.

P/F : bit chỉ thị. P =1 trả lời trực tiếp cho sơ cấp.

F =1 như P nhưng chấm dứt frame cho thứ cấp.

S : chỉ ra các chức năng giám sát.

Chú ý rằng cơ sở cho vùng kiểm tra cho S frame và I frame ta dùng 3 bit liên tiếp. Với vùng kiểm tra mở rộng ta dùng 7 bit.

I frame cũng chứa bit Poll / Final (P/F). Bit đó là Poll nếu là điều khiển (từ sơ cấp) và là bit Final khi trả lời (từ thứ cấp).

Trong NRM: sơ cấp sử dụng Poll cấp cho phép và gửi tín hiệu để làm cho poll =1, final =1 đối với frame cuối cùng mà thứ cấp trả lời.

Trong ARM và ABM. Bit P/F có khi được sử dụng thích hợp cho sự trao đổi của S frame và U frame.

- *Super visory frames (S.frame)*: được coi như dùng ARQ nếu như không có dấu hiệu gì đặc biệt. S frame dùng cho kiểm tra dòng và kiểm tra sai. Với ARQ – trở lại N và ARQ – trở lại có lựa chọn nó đều chấp nhận. ARQ - trở lại có lựa chọn ít dùng vì nó yêu cầu bộ nhớ.

Một frame có thể ACK khi bộ thu sẵn sàng (Receiver Ready) RR. Ngược lại nếu RNR (Receiver Not Ready) thì frame đó được chấp nhận nhưng nó sẽ không chấp nhận thêm I frame nào nữa cả cho đến khi có RR.

P/F bit trong trường hợp này có thể chấp nhận. Sơ cấp có thể đưa P bit trong RR frame lên 1 để poll thứ cấp. Thứ cấp sẽ trả lời với một I frame nếu nó có một frame, nếu không nó sẽ gửi RR với F bit biểu thị nó không có dữ liệu gửi sơ cấp (trạm tổ hợp) có thể đặt P =1 trong lệnh điều khiển RNR(Receiver not Ready).

- *Unnumbered frames (U frame)*: biểu thị các chức năng điều khiển đường dây nối.

Một hoặc hai bit đầu của vùng dùng để nhận dạng frame, các thông tin nó biểu thị trong vùng kiểm tra hay trong vùng kiểm tra mở rộng được chỉ ra như hình vẽ.

U frame dùng để thay đổi chức năng kiểm tra. Như tên gọi của nó, frame này không mang số thứ tự và cũng không sửa đổi thứ tự hoặc dùng của I frames. U frame được chia thành các nhóm:

Cách đặt điều khiển hoặc trả lời.

Điều khiển truyền thông tin và trả lời.

Lấy lại điều khiển và trả lời.

Trộn sự điều khiển và trả lời.

Cách đặt điều khiển được gửi đi do sơ cấp/trạm tổ hợp để thiết lập hoặc đổi cách của thứ cấp/ trạm tổ hợp. Trạm thứ cấp/ trạm tổ hợp chấp nhận với việc trả lời với frame không đánh số chấp nhận (UA frame). UA frame được đặt F bit giống như P bit trong khi thu.

Những lệnh điều khiển SNRM, SNRME, SARM, SARME, SABM và SABME đều tự giải thích. Như giải thích ở trên trong cả 2 hướng với I frame số thứ tự =0. Lệnh điều khiển trong khi xét thiết lập ban đầu để khởi động một trạm đặc biệt.

Điều khiển truyền thông tin và trả lời: Dùng để trao đổi thông tin giữa các trạm. Nó được cung cấp trước tiên thông qua thông tin không đánh số. (UI) điều khiển / trả lời. Những ví dụ về UI frame đều là những tình trạng mức độ cao (higher level status) như: tác động ngắt, thời gian trong ngày, tham số khởi động đường nối.

Những lệnh điều khiển poll không đánh số dùng để xin sự trả lời không đánh số như là: sự thiết lập tình trạng một đường của trạm địa chỉ hóa.

Lệnh lấy lại điều khiển và trả lời được dùng trong trường hợp ARQ không chấp nhận hoặc không làm việc. Frame trả lời lại (FRMR) được dùng để giải trình sai trong frame đã nhận như:

- Vùng kiểm tra không đúng.
- Vùng dữ liệu quá dài
- Vùng dữ liệu không chấp nhận với loại frame đã thu.
- Bộ đếm thu không ổn định.

Lệnh Reset dùng để xóa FRMR, RSET thông báo rằng trạm phát được reset số thứ tự nó gửi đi và trạm thu (có địa chỉ) sẽ reset số thứ tự thu.

Trộn sự điều khiển và trả lời: cuối cùng ta có 2 loại: điều khiển hỗn hợp và điều khiển trả lời.

Lệnh điều khiển trả lời nhận dạng sự trao đổi được dùng cho 2 trạm để nhận dạng trạm trao đổi và những đặc tính của 2 trạm. Những thông tin trao đổi là phần thêm vào độc lập. Điều đó có thể nghĩ đơn giản là trạm phát và nhận đều đang làm việc.

Vùng thông tin

Vùng thông tin thông thường chỉ giới hạn trong một frame và một số Unnumbered frame. Vùng này có thể chứa liên tiếp các bit với độ dài không được định nghĩa trong các chuẩn nhưng thông thường bị giới hạn. Thường độ dài của nó là một số lần 8 bit.

Vùng kiểm tra (FCS)

Thông thường FCS có 16 bit CRC. Theo qui định của CCITT cũng có khi người ta chọn 32 bit CRC tùy theo độ dài của frame và đường dây.

Bảng sau cho ta cụ thể về các lệnh ở 3 dạng U, I, S trong vùng kiểm tra.

HDLC Control Field Format:

I	Information	NR0	Non-Reserved 0
RR	Receive Read	NR1	Non-Reserved 1
REJ	Reject	NR2	Non-Reserved 2
RNR	Receive Not Ready	NR3	Non-Reserved 3
SREJ	Selective Reject	SIM	Set Initialization Mode
UI	Unnumbered Information	RIM	Request Initialization Mode
SNRM	Set Normal Response Mode	PRMR	Frame Reject
DISC	Disconnect	SARM	Set Async Response Mode
RD	Request Disconnect	SARME	Set ARM Extended Mode
UP	Unnumbered Poll	SNRM	Set Normal Response Mode
RSET	Reset	SNRME	Set NRM Extended Mode

XID	Exchange Identification	SAMB	Set NRM Async Balance Mode
DM	Disconnect Mode	SAMBE	Set AMB Extended Mode
The P/F	Bit	TEST	Test
C	Command	R	Response

SNRM (Set normal reponse mode): Đặt thứ cấp vào Mode NRM. NRM giải phóng thứ cấp từ việc gửi những frames không yêu cầu. Điều đó có nghĩa rằng trạm sơ cấp kiểm tra tất cả dòng frame trên đường.

DISC (Discon nect): đưa trạm thứ cấp vào Mode không nối mạch. Lệnh này có giá trị cho đảo đường nối. Nó có giá trị như là việc gác điện thoại.

DM (Discon nect mode): được truyền từ thứ cấp đến trạm chỉ thị, nó không có thao tác gì.

TEST (Frame test): để xác định sự kiểm tra trả lời từ thứ cấp. HDLC không qui ước frame test như thế nào được dùng. Một ví dụ như: có thể dùng vùng I để chuẩn đoán điều mình muốn.

SARM: chấp nhận trạm thứ cấp truyền ngoài sự yêu cầu của sơ cấp. Nó đặt trạm thứ cấp vào trạng thái truyền thông tin của ARM.

SABM (Set Asynchronnous Reponse Mode): đặt 2 trạm tương đương vào chế độ ABM. Không cần có P bởi vì 2 trạm đều là trạm tổ hợp.

SNRME (Set Normal Reponse Mode Estanded): set NRM với 2 bytes ở vùng kiểm tra, lệnh này để mở rộng vùng đếm thứ tự. Cho phép N(R) Ns có độ dài 7 bit, tức là cửa sổ đếm từ 1- 127.

SABME (Set Asynchronnous Balanced Mode Estanded): Set ABME với 2 bytes vùng đếm thứ tự.

UP (Unnumbered Poll): yêu cầu 1 trạm không quan tâm đến số thứ tự đếm hoặc ACK. Sự trả lời là tùy ý, nếu như P =0.

RSET (Reset) dùng như sau: trạm phát reset N(S) và trạm thu reset V(R).

