

## CHƯƠNG 11

# GIAO THỨC KẾT NỐI DỮ LIỆU (DATA LINK PROTOCOLS)

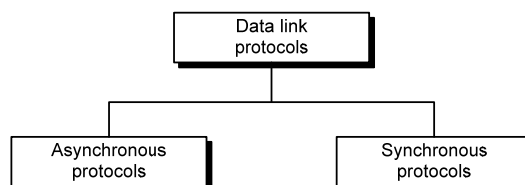
Giao thức (protocol) được hiểu là **tập các luật hay qui ước nhằm thực hiện một nhiệm vụ đặc thù**, trong nghĩa hẹp hơn giao thức là tập các luật hay đặc tính được dùng để thiết lập một hay nhiều lớp trong mô hình OSI.

**Giao thức trong truyền số liệu là tập các luật hay đặc tính được dùng để thiết lập một hay nhiều lớp trong mô hình OSI.**

**Giao thức kết nối dữ liệu là tập các đặc tính được dùng để thiết lập lớp kết nối dữ liệu**

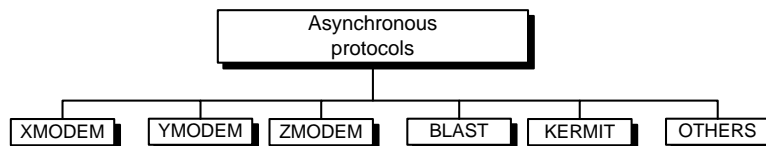
Giao thức kết nối dữ liệu chia ra hai nhóm con:

- **Giao thức không đồng bộ** xử lý các ký tự trong dòng bit một cách độc lập.
- **Giao thức đồng bộ** dùng nguyên dòng bit để chuyển sang thành ký tự có cùng chiều dài.



Hình 11.1

## 11.1 GIAO THỨC KHÔNG ĐỒNG BỘ



Hình 11.2

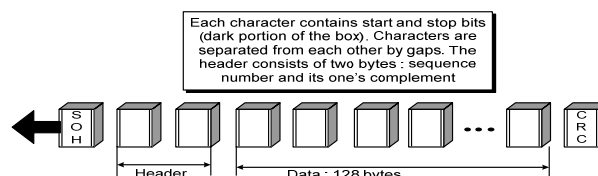
Các giao thức này chủ yếu được **dùng trong các modem**.

**Phương thức này có yếu điểm là truyền chậm** (do tồn tại start bit, stop bit và khoảng trống giữa các frame) nên hiện nay, đã có các giao thức truyền tốc độ cao dùng cơ chế đồng bộ.

### 11.1.1 XMODEM

Truyền file dùng đường truyền điện thoại giữa các PC. Giao thức này, được gọi là XMODEM:

- Là giao thức **stop and wait ARQ**
- Truyền bán song công (half-duplex)



- ✓ Trường **đầu tiên là một byte** tiêu đề header (start of header: SOH).
- ✓ Trường thứ hai là **header gồm 2 byte**: byte đầu là một chuỗi bit mang giá trị số frame và byte thứ hai được dùng để kiểm tra giá trị hợp pháp của chuỗi bit.
- ✓ Trường cố định gồm **128 byte dữ liệu** (binary, ASCII, Boole, text)
- ✓ **Trường cuối cùng là CRC**, chỉ dùng kiểm tra lỗi trong trường dữ liệu.
- Bắt đầu truyền bằng cách gửi một frame **NAK** từ máy thu đến máy phát.
- Mỗi khi máy phát gửi đi một frame thì phải chờ tín hiệu **ACK** trước khi gửi tiếp frame kế.
- Nếu máy phát nhận được NAK thì frame vừa gửi sẽ được gửi lại.
- Một frame cũng có thể được gửi lại nếu máy phát không nhận được tín hiệu xác nhận sau một thời gian định trước.
- Ngoài tín hiệu ACK và NAK, máy thu còn có thể nhận được tín hiệu **CAN** (cancel), yêu cầu hủy việc truyền.

### 11.1.2 YMODEM

Dùng giao thức **tương tự như XMODEM**, ngoài một số điểm khác biệt sau:

- Đơn vị dữ liệu là **1024 byte**.
- Dùng hai tín hiệu **CAN** để hủy việc truyền tin.
- Dùng phương pháp kiểm tra lỗi **ITU-T, CRC-16**.
- Có thể truyền đồng thời nhiều file.

### 11.1.3 ZMODEM

Giao thức mới, kết hợp cả hai giao thức XMODEM và YMODEM.

### 11.1.4 BLAST

**Blocked asynchronous transmission (BLAST)** mạnh hơn XMODEM. Giao thức dùng chế độ **song công (full-duplex)** dùng phương pháp **kiểm soát lưu lượng dạng cửa sổ trượt (sliding window)**.

### 11.1.5 KERMIT

Hiện là **giao thức không đồng bộ được dùng nhiều nhất hiện nay**.

Giao thức truyền file này **tương tự như hoạt động của X MODEM**,

Máy phát chờ NAK trước khi bắt đầu truyền.

Kermit cho phép truyền các ký tự kiểm tra dạng text theo hai bước:

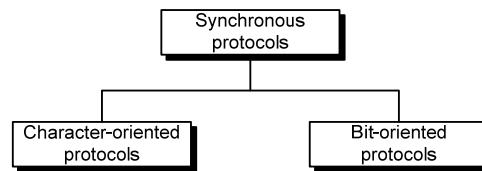
**Đầu tiên**, ký tự kiểm tra được dùng dạng text, được chuyển thành các ký tự in được thông qua việc thêm vào một số cố định và mã ASCII được dùng biểu diễn.

**Bước hai**, thêm ký tự # vào phía trước ký tự vừa chuyển đổi. Theo cách này, ký tự kiểm tra dùng như text được gửi đi như hai ký tự. Khi máy thu gặp ký tự #, thì biết phải bỏ đi và ký tự kế chính là ký tự kiểm tra. Nếu máy phát muốn phát ký tự #, thì cần phải gửi đi hai ký tự này.

## 11.2. GIAO THỨC ĐỒNG BỘ

Tốc độ truyền đồng bộ là chọn lựa tốt hơn so với trường hợp không đồng bộ, trong công nghệ LAN, WAN hay MAN.

Các giao thức đồng bộ truyền dữ liệu được chia thành hai dạng: **giao thức theo hướng ký tự** và **giao thức theo hướng bit**.



**Giao thức theo hướng ký tự** (còn gọi là giao thức theo hướng byte) diễn dịch các frame hay gói cần truyền thành các ký tự liên tiếp nhau, mỗi ký tự gồm một byte (8 bit). Tất cả các thông tin đều ở dạng hiện hữu của ký tự (mã ASCII)

**Giao thức theo hướng bit** diễn dịch dữ liệu hay gói cần truyền thành của **các bit đơn**, tạo nghĩa cho chúng bằng cách sắp xếp vị trí trong frame và bằng phương thức xếp đặt chúng với các bit khác. Các thông tin điều khiển trong giao thức này có thể dùng một hay nhiều bit, tùy theo kiểu thông tin trong mẫu

*Trong giao thức theo hướng ký tự, các frame hay gói được chuyển thành chuỗi các ký tự. Trong giao thức theo hướng bit, các frame hay gói được diễn dịch thành chuỗi các bit.*

### 11.2.1. CÁC GIAO THỨC THEO HƯỚNG KÝ TỰ BINARY SYNCHRONOUS COMMUNICATION (BSC)

#### a. giới thiệu:

- Dùng cho **cấu hình điểm - điểm** và **đa điểm**
- Cơ chế truyền **bán song công (half-duplex)**
- Dùng phương pháp kiểm tra lỗi và điều khiển lưu lượng **stop and wait ARQ**  
(BSC không hỗ trợ chế **độ full-duplex** hay **giao thức cửa sổ trượt**)

#### b. Các ký tự điều khiển

- **Ký tự ACK** không được dùng trong giao thức này.
- BSC dùng phương pháp **stop and wait ARQ** và ACK phải là **ACK0** hay **ACK1** cho các frame dữ liệu liên tiếp nhau.

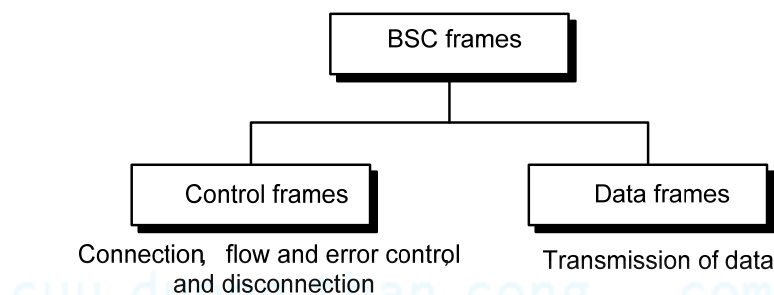
Trong bảng biểu diễn các ký tự dùng mã ASCII, chú ý là các ký tự điều khiển có thể biểu diễn bằng nhiều ký tự

Character	ASCII Code	Function
ACK 0	DLE and 0	Good <b>even frame</b> received or ready to receive
<b>ACK 1</b>	DLE and 1	Good <b>odd frame</b> received
DLE	DLE	Data transparency marker
ENQ	ENQ	Request for a response
EOT	EOT	Sender terminating

ETB	ETB	End of transmission block; ACK required
ETX	ETX	End of text in a message
ITB	US	End of intermediate block in a multiblock transmission
NAK	NAK	Bad frame received or nothing to send
NUL	NULL	Filler character
RVI	DLE and <	Urgent message from receiver
SOH	SOH	Header information begins
STX	STX	Text begins
SYN	SYN	Alerts receiver to incoming frame
TTD	STX and ENQ	Sender is pausing but not relinquishing the line
WACK	DLE and :	Good frame received but not ready to receive

c. Mã ASCII:

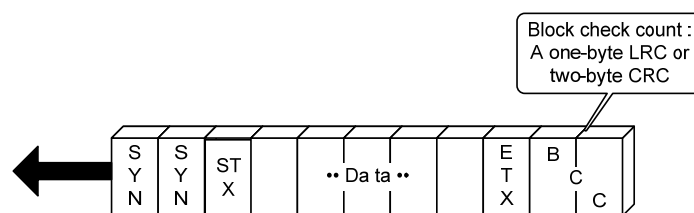
d. BSC frames:



Hình 11.3

- **Control frame:** chỉ chứa các thông tin về điều khiển.
- **Data frame:** chứa các thông tin về dữ liệu, nhưng cũng có các thông tin điều khiển dùng trong thông tin này.

e. Data Frame:



Hình 11.4

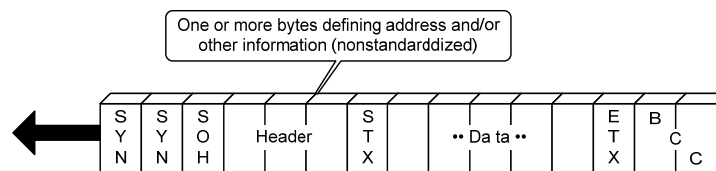
- Chiều mũi tên là chiều truyền.
- Frame có hai ký tự đồng bộ hay nhiều hơn. Các ký tự này cảnh báo máy thu là frame mới đến và cung cấp bit pattern cho máy thu nhằm đồng bộ thời gian với máy phát. Ví dụ mã ASCII của **SYN** là **00010110**. Bit đầu (thứ 8) của byte thường được thêm vào các số 0. Hai ký tự SYN cùng nhau sẽ có dạng **0001011000010110**.
- Tiếp sau 2 ký tự đồng bộ thì bắt đầu ký tự văn bản (STX: **00000010** start of text). Các ký tự này báo cho máy thu là đã hết thông tin điều khiển và byte kế tiếp là

dữ liệu. Dữ liệu hay văn bản có thể là một số các ký tự. Ký tự chấm dứt text (end of text: ETX: 00000011) cho biết có sự chuyển tiếp từ văn bản sang nhiều ký tự điều khiển.

- Sau cùng, một hay hai ký tự được gọi là khối đếm-kiểm tra (block check count: BCC) được thêm vào để kiểm tra lỗi. **Trường BCC có thể có một ký tự kiểm tra lỗi dạng LRC hay hai ký tự kiểm tra lỗi CRC.**

#### f. Trường tiêu đề (Header Fields)

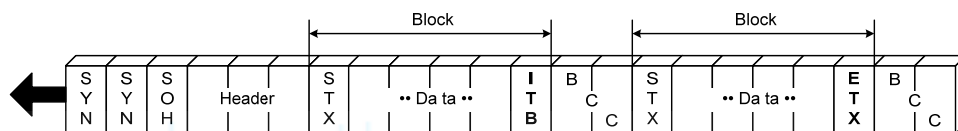
Một frame đơn như vừa mô tả thường ít được dùng, do phải có thêm địa chỉ của thiết bị thu, địa chỉ của thiết bị gửi, và số nhận dạng **frame (0 hay 1)** cho trường hợp stop and wait ARQ, xem hình bên dưới. Các thông tin này thường được chứa trong một trường đặc biệt gọi là tiêu đề (header), được bắt đầu bằng ký tự **start of header (SOH)**. Tiêu đề này đến sau ký tự **SYN và trước ký tự STX**; mọi thông tin nhận sau trường SOH nhưng trước ký tự STX là các thông tin tiêu đề.



Hình 11.5

#### g. Multiblock Frame

Khi chiều dài của khối (block) tăng thì xác suất xuất hiện lỗi cũng gia tăng theo. Càng nhiều bit trong một frame thì khả năng bị lỗi càng cao, làm cho việc phát hiện lỗi càng trở nên khó khăn. Do đó, các văn bản trong bản tin thường được chia ra thành nhiều block. Mỗi block (trừ block cuối cùng) đều **bắt đầu với một ký tự STX** và chấm dứt bằng một khối text trung gian (ITB: intermediate text block). Block cuối bắt đầu dùng STX nhưng tận cùng dùng **ETX**. Liên kế ngay sau mỗi ITB hay ETX là trường BCC. Theo cách này, thì máy thu có thể kiểm tra lỗi cho từng block riêng biệt, cho phép gia tăng khả năng phát hiện lỗi. Nếu một block có lỗi thì cả frame phải được chuyển lại. Sau khi ETX đã đến và BCC cuối cùng đã được kiểm tra xong, máy thu mới gửi độc một xác nhận cho toàn frame. Hình vẽ bên dưới minh họa cấu trúc của frame nhiều block; thí dụ này chỉ dùng hai block, tùy nhiệm trong thực tế thì có thể có nhiều hơn hai.

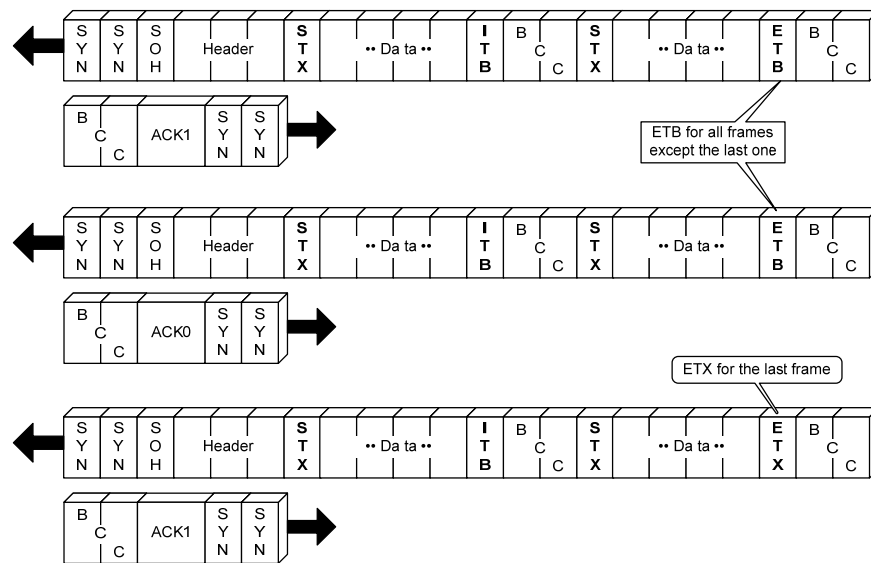


Hình 11.6

#### Truyền nhiều frame (Multiframe Transmission)

Trong thí dụ vừa rồi, một frame đơn mang toàn bản tin. Sau mỗi frame, bản tin được chấm dứt và kiểm tra chuyển sang đường thứ hai (thí dụ trong chế độ full-duplex). Một số bản tin, thường dài để có thể đặt vào format của một frame đơn, như thế, máy phát sẽ chia bản tin ra không những theo nhiều block mà còn thành nhiều frame. Nhiều frame có thể chuyển liên tục một bản tin. Để máy thu biết được là phần cuối của frame chưa phải là phần cuối của bản tin, thì ký tự ETX trong tất cả các frame (trừ frame cuối cùng) được thay thế bằng ký tự

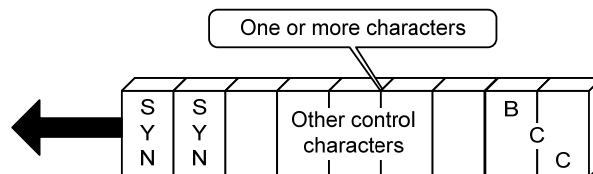
end of transmission block (ETB). Máy thu phải xác nhận mỗi frame riêng biệt nhưng không thể điều khiển toàn kết nối cho đến khi tìm được ký tự ETX tại frame cuối.



Hình 11.7

### Frame điều khiển (Control Frames)

Một frame kiểm tra không thể bị hiểu lầm thành một ký tự kiểm tra. Một frame điều khiển được một thiết bị dùng để gửi tín hiệu điều khiển, để củng cố thông tin, cho thiết bị khác. Một frame điều khiển chứa các ký tự điều khiển nhưng không có data; chúng chứa các thông tin đặc biệt để vận hành lớp kết nối dữ liệu.



Hình 11.8

Control frame có ba mục đích:

- Thiết lập kết nối
- Duy trì lưu lượng và kiểm tra lỗi trong khi truyền dẫn
- Kết thúc kết nối.

### Connection establishment

<table><tr><td>S</td><td>S</td><td>E</td></tr><tr><td>Y</td><td>Y</td><td>N</td></tr><tr><td>N</td><td>N</td><td>Q</td></tr></table>	S	S	E	Y	Y	N	N	N	Q	<b>Bid</b> Point-to-point connection request	<table><tr><td>S</td><td>S</td><td>E</td><td>Polling address</td></tr><tr><td>Y</td><td>Y</td><td>N</td><td></td></tr><tr><td>N</td><td>N</td><td>Q</td><td></td></tr></table>	S	S	E	Polling address	Y	Y	N		N	N	Q		<b>Poll</b> Primary polls secondary
S	S	E																						
Y	Y	N																						
N	N	Q																						
S	S	E	Polling address																					
Y	Y	N																						
N	N	Q																						
<table><tr><td>S</td><td>S</td><td>E</td><td>Selecting address</td></tr><tr><td>Y</td><td>Y</td><td>N</td><td></td></tr><tr><td>N</td><td>N</td><td>Q</td><td></td></tr></table>	S	S	E	Selecting address	Y	Y	N		N	N	Q		<b>Select</b> Primary selects secondary	<table><tr><td>S</td><td>S</td><td>ACK0</td></tr><tr><td>Y</td><td>Y</td><td></td></tr><tr><td>N</td><td>N</td><td></td></tr></table>	S	S	ACK0	Y	Y		N	N		<b>Positive response to select or bid</b> Ready to receive data
S	S	E	Selecting address																					
Y	Y	N																						
N	N	Q																						
S	S	ACK0																						
Y	Y																							
N	N																							
<table><tr><td>S</td><td>S</td><td>N</td></tr><tr><td>Y</td><td>Y</td><td>A</td></tr><tr><td>N</td><td>N</td><td>K</td></tr></table>	S	S	N	Y	Y	A	N	N	K	<b>Negative response to select or bid</b> Not ready to receive data	<table><tr><td>S</td><td>S</td><td>E</td></tr><tr><td>Y</td><td>Y</td><td>O</td></tr><tr><td>N</td><td>N</td><td>T</td></tr></table>	S	S	E	Y	Y	O	N	N	T	<b>Negative response to poll</b> Not ready to send data			
S	S	N																						
Y	Y	A																						
N	N	K																						
S	S	E																						
Y	Y	O																						
N	N	T																						

### Flow and error control

<table><tr><td>S</td><td>S</td><td rowspan="3">ACK0</td></tr><tr><td>Y</td><td>Y</td></tr><tr><td>N</td><td>N</td></tr></table>	S	S	ACK0	Y	Y	N	N	<b>Positive ACK of even frames</b> Frame number 0 received	<table><tr><td>S</td><td>S</td><td rowspan="3">ACK1</td></tr><tr><td>Y</td><td>Y</td></tr><tr><td>N</td><td>N</td></tr></table>	S	S	ACK1	Y	Y	N	N	<b>Positive ACK of odd frames</b> Frame number 1 received		
S	S	ACK0																	
Y	Y																		
N	N																		
S	S	ACK1																	
Y	Y																		
N	N																		
<table><tr><td>S</td><td>S</td><td>N</td></tr><tr><td>Y</td><td>Y</td><td>A</td></tr><tr><td>N</td><td>N</td><td>K</td></tr></table>	S	S	N	Y	Y	A	N	N	K	<b>Negative ACK of frames</b> Error in the frame received	<table><tr><td>S</td><td>S</td><td rowspan="3">WACK</td></tr><tr><td>Y</td><td>Y</td></tr><tr><td>N</td><td>N</td></tr></table>	S	S	WACK	Y	Y	N	N	<b>Wait &amp; ACK</b> ACK of previous frame not ready to receive more
S	S	N																	
Y	Y	A																	
N	N	K																	
S	S	WACK																	
Y	Y																		
N	N																		
<table><tr><td>S</td><td>S</td><td rowspan="3">RVI</td></tr><tr><td>Y</td><td>Y</td></tr><tr><td>N</td><td>N</td></tr></table>	S	S	RVI	Y	Y	N	N	<b>Reverse interrupt</b> Request for interruption, urgent data to send	<table><tr><td>S</td><td>S</td><td rowspan="3">TTD</td></tr><tr><td>Y</td><td>Y</td></tr><tr><td>N</td><td>N</td></tr></table>	S	S	TTD	Y	Y	N	N	<b>Temporary delay</b> Temporarily delayed but does not relinquish the line		
S	S	RVI																	
Y	Y																		
N	N																		
S	S	TTD																	
Y	Y																		
N	N																		

### Connection termination

S	S	E
Y	Y	O
N	N	T

**End of transmission**  
Station finished sending data

## DATA TRANSPARENCY

Nếu một trường text khi truyền gồm mẫu 8 bit giống như ký tự điều khiển của BSC, thì máy thu sẽ biên dịch thành một ký tự và hủy ý nghĩa của bản tin.

Thí dụ: máy thu nhìn thấy một chuỗi bit 0000011 được đọc là ký tự ETX, nên như ta đã biết, khi máy thu nhận ra ký tự ETX, nó sẽ cho rằng hai byte kế tiếp là BCC và bắt đầu kiểm tra lỗi. Thực ra mẫu 0000011 ở đây là dữ liệu chứ không phải là thông tin điều khiển. Hiểu lầm này được gọi là thiếu thông tin minh bạch (transparency).

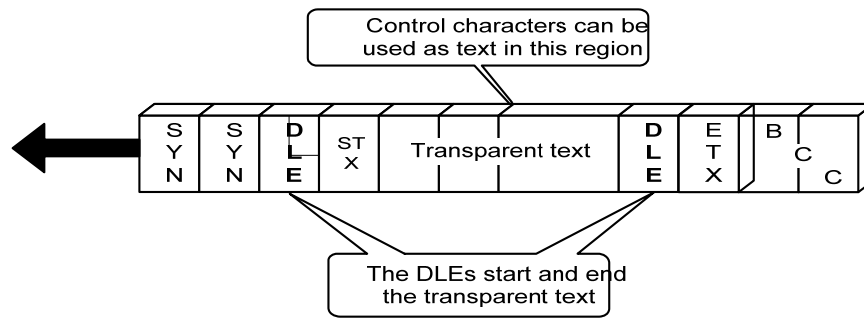
Để một giao thức là hữu ích thì giao thức này phải là minh bạch - tức là có thể mang bất kỳ tổ hợp bit như là dữ liệu mà không bị hiểu lầm là thông tin điều khiển.

*Data transparency trong thông tin số liệu được hiểu là ta có thể truyền các tổ hợp bit dữ liệu bất kỳ.*

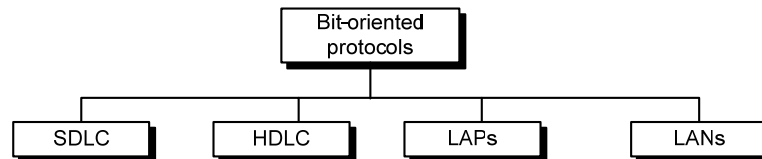
Tính minh bạch của BSC có thể được thực hiện thông qua quá trình bit nhồi (bit stuffing). Bao gồm hai tác động: định nghĩa vùng văn bản transparency dùng ký tự data link escape (DLE) và xử lý các ký tự DLE trong vùng transparency bằng các ký tự DLE extra.

Để định nghĩa vùng transparency, ta chèn vào một ký tự DLE ngay trước ký tự STX tại lúc bắt đầu trường text và DLE khác ngay trước ETX (hay ITB hay ETB) tại cuối trường text. DLE đầu cho máy thu biết là text có thể chứa các ký tự điều khiển và phải bỏ qua chúng. DLE cuối cho máy thu biết là vùng transparency đã chấm dứt.





### 11.2.2 CÁC GIAO THỨC THEO HƯỚNG BIT



Các bit được nhóm thành các mẫu tạo ký tự.

So sánh với phương pháp theo hướng byte thì giao thức theo hướng bit có thể đóng gói nhiều thông tin hơn trong một frame ngắn hơn và **tránh được vấn đề transparency** (minh bạch)

Các giao thức theo hướng bit ngày nay ngày càng nhiều và dần phát triển thành các chuẩn. Đa số chúng được các nhà sản xuất thiết kế nhằm hỗ trợ cho các sản phẩm của mình. Trong số đó, chuẩn **HDLC** do ISO thiết kế và ngày càng trở thành cơ sở của các giao thức theo hướng bit hiện nay.

Đầu năm 1975, IBM đã đi đầu trong việc phát triển giao thức theo hướng bit với giao thức **synchronous data link control (SDLC)** và yêu cầu ISO chấp nhận để đưa vào làm chuẩn. Đầu năm 1979, ISO trả lời bằng cách đưa ra **high-level data link control (HDLC)**, phát triển từ SDLC. Việc ISO chấp nhận chuẩn HDLC làm giao thức này được nhiều tổ chức chấp nhận và mở rộng. ITU-T là tổ chức đầu tiên chấp nhận HDLC. Từ 1981, **ITU-T** đã phát triển một tập các giao thức được gọi là **link access protocol (LAPs, LAPB, LAPD, LAPM, LAPX, v.v...)** đều dựa trên HDLC. Các giao thức khác (thí dụ Frame Relay, PPP, v.v...) được cả ITU-T và ANSI cũng dựa trên HDLC, và làm giao thức cho mạng LAN. Hầu hết các giao thức theo hướng bit đều xuất phát từ HDLC, nên đó chính là nền tảng để tìm hiểu các giao thức khác.

**Tất cả các giao thức theo hướng bit đều xuất phát từ HDLC (high-level data link control), là dạng giao thức theo hướng bit do ISO công bố. HDLC hỗ trợ cả chế độ song công và bán song công trong cấu hình điểm-điểm hoặc điểm nối nhiều điểm.**

#### HDLC

HDLC là giao thức kết nối dữ liệu theo hướng bit được thiết kế nhằm hỗ trợ cho các chế độ bán song công và song công, cấu hình điểm nối điểm hay điểm nối nhiều điểm. Hệ thống dùng HDLC có thể được đặc trưng hoá bởi dạng trạm, cấu hình và chế độ đáp ứng.

#### Các dạng trạm

HDLC chia thành các trạm : sơ cấp, thứ cấp và kết hợp

**Trạm sơ cấp trong chức năng HDLC:** tương tự như chức năng thiết bị sơ cấp của phương pháp kiểm soát lưu lượng. Sơ cấp là thiết bị kiểm soát mạng theo các cấu hình kết nối



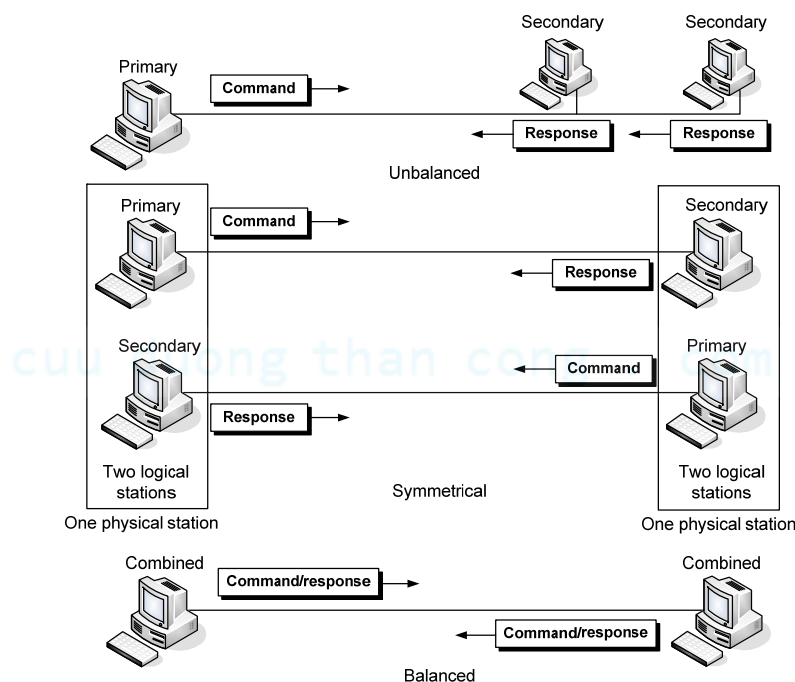
điểm – điểm hay điểm nhiều điểm. **Sơ cấp** chuyển tín hiệu điều khiển đến các trạm **thứ cấp**. Sơ cấp - điều khiển, thứ cấp-đáp ứng.

**Trạm hỗn hợp** là trạm có thể điều khiển và đáp ứng. Một trạm kết hợp có thể là một tập các thiết bị đồng cấp kết nối được lập trình để có thể vận hành như sơ cấp hay thứ cấp tùy theo bản chất và chiều truyền dẫn.

*Trạm trong HDLC có ba dạng : sơ cấp, thứ cấp và kết hợp. Trạm sơ cấp gửi tín hiệu điều khiển, thứ cấp gửi tín hiệu đáp ứng. Trạm kết hợp gửi cả tín hiệu điều khiển và đáp ứng.*

### Cấu hình (Configuration)

Từ cấu hình nói lên quan hệ của các thiết bị phần cứng trong kết nối. Các trạm sơ cấp, thứ cấp và kết hợp có thể được **cấu hình theo 3 cách**: **không cân bằng**, **đối xứng** và **cân bằng**. Các cấu hình này đều hỗ trợ cho phương thức truyền song công và bán song công.



Hình 11.9

**Cấu hình không cân bằng** (hay còn gọi là cấu hình **master/slave**) trong đó có một thiết bị là sơ cấp và các thiết bị khác là thứ cấp. Cấu hình không cân bằng còn được gọi là cấu hình **điểm -điểm** nếu chỉ có hai thiết bị, và thường là **điểm -nhiều điểm** trong đó một thiết bị sơ cấp điều khiển nhiều thiết bị thứ cấp.

**Cấu hình đối xứng**, trong đó mỗi trạm vật lý trên mạng gồm hai trạm luận lý, một là sơ cấp và một là thứ cấp. Các dây riêng biệt nối **sơ cấp của một trạm vật lý đến thứ cấp** của một trạm vật lý khác. Cấu hình đối xứng hoạt động tương tự như cấu hình không cân bằng trừ việc điều khiển mạng có thể được cả hai mạng thực hiện.

**Cấu hình cân bằng**, trong đó có một trạm dạng hỗn hợp, trong số các trạm cấu hình **điểm - điểm**. Các trạm được kết nối dùng một dây và được điều khiển từ các trạm khác.

HDLC không hỗ trợ chế độ cân bằng nhiều điểm. Điều này đưa ra nhu cầu cho việc thiết lập các giao thức truy cập môi trường cho mạng LA.

### Chế độ thông tin

Hoạt động của HDLC dựa trên quan hệ giữa hai thiết bị cần trao đổi thông tin: Chế độ này cho biết ai điều khiển mạng. Trao đổi trong cấu hình không cân bằng thường được thực hiện trong chế độ đáp ứng bình thường. Trao đổi trong cấu hình đối xứng hay cân bằng có thể được thiết lập ở các chế độ đặc biệt dùng các frame được thiết kế để mang lệnh điều khiển (sẽ thảo luận trong phần U-frame). *HDLC hỗ trợ ba chế độ thông tin giữa các trạm:*

*chế độ đáp ứng bình thường (normal response mode: **NRM**).*

*chế độ đáp ứng không đồng bộ (asynchronous response mode: **ARM**).*

*và chế độ cân bằng không đồng bộ (asynchronous balanced mode: **ABM**).*

**NRM:** là chuẩn về quan hệ sơ cấp-thứ cấp. Trong chế độ này, thiết bị thứ cấp phải cho phép từ thiết bị sơ cấp thì mới có thể gửi tin. Khi đã có phép rồi thì thiết bị thứ cấp có thể khởi tạo một đáp ứng truyền một hay nhiều frame dữ liệu.

**ARM:** thiết bị thứ cấp có thể khởi tạo việc truyền không cần sự cho phép của thiết bị sơ cấp khi nào kênh trống. Các trường hợp khác thì quan hệ master/slave vẫn được duy trì. Mọi thông tin truyền từ thiết bị thứ cấp (hay từ một thiết bị thứ cấp khác trong đường truyền) vẫn phải dùng thiết bị sơ cấp làm relay để đi đến đích.

**ABM:** mọi thiết bị đều đồng quyền nên cần có các trạm hỗn hợp điểm nối điểm. Các trạm hỗn hợp có thể gửi tin đến các trạm hỗn hợp khác mà không cần có phép.

■ *Normal response mode (NRM)*

■ *Asynchronous response mode (ARM)*

■ *Asynchronous balanced mode (ABM)*

Các chế độ HDLC:

	<b>NRM</b>	<b>ARM</b>	<b>ABM</b>
<b>Station type</b>	Primary & secondary	Primary & secondary	Combined
<b>Initiator</b>	Primary	Either	Any

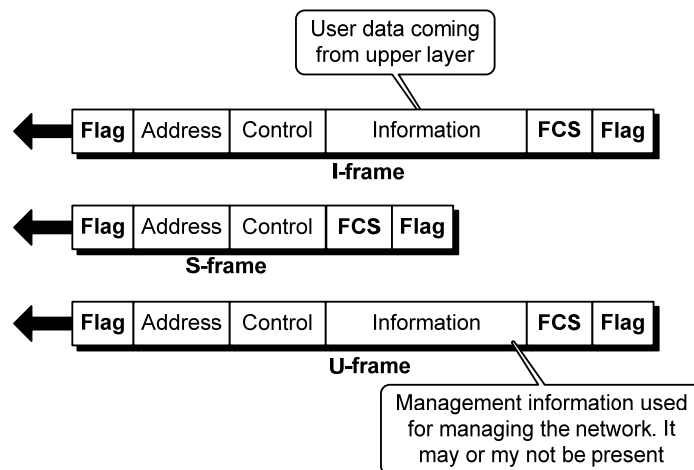
### FRAMES

Ả hằm cung cấp hỗ trợ mềm dẻo cho tất cả các trường hợp về chế độ và cấu hình đã nói trên, **HDLC định nghĩa 3 dạng frame:** frame thông tin (**I-frame**: information frame), frame giám sát (**S-frame**: supervisory frame) và frame không đánh số (unnumbered frame **U-frame**). Mỗi dạng frame hoạt động như lớp vỏ để truyền thông tin đến nhiều dạng bản tin.

**I-frame:** được dùng để vận chuyển dữ liệu của người dùng (user) và thông tin điều khiển liên quan đến người dùng.

**S-frame:** chỉ dùng để vận chuyển các thông tin điều khiển, lưu lượng của lớp kết nối dữ liệu và kiểm tra lỗi.

**U-frame:** được dùng dự phòng cho quản lý hệ thống. Thông tin do U-frame thường được dùng cho việc tự quản lý mạng.

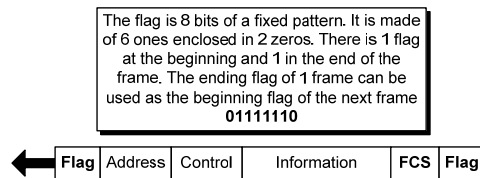


Hình 11.10

Mỗi frame trong HDLC có thể chứa đến 6 trường: trường bắt đầu flag, trường địa chỉ, trường điều khiển, trường thông tin, trường kiểm tra sequence của frame (FCS: frame check sequence) và trường cuối flag. Khi truyền nhiều frame, flag cuối có thể là một frame đôi và làm flag bắt đầu cho frame kế tiếp.

### Flag field

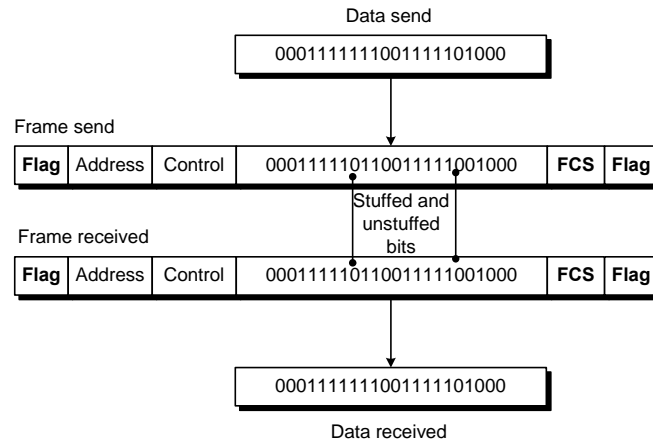
Trường flag của HDLC là chuỗi 8 bit có mẫu **01111110** nhằm nhận dạng cả phần đầu, phần cuối của frame và mẫu đồng bộ cho máy thu. Hình dưới đây vẽ sắp xếp của hai trường flag trong một I-frame



Hình 11.11

Trường flag là trường gần nhất mà HDLC đến một ký tự điều khiển dễ bị máy thu đọc sai nhất. ả hư thế, trường flag có thể là nguyên nhân cho vấn đề transparency. Khi một trạm nhận ra flag, xác định trường này được định địa chỉ đến mình, thì bắt đầu đọc quá trình truyền, và chờ flag kế tiếp nhằm cho biết phần kết thúc của frame. Còn có khả năng là chuỗi bit, dù là thông tin điều khiển hay dữ liệu, đều có thể chứa mẫu **01111110**. ả ếu điều này xuất hiện trong dữ liệu thì máy thu sẽ tìm và giả sử là sắp đạt tới phần cuối của frame (với kết quả là sai).

Để bảo đảm là flag không xuất hiện một cách không báo trước (inadvertently) trong frame, HDLC đã trù tính một quá trình được gọi là bit nhồi (bit stuffing). Mỗi lần máy phát muốn gửi một chuỗi bit có hơn 5 bit 1 liên tiếp, thì nó sẽ chèn (nhồi) thêm một bit thừa 0 sau 5 số 1. Thí dụ, chuỗi **01111111000** sẽ trở thành **0111110111000**. Số 0 thêm vào bit 1 thứ 6 cho biết là đã có bit nhồi, và máy thu khi nhận được sẽ loại bỏ đi

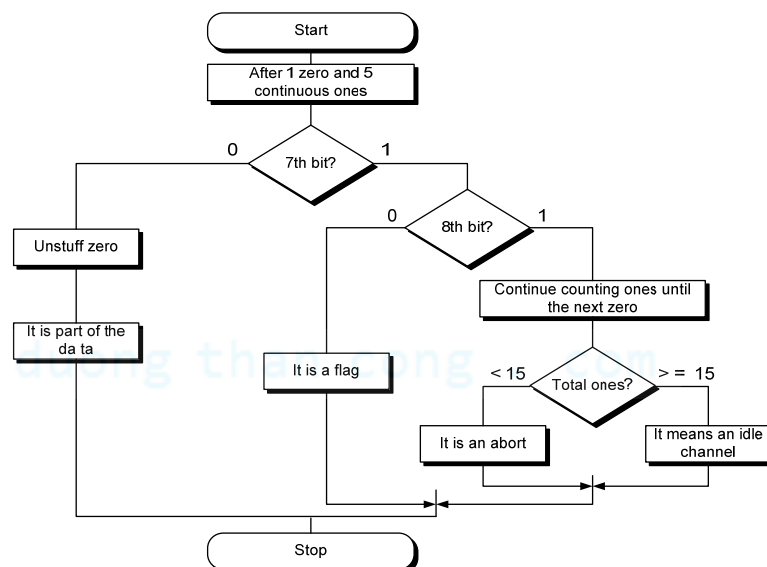


Hình 11.12

Quá trình này có ba ngoại lệ: khi chuỗi bit thực sự là flag, khi việc truyền bị hủy bỏ và khi kênh truyền không còn được sử dụng. Lưu đồ dưới đây các bước mà máy thu dùng để nhận dạng và loại bit nhồi. Khi máy thu đọc các bit thu được, thì bắt đầu đếm số bit 1, Sau khi đã tìm ra 5 bit 1 tiếp đến là bit 0, thì tiếp tục kiểm tra 7 bit tiếp. ả ếu bit thứ bảy là 0, máy thu xác nhận đó là bit nhồi , và reset lại bộ đếm. ả ếu bit thứ 7 là bit 1, thì máy thu kiểm tra bit thứ 8. ả ếu bit thứ 8 tiếp tục là bit 1, thì máy thu tiếp tục đếm. Giá trị tổng là 7 hay 14 bit 1 liên tiếp, cho chỉ thị loại bỏ. Khi tổng này là 15, tức là kênh trống.

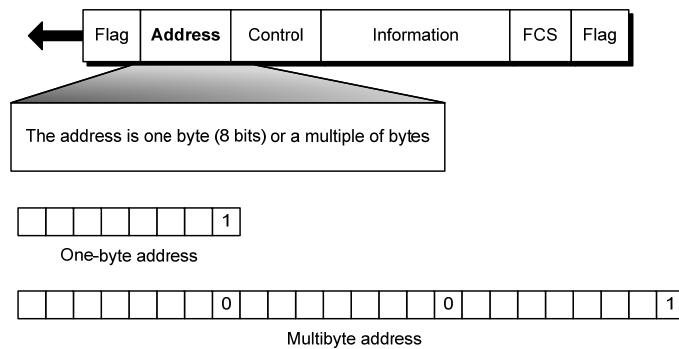
### Address field

Trường thứ hai của frame HDLC chứa địa chỉ của trạm thứ cấp, có thể là originator hay destination của frame (hay trạm đóng vai trò trạm thứ cấp trong trường hợp trạm hỗn hợp. ả ếu trạm thứ cấp tạo ra một frame, thì frame này chứa *from address*. Trường địa chỉ có thể dài một byte hay nhiều byte, tùy theo nhu cầu của mạng. Mạng càng lớn thì đòi hỏi trường địa chỉ với nhiều byte hơn



Hình 11.13

Hình vẽ tiếp theo đây cho thấy quan hệ của trường địa chỉ với các phần khác của frame.



Hình 11.14

Ấu trường địa chỉ chỉ gồm một byte, thì bit cuối cùng thường là bit 1. ấu trường này dài hơn một byte, tất cả các byte đều có bit cuối có giá trị 0, trừ byte cuối tận cùng bằng bit 1. Các byte trung gian được tận cùng là bit 0, nhằm báo cho máy thu biết là còn nhiều byte địa chỉ khác đến.

### Control field

Trường điều khiển là đoạn gồm một hay nhiều byte của frame được dùng để quản lý. Đầu tiên, khảo sát trường hợp một byte, rồi sẽ phát triển thành trường hợp 2 byte, được gọi là chế độ mở rộng.

Tùy theo dạng frame mà trường điều khiển có thể khác nhau.

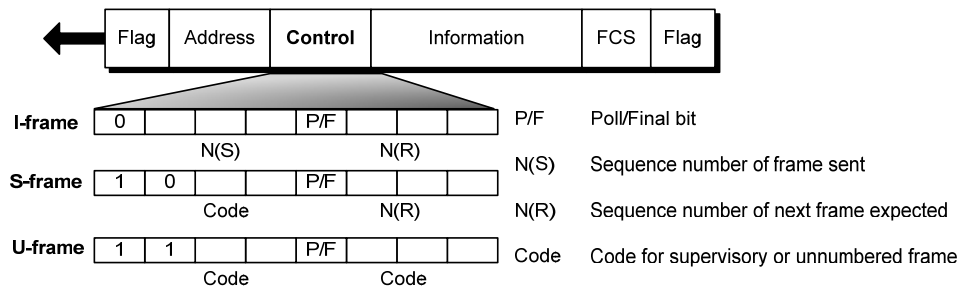
- Ấu bit đầu tiên của trường điều khiển là 0, thì đó là I-frame.
- Ấu bit đầu là 1 và bit kế là 0 thì đó là S-frame.
- Ấu cả hai bit đầu và kế đều là 1, thì đó là U-frame.

Trường điều khiển của cả ba dạng frame đều chứa một bit được gọi là poll/final (P/F)-bit thứ 5.

Một I-frame chứa 2 chuỗi 3 bit điều khiển lưu lượng và kiểm tra lỗi, được gọi là ấ (S) và ấ (R), nằm giữa bit (P/F). ấ (S) cho biết số frame mong muốn gửi trả về trong trường hợp trao đổi hai chiều; còn ấ (R) cho biết số frame kế tiếp trong chuỗi. ấ frame cuối không được nhận chính xác, thì số ấ (R) sẽ là số các frame bị hỏng, cho thấy nhu cầu cần truyền lại.

Trường điều khiển trong S-frame có chứa trường ấ (R) nhưng không chứa trường ấ (S). S-frame được dùng để gửi về ấ (R) khi máy thu không có dữ liệu riêng để gửi đi. Mặt khác, tín hiệu xác nhận chứa trong trường điều khiển của một I-frame (nói trên).

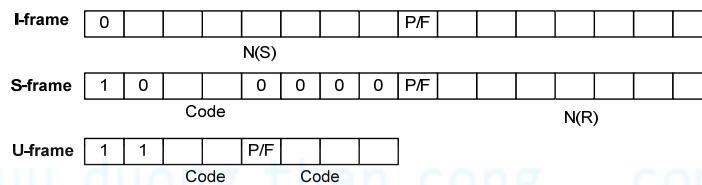
S-frame không truyền dữ liệu nên không cần trường ấ (S) để nhận dạng chúng. Hai bit nằm trước bit P/F trong S-frame được dùng mang mã lưu lượng (code flow) và thông tin kiểm tra lỗi, sẽ được thảo luận ở phần sau



Hình 11.15

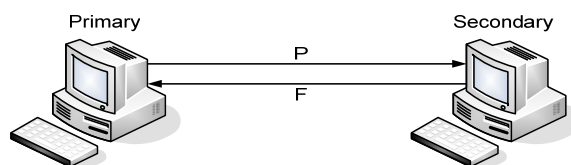
U-frame thì không có các trường **ảnh (S)** và **ảnh (R)**, và không được thiết kế để trao đổi dữ liệu của người dùng hay tín hiệu xác nhận. Thay vào đó, U-frame có hai trường code, một gồm hai bit, và một là ba bit, chen giữa bởi bit P/F. Các mã này được nhận ra dạng của U-frame cùng các chức năng (thí dụ thiết lập các chế độ của trao đổi).

Hình dưới đây mô tả trường điều khiển trong chế độ mở rộng. Chú ý là trong chế độ mở rộng, trường điều khiển của I-frame và S-frame có chiều dài hai byte cho phép dùng 7 bit dùng cho trường hợp phát và số chuỗi thu ( số này có thể nằm giữa 0 và 127). Tuy nhiên, U-frame vẫn là một byte.



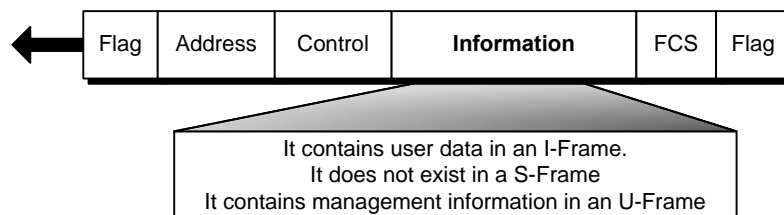
Hình 11.16

Trường P/F là một bit đơn có hai mục đích. **ảnh ó chỉ có nghĩa khi thiết lập với (bit=1)** và có thể cho biết là poll hay final. **ảnh ó là poll khi frame được trạm sơ cấp gọi đi** (tức là khi trường địa chỉ chứa địa chỉ máy thu) và là final khi frame được trạm sơ cấp gọi về như trường hợp hình bên dưới.



Hình 11.17

### Information field



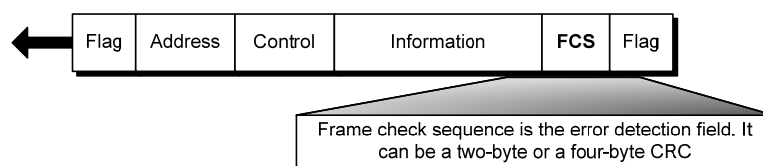
Hình 11.18

Trường thông tin chứa dữ liệu người dùng trong **I-frame**, và mạng quản lý thông tin trong **U-frame**. Chiều dài của frame thay đổi tùy thuộc vào từng dạng mạng nhưng giữ cố định trong cùng một mạng. **S-frame không có trường thông tin**.

Ả hử đã thấy trong các trường hợp trên thì thường có **khả năng đặt các thông tin về lưu lượng, lỗi và các thông tin khác trong một I-frame tức là frame có chứa dữ liệu**. Thí dụ, trong phương thức trao đổi hai chiều (half hay full duplex), **trạm 2 có thể xác nhận dữ liệu nhận được từ trạm 1 trong trường điều khiển của chính frame dữ liệu của mình thay vì gửi các frame xác nhận riêng**. Kết hợp dữ liệu gửi vào thông tin điều khiển theo cách này được gọi là piggybacking (cồng, cỡi trên lưng người khác)

*Piggybacking (cồng) là phương thức kết hợp dữ liệu truyền và xác nhận vào trong một frame đơn.*

### FCS Field



Hình 11.19

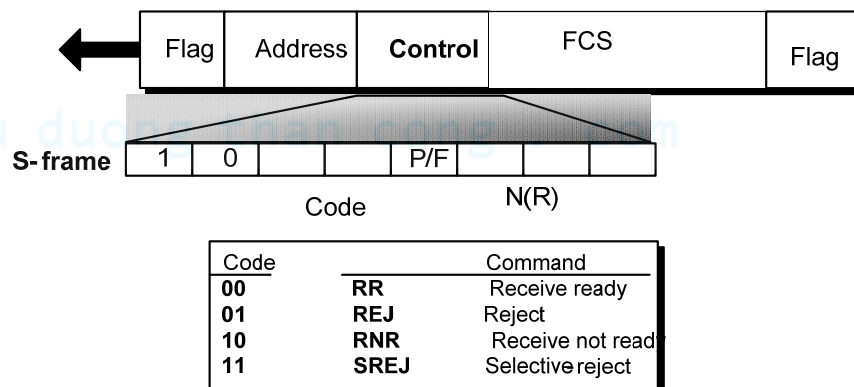
**Frame Check Sequence (FCS)** nằm trong trường kiểm tra lỗi của HDLC, trong đó chứa từ 2 đến 4 byte CRC.

### NÓI THÊM VỀ FRAME

Trong ba frame của HDLC thì **I-frame là đơn giản nhất**, do được thiết kế để vận chuyển các thông tin của người dùng (user) và piggybacking xác nhận. Do đó, tầm biến động của I-frame – các khác biệt liên quan đến dữ liệu (nội dung và CRC), nhằm để nhận dạng số frame hay để xác nhận các frame nhận được.

Trái lại, S-frame và U-frame thì chứa các trường con trong frame điều khiển. ả hử đã thảo luận ở phần trường điều khiển, thì các trường con này chứa mã nhằm thay đổi ý nghĩa của frame. Thí dụ, mã của S-frame dùng cho **selective-reject (SREJ)** không thể được dùng thay cho mã của S-frame dùng cho receive ready (RR).

### S-FRAME



Hình 11.20



Frame giám sát được dùng để xác nhận, điều khiển lưu lượng, và kiểm tra lỗi khi việc piggybacking vào I-frame là không thể được hoặc không thích hợp (khi trạm không có dữ liệu để gửi hay khi cần gửi các tín hiệu điều khiển, hay đáp ứng thay cho các tín hiệu xác nhận).

S-frame không có trường thông tin, mà nằm trong các bản tin được gửi đến trạm thu. Các bản tin này tùy theo dạng của S-frame và context của tin truyền, dạng của mỗi S-frame được xác định từ một mã gồm hai bit thiết lập trong trường điều khiển, ngay trước bit P/F. Có 4 dạng S-frame:

- Thu, sẵn sàng thu (RR).
- Chưa sẵn sàng thu (R<sup>h</sup> R).
- Loại (REJ)
- Chọn-lọc (SREJ)

### Receive Ready

S-frame chứa các mã cho RR (00) có thể được dùng trong 4 trường hợp khác nhau:

- **ACK:** Tín hiệu RR được trạm thu dùng gửi trả về một xác nhận khi nhận được I-frame khi máy thu không có dữ liệu riêng để gửi (không có I-frame để piggybacking tín hiệu xác nhận). Trong trường hợp này, trường  $\hat{a}$  (R) của frame điều khiển chứa các số của chuỗi của frame kế cần nhận. Trong trường điều khiển một byte, trường  $\hat{a}$  (R) có 3 bit, cho phép xác nhận đến 8 frame. Trong chế độ mở rộng, trường  $\hat{a}$  (R) có 7 bit cho phép xác nhận đến 128 frame
- **Poll:** Khi trạm sơ cấp truyền (hay trường hợp trạm hỗn hợp đóng vai trò sơ cấp), với bit P/F được thiết lập ở chức năng poll hay bit P, RR sẽ hỏi trạm thứ cấp có gì gửi không?
- **Negative response to poll:** Khi gửi bằng trạm thứ cấp dùng bit P/F được thiết lập ở final hay bit F, RR sẽ báo cho trạm phát biết là trạm thu không có gì để gửi. Nếu trạm thứ cấp có dữ liệu cần truyền, thì sẽ đáp ứng với poll thông qua I-frame, chứ không dùng S-frame.
- **Positive response to poll:** Khi trạm thứ cấp có khả năng thông tin truyền từ sơ cấp, thì nó gửi về một frame RR trong đó bit P/F được thiết lập ở 1 (bit F)

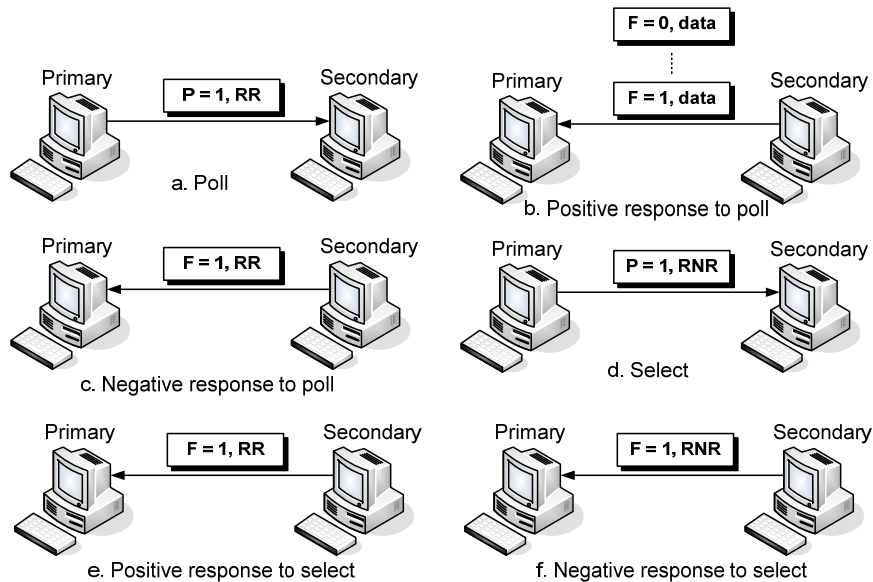
**Receive not ready:** Frame R<sup>h</sup> R có thể được dùng theo 3 cách:

- **ACK:** Tín hiệu R<sup>h</sup> R từ máy thu gửi về máy phát nhằm xác nhận về tất cả các frame đã nhận, nhưng không bao gồm frame được chỉ trong trường  $\hat{a}$  (R) nhưng yêu cầu là không gửi thêm frame nào nữa cho đến khi có frame RR được gửi đi.
- **Select:** Khi trạm sơ cấp muốn truyền dữ liệu to một trạm thứ cấp đặc thù, nó cảnh báo cho thứ cấp bằng cách gửi frame R<sup>h</sup> R với bit P/F được thiết lập ở bit P. Mã R<sup>h</sup> R báo cho máy thứ cấp đừng gửi dữ liệu riêng của mình nữa, do frame đã được thiết lập ở chế độ select chứ không phải là poll.
- **Negative response to select.** Khi trạm thứ cấp được chọn không có khả năng nhận dữ liệu, thì nó gửi trả về frame R<sup>h</sup> R với bit P/F được thiết lập ở bit F.

**Reject.** Dạng thứ 3 của S-frame là reject (REJ). REJ là tín hiệu không xác nhận được máy thu gửi trả về trong hệ thống sửa lỗi go-back-n ARQ, với trường hợp máy thu không có

dữ liệu để piggybacking đáp ứng. Trong frame REJ, trường  $\mathbf{\hat{a}}$  (R) chứa số của frame bị hỏng để cho biết là frame này và tất cả các frame tiếp sau đều phải được truyền lại.

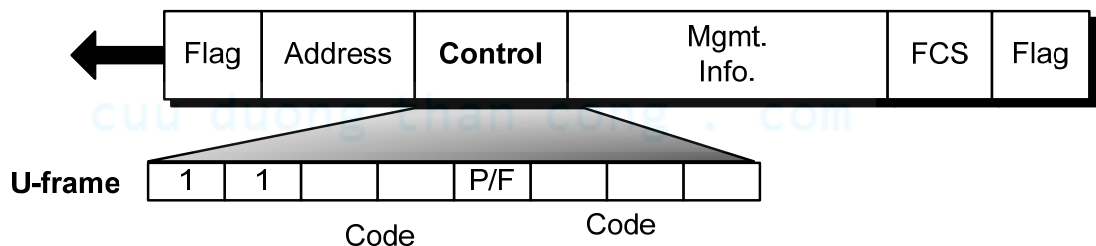
**Selective-Reject:** Frame selective-reject (SREJ) là tín hiệu không xác nhận trong hệ thống selective-reject ARQ.  $\hat{\mathbf{a}}$  ó được máy thu gửi về máy phát cho biết một frame nhận được đã bị hỏng (số nằm trong trường  $\hat{\mathbf{a}}$  (R)) và yêu cầu gửi lại frame này.



Hình 11.21

## U-FRAME

Các frame không đánh số được dùng để trao đổi các thông tin về quản lý và điều khiển giữa các thiết bị đang kết nối. Khác với S-frame, **U-frame có chứa trường thông tin, nhưng là các thông tin quản lý hệ thống chứ không phải là dữ liệu của user.** Tương tự như S-frame, nhiều thông tin do U-frame mang được chứa trong mã đặt ở trường điều khiển. Mã của U-frame được chia thành hai phần: **một prefix gồm hai bit đặt trước bit P/F** và **một suffix 3 bit sau bit P/F**. Hai phân đoạn này (5 bit) cùng được dùng để **tạo ra 32 dạng U-frame**. Một số tổ hợp được minh họa trong hình dưới đây



Hình 11.22

Code	Command	Response
00 001	SNRM	
11 011	SNRME	
11 000	SARM	DM
11 010	SARME	
11 100	SABM	
11 110	SABME	
00 000	UI	UI
00 110		UA
00 010	DISC	RD
10 000	SIM	RIM
00 100	UP	
11 001	RSET	
10 101	XID	XID
10 001		FRMR

Các lệnh trong U-frame được ghi trong bảng có thể chia **thành 5 phạm trù chức năng cơ bản: thiết lập chế độ, trao đổi không đánh số, ngừng kết nối, khởi tạo, và các chức năng khác(hỗn hợp):**

#### - Mode setting

Các lệnh thiết lập chế độ được trạm sơ cấp, hay do trạm hỗn hợp đóng vai trò sơ cấp gửi đi nhằm điều khiển quá trình trao đổi, nhằm thiết lập kiểm soát kết nối. **Frame thiết lập chế độ của U-frame thông báo cho trạm thu biết về format của quá trình sắp truyền.** Thí dụ, một trạm hỗn hợp muốn thiết lập một quan hệ sơ cấp -thứ cấp tạm thời với một trạm khác, thì nó gửi đi một U-frame chứa mã **00 001** (nhằm thiết lập đáp ứng thông thường). Trạm có địa chỉ nhận hiệu được là mình được chọn để nhận tin (từ sơ cấp) nên tự chỉnh định cho thích hợp.

#### - Unnumbered-Exchange

**Các mã về trao đổi không đánh số được dùng để truyền hay cũng có phần đặc thù về thông tin kết nối dữ liệu giữa hai thiết bị.** Mã poll không đánh số (**UP: unnumbered poll**) **00 100** được trạm sơ cấp (hay trạm hỗn hợp đóng vai trò sơ cấp) truyền đi trên mạng nhằm thiết lập trạng thái của trạm có địa chỉ trong quá trình trao đổi không đánh số này. Mã thông tin không đánh số (**UI: unnumbered information**) **00 000** được dùng để truyền đi phần đặc thù của thông tin như time/date dùng cho đồng bộ. Frame UI có thể được truyền đi như các lệnh (list các tham số cho quá trình truyền) hay đáp ứng (mô tả về khả năng của trạm có địa chỉ để nhận tin). Mã của xác nhận không đánh số (**UA: unnumbered acknowledgment**) **00 110** được máy thu gửi trả về nhằm trả lời cho một unnumbered poll, xác nhận cho một unnumbered request frame (thí dụ RD: request disconnect) hay là để chấp nhận lệnh thiết lập chế độ (xem lại bảng).

#### Disconnection

**Có ba mã ngừng kết nối**, một là **lệnh từ trạm đóng vai trò sơ cấp hay trạm hỗn hợp**, còn lại là **hai đáp ứng từ trạm thu**. Lệnh đầu tiên, disconnect (**DISC, 00 010**) được trạm thứ nhất gửi đến trạm thứ hai để thông báo ngừng kết nối. Lệnh thứ hai: do máy thứ hai gửi yêu cầu ngừng kết nối request disconnect (**RD, 00 010**) về máy thứ nhất sau khi nhận được DISC. Lệnh thứ ba chế độ ngừng kết nối (**DM: disconnect mode 11 000**) được máy có địa chỉ nhận gửi đến máy phát như một negative response cho lệnh thiết lập chế độ (xem bảng).

### Initialization Mode

Mã **10 000**, được dùng làm lệnh (do trạm thứ nhất gửi đến trạm thứ hai) nhằm thiết lập chế độ khởi tạo (SIM: set initialization mode) nhằm chuẩn bị cho trạm thu chuẩn bị khởi tạo các chức năng điều khiển kết nối dữ liệu. Lệnh SIM và tiếp theo là trường UI chứa các chương trình hay các tham số được thiết lập. Cùng mã này 10 000, được dùng làm đáp ứng (do máy thứ hai gửi về máy thứ nhất), cho biết chế độ yêu cầu khởi tạo (**RIM**: request initialization mode) và cũng có lệnh **SIM** do trạm thứ nhất gửi đến. Lệnh này được dùng để đáp ứng lệnh thiết lập chế độ khi trạm thứ hai không thể hoạt động được theo lệnh without first receiving a SIM (xem bảng).

### Miscellaneous

Trong ba lệnh trên thì hai lệnh đầu: **reset** (RSET, **11 001**) và **trao đổi ID** (XID, **11 101**) là lệnh được gửi từ máy phát đến máy thu theo địa chỉ. Lệnh thứ ba, **frame reject** (FRMR, 10 001) là đáp ứng từ trạm nhận gửi về trạm phát:

**RSET**: cho trạm thứ hai biết là trạm thứ nhất đã reset send sequence numbering và thông báo cho trạm thứ hai để làm các bước tương tự. Lệnh này thường được gửi đi khi nhận được FRMR.

**XID**: yêu cầu trao đổi dữ liệu nhận dạng từ máy thứ hai (Địa chỉ của bạn là gì?)

**FRMR**: báo cho hệ thống thứ nhất là U-frame do trạm thứ hai nhận được có syntax bị sai (điều này không giống như frame HDLC). Thí dụ, tín hiệu này được gửi về khi một frame được nhận dạng là S-frame nhưng lại có chứa trường thông tin.

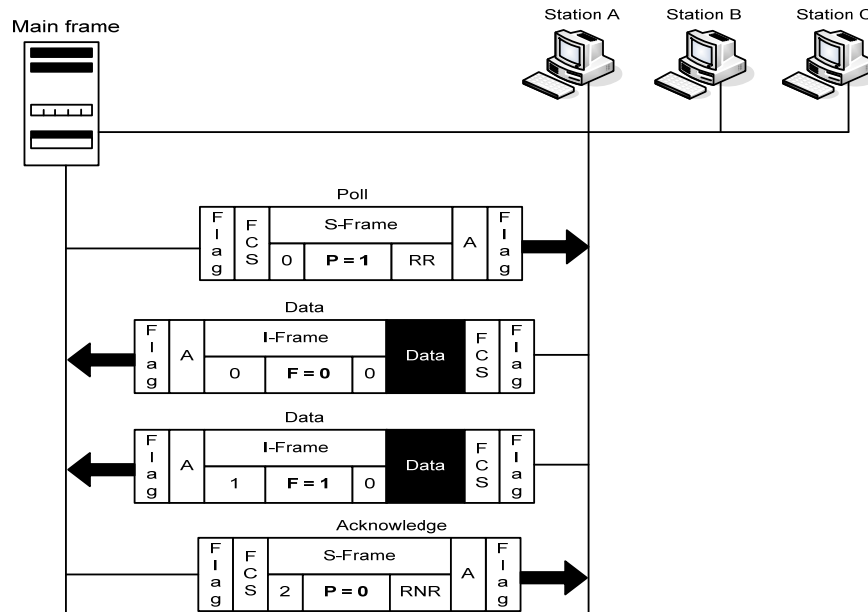
**CÁC THÍ DỤ**: Sau đây là một số thí dụ về phương pháp thông tin dùng HDLC.

#### Thí dụ 1: Poll/Response

Trong hình bên dưới thì tiết bị sơ cấp (mainframe) trong hệ nhiều điểm gửi poll đến thiết bị thứ cấp (A) bằng S-frame chứa mã của poll. Đầu tiên là trường flag, tiếp đến là địa chỉ của thứ cấp cần được poll, trường hợp này là A. Trường thứ ba, điều khiển chứa mã nhận dạng frame là S-frame, theo sau là các mã RR (receive ready), trạng thái máy phát, bit P/F được thiết lập ở poll, và trường  $\text{R} = 0$ . Sau khi trường điều khiển là FCS error detection code và trường ending flag.

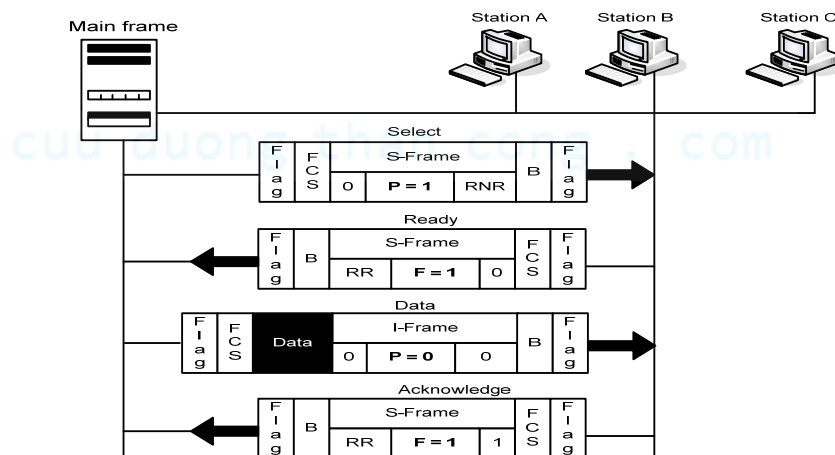
Trạm A có dữ liệu cần gửi, nên trả lời bằng một I-frame đánh số 0 và 1. Frame thứ hai có bit P/F thiết lập về final cho biết chấm dứt dữ liệu. Trạm sơ cấp xác nhận về cả hai frame cùng một lúc dùng S-frame chứa số 2 trong trường  $\text{R}$  cho trạm A biết là frame 0 và 1 đã được nhận và nếu A còn gửi thêm frame nào, thì trạm sơ cấp mong nhận được frame số 2 kế tiếp

cuu duong than cong . com



Hình 11.23

### Thí dụ 2: Select/Response



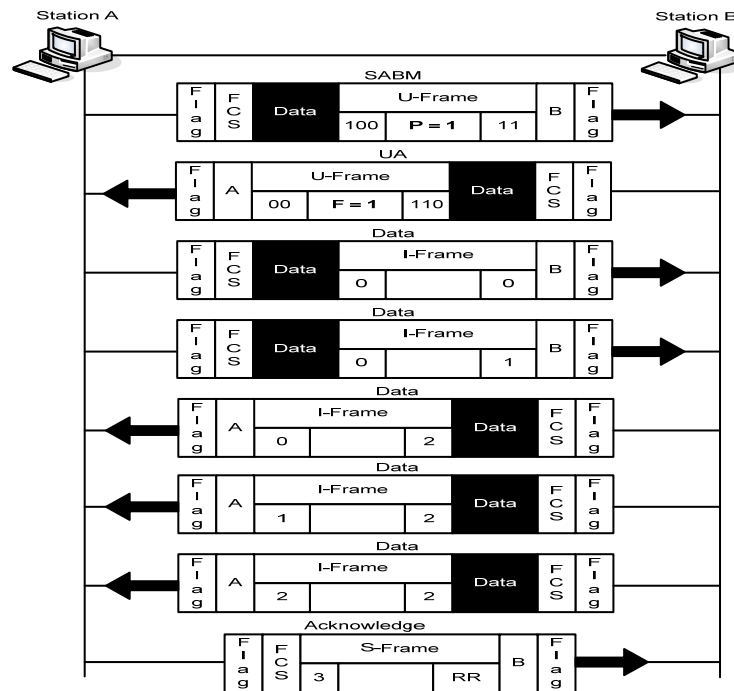
Hình 11.24

Thí dụ này cũng dùng cấu hình nhiều điểm cho thấy cách sơ cấp chọn lựa trạm thứ cấp, trạm B để nhận tín hiệu truyền.

Đầu tiên, sơ cấp gửi S-frame đến trạm địa chỉ B có chứa mã select. Frame select này tương tự như frame poll, nhưng trạng thái RR trong trường điều khiển đã được thay bằng RNR, cho thứ cấp biết để sẵn sàng nhưng chưa gửi. Trạm B trả lời dùng một S-frame khác, định địa chỉ từ B, chứa mã RR cùng với bit final, cho biết là máy đã sẵn sàng nhận và đây là frame cuối.

Sơ cấp gửi I-frame có chứa dữ liệu. Frame này được gửi cho địa chỉ B, trường S nhận dạng là frame số 0, bit P chưa được thiết lập cho thấy frame không phải là poll, và trường R cho thấy là nếu I-frame bị trả về, thì cũng mong được đánh số là 0. Trạm B trả lời dùng frame RR với hai mục tiêu: bit final được thiết lập cho sơ cấp biết là B không có gì để gửi và R=1 cho thấy là B mong nhận được frame 1.

### Thí dụ 3: Peer Devices



Hình 11.25

Thí dụ nhằm minh họa quá trình truyền dẫn trao đổi trong chế độ cân bằng không đồng bộ (ABM) dùng phương pháp xác nhận piggybacking. Hai trạm là đồng quyền và kết nối điểm-điểm.

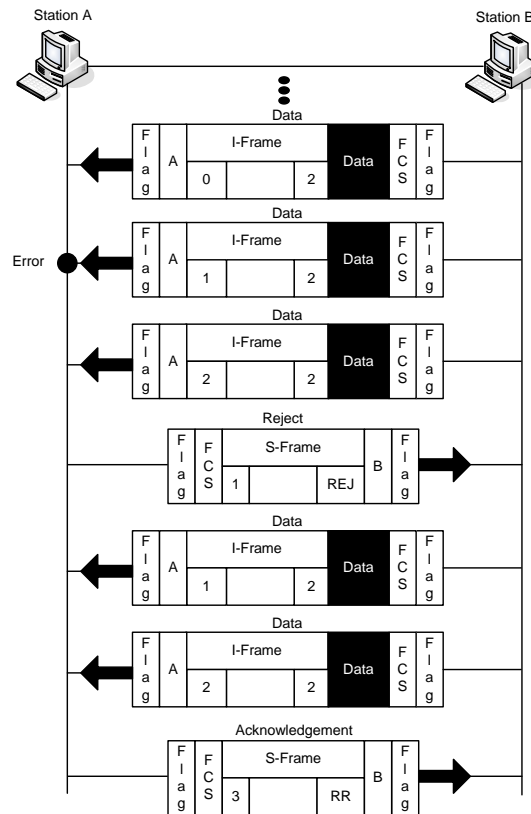
Trạm A gửi U-frame chứa mã của SABM để thiết lập kết nối ở chế độ cân bằng không đồng bộ. Bit P được thiết lập cho biết trạm A muốn điều khiển kết nối và được phát đầu tiên. Trạm B đồng ý yêu cầu bằng cách gửi về U-frame chứa mã UA, thiết lập bit F. Chấp nhận truyền ở chế độ cân bằng không đối xứng, cả hai trạm đều đang ở chế độ trạm hỗn hợp, thay vì sơ cấp-thứ cấp, nên bit P/F không còn giá trị và được bỏ qua trong frame kế tiếp.

Trạm A bắt đầu trao đổi thông tin với I-frame đánh số 0 tiếp theo là I-frame đánh số 1. Trạm B gửi xác nhận piggybacking của cả hai frame này vào trong I-frame của mình. I-frame đầu tiên của B cũng được đánh số 0 (trường N (S)) và chứa 2 trong trường R, xác nhận đã nhận frame số 1 và số 0 của A và cho biết mong muốn nhận được frame 2. Trạm B gửi tiếp frame thứ hai và ba (đánh số 1 và 2) trước khi chấp nhận thêm các frame đến từ A. Các thông tin R, như thế thì chưa thay đổi. Các frame 1 và 2 của B cho biết là trạm B còn mong nhận được frame 2 từ A.

Trạm A đã gửi hết dữ liệu. ả hư thế, không có thể piggybacking xác nhận vào trong I-frame nên phải gửi S-frame thay vào. Mã RR cho thấy là A vẫn còn sẵn sàng nhận. Số 3 trong trường N (R) cho B biết là các frame 0, 1, và 2 đã được chấp nhận và hiện đang chờ số 3.

### Thí dụ 4: Peer Communication with Error

Trong thí dụ vừa qua, hãy giả sử là frame 1 gửi từ B đến A là có lỗi. Trạm A, cần báo cho B để gửi lại frame 1 và 2 (hệ thống dùng giao thức go-back-n). Trạm A gửi một frame reject supervisory nhằm thông báo lỗi ở frame 1.



Hình 11.26

## 11.2 THỦ TỤC TRUY CẬP KẾT NỐI MẠNG

Ấp hiệu giao thức dựa trên phạm trù **link access procedure (LAP)** đã được phát triển. Các giao thức đặc thù này là các tập con của HDLC trong các ứng dụng cụ thể, trong đó LAPB, LAPD và LAPM là thông dụng nhất.

### LAPB

Link **access Procedure, Balanced (LAPB)**, đơn giản là tập con của HDLC và chỉ được dùng để **kết nối một trạm với mạng**. Chuẩn này chỉ cung cấp những chức năng cơ bản cần thiết để thông tin giữa **DTE và DCE** (tức là không có các ký tự poll và select)

LAPB chỉ được dùng trong các cấu hình đối xứng giữa hai thiết bị. Thông tin thường ở chế độ không đối xứng cân bằng. LAPB hiện được dùng trong **ISD**, **kênh B**

### LAPD

**Link access procedure for D channel (LAPD)** là một tập con đơn giản của HDLC dùng trong ISD và dùng với chế độ không đồng bộ cân bằng (ABM)

### LAPM

**Link access procedure for modems (LAPM)** là tập con đơn giản của HDLC dùng cho modem. Được thiết kế cho **chuyển đổi không đồng bộ-đồng bộ, phát hiện lỗi, và chuyển lại**. Được phát triển cho các ứng dụng của HDLC trên modem