



# Chương 3 :

# Các Nghi Thức Lớp Liên Kết

# Dư Liệu

cuu duong than cong . com



# NỘI DUNG CHÍNH

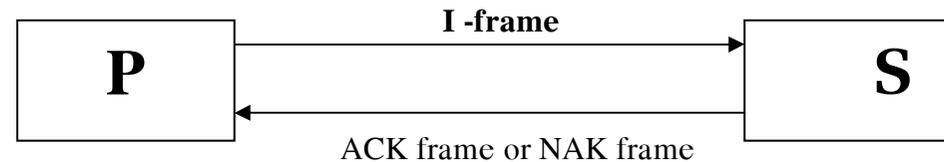
---

- Kiểm soát lỗi (Error Control).
- Kiểm soát luồng ( Flow Control)
- Quản lý kết nối ( Connection management).
- Nghi thức Lớp liên kết dữ liệu ( DATA LINK PROTOCOL)

[cuduongthancong.com](http://cuduongthancong.com)



## Một số ký hiệu



- P : Primary (phía sơ cấp) là phía gửi đi các frame dữ liệu.
- S : Secondary (phía thứ cấp) là phía thu các frame dữ liệu từ P.
- I : frame (information frame): khung thông tin chứa dữ liệu phía phát truyền cho phía thu.
- I(N) : Số tuần tự của khung thông tin đó
- ACK frame (Acknowledge frame) : S truyền tới P để báo là đã nhận dữ liệu tốt (không bị lỗi).
- NAK frame (Negative Acknowledge frame): S truyền tới P để báo là đã nhận dữ liệu sai (bị lỗi).



# Kiểm soát lỗi (Error Control).

- Phía nhận khi nhận được frame sẽ kiểm tra có lỗi hay không, sau đó có 2 khả năng
  - Gởi lại phía phát bản tin điều khiển để xác nhận là khung tin không lỗi.
  - Gởi lại phía phát bản tin điều khiển để yêu cầu phát lại khung tin nếu khung tin lỗi.
- Quá trình này diễn ra tự động nên gọi là Automatic Repeat Request (ARQ)

cuu duong than cong . com



# Kiểm soát lỗi

- Các phương pháp kiểm soát lỗi:
  - Idle RQ ( Stop and Wait )
    - Implicit ( Hiểu ngầm )
    - Explicit ( Tương minh )
  - Continuous RQ
    - Selective Repeat
      - Implicit ( Hiểu ngầm )
      - Explicit ( Tương minh )
    - Go back N



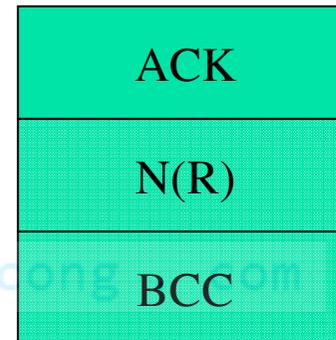
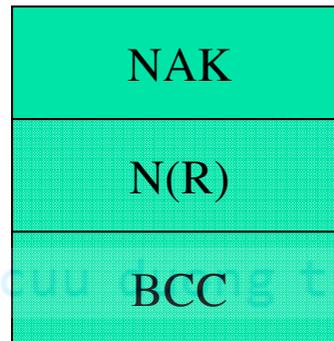
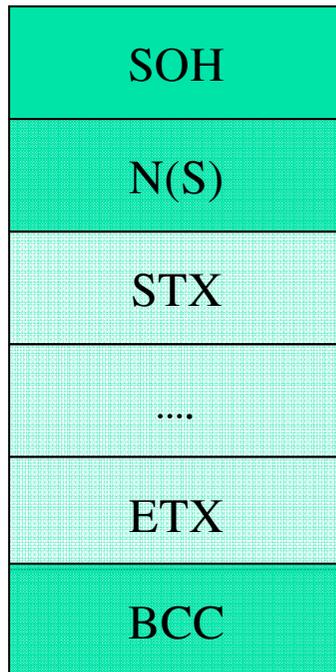
## Idle RQ ( Stop and Wait )

- Ứng dụng:
  - Sử dụng trong kiểu truyền số liệu định hướng ký tự (*character-oriented*).
  - Hoạt động theo chế độ bán song công.
- Định dạng của các frame trong Idle RQ như sau:
  - Có 3 loại frame : I-frame, ACK-frame, NAK-frame.
  - Các frame này gọi là PDU (Protocol Data Unit) trong Idle RQ



# Idle RQ ( Stop and Wait )

## PDU – Protocol Data Units



NAK- frame format    ACK- frame format

- N(S) – Send Sequence Number
- N(R) – Receive Sequence Number
- SOH – Start of Header
- STX – Start of Text
- ETX – End of Text
- BCC – Block (sum) Check Character
- ACK – Acknowledge
- NAK – Negative Acknowledge



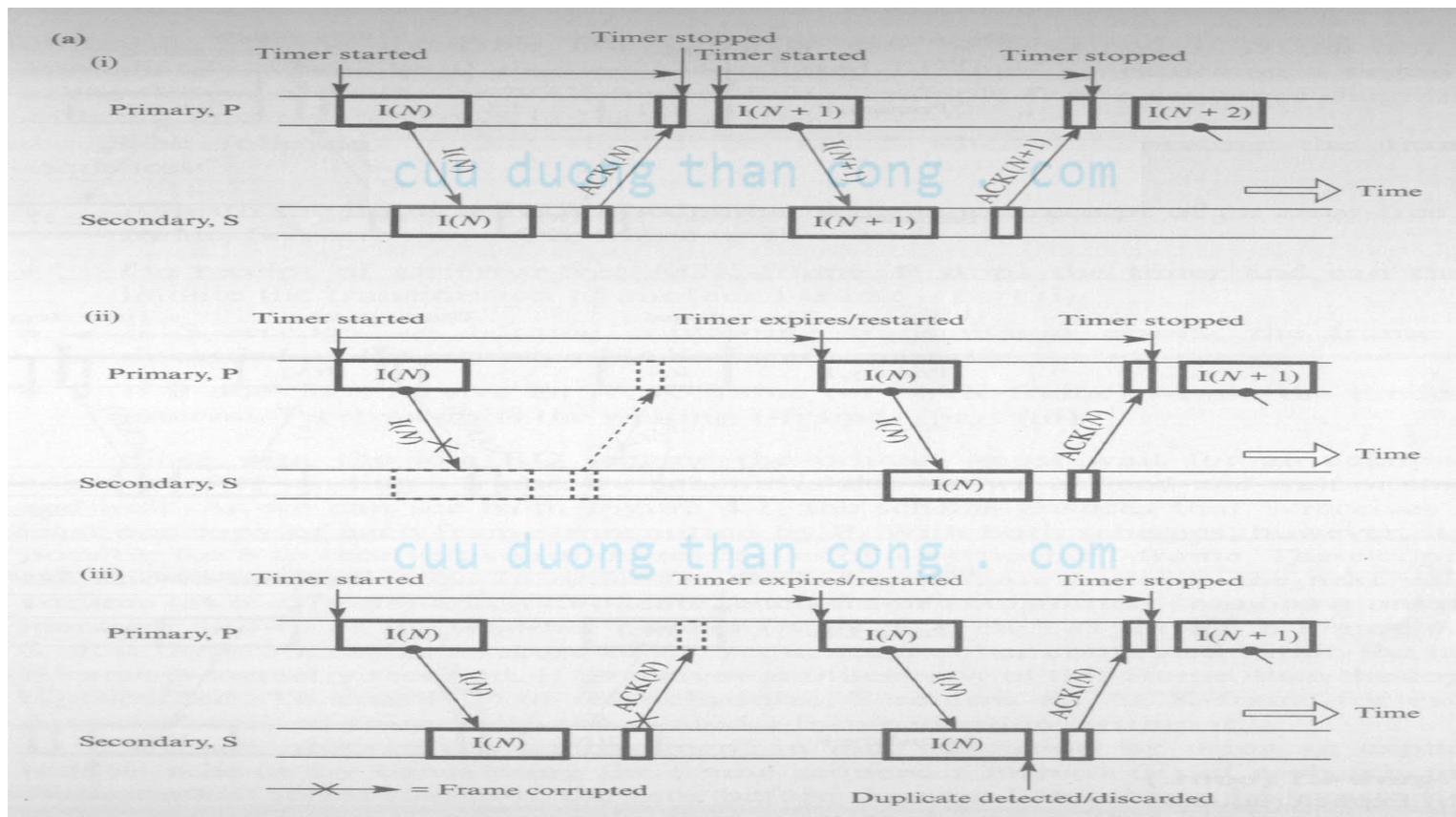
## Idle RQ ( Stop and Wait )

- Đặc điểm :
  - P chỉ có một I – frame đang chờ ACK tại một thời điểm
  - Khi nhận một I – frame không bị lỗi, S truyền lại P một ACK frame, khi P nhận được ACK của frame N, P sẽ tiếp tục truyền I – frame kế tiếp ( $N+1$ ).
  - Khi P bắt đầu truyền I – frame, nó sẽ khởi động bộ định thời (Timer start), nếu quá khoảng thời gian giới hạn (time expires/restarts ) mà không nhận được frame trả lời từ S thì P sẽ truyền lại frame đó.
  - Nếu S nhận được cùng 1 frame 2 lần thì sẽ loại bỏ bản copy. Điều này thực hiện được do trong mỗi I-frame P đều truyền kèm theo số tuần tự của frame.
  - Không tốn nhiều bộ nhớ đệm
  - Hiệu suất sử dụng đường truyền thấp



# Idle RQ ( Stop and Wait )

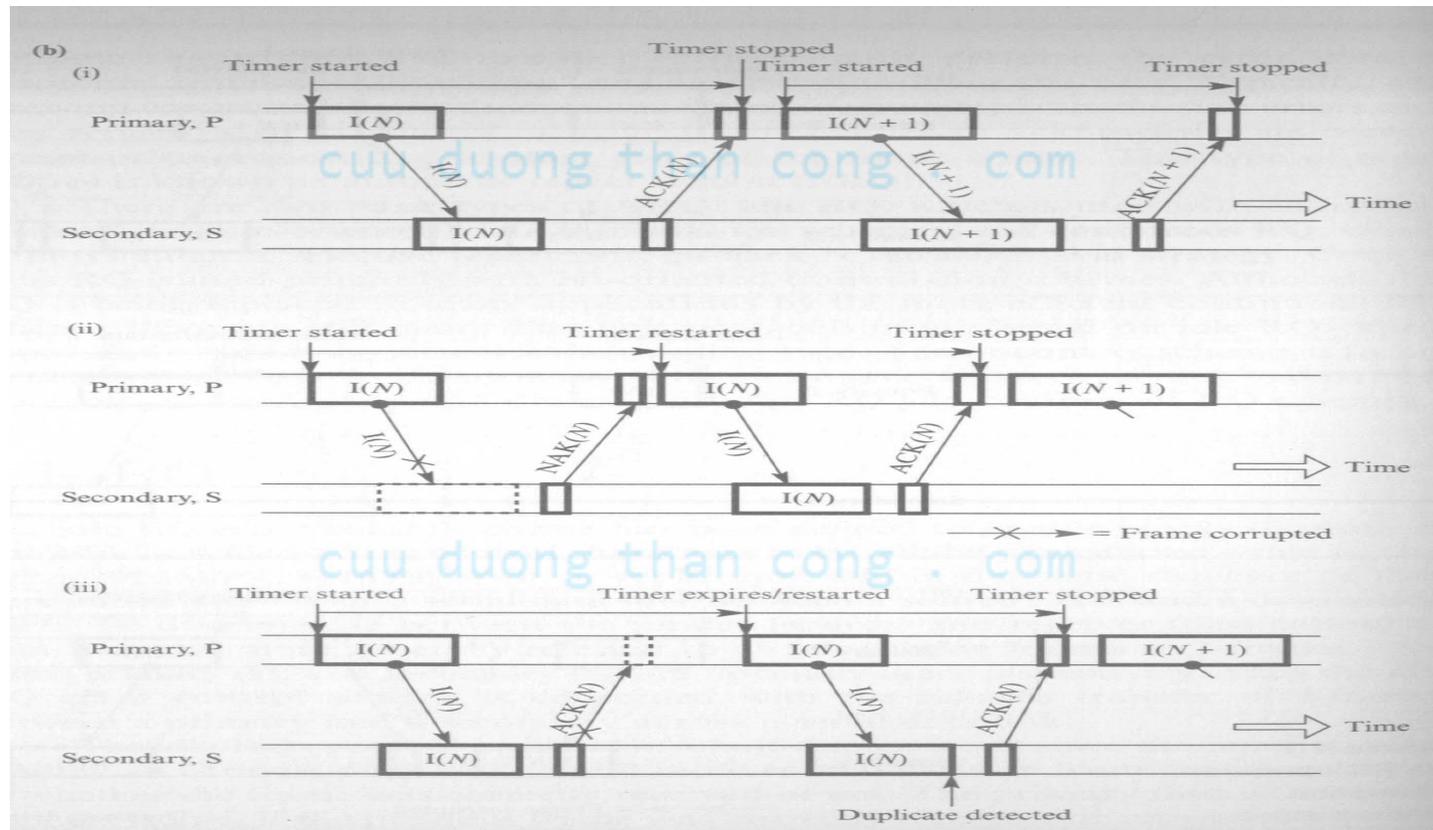
- Idle RQ - Implicit ( Hiểu ngầm )
  - **Ví dụ:** Khi 1 khung I(N) bị lỗi và khi ACK (N) bị lỗi





# Idle RQ ( Stop and Wait )

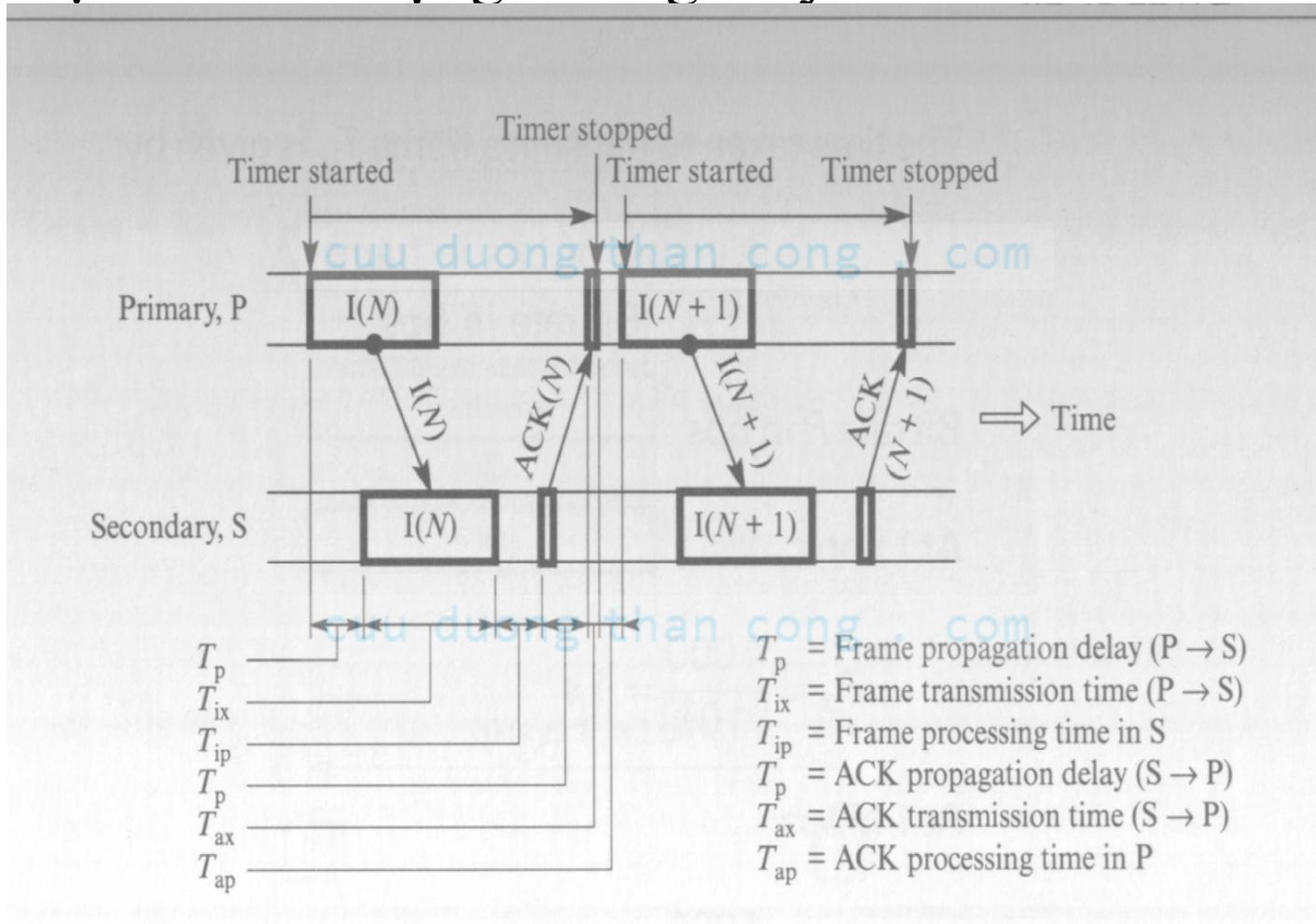
- Idle RQ - Explicit ( Tường minh )
  - Ví dụ:** Khi 1 khung I(N) bị lỗi và khi ACK (N) bị lỗi





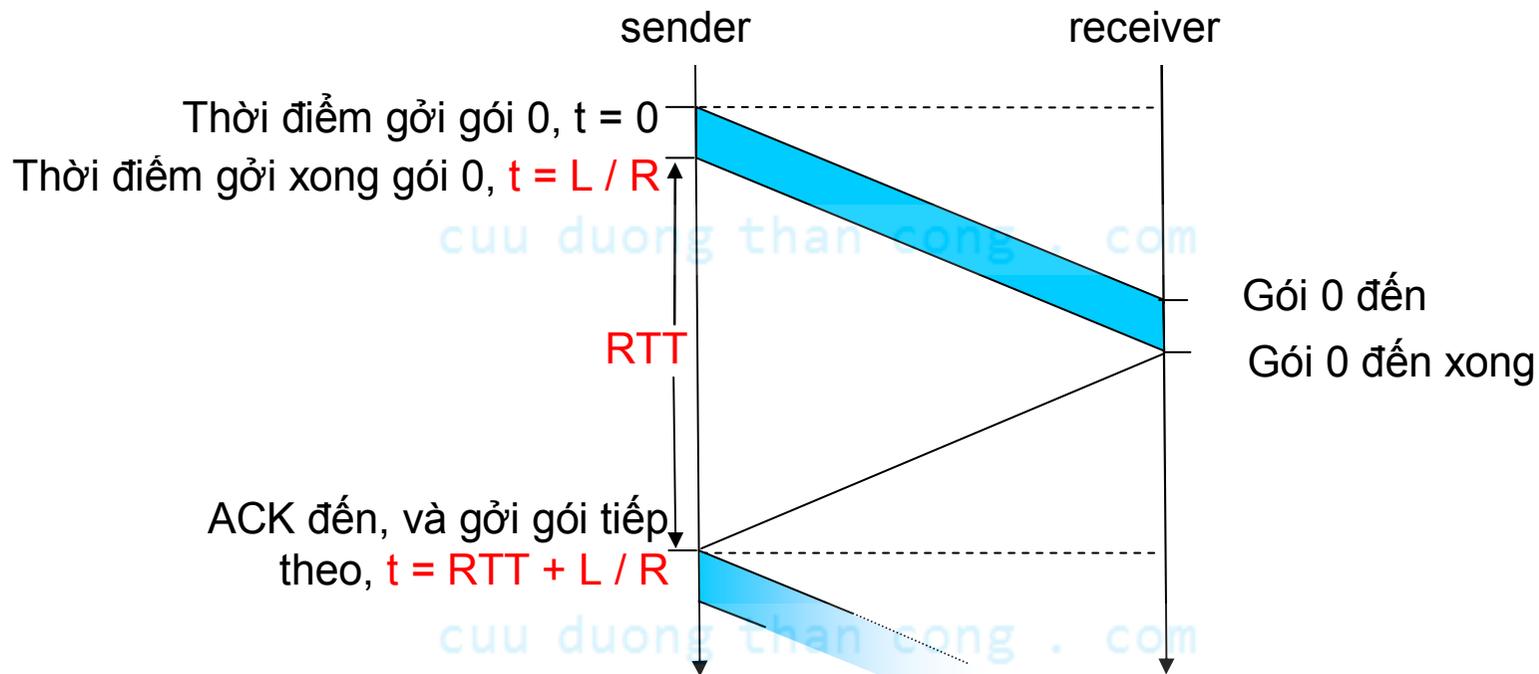
# Idle RQ ( Stop and Wait )

- Hiệu suất sử dụng đường truyền





# Idle RQ ( Stop and Wait )





## Idle RQ ( Stop and Wait )

- Trường hợp truyền không có lỗi, thì thời gian tổng để hoàn thành việc truyền và xử lý một I – frame:

$$T_t = T_{ix} + 2T_p + T_{ap} + T_{ip} + T_{ax}$$

- Thông thường  $T_{ap}$ ,  $T_{ip}$  và  $T_{ax}$  rất nhỏ so với  $T_p$  và  $T_{ix}$  do đó:

$$T_t \approx T_{ix} + 2T_p$$

- Hiệu suất liên kết được định nghĩa là tỷ số của thời gian P phát một frame  $T_{ix}$  trên thời gian tổng để hoàn thành việc truyền một frame đó  $T_t$

$$U = \frac{T_{ix}}{T_t} \approx \frac{T_{ix}}{T_{ix} + 2T_p} = \frac{1}{1 + 2a}$$

Với

$$a = \frac{T_p}{T_{ix}}$$



## Idle RQ ( Stop and Wait )

- Trường hợp có lỗi, các frame bị lỗi phải thực hiện việc truyền lại. Giả sử để truyền thành công 1 frame thì trung bình có  $N_r$  frame truyền lại, do đó xác suất một frame không lỗi là  $1/N_r$  và thời gian truyền tổng cộng :

$$T_i = N_r T_{ix} + 2N_r T_p$$

- Hiệu suất: 
$$U = \frac{T_{ix}}{T_i} = \frac{T_{ix}}{N_r T_{ix} + 2N_r T_p} = \frac{1}{N_r \left( 1 + 2 \frac{T_p}{T_{ix}} \right)} = \frac{1}{N_r (1 + 2a)}$$
- Gọi P là xác suất một bit bị lỗi, khi đó xác suất một frame (chiều dài  $N_i$ ) bị lỗi là:

$$P_f = 1 - (1 - P)^{N_i} \approx N_i P, \text{ nếu } N_i P \ll 1$$

- Khi đó, xác suất frame không bị lỗi là  $1 - P_f$  và do vậy

$$N_r = \frac{1}{1 - P_f}$$

$$U = \frac{1}{N_r (1 + 2a)} = \frac{1 - P_f}{1 + 2a}$$



## Continuous RQ

- Đặc điểm:
  - P truyền các I frame tới S một cách liên tục mà không dừng lại để chờ ACK frame truyền về từ S.
  - Khi có nhiều hơn 1 I-frame chờ ACK, P giữ lại bản sao của các I frame trong bộ đệm truyền lại (*retransmission list*) hoạt động theo nguyên tắc FIFO
  - S trả về ACK frame cho mỗi I frame nhận đúng.
  - Mỗi I frame chứa số thứ tự được trả về trong ACK
  - Khi P nhận được ACK thì sẽ loại bỏ I – frame tương ứng ra khỏi danh sách.
  - Các I frame nhận được không lỗi được S chứa trong bộ đệm thu (*link receive list*) để chờ xử lý.
  - S luôn chờ các I frame kế tiếp theo thứ tự để xử lý. Trong trường hợp frame nhận được không đúng thứ tự (giả sử trước đó nhận frame N kế đến nhận frame N+2 ) thì S sẽ giữ lại tất cả các I frame trong bộ đệm thu cho đến khi nhận lại được frame theo đúng thứ tự (frame N+1). Ngoài trừ nghi thức Go Back N, bộ đệm thu bên S luôn luôn chỉ giữ lại đúng 1 I- frame vừa nhận được.

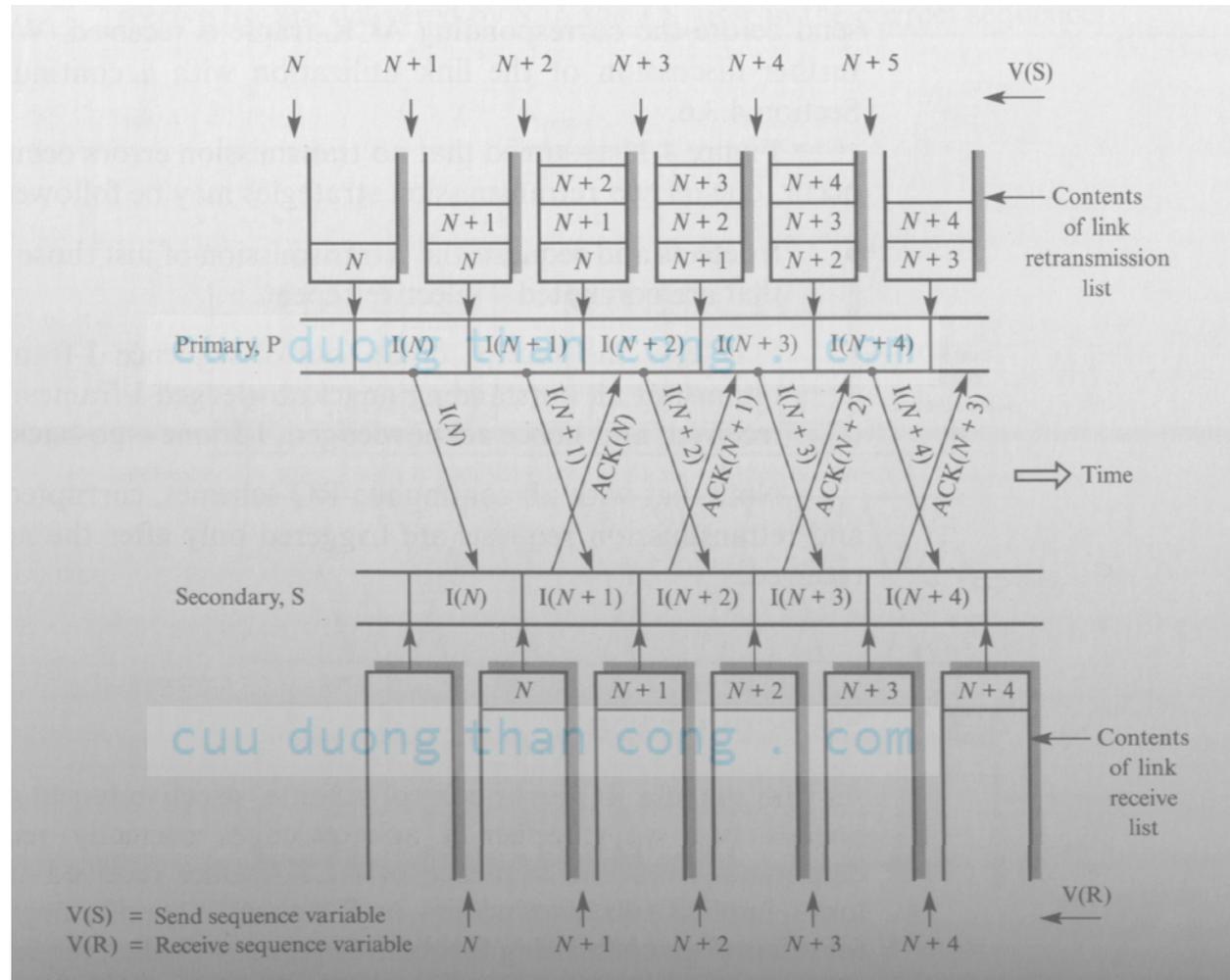


## Continuous RQ

- Trong trường hợp có lỗi xảy ra trong quá trình truyền dẫn, có 2 cách truyền lại được áp dụng như sau:
  - S phát hiện và yêu cầu P truyền lại chỉ những frame bị lỗi. Kiểu truyền lại này được gọi là truyền lại có lựa chọn (*selective – repeat*).
  - S phát hiện và yêu cầu P truyền lại những frame chưa được trả lời ACK, nghĩa là tất cả các frame kể từ frame cuối cùng nhận đúng. Kiểu truyền lại này được gọi là lặp lại N (*go-back-N*).
- Hiệu suất sử dụng đường truyền cao.
- Cần bộ đệm lớn.



# Continuous RQ





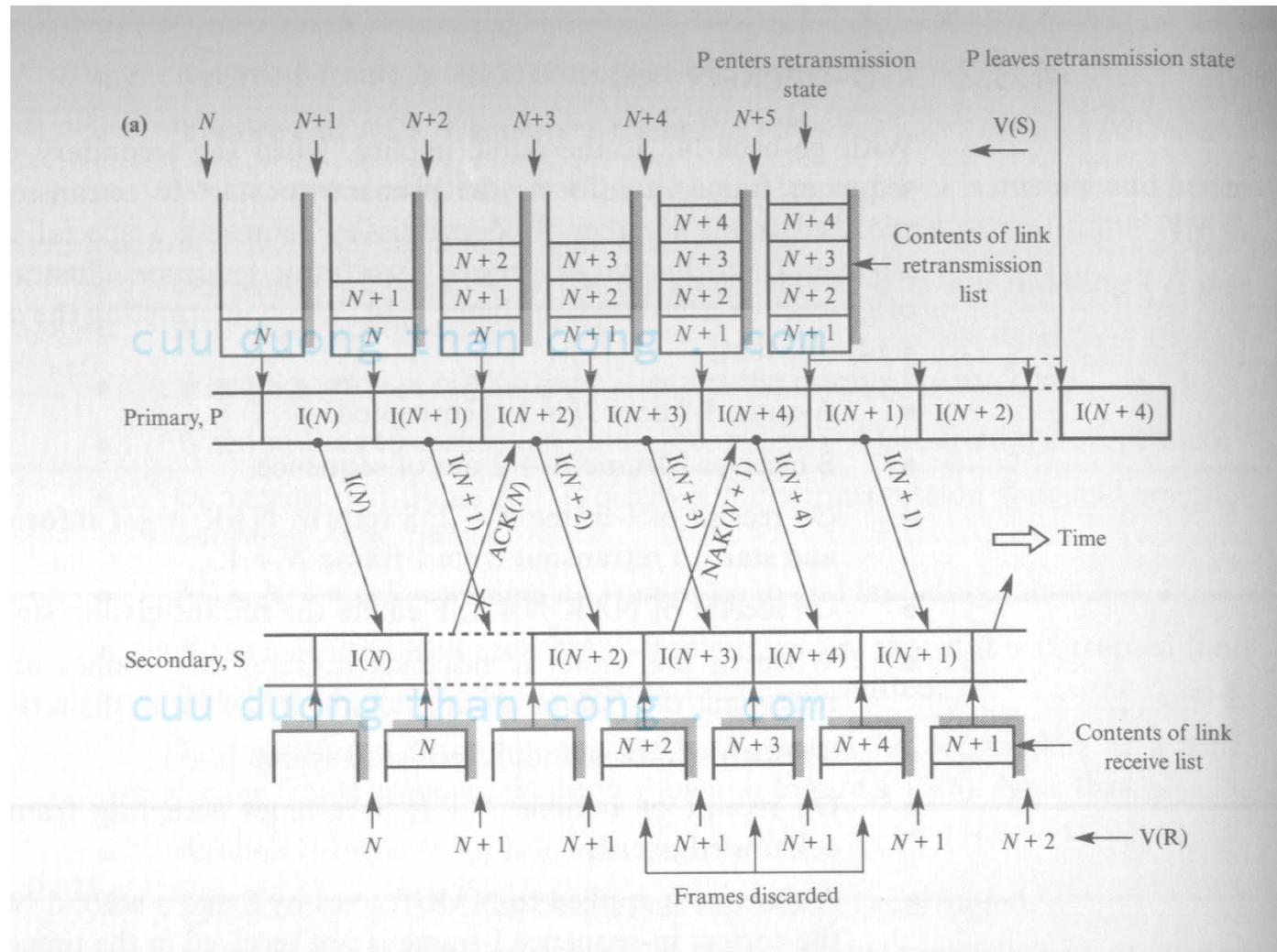
# Continuous RQ

- Go back N
  - Giả sử  $I(N+1)$  bị lỗi
  - S nhận  $I(N+2)$  không đúng thứ tự (chưa nhận được frame  $N+1$ )
  - S gửi NAK ( $N+1$ ) cho P để báo P bắt đầu truyền lại từ frame  $N+1$  và bắt đầu khởi động timer để chờ nhận  $I(N+1)$ , nếu quá một khoảng thời gian xác định mà không nhận được  $I(N+1)$  thì S truyền lại NAK( $N+1$ ) ( để phòng trường hợp NAK( $N+1$ ) bị lỗi).
  - S vào trạng thái truyền lại (Retransmission), tạm thời không trả lời ACK cho bất kỳ frame nào nhận được và chờ  $I(N+1)$ .
  - Khi nhận được frame  $N+1$ , S trả lời ACK ( $N+1$ ) và ra khỏi trạng thái truyền lại.
  - Bên P khi gửi  $I(N+1)$  – frame thì cũng khởi động timer. Sau khoảng thời gian Time Expires mà không nhận được tín hiệu trả lời của frame này thì sẽ truyền lại frame đó.
  - Bộ đệm thu không cần dung lượng lớn.



# Continuous RQ

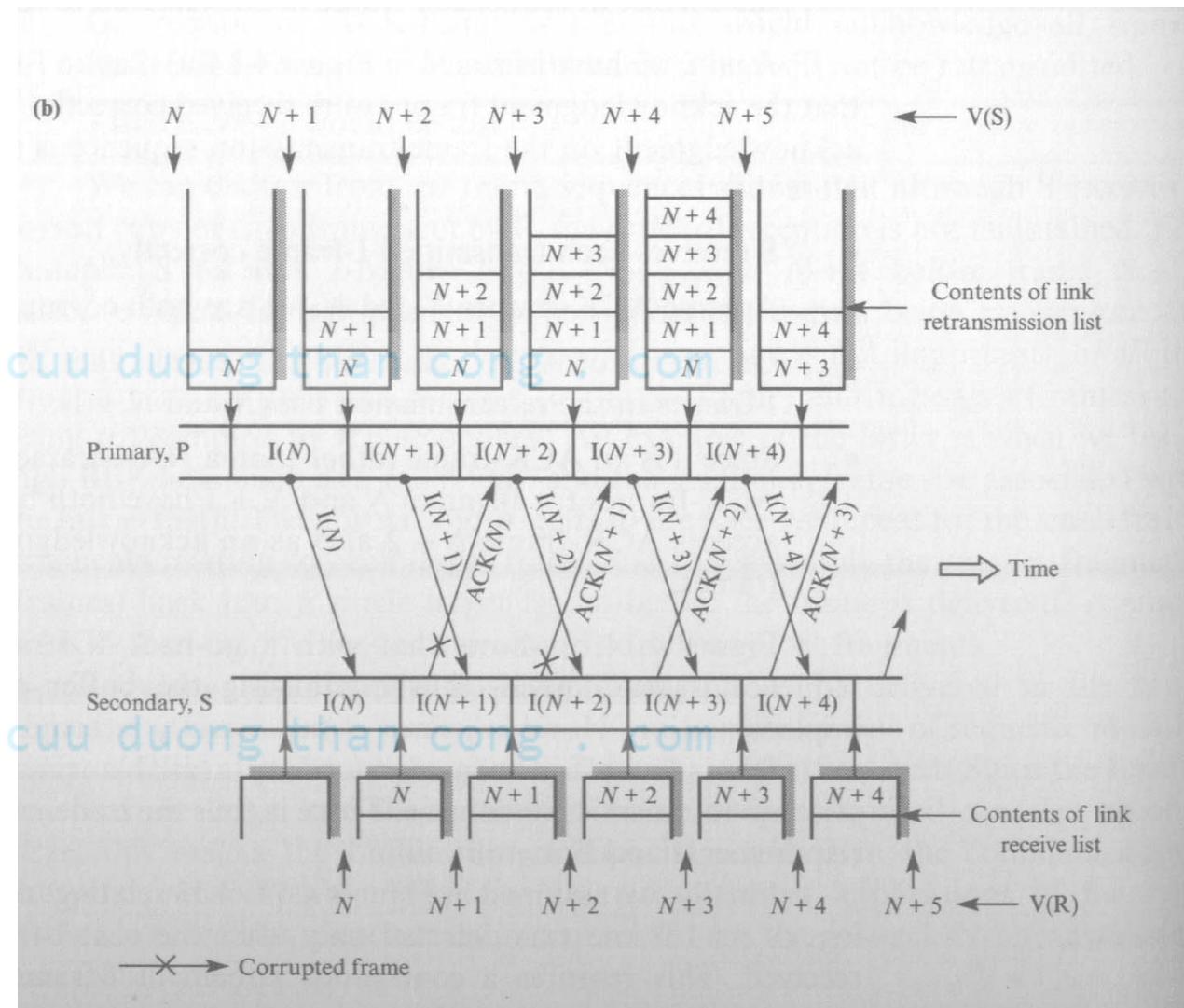
**Ví dụ:** Khi  
1 khung  
 $I(N+1)$  bị  
lỗi





# Continuous RQ

**Ví dụ:** Khi  
1 khung  
ACK bị lỗi





# Continuous RQ

## ■ Selective repeat

- Bên phát chỉ phát lại các frame bị lỗi, còn các frame trước đó không bị lỗi sẽ không phát lại. Có 2 cách thực hiện điều này :

- **Implicit Retransmission:**

- **Giả sử I-frame N+1 bị lỗi :**

- S trả về ACK frame cho những I-frame đúng (N, N+2, N+3,...).
    - Khi nhận được ACK của I-frame N+2, P nhận thấy ACK (N+1) chưa nhận được -> bị lỗi -> P xoá I-frame N+2 ra khỏi bộ đệm và truyền lại frame N+1.

- **Giả sử ACK(N) bị lỗi :**

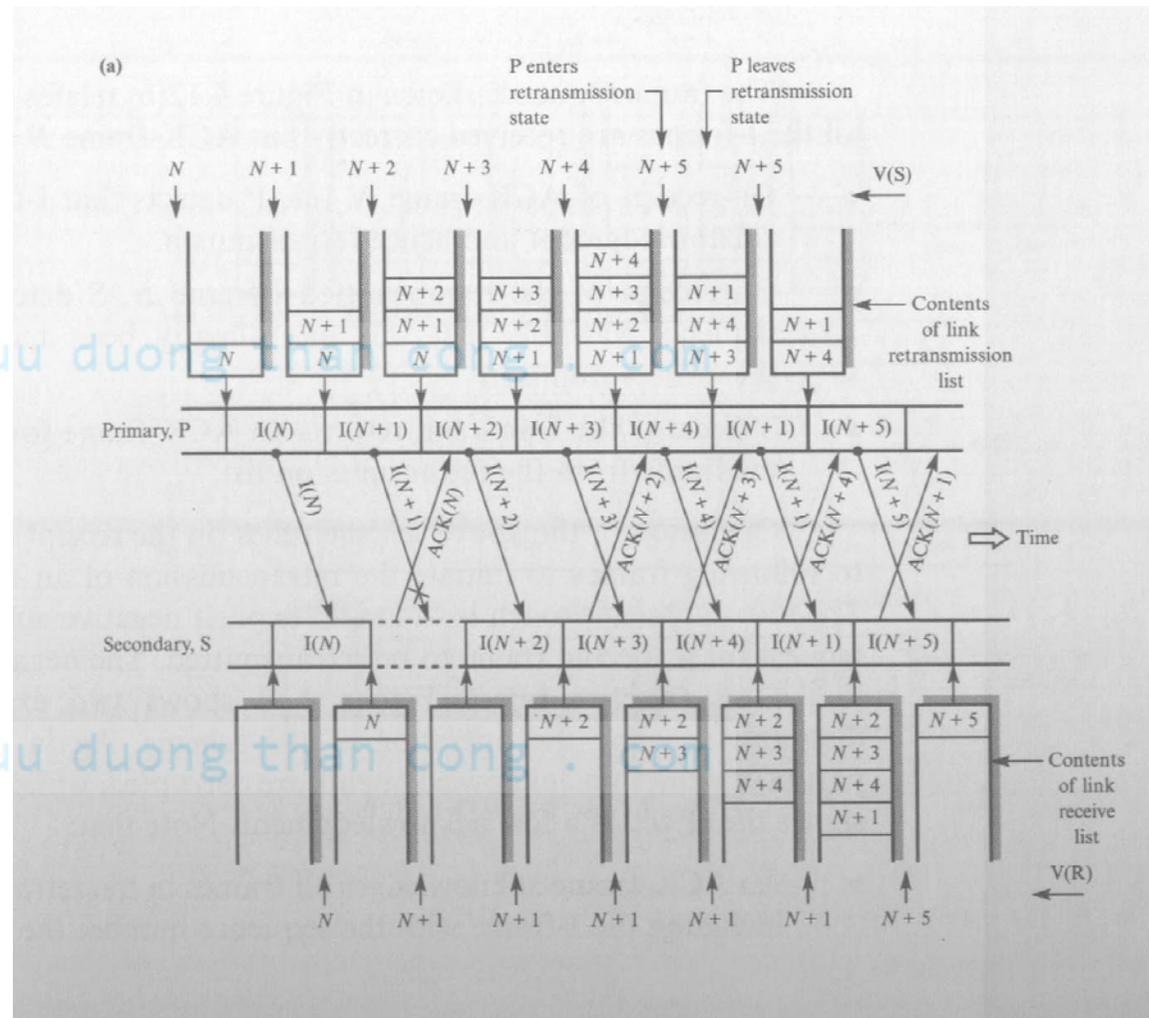
- Khi nhận được ACK của frame N+1, P phát hiện ACK(N) chưa nhận được, có nghĩa là frame N bị lỗi do đó P truyền lại frame N.
    - Khi nhận frame N lần thứ 2, S xác định được sự trùng lặp và do đó bỏ qua, tuy nhiên S vẫn truyền trở về ACK(N) để đảm bảo P xoá I-frame N ra khỏi bộ đệm. â2n bộ đệm thu lớn

- Bộ đệm thu cần dung lượng lớn.



# Continuous RQ

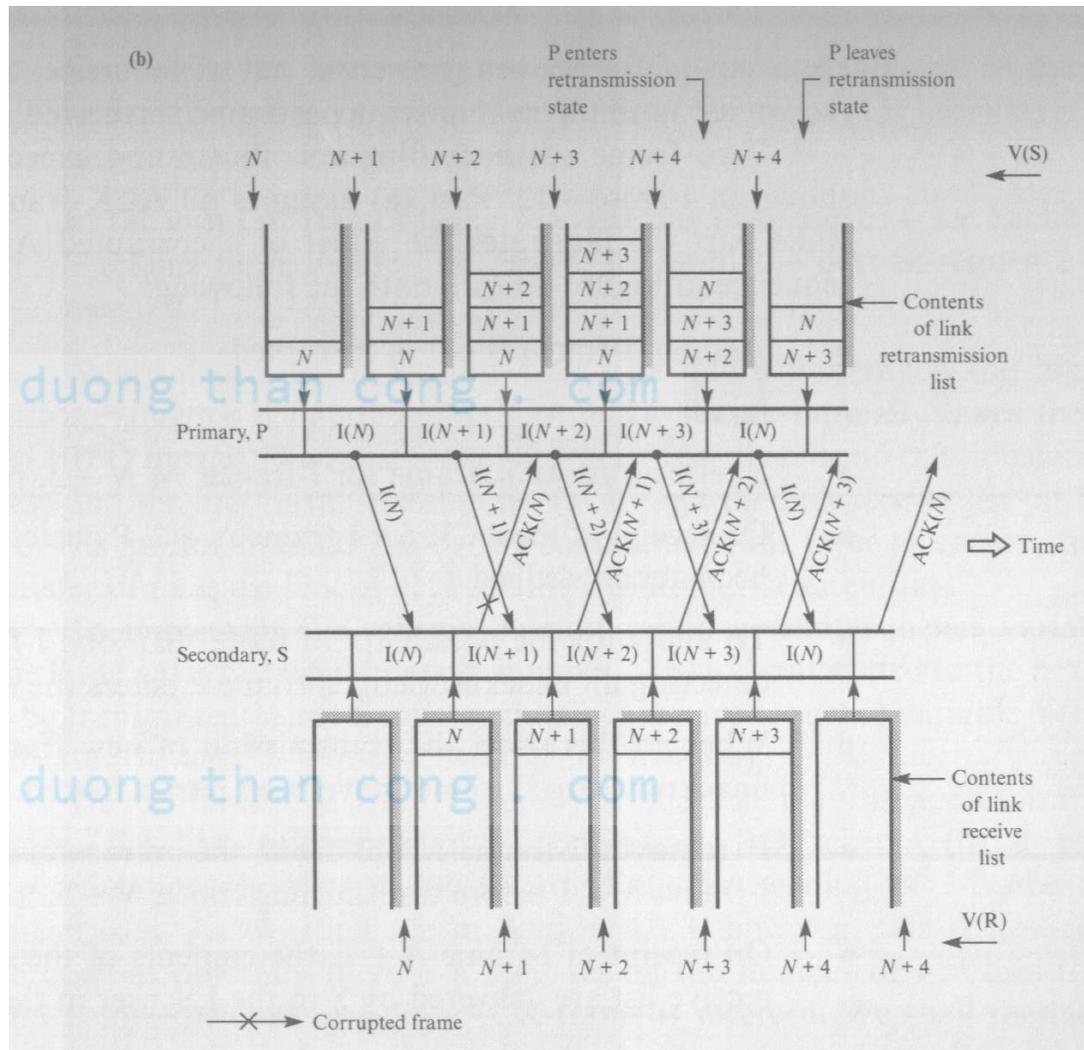
**Ví dụ:** Khi  
1 khung  
 $I(N+1)$  bị  
lỗi





# Continuous RQ

**Ví dụ:** Khi  
1 khung  
ACK bị lỗi





- **Selective repeat**

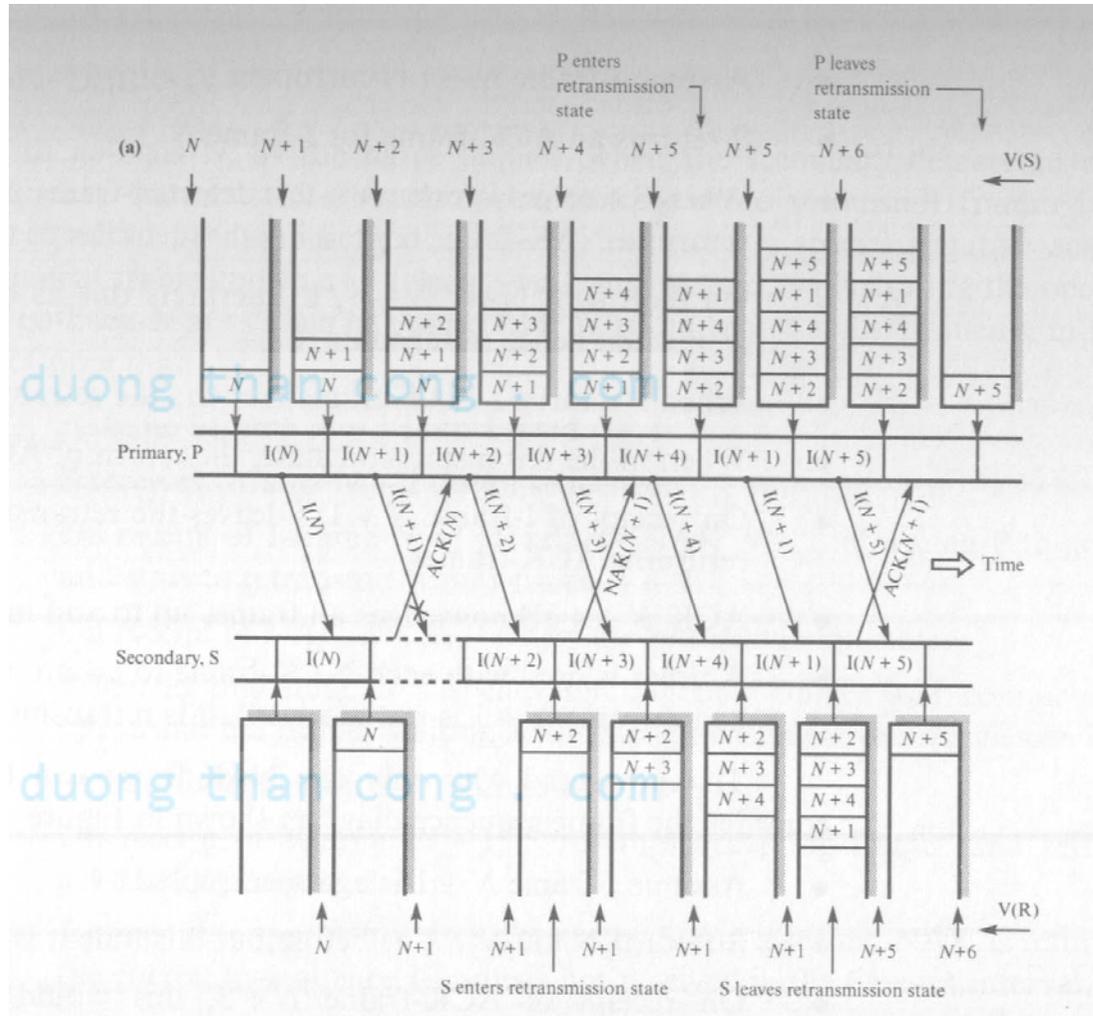
- **Explicit Retransmission**

- Bên P khi phát một I frame sẽ giữ lại bản copy của I frame đó trong bộ đệm để chờ tín hiệu trả lời.
- Khi nhận được frame không lỗi, S sẽ trả lời ACK.
- Khi P nhận ACK (N), P sẽ loại bỏ tất cả các I-frame trước I(N) và chính nó ra khỏi bộ đệm.
- Khi S không nhận được frame bất kỳ giả sử I(N+1), S sẽ gửi P NAK(N+1), và chuyển sang chế độ Retransmission (*trong chế độ này S sẽ không trả lời ACK cho bất kỳ I-frame nào nhận được*), đồng thời khởi động timer (*Để phòng trường hợp NAK lỗi thì sau thời gian timeout sẽ truyền lại cho đến khi nhận được I(N+1). Nếu không truyền lại thì có khả năng I(N+1) sẽ không bao giờ thu được khi NAK(N+1) bị lỗi (hình b)*).
- Khi nhận được NAK(N+1) thì P gửi lại I(N+1).
- Khi S nhận được I(N+1) thì gửi lại P ACK(N+1) và thoát khỏi trạng thái Retransmission.



# Continuous RQ

**Ví dụ:** Khi  
1 khung  
 $I(N+1)$  bị  
lỗi







- Continuous RQ
  - Trường hợp đường truyền lỗi
    - Selective Repeat

- $K \geq 1 + 2a$  :  $U = 1 - P_f$

- $K < 1 + 2a$

$$U := \frac{K}{N_r(1+2a)} = \frac{K(1-P_f)}{1+2a}$$

- Go Back N

- $K \geq 1 + 2a$  :

$$U = \frac{1 - P_f}{1 + P_f(K - 1)}$$

- $K < 1 + 2a$

$$U = \frac{K(1 - P_f)}{(1 + 2a)(1 + P_f(K - 1))}$$



# Continuous RQ

## Ví dụ:

- Một chuỗi các khung dữ liệu  $L=1000$  bit được truyền theo nghi thức RQ rảnh. Biết tốc độ lan truyền sóng trên liên kết là  $V=2.108\text{m/s}$ . Hãy xác định hiệu suất liên kết với các tuyến liên kết sau:
- Tuyến liên kết có chiều dài  $S=10\text{km}$ , tỷ lệ lỗi bit là  $\text{BER}=10^{-4}$  và tốc độ truyền dữ liệu là  $R=9600\text{bps}$
- Tuyến liên kết có chiều dài  $S=100\text{km}$ , tỷ lệ lỗi bit là  $\text{BER}=10^{-4}$  và tốc độ truyền dữ liệu là  $R=10\text{Mbps}$



# Continuous RQ

## Bài giải:

- $S=10\text{km}$ ,  $\text{BER}=10^{-4}$ ,  $R=9600\text{bps}$

- Thời gian lan truyền sóng:  $T_p = \frac{S}{V} = \frac{10 * 10^3}{2 * 10^8} = 5 * 10^{-5} \text{(s)}$

- Thời gian phát một khung dữ liệu:  $T_{ix} = \frac{L}{R} = \frac{1000}{9600} \approx 0.1 \text{(s)}$

- Xác suất truyền đúng mỗi khung:  $P_f = (1 - \text{BER})^L = (1 - 10^{-4})^{1000} \approx 0.905$

- Hiệu suất liên kết:  $U_1 = \frac{T_{ix} * P_f}{(T_{ix} + 2 * T_p)} = \frac{0.1 * 0.905}{(0.1 + 2 * 5 * 10^{-5})} \approx \mathbf{0.905}$

- $S=100\text{km}$ ,  $\text{BER}=10^{-4}$ ,  $R=10\text{Mbps}$

$$U_2 = \frac{T_{ix} * P_f}{(T_{ix} + 2 * T_r)} = \frac{10^{-4} * 0.905}{(10^{-4} + 2 * 50 * 10^{-5})} \approx \mathbf{0.0905} \ll U_1$$



# KIỂM SOÁT LƯỒNG ( FLOW CONTROL)

## ■ Mục đích :

- Điều khiển việc truyền dữ liệu giữa bên phát và bên thu sao cho đảm bảo bên thu luôn luôn có thể nhận được dữ liệu vào bộ đệm trước khi xử lý.
- Theo các cơ chế điều khiển lỗi trên, có những thời điểm bộ đệm bên thu bị quá tải do lượng lớn dữ liệu truyền đến nhưng chưa được xử lý. Do đó việc mất dữ liệu có thể xảy ra. Vì vậy phải báo bên phát biết để ngưng phát và chỉ phát lại khi bên thu đã sẵn sàng nhận.
- Có 2 phương pháp điều khiển luồng :
  - X-ON/X-OFF
  - Sliding Window



# KIỂM SOÁT LUỒNG

## ■ X-ON/X-OFF

- Khi bộ đệm bên S bị quá tải sẽ gửi ký hiệu X-OFF về P, P sẽ ngưng việc truyền dữ liệu.
- Khi S thoát khỏi trạng thái quá tải thì sẽ gửi ký hiệu X-ON về P, P tiếp tục quá trình phát dữ liệu.
- Ví dụ trong RS232, chân RTS và CTS được dùng để điều khiển luồng.

cuu duong than cong . com



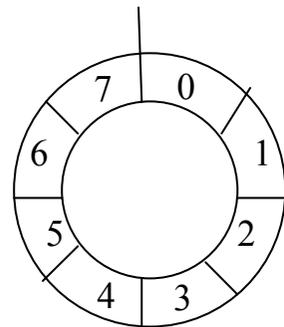
# KIỂM SOÁT LƯỠNG

- Sliding Window
  - Cơ chế :
    - Các I frame đã phát đi và đang đợi ACK được đặt trong cửa sổ gọi send Window.
    - Giới hạn số lượng I frame được phát trước khi nhận ACK là K (K là kích thước Send Window).
    - Mỗi khi phát một I Frame, cạnh trên cửa sổ UWE tăng lên 1.
    - Mỗi khi nhận một ACK, cạnh dưới cửa sổ LWE tăng lên 1.
    - Bên phát sẽ ngưng truyền nếu  $UWE - LWE = K$
    - Bên thu cũng thực hiện tương tự, các frame trong bộ đệm thu được quan sát bởi cửa sổ nhận gọi là Receive Window
  - Nguyên tắc chọn K đảm bảo rằng S sẽ nhận được tất cả các I-frame truyền tới. Các thông số ảnh hưởng tới việc chọn K là:
    - Kích thước frame
    - Dung lượng bộ đệm
    - Thời gian trễ do lan truyền sóng.
    - Tốc độ phát

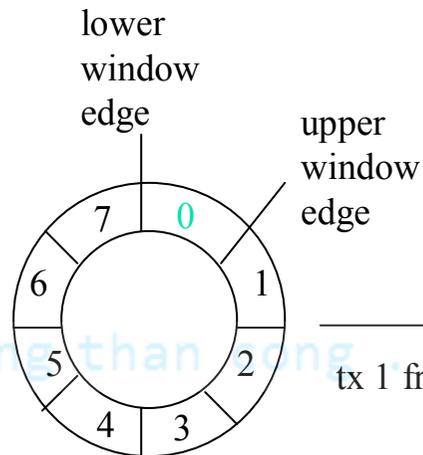


# KIỂM SOÁT LƯỒNG

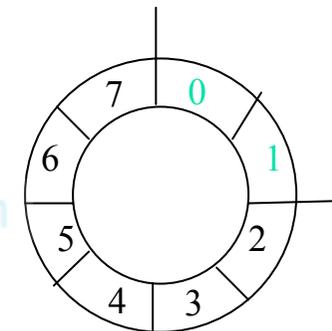
Window size = 7



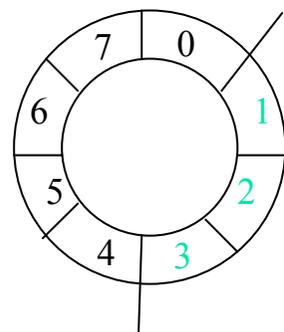
tx 1 frame



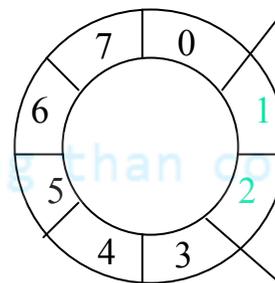
tx 1 frame



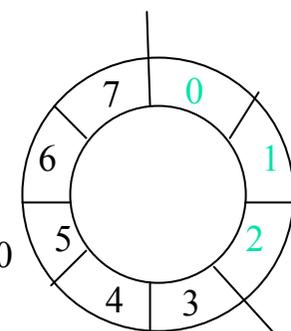
tx 1 frame



tx 1 frame



receive ACK0



max window size  
can't transmit

max window size  
can't transmit



# KIỂM SOÁT LUỒNG