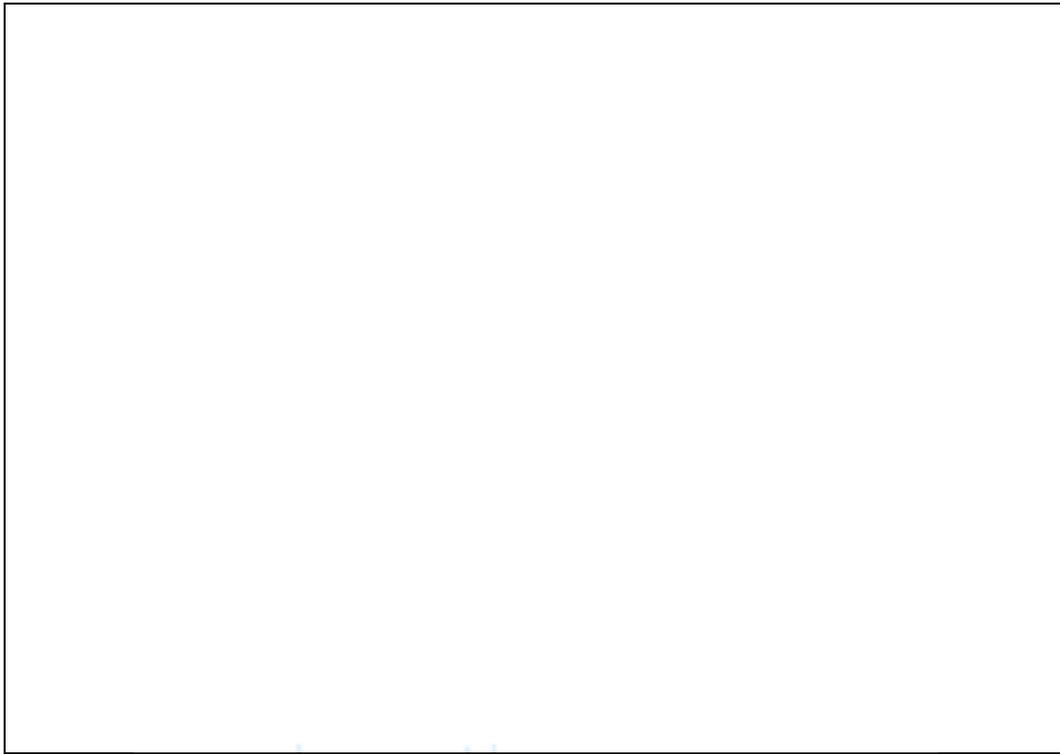
RIM (Request Initalization Mode): yêu cầu từ thứ cấp. Thứ cấp yêu cầu sơ cấp thiết lập. Một trạm thứ cấp truyền RIM, nó có thể hiển thị frames hoặc có thể trả lời SIM, DISC, TEST, XID.

UI (Unnumbered information): chấp nhận sự truyền frame dữ liệu không số hiệu (không số thứ tự). Nó thực chất là 1 dạng của connectionless - mode protocol. Sự vắng mặt của N(R), N(S) để tránh kiểm tra dòng và các frames xác nhận (ACK).

REJ (Reject): dùng để yêu cầu truyền lại những frames với frame bắt đầu có số thứ tự N(R). Các frame từ N(R) -1 về trước được chấp nhận.

SREJ (Selective Reject): với frame SREJ yêu cầu truyền lại frame sai. Mỗi frame là do sai phân biệt, chỉ có frame SREJ có thể không theo thời gian bởi vì vùng N(R) trong frame bao gồm ACK

các frame trong quá trình. Đi gửi frame SREJ thứ 2 cần phải khác biệt với frame trước vì tất cả I frames với N(s) thấp hơn N(R) của SREJ thứ 2 sẽ được chấp nhận.



Quá trình tạo thành frame HDLC. Hình vẽ cho ta tổng thể sự tạo thành HDLC trong cấp đường nối dữ liệu. Những vùng của HDLC được tạo nên bên phía truyền đi và được giải phóng frame ở bên phía thu. Việc tổ chức các thao tác rất quan trọng. Ở đây ta không xét quá trình Bit Stuffing. Các thao tác được thực hiện sau khi Bit Stuffing ở bên phát.

Dưới đây là một số ví dụ về sự trao đổi điều khiển và thông tin trong HDLC. Những ký hiệu cho sự truyền như đã cho ở trên:

cuu duong than cong . com

3.8 Kiểm soát đường nối dữ liệu (Data link control)

Chúng ta đã thảo luận vấn đề gửi những thông báo qua đường dây truyền. Để sự liên lạc có hiệu quả điều cần thiết là kiểm tra và quản lý sự trao đổi thông báo đó. Chúng ta sẽ khảo sát quá trình gửi dữ liệu qua đường dây liên lạc số. Trong chương trình này chủ yếu là kiểm tra đường nối dữ liệu và protocol nối dữ liệu. Khi ta dùng quá trình kiểm tra đường nối dữ liệu thì môi trường truyền giữa các trạm là đường nối dữ liệu.

Để thấy được các yêu cầu cho sự kiểm tra đường nối dữ liệu, chúng ta liệt kê một số yêu cầu và mục tiêu đảm bảo hiệu quả sự liên lạc số giữa hai trạm nối và truyền nhận trực tiếp.

- Đồng bộ khung (frame syn-chronization). Dữ liệu được gửi thành các khối gọi là frame. Sự bắt đầu và kết thúc mỗi frame cần vạch rõ.
- Các phương án cấu hình đường dây được dùng.
- Kiểm tra dòng: Trạm gửi cần phải gửi những frames có tốc độ không nhanh hơn tốc độ mà trạm thu có thể nhận được.
- Kiểm tra sai: Những bit sai trong khi truyền hệ thống cần phải sửa.
- Địa chỉ hóa: Với đường dây nhiều điểm, sự nhận dạng hai trạm truyền cần biết rõ.
- Kiểm soát và dữ liệu trên cùng đường.
- Quản lý đường nối: Sự khởi động, duy trì và kết thúc sự trao đổi dữ liệu yêu cầu sự phối hợp giữa những trạm. Cần phải quản lý quá trình trao đổi đó.

Những yêu cầu đó sự giao tiếp vật lý không thỏa mãn được, chúng ta sẽ phải thảo luận trong phần này.

Để bắt đầu ta giới thiệu một số chìa khóa cơ bản cho các nghi thức (protocol) kiểm soát đường nối dữ liệu như: Kiểm tra cấu hình đường dây, kiểm tra dòng, kiểm tra sai.

3.8.1 Cấu hình đường truyền:

Ba đặc trưng để phân biệt phương án cho cấu hình đường dây nối là: đồ hình, độ duplex và nguyên tắc đường dây.

FDồ hình

Như chúng ta biết, đồ hình của đường dây nối dữ liệu là sự chuẩn bị về đường nối vật lý từ trạm lên đường dây.

- Nếu như chỉ có 2 điểm thì ta gọi là: điểm nối điểm.
- Nếu như nhiều hơn 2 điểm ta gọi là nhiều điểm.

Hình 3.36 *Cấu hình truyền thông của máy tính và terminal.*

Thông thường, đường nối nhiều điểm được dùng trong trường hợp một máy tính (trạm sơ cấp) và nhiều trạm terminal (trạm thứ cấp). Nhiều đồ hình nhiều điểm được tìm thấy trong mạng nội bộ.

Hình vẽ cho ta thấy sự ưu việt của cấu hình nhiều điểm.

Khi một máy tính cần nối đến nhiều terminal, nếu mỗi terminal có đường nối điểm đối điểm với máy tính thì máy tính phải có I/O port cho từng terminal như vậy nó có các đường dây nối riêng lẻ từ máy tính đến terminal. Do đó người ta dùng cấu hình nhiều điểm.

Trong cấu hình nhiều điểm máy tính chỉ cần một port I/O, và một đường dây để nối đến các terminal. Điều đó tiết kiệm được mạch và giảm được giá thành.

Trạm sơ cấp và thứ cấp.

Người ta còn chia các trạm theo chức năng của nó. Có 2 loại trạm sơ cấp (P) và thứ cấp (S).

Trạm sơ cấp (còn gọi là trạm điều khiển): là trạm có nhiệm vụ gửi các lệnh điều khiển đến các trạm khác, dịch và trả lời những yêu cầu từ những trạm khác gửi về. Thực hiện các thao tác:

Tổ chức trao đổi dữ liệu.

Bảo đảm, giám sát sự nối.

Bảo đảm phục hồi sự nối khi có nhiễu loạn làm gián đoạn sự nối.

Hình 3.37 Phân cấp giữa các trạm.

Trạm thứ cấp (còn gọi là trạm phụ): có chức năng thực hiện các điều khiển do sơ cấp gửi đến và sau khi thực hiện nó gửi những trả lời đến sơ cấp.

Trạm chủ và tớ: Trên thực tế sơ cấp hoặc thứ cấp đều có thể là nơi phát thông tin (đặc biệt trong trường hợp phát text), lúc đó người ta gọi trạm phát thông tin là trạm chủ và trạm thu thông tin là trạm tớ. Vậy: sơ cấp và thứ cấp đều có thể là chủ và là tớ.

F Độ duplex:

Người ta dùng độ duplex để chỉ ra hướng và dòng tín hiệu. Có 3 dạng truyền:

- Truyền đơn giản: dòng tín hiệu chỉ đi một hướng. Ví dụ như một máy tính và một máy in: thông tin chỉ có thể xuất ra từ máy tính đưa đến máy in và máy in chỉ có nhiệm vụ nhận thông tin và in ra. Không thể có chiều ngược lại. Cách truyền đơn giản (1chiều) không được dùng rộng rãi bởi nó không có khả năng gửi tín hiệu báo sai hoặc tín hiệu kiểm tra trở về nguồn.
- Truyền 2 chiều gián đoạn (half duplex) trạm có thể truyền và nhận thông tin nhưng không đồng thời.
- Truyền 2 chiều toàn phần: 2 trạm có thể đồng thời gửi và nhận dữ liệu của nhau. Có nghĩa là 2 chiều đồng thời.

Với tín hiệu số yêu cầu phải có định hướng khi truyền. Khi truyền 2 chiều toàn phần yêu cầu có 2 đường khác nhau, trong khi với cách truyền 2 chiều gián đoạn chỉ cần có một đường truyền.

Với tín hiệu analog, vấn đề 2 chiều phụ thuộc vào tần số. Nếu một trạm truyền và nhận trên cùng tần số, bắt buộc phải xử dụng 2 chiều gián đoạn. Nếu một trạm truyền ở một tần số và nhận ở tần số khác nó có thể xử dụng 2 chiều toàn phần.

Nhiều khả năng có thể tổ hợp giữa đồ hình và khả năng duplex. Hình vẽ cho ta một số trường hợp. Hình vẽ cũng cho ta thấy một trạm sơ cấp đơn giản (P) có thể có một hay nhiều trạm thứ cấp (S).

Với phương án điểm - điểm: có thể dùng cách truyền 2 chiều gián đoạn hoặc 2 chiều toàn phần.

Với đường nối nhiều điểm. Có 3 cấu hình.

- Sơ cấp: 2 chiều toàn phần. Thứ cấp: gián đoạn.

- Cả 2 sơ cấp và thứ cấp đều gián đoạn.
- Cả 2 sơ và thứ cấp đều toàn phần.

Hình 3.38 Cấu tạo đường truyền.

F Nguyên tắc đường dây:

Một số nguyên tắc cần thiết cho đường nối truyền. Trên đường dây hai chiều gián đoạn, tại một thời điểm chỉ có một trạm truyền, trạm đối diện có thể là hai chiều gián đoạn hoặc hai chiều toàn phần. Một trạm chỉ truyền nếu nó biết trạm đối diện nó sẵn sàng nhận.

Đường nối điểm - điểm:

Với đường dây nối điểm - điểm thì đơn giản. Ta giả thiết trước tiên đường nối hai chiều gián đoạn với trạm kia được khởi động trao đổi và nó được chỉ ra như hình vẽ 3-39. Nếu trạm muốn gửi dữ liệu đến trạm khác, yêu cầu đầu tiên là trạm kia sẵn sàng nhận.

Trạm thứ 2 trả lời ACK để chỉ ra nó đã sẵn sàng. Trạm thứ nhất sẽ gửi một số dữ liệu trong dạng frame.

Trong trường hợp truyền không đồng bộ, dãy dữ liệu sẽ là dòng các ký tự. Trong một số trường hợp sau khi truyền một số dữ liệu, trạm thứ nhất sẽ tạm dừng để chờ kết quả. Trạm thứ hai chấp nhận kết quả (ACK).

Trạm thứ nhất sẽ truyền tín hiệu kết thúc truyền EOT và chấm dứt sự trao đổi, hệ thống trả về trạng thái ban đầu, như hình vẽ.

Có 3 pha trong suốt quá trình điều khiển.

- *Thiết lập:* Kết thúc xác lập trạm nào phát, trạm nào nhận và trạm nhận chuẩn bị nhận.
- *Truyền dữ liệu:* Dữ liệu được truyền thành một hay nhiều khối được chấp nhận.
- *Kết thúc:* Kết thúc sự nối logic (tương quan giữa trạm phát và thu).

Chúng ta có thể xem 2 trạm trên là sơ cấp và thứ cấp. Trạm sơ cấp có nhiệm vụ thiết lập sự trao đổi (ví dụ như một máy tính) và trạm còn lại là terminal (thứ cấp).

Hình 3.39 Kiểm tra đường nối điểm - điểm.

Hình đã vẽ cho ta thứ tự lần lượt để trạm sơ cấp truyền dữ liệu cho trạm thứ cấp. Trạm sơ cấp (trạm 1) phát ENQ để báo cho trạm thứ cấp biết nó cần truyền dữ liệu. Khi trạm thứ cấp chưa chấp nhận nó truyền NAK cho sơ cấp. Ngược lại, nó truyền ACK. Nhận được ACK từ trạm thứ cấp, trạm sơ cấp sẽ truyền một frame cho trạm thứ cấp. Khi nhận ACK nó sẽ đưa quá trình truyền vào kết thúc. Nếu như trạm thứ cấp muốn truyền dữ liệu nó phải chờ trạm sơ cấp yêu cầu và chỉ có lúc đó mới được vào pha truyền dữ liệu.

Nếu một đường nối là full - duplex, dữ liệu và tín hiệu điều khiển có thể truyền đồng thời về 2 phía. Ta sẽ thảo luận về các ưu điểm của nó khi xét về kiểm tra dòng và kiểm tra sai.

Đường nối nhiều điểm:

Sự lựa chọn nguyên tắc đường dây cho đường nối nhiều điểm phụ thuộc trước tiên nó là trạm sơ

cấp hay thứ cấp. Nếu như nó là trạm sơ cấp thì sự trao đổi dữ liệu chỉ là sự trao đổi giữa sơ cấp và thứ cấp chứ không phải giữa 2 trạm thứ cấp. Để hiểu rõ hơn trường hợp đó ta sử dụng 2 khái niệm:

- Poll: Sơ cấp yêu cầu dữ liệu từ thứ cấp.
- Select: Sơ cấp có dữ liệu cần gửi và thông báo cho thứ cấp dữ liệu sẽ đến.

Hình vẽ chỉ cho ta các trường hợp đó.

Hình a cho ta thấy sơ cấp yêu cầu thứ cấp gửi thông báo nhưng thứ cấp không có gì để gửi nên nó gửi NAK. Tổng thời gian như chỉ ra ở hình vẽ.

Hình b cho ta trường hợp yêu cầu có hiệu quả. B gửi thông báo cho A.

Hình c chỉ cho ta chức năng Select. A yêu cầu gửi thông báo cho B. B chấp nhận và A gửi xong, B báo kết quả nhận được tốt.

Hình d chỉ cho ta trường hợp fast select.

Hình 3.40 Quá trình POLL và SELECT.

Trong trường hợp này thông báo select gửi kèm cả dữ liệu. Sự nhận được đầu tiên từ thứ cấp là ACK chỉ ra rằng trạm thứ cấp sẵn sàng nhận thông tin và đã nhận xong. Fast select thông thường dùng trong trường hợp ta có thông báo ngắn và thời gian truyền không dài hơn thời gian trả lời.

Chú ý rằng trong cả 4 trường hợp trên đều do sơ cấp yêu cầu thứ cấp.

Như phần trước cho thấy. Với cách nối đường dây nhiều điểm, một sơ cấp có nhiều thứ cấp. Trên đường dây có n điểm thời gian poll cần n lần cho n địa chỉ khác nhau. Người ta sử dụng Hub polling. Kỹ thuật này yêu cầu các thứ cấp cùng tham gia quá trình polling. Hai đường dữ liệu đồng thời được dùng và mỗi thứ cấp yêu cầu nhận ở cả 2 đường đồng thời. Thao tác như sau:

Sơ cấp truyền poll đồng thời đến nhiều thứ cấp. Nếu thứ cấp có dữ liệu truyền nó truyền dữ liệu cho sơ cấp và sau đó nó gửi poll cho thứ cấp tiếp theo trên đường dây. Nếu như thứ cấp không có dữ liệu để truyền, nó truyền trực tiếp poll cho trạm tiếp theo. Thứ cấp cuối cùng trên đường dây gửi poll cho sơ cấp và bắt đầu chu trình mới trong suốt quá trình đó. Sơ cấp có thể gửi dữ liệu đến những trạm thứ cấp trên đường dây output.

Một dạng nữa của đường dây nguyên tắc là: cạnh tranh. Trong cách này, không có sơ cấp mà chỉ là chọn những trạm, một trạm có thể truyền nếu như đường dây rảnh, nếu không nó phải chờ. Phương pháp này được dùng trong mạng LAN và hệ thống vệ tinh.

Đặc trưng của nguyên tắc đường dây nhiều điểm là cần có địa chỉ. Trong trường hợp polling, sự truyền từ sơ cấp cần có địa chỉ trạm thứ cấp đến, sự truyền từ thứ cấp cũng cần chỉ ra trạm thứ cấp. Trong trường hợp các trạm đẳng quyền sự truyền và sự nhận cũng cần phải nhận dạng. Ta có 3 trường hợp:

- Điểm - điểm: Không cần địa chỉ.
- Sơ cấp - thứ cấp nhiều điểm: Một địa chỉ thứ cấp cần thiết để nhận dạng thứ cấp.
- Nhiều điểm đẳng quyền: Cần 2 địa chỉ để nhận dạng phát và thu.

Trong thực tế, trường hợp thứ nhất là cá biệt của trường hợp thứ 2. Nhiều protocol kiểm tra đường

nối dữ liệu yêu cầu 1 địa chỉ cũng dùng cho điểm - điểm.

Trường hợp đăng quyền ta thấy như trong mạng LAN. Ở đây ta chỉ khảo sát 2 loại đầu.

3.8.2 Kiểm tra dòng (Flow control):

Kiểm tra dòng thông tin là kỹ thuật bảo đảm cho một trạm truyền dữ liệu cho trạm thu không bị tràn, bảo đảm trạm thu tạm giữ dữ liệu với độ dài cực đại. Khi dữ liệu đến bộ thu phải có được giá trị của quá trình trước khi xóa bộ nhớ đệm để nhận thông tin tiếp theo. Khi không có kiểm tra dòng thì bộ nhớ đệm của bộ thu có thể bị tràn vì còn dữ liệu cũ.

Giả thiết, ta kiểm tra dòng trong trường hợp không có lỗi khi truyền. Cách làm của ta như hình đã cho.

Hình 3.42 Cách truyền các frame.

Theo thứ tự thời gian ta có thể nhìn thấy sự phụ thuộc, sự đúng đắn và tương quan giữa gửi và nhận.

Mỗi mũi tên đặc trưng cho sự truyền một frame dữ liệu đơn giản giữa 2 trạm.

Như ta biết trong khi truyền đồng bộ, dữ liệu được truyền bằng những frame liên tiếp mà mỗi frame gồm có dữ liệu và một số thông tin kiểm tra, giả thiết tất cả các frames đều nhận được đầy đủ không có frame nào bị mất và bị sai. Hơn nữa các frames đều đến đúng thứ tự sắp xếp gửi đi. Tuy nhiên, mỗi frame được truyền bỏ qua thời gian trễ trước khi nhận.

Dạng kiểm tra đơn giản cho kiểm tra dòng là: Kiểm tra dòng dừng và chờ, nó làm việc như sau: Nguồn truyền một frame, sau khi nhận bộ phận nhận sẽ gửi tín hiệu báo chấp nhận frame khác với tín hiệu ACK cho frame trước. Nguồn phải chờ cho đến khi nó nhận được ACK mới truyền frame mới. Bộ phận nhận vì vậy dừng nhận dữ liệu để phát ACK. Quá trình làm việc như vậy có thể diễn ra với một thông báo dài. Tuy nhiên, thường nguồn ngắt những thông báo dài đó thành các khối nhỏ và truyền dữ liệu đó thành nhiều frames, vì những lý do sau:

- Với độ dài càng lớn trong khi truyền sẽ dễ sinh ra sai, cần truyền lại nguyên cả khối. Với các khối nhỏ ít sai.
- Nếu có sai ta cũng chỉ truyền lại một khối nhỏ, ít tốn thời gian hơn.
- Trên đường nối nhiều điểm người ta không cho phép một trạm chiếm nhiều thời gian quá bởi vì nó sẽ làm chậm trễ cho trạm khác.
- Độ lớn của bộ nhớ đệm ở bộ thu có hạn.

Chúng ta tạm thời đưa ra những lý do trên cho việc phải truyền nhiều frames cho một thông báo. Cụ thể ta sẽ khảo sát ở phần sau.

Nghi thức cửa sổ trượt (The Sliding Window Protocol).

Vấn đề chính cần quán triệt là tại một thời điểm chỉ có một frame được truyền. Ta giả thiết với điều kiện là độ dài bit của đường nối lớn hơn độ dài của frame ($a > l$; a là độ dài bit đường truyền và l là độ dài bit của frame), sự truyền không bị ảnh hưởng gì. Điều cần giải quyết là nhiều frames cùng được truyền liên tiếp tại một thời điểm nào đó.

Ta có thể khảo sát sự làm việc của 2 trạm. Có 2 trạm A và B nối qua đường nối duplex. Trạm B có bộ nhớ đệm cho n frames. Có nghĩa là B có thể nhận được n frames và A có thể truyền n frames mà không chờ ACK. Để đánh dấu frames nào đã được ACK mỗi frame có nhãn với số thứ tự. B chấp nhận 1 frame và gửi ACK với số thứ tự frame tiếp theo. Với ACK đó thông báo rằng B chuẩn bị nhận n frames tiếp theo bắt đầu từ số thứ tự đã chỉ.

Ví dụ: B có thể nhận frames 2,3,4 nhưng nó giữ ACK cho đến khi nào frame 4 đã đến, nó sẽ cấp ACK với số thứ tự 5 (ACK₅), có nghĩa là B đã chấp nhận các frames 2,3,4. A gửi một bảng số thứ tự các frame mới mà nó lần lượt gửi, và B giữ bảng số thứ tự mà nó chuẩn bị nhận. Mỗi một bảng đó ta có thể coi như là cửa sổ của các frames. Những thao tác đó được gọi là phương pháp kiểm tra đồng bằng cửa sổ trượt.

Một sự hạn chế rất quan trọng cần biết là: do số thứ tự nó chiếm một vùng trong frame là N(s) và N(r) trong vùng C của Protocol, có độ lớn giới hạn. Một vùng có k bit, giới hạn của cửa sổ thứ tự biểu diễn là từ 0 đến 2^k-1 . Tương ứng với nó số thứ tự theo modulo 2^k , như vậy sau giá trị 2^k-1 là 0.

Hình vẽ mô tả cho ta quá trình biến đổi cửa sổ trượt.

Nếu trong vùng C của protocol N(s) và N(r) có 3 bit để đánh số thứ tự như vậy frames được đánh số từ 0 đến 7.

Hình 3.43 Nguyên tắc cửa sổ trượt.

Hình vẽ (3-43) chỉ cho ta có thể truyền 7 frames bắt đầu bằng frame 0. Mỗi lần truyền một frame thì bóng sẽ chuyển đi.

Hình vẽ (3-44) cho ta ví dụ cụ thể về sự dịch chuyển số thứ tự frame ở bên phát và thu.

Ví dụ cho ta vùng số thứ tự là 3 bit và độ cực đại của độ lớn cửa sổ là 7. Ban đầu A và B có sự chỉ thị trên cửa sổ là A có thể truyền 7 frames, bắt đầu từ frame 0 (F₀). Sau khi truyền 3 frames (F₀, F₁, F₂) không có ACK. Số vùng cửa sổ nhảy đến frame thứ 4. Cửa sổ chỉ ra là A còn có thể truyền 4 frames bắt đầu với frame 3. B truyền cho A tín hiệu ACK₃, điều đó có nghĩa là: } Tôi đã nhận tất cả frames cho đến frame 2 và tôi sẵn sàng nhận frame 3. Tôi chuẩn bị nhận 7 frames bắt đầu với frame 3~ .

Với ACK₃ thì A ghi nhận truyền 7 frames, bắt đầu từ F₃ A xúc tiến truyền F₃, F₄, F₅ và F₆. B báo lại ACK₄ sau khi nhận F₃ và có thể nhận sự truyền từ F₄ đến F₂. Nhưng tại thời điểm ACK₃ đến A thì F₄, F₅, F₆ đã truyền đi. Kết quả là A chỉ có thể mở cửa sổ cho phép truyền 4 frames bắt đầu từ F₇.

Sự kiểm tra đó cho ta một cách kiểm tra đồng: Bộ phận thu muốn chỉ có khả năng nhận n frames mà một frame đã được chấp nhận. Thêm vào đó một số protocol thừa nhận việc cắt dòng frames từ một trạm khác bằng cách gửi một tín hiệu: không sẵn sàng nhận (RNR) thông báo để cấm truyền các frames tiếp theo. Ví dụ RNR₅ có nghĩa là: Tôi đã nhận tất cả các frames ... cho đến F₄ nhưng tôi không chấp nhận tiếp một frame nào nữa bắt đầu từ frame 5. Tại thời điểm phụ nào đó, trạm bắt buộc phải gửi một tín hiệu ACK bình thường để mở lại cửa sổ (ACK₅).

Hiệu quả của đường dây bây giờ phụ thuộc vào các giá trị: N , và a .

N : độ lớn của cửa sổ.

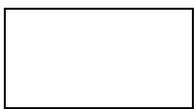
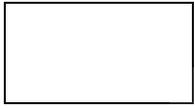
a : thời gian lan truyền trên đường dây từ A đến B.

Hình 3.44 Ví dụ về cửa sổ trượt.

Ta giả thiết độ dài bit của một frame có giá trị là 1. Hình vẽ cho ta hiệu quả của đường dây full duplex điểm - điểm. Trạm A bắt đầu gửi thông tin tại t_0 . Thời điểm bit đầu tiên của frame đầu tiên đến B là t_0+a . Frame đầu tiên mà B nhận hết tại thời điểm t_0+a+1 . Sau khi nhận được B truyền ACK₁, thời điểm ACK₁ đạt đến A là: t_0+2a+1 .

Hình 3.45 Thời gian cho cửa sổ trượt.

Có 2 trường hợp có thể xảy ra:

-  ACK cho frame 1 đến A trước khi độ lớn cửa sổ bị chiếm hết. A có thể truyền tiếp tục mà không dừng. Chu kỳ chỉ hoàn thành khi nào truyền hết n frames.
-  Trạm A đã truyền hết N frames và không có n frames để truyền tiếp, trên đường dây bị trống. Chu kỳ hoàn thành trong thời gian $2a + 1$.

3.8.3 Phát hiện và sửa sai

3.6.3.1 Mở đầu

Những thông tin truyền từ A ® B như ta đã thấy thông thường bị sai: $d_i(t) \text{ ® } d_i(t) \pm d'_i(t)$

$$d_i(t) = (\dots, d_{i-1}, d_i, d_{i+1} \dots)$$

Sự sai số đó do nhiều nguyên nhân: đường dây truyền, lưu lượng truyền, loại mã dùng, loại điều chế, loại thiết bị phát, loại thiết bị thu... Thường sai số đó cho phép trong khoảng: 10^{-4} 10^{-7} và nhóm sai phụ thuộc loại mạch.

Ví dụ: một trang thông tin ta dùng phương pháp để nén lại, thông thường còn 10^5 ký hiệu nhị phân.

Giả thiết các bit sai phân bố như hình vẽ.

Hình 3.46 Sai số do đường dây truyền.

Nếu trang thông tin truyền trên đường dây mạng điện thoại công cộng với độ sai số 10^{-4} thì tổng số các ký hiệu nhị phân sai là: $10^5/10^{-4} = 10$

Mục đích chương này là:

- Miêu tả phương pháp thường dùng để bảo vệ thông tin phát đi, để chống sai do đường dây

gây nên.

- Những phương pháp bảo vệ được dùng phần cứng (bộ mã hóa ở bộ phận phát đi và giải mã ở bộ phận thu). Phương pháp chung để giải quyết là đặt giữa nguồn phát và bộ phận thu một thiết bị để mã hóa và giải mã.

Hình 3.47 Các tín hiệu trên đường truyền.

Về nguyên lý ta có:

- Bộ phận mã hóa chuyển dãy ký hiệu nhị phân $\{ d_i \}$ thành dãy nhị phân khác với: $\dots, a_i, a_{i+1} \dots a_i \in \{0, 1\}$

Sau khi truyền ta có: $\{ \hat{a}_i \}$ với $\{ \hat{a}_i \} \in \{0, 1\}$

Sự chuyển đổi đó được gọi là mã hóa. Giữa $\{ a_i \}$ và $\{ \hat{a}_i \}$ có sự khác nhau do sai số khi truyền. Người ta nói rằng:

Sai số $\hat{a}_i - a_i$ là độc lập (không phụ thuộc), nếu dãy ngẫu nhiên $\{ \hat{a}_i - a_i \}$ là dãy độc lập. Có khi dãy này không độc lập và sự sai số thành nhóm, người ta gọi là sai số theo gói có độ dài l.

- Bộ phận giải mã thực hiện việc biến đổi dãy \hat{a}_i thành d_i và ta gọi thao tác đó là giải mã.

Có nhiều cách tạo sự nối dữ liệu với bộ mã hóa và giải mã. Một trong những cách như sơ đồ sau:

Hình 3.48 Tín hiệu trên đường truyền có điều chế.

Sau khi truyền để lấy lại thông tin cần thiết ta cần giải mã.

Việc dùng loại mã nào được mã hóa và hệ thống giải mã nào phụ thuộc vào hệ thống truyền số liệu.

Có được $\{ d_i \}$ ở bộ phận thu, $\{ d_i \}$ có thể sai. Khi phát hiện sai $\{ d_i \} \neq d_i$. Cách bảo vệ đơn giản nhất là báo động có sai số, cần phải sửa sai. Thường có 2 cách:

Nếu bộ giải mã tự sửa được nó sẽ sửa trực tiếp.

Nếu bộ giải mã không tìm được sai thì cần phải truyền lại một bộ phận dữ liệu để thực hiện sự sửa sai. Người ta còn gọi cách đó là sửa sai bằng cách truyền lại và gọi tắt là ARQ (Automatic Repeat Request)

Sự truyền lại đó có thể thực hiện bằng 3 cách như được chỉ ra ở hình vẽ 4.4.

Những phương pháp sửa sai ARQ ta sẽ khảo sát kỹ trong chương sau. Trong chương này ta chỉ khảo sát về các phương pháp kiểm tra frames thông tin truyền và các loại mã giảm bớt sai trong khi truyền.

Hình 3.49 Những khả năng phát hiện và sửa sai.

3.6.3.2 Những ví dụ đơn giản về mã:

a. Kiểm tra chẵn lẻ theo ký tự

Trong chương 1 chúng ta đã thấy trong mã số 5 của CCITT (mã ASCII) mỗi ký tự gồm có 7 bit và 1 bit kiểm tra. Với kiểm tra chẵn, giá trị của bit kiểm tra là 0 nếu số lượng các bit có giá trị 1 trong 7 bit là chẵn và nó có giá trị 1 trong trường hợp ngược lại. Còn nếu kiểm tra lẻ thì ngược lại. Thông thường người ta sử dụng kiểm tra chẵn và bit kiểm tra gọi là P. Giá trị kiểm tra đó cho phép ở bộ thu phát hiện ra những sai sót đơn giản.

Ví dụ:

Ký tự	Mã ASCII	Từ mã	
A	1000001	10000010	
E	1010001	10100011	bit P
V	0110101	01101010	

Vấn đề là trong khi dùng bit kiểm tra, thông thường nhiều xung kéo dài không phải chỉ có độ dài 1 bit mà là nhiều bit, nhất là khi tốc độ truyền tăng cao. Với cách kiểm tra trên nếu số bit sai là chẵn (2,4,6 bit) ta không thể phát hiện được.

Hình 3.50 Khả năng sinh bit sai thứ hai.

Hình vẽ cho ta thấy khả năng sinh ra sai số ở bit thứ hai sau bit đầu tiên với đường truyền có tốc độ 1200 bps. Để giảm bớt sai số đó người ta dùng mã kiểm tra khối LRC (BCC).

b. Kiểm tra chẵn lẻ theo độ dài và ngang

Quá trình bảo vệ ký tự có thể giảm thiểu sai nếu như người ta dùng cách kiểm tra theo chiều ngang (theo từng ký tự - VRC) và đồng thời theo chiều dọc (LRC) khi ta truyền một khối thông tin người ta gọi là kiểm tra khối (BCC). Nếu như một sai sót nào đó sinh ra tại 1 bit bất kỳ, dựa vào các bit kiểm tra ngang và dọc ta có thể xác định được tọa độ của nó và đương nhiên như vậy ta có thể tìm ra và sửa nó.

Một frame coi như là một khối ký tự sắp xếp có hai chiều. Mỗi ký tự có bit kiểm tra chẵn lẻ (P). Nếu ta sắp các bit của các ký tự đúng vị trí tương ứng từ trên xuống thì ta có một khối các ký tự. Nếu tính theo chiều ngang.

$$R_j = b_{1j} \ b_{2j} \ \dots \ b_{nj}$$

với

R_j : bit kiểm tra thứ tự thứ j.

b_{ij} : bit thứ i của ký tự thứ j.

n : số lượng bit trong một ký tự.

Nếu tính theo chiều dọc ta có:

$$C_i = b_{i1} \ b_{i2} \ \dots \ b_{im}$$

với

C_i : bit kiểm tra cột thứ i .

m : số lượng ký tự trong frame.

Người ta gọi R_j là VRC, còn C_i là LRC. Hình 4.6 cho ta ví dụ đó.

Hình 3.51 Kiểm tra VRC và LRC.

Tuy nhiên phương pháp này cũng không phải hoàn toàn hiệu quả. Giả sử bit thứ nhất (b_{11}) và bit thứ ba (b_{13}) của ký tự thứ nhất bị sai, như vậy R_1 sẽ không sai. Tuy nhiên, nếu dùng kiểm tra LRC ta sẽ thấy bit thứ nhất (c_1) và bit thứ ba (c_3) của LRC bị sai. Ta biết được sự truyền bị sai nhưng không biết sai ở ký tự nào. Bây giờ giả thuyết rằng bit thứ nhất và bit thứ ba của ký tự thứ năm cũng bị sai đồng thời với bit thứ nhất và bit thứ ba của ký tự thứ nhất, lúc đó ta sẽ không phát hiện được LRC bị sai, kết quả thu được có sai nhưng ta không phát hiện được.

Như vậy VRC và LRC chỉ phát hiện được số lẻ bit theo chiều ngang hay theo chiều dọc bị sai. Đó cũng là lý do tại sao trước kia có một số protocol dùng BCC để kiểm tra mà hiện tại các protocol mới không sử dụng. Để đạt được sự kiểm tra tốt hơn trong protocol SDLC-HDLC sử dụng FCS là mã CRC để kiểm tra sai.

3.8.3.3 Mã vòng và tính CRC

Việc nghiên cứu mã tuyến tính trên đây nhằm mục đích là sao ta có thể tìm một họ mã có đủ các tính chất sau:

- mã hóa và giải mã đơn giản.
- khả năng tìm và sửa lỗi độc lập theo các gói đã truyền.

Ở phần trên chúng ta đã thấy việc kiểm tra theo VRC hoặc LRC còn những sai sót mà ta khó tránh khỏi, với mã VRC, nếu sai số theo chiều V là số chẵn bit thì bit kiểm tra cũng không có gì thay đổi.

Tương tự ta có nếu số bit sai là $2n$, ta cũng không thể phát hiện gì được.

Mã vòng với kiểm tra CRC (Cyclic Redundancy Check) có nhiều hiệu quả hơn trong việc thông báo lỗi.

a. Định nghĩa:

Mã vòng (n,k) là một mã tuyến tính (n,k) với sự hoán vị vòng của một từ mã là một từ mã.

$$c = (C_{n-1}, \dots, C_0) \text{ } \Upsilon \text{ } C^2 \text{ } (C_{n-2}, \dots, C_0, C_{n-1}) \text{ } \Upsilon \text{ } C$$

Ta có thể viết một từ mã dưới dạng đa thức:

$$C(x) = C_{n-1} x^{n-1} + \dots + C_1 x + C_0.$$

và theo định nghĩa trên ta có:

$$C(x) \text{ } \Upsilon \text{ } C^2 \text{ } x^1. C(x) \text{ modulo } (x^n+1) \text{ } \Upsilon \text{ } C$$

b. Tính chất:

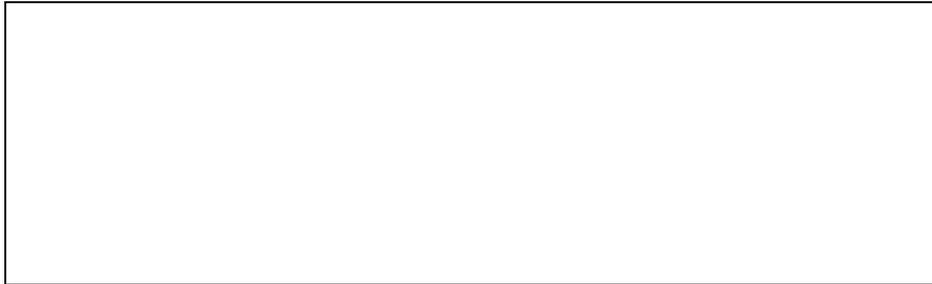
Đa thức phát

Sự nhận biết ma trận phát G của một mã vòng được đưa đến từ sự nhận biết của đa thức bậc n-k.

Do sự chính xác của tính chất sau:

Tính chất: Tất cả các từ mã vòng (n, k) là sự nhân của đa thức g(x) bậc (n-k) kết hợp từ dòng cuối của G và gọi là đa thức phát.

Ví dụ: Với mã tuyến tính biểu diễn ở trên, ta có ma trận



Vậy:

$$g(x) = x^3 + x + 1$$

c. Tạo CRC:

Phương pháp tạo CRC bao gồm việc dịch thông báo sang trái và chia cho một hàm cho trước với modul 2. Kết quả dư lại của phép chia chính là CRC và được truyền theo thông báo. Bên thu sau khi nhận được thông báo người ta cũng đem chia cho hàm biết trước như bên phát. Nếu kết quả bằng 0, phép truyền không có sai số.

Trong thực tế, nếu dùng cách kiểm tra CRC kết hợp với ARQ thì hiệu quả rất cao.

Nếu dùng CRC với 16 bit thì có thể chỉ sai 1 bit khi ta truyền 10^{14} bit.

Cách tính CRC gồm bốn bước. Ví dụ cần tính CRC cho thông báo $M(x) = 110101$.

Ta có thể biểu diễn $M(x)$ dạng đa thức.

Các phép tính CRC được thực hiện 4 bước sau với các phép tính thực hiện với modul 2.

Bước 1:

Chuyển thông báo nhị phân thành đa thức

$$M(x) = (1).x^5 + (1).x^4 + (0).x^3 + (1).x^2 + (0).x^1 + (1).x^0$$

Bậc cao nhất là $(n-1) = 5$

$$M(x) = x^5 + x^4 + x^2 + 1$$

Ta chọn độ dài của CRC. Nếu ta chọn CRC có độ dài là c bit.

Ta chọn ra hàm $G(x) = x^c + 1$. [Thường hàm $G(x)$ cho trước tùy giá trị c , và có nơi gọi $G(x)$ là $P(x)$].

Bước 2:

Nhân $M(x).x^c/G(x)$

Ví dụ ta chọn $c = 3$

Ta có:

Bước 3:

Thực hiện phép tính $M(x).x^0/G(x)$ với modul 2, ta có được:

cuu duong than cong . com

Bước 4:

Lập $T(x)$

Với $T(x) = x^c.M(x) + R(x)$

$T(x)$ chính là bảng thông báo cần truyền đi.

Ví dụ: Cần truyền thông tin 110101

1. Tạo $M(x)$

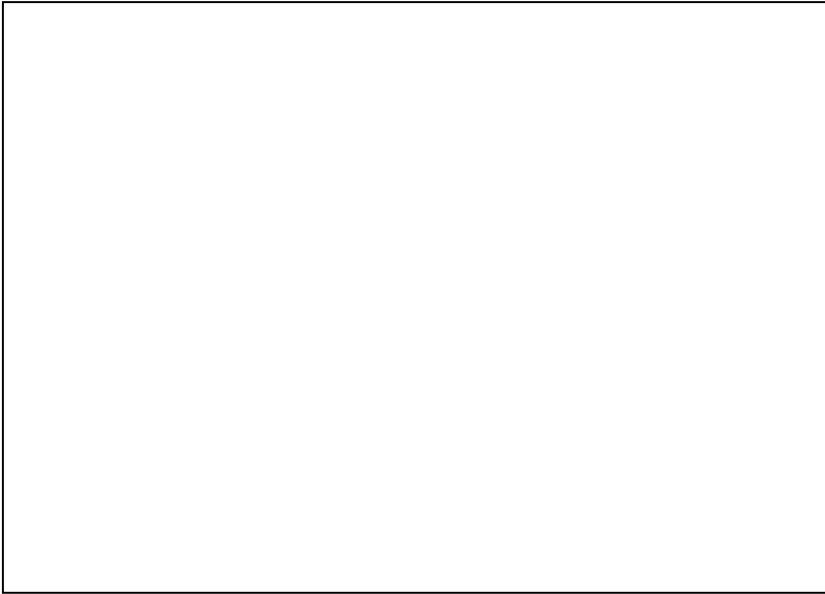
$$M(x) = x^5 + x^4 + x^2 + 1$$

cuu duong than cong . com

Chọn $C = 3$ ® $G(x) = x^3 + 1$

2. Tạo $M(x).x^3/G(x)$

3. Tính



4. Tính $T(x) = x^C M(x) + R(x)$

$$= x^8 + x^7 + x^5 + x^3 + x + 1$$

Thông tin cần truyền là:

1 1 0 1 0 1 0 1 1

d. Thu và kiểm tra CRC:

Để kiểm tra sai số khi truyền, bộ phận thu đem khối thông tin thu được chia cho $G(x)$ theo modul 2.

Nếu phần dư còn lại là 0, mã nhận được là đúng (không sai), nếu phần dư khác 0 thì kết quả nhận được không đúng.

Thử CRC

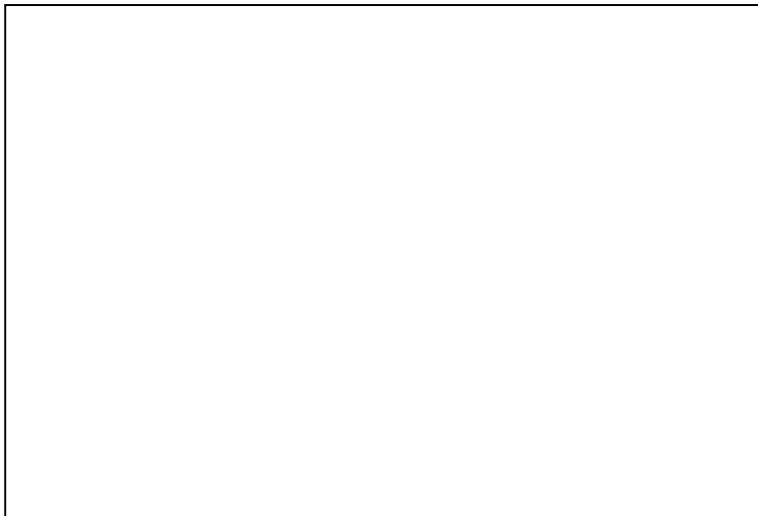
Ta có:

Và

Ta có:

Mà

Ví dụ ta có:



e. Mạch tạo mã CRC:

Để tạo mã CRC người ta có thể dùng phần mềm, tuy nhiên trên thực tế để nhanh chóng hơn và giảm bớt thời gian sử dụng cho m P, người ta thường dùng phần cứng để tạo CRC và kiểm tra.

Thao tác cộng MOD-2 và dịch có thể dùng một mạch ghi dịch phản hồi về một cổng EXOR. Số lượng cột của bộ ghi dịch phụ thuộc vào giá trị C đã chọn cho $G(x)$ và cũng là số bit cho mã CRC.

Hình vẽ cho ta mạch tạo CRC 3 bit.

Hình 3.52 Mạch tạo CRC dùng bộ ghi dịch với $G(x)=x^3-1$.

Ví dụ 2: Ta cần truyền thông tin:

$M(x) = 1010001101$ với hàm

$$G(x) = x^5 + x^4 + x^2 + 1$$

Ta có mạch dịch và các bước thực hiện như hình vẽ.

Hình 3.53 Mạch ghi dịch với hàm $G(x) = x^5 + x^4 + x^2 + 1$.

Người ta có thể tạo mã CRC dài 12 bit, 16 bit, 32 bit ... Vấn đề là chọn $G(x)$ sao cho phù hợp.

Thường cho 16 bit người ta chọn:

$$G(x) = x^{16} + x^{15} + x^2 + 1$$

Hoặc $G(x) = x^{16} + x^{12} + x^5 + 1$

Hình 3.54 Mạch dịch với $G(x) = x^{16} + x^{15} + x^2 + 1$.

Mạch tạo CRC cho 16 bit như hình vẽ:

Trong mạng Ethernet, người ta truyền 1500 bytes cần CRC là 32 bit.

Và hàm $G(x)$ được sử dụng là:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Và dùng Intel 82586 coprocessor on chip.

Từ những ví dụ trên ta có thể nhận xét một cách máy móc rằng: để tạo ra mạch dịch với các cổng EXOR và các FF cho một hàm có sẵn ở đầu vào các thành phần x có giá trị 1 sẽ có cổng EXOR, tín hiệu vòng về sẽ được đưa đến các mạch EXOR đó, số lượng các FF trong bộ ghi dịch là C đã chọn của hàm $G(x)$.

3.8.3.4 Mã sửa sai - Mã Hamming

Vấn đề là: nếu khi mã hóa mỗi ký tự ta sử dụng n bit thì với n bit ta có được 2^n ký tự. Nhưng nếu như vậy thì vấn đề tìm sai và sửa sai không thể thực hiện được.

Nếu ta thêm 1 bit để kiểm tra thì có thể tìm được những sai sót đơn giản (như VRC, LRC).

Ông Hamming của phòng thí nghiệm Bell đưa ra một đề án về mã khoảng cách.

Theo Hamming: khoảng cách Hamming giữa 2 từ mã bằng số lượng bit khác nhau giữa chúng.

Ví dụ:

$$B = 1000010$$

$$C = 1000011 \text{ Khoảng cách là } 1$$

$$\text{Nếu có } P \otimes B = 01000010$$

$$C = 11000011 \text{ Khoảng cách là } 2$$

Điều đó cho ta khả năng ở bộ thu sau khi kiểm tra có thể sửa sai trong một giới hạn nào đó bit sai.

Hamming đã phát triển nó thành mã Hamming. Để có thể thấy được rõ ràng tác dụng đó, ta có thể lấy ví dụ cho hệ 16 ($0_{16} \text{ } ^{3/4} F_{16}$). Thông thường để biểu thị từ 0 – F ta chỉ cần 4 bit. Nếu dùng 7 bit để biểu thị, ở bộ thu chúng ta có thể phát hiện được 2 bit sai và có thể sửa từng bit.

7 bit của từ mã bao gồm: 4 bit biểu thị thông báo $m_3 m_2 m_1 m_0$ và 3 bit kiểm tra $C_2 C_1 C_0$.

Mỗi từ mã có dạng $m_3 m_2 m_1 C_2 m_0 C_1 C_0$.

Các giá trị của $C_2 C_1 C_0$ được tính như sau:

$$C_0 = m_0 \oplus m_1 \oplus m_3$$

$$C_1 = m_0 \oplus m_2 \oplus m_3$$

$$C_2 = m_1 \oplus m_2 \oplus m_3$$

Ví dụ: mã cho chữ A_{16} là: $m_3 m_2 m_1 m_0 = 1010 (A_{16})$

$$C_0 = \overset{\circ}{0} \overset{\circ}{1} \overset{\circ}{1} = 0$$

$$C_1 = \overset{\circ}{0} \overset{\circ}{0} \overset{\circ}{1} = 1$$

$$C_2 = \overset{\circ}{1} \overset{\circ}{0} \overset{\circ}{1} = 0$$

Vậy A theo mã Hamming là: 1010010

Khi bộ phận thu thu được một từ mã, có thể tính lại 3 giá trị kiểm tra: $P_0; P_1; P_2$ như sau:

$$P_0 = C_0 \overset{\circ}{m_0} \overset{\circ}{m_1} \overset{\circ}{m_2} + m_3$$

$$P_1 = C_1 \overset{\circ}{m_0} \overset{\circ}{m_2} \overset{\circ}{m_3} + m_1$$

$$P_2 = C_2 \overset{\circ}{m_1} \overset{\circ}{m_2} \overset{\circ}{m_3} + m_0$$

Nếu $P_0 = P_1 = P_2 = 0$ thì phép truyền không gây sai số. Nếu có sai ta dựa vào giá trị nhị phân của $P_2; P_1; P_0$ mà xác định vị trí sai.

Ví dụ: ta nhận được thông tin 1110010 CuuDuongThanCong.com

Ta có:

$$P_0 = \overset{\circ}{0} \overset{\circ}{0} \overset{\circ}{1} \overset{\circ}{1} = 0$$

$$P_1 = \overset{\circ}{1} \overset{\circ}{0} \overset{\circ}{1} \overset{\circ}{1} = 1 \text{ 110 } \text{Ø } 6$$

$$P_2 = \overset{\circ}{0} \overset{\circ}{1} \overset{\circ}{1} \overset{\circ}{1} = 1$$

Vậy bit thứ 6 sai.

3.8.4 Kiểm tra sai (Error control)

Kiểm tra sai dùng để kiểm tra, tìm và sửa sai trong frames được truyền. CuuDuongThanCong.com

Trước tiên, dữ liệu được truyền tạo thành các frames và truyền liên tiếp các frames với nhau. Frames đến đồng thời hay đến lần lượt khi truyền, mỗi frames bảo đảm sự độc lập và giá trị cố định khi nhận. Ta thấy có 2 loại sai:

- Mất frame: 1 frame bị mất trước khi đến bên kia hoặc nhiều làm cho frame hư, làm cho bộ nhận cho rằng frame chưa truyền.
- Frame bị hư: Frame được xác định đến mà một số bit bị sai trong khi truyền.

Hầu hết các kỹ thuật hiện nay cho việc kiểm tra sai dựa trên cơ sở sau:

- Phát hiện sai: như phần trước ta dùng FCS để kiểm tra mà nội dung chính là tìm CRC.
- Sự thừa nhận tích cực: Bộ phận nhận truyền tín hiệu ACK tích cực khi có kết quả, frame không có sai.
- Truyền lại sau thời gian nghỉ: nguồn sẽ truyền lại một frame nào đó mà chưa có được ACK, trước khi kết thúc thời gian truyền.
- Sự thừa nhận không tích cực và truyền lại: Bộ phận nhận trả lại tín hiệu NAK khi có một frame nào đó có sai, nguồn sẽ truyền lại frame đó.

Tổng hợp lại ta dùng phương pháp yêu cầu phát lại ARQ (Automatic Repeat reQuest). ARQ có 3 loại:

- Dừng và chờ ARQ.
- Trở lại N, ARQ.
- Truyền lại có lựa chọn ARQ.

Tất cả dạng đó đều dựa vào kỹ thuật kiểm tra dòng mà ta đã mô tả.

3.8.4.1 Dừng và chờ ARQ (Stop and wait ARQ):

Dừng và chờ ARQ dựa trên kỹ thuật kiểm tra lưu thông. Nó được miêu tả như hình vẽ 5-11.

Trạm nguồn truyền một frame đơn giản và nó chờ tín hiệu ACK. Các frame khác không được truyền cho đến khi trạm nhận cung cấp tín hiệu đó cho trạm nguồn.

Frame được truyền do nguồn có thể mất hoặc sai, nếu như bộ phận nhận phát hiện sai, nó bỏ frame đó và truyền tín hiệu NAK.

Do đó nguồn truyền lại frame bị hư. Mặt khác, nếu frame nguồn bị mất do nhiễu thì bộ phận thu không cho trả lời, nguồn phải chờ một thời gian. Sau mỗi frame được truyền, nguồn chờ ACK hoặc NAK. Nếu không nhận được ACK trong thời gian dừng chờ thì frame đó được truyền lại. Chú ý rằng nguồn sẽ giữ lại một copy của frame truyền cho đến khi nó nhận được ACK.

Hình 3.55 ARQ dừng và chờ.

Một điều nữa cần phải nhắc trước khi kết thúc vấn đề là: Nếu một frame truyền đúng mà trong khi truyền ACK bị hư thì sau thời gian dừng chờ không có tín hiệu ACK nguồn sẽ truyền lại frame đó. Bộ phận nhận bây giờ sẽ nhận 2 bản sao của một frame. Để tránh vấn đề đó, frame luôn thay đổi giá trị 0 và 1: ta có ACK₀ và ACK₁. ACK₀(ACK₁) nhận được của frame 1(0) và chỉ ra rằng bộ phận nhận đã sẵn sàng cho frame 0(1).

Về nguyên tắc, dừng và chờ ARQ có ưu điểm là đơn giản. (Nhược điểm ta đã thảo luận ở trên).

3.8.4.2 Trở lại N-ARQ (Goback N- ARQ):

Một phương án của ARQ tiếp tục là trở lại N-ARQ: Với kỹ thuật này một trạm có thể gửi đi một loạt các frames với các khung cửa sổ trượt sử dụng cho kỹ thuật kiểm tra dòng. Khi mà không có sai thì bộ phận nhận sẽ dùng ACK.

Giả thiết là trạm A gửi một số frames sang B. Sau mỗi lần truyền 1 frame A thiết lập tín hiệu ACK.

Kỹ thuật trở về N đưa vào những khả năng có thể xảy ra sau:

* **Mất frame:** có 3 trường hợp.

a. A truyền frame thứ i đến B.

B phát hiện ở frame có sai.

B gửi NAK i chỉ rằng frame i phải truyền lại. Khi A nhận được NAK frame thứ i và tất cả các frames tiếp theo đều phải truyền lại.

b. Frame i bị mất trong khi truyền, frame($i+1$) sẽ gửi đi. B nhận frame($i+1$) không đúng thứ tự ® gửi NAK $_i$ đến A.

c. Frame i bị mất trong khi truyền nhưng sau đó A không truyền frame ($i+1$). B không nhận được gì và trả lại ACK hay NAK. A sẽ dừng và sau thời gian qui định nó truyền lại frame i .

* **Mất ACK:** Có 2 trường hợp.

a. B nhận được frame i và truyền ACK $_{(i+1)}$ và nó bị mất trong khi truyền.

Hình vẽ chỉ cho ta các dòng frame cho phương pháp trở lại N-ARQ trên đường truyền full - duplex. Ta cho rằng có 3 bit để chỉ thứ tự frame (0-7).

Với phương pháp trở lại N-ARQ, không yêu cầu sau khi gửi đi một frame phải có ACK. Ví dụ trạm A gửi các frames 0,1,2,3 và trạm B sẽ trả lời với ACK $_1$ sau khi nhận frame 0 nhưng nó chưa trả lời cho frame 1 và 2. Sau khi nhận frame 3, B sẽ gửi ACK $_4$ chỉ ra rằng tất cả các frame khác được chấp nhận.

Hình 3.56 Trở lại N-ARQ.

b. Trường hợp frame 2 bị sai. B sẽ gửi NAK $_2$ đến A cho dù frames 3,4,5 đã gửi đi nhưng A sẽ truyền lại từ frame 2, frames 3,4,5 đã nhận sẽ bị hủy bỏ.

Bây giờ chúng ta làm rõ vấn đề vì sao ta có 2^n mà độ lớn cửa sổ chỉ 2^n-1 . Đó là do ảnh hưởng lẫn nhau của kiểm tra sai và ACK. Một trạm sẽ gửi ACK với frame. Do độ dài ACK có n bits trong frame khi gửi frame đi thì ACK đầu tiên bị mất, có thể ACK tiếp theo qua. Ta giả thiết nếu trạm truyền frame 0 và nhận ACK $_1$ với truyền frame 1,2,3,4,5,6,7,0 và nhận các ACK khác. Điều đó có nghĩa là 8 frames đều nhận đúng và trạm nhận bắt đầu lại ACK $_1$.

3.8.4.3 Truyền lại có lựa chọn ARQ (Selective Reject ARQ):

Hình 3.57 ARQ có lựa chọn.

Với sự truyền lại có lựa chọn ARQ khi phát hiện sai ta chỉ truyền lại frame nào mà ta nhận được NAK hoặc với thời gian tạm nghỉ.

Ta thấy sự sai hoàn toàn giống như trường hợp trước mà nó truyền lại có lựa chọn. So với trở lại N-ARQ nó hiệu quả hơn nhiều vì nó chỉ truyền lại frame hư, ít tốn thời gian hơn. Mặc khác, bộ phận thu cần giữ lại frame hư có NAK cho đến khi nó được truyền lại vì nó có giá trị logic cho frame mới đưa vào theo thứ tự. Sự truyền lại sẽ yêu cầu phức tạp hơn khi ta gửi các frame không theo thứ tự.

Do sự phức tạp đó mà phương thức truyền lại có lựa chọn ít được sử dụng.

Ta giả thiết frame 2 bị sai, do sự làm trễ khi truyền. Frame bị sai mãi đến sau khi truyền frames 3,4,5 mới tới bộ thu. Nếu theo giải thuật trở lại N-AKQ thì các frame đó bị bỏ đi và buộc phải truyền lại. Với giải thuật truyền lại có lựa chọn ARQ, các frames đó được giữ lại ở bộ phận thu. Chú ý rằng trong trường hợp tiếp theo, các frames không thể ACK ngoài thứ tự. Có nghĩa là frame 2 sau khi truyền lại, sau khi bộ thu đã cho ACK₆ mới cho ACK₃, ACK₄, ACK₅ được.

cuu duong than cong . com

cuu duong than cong . com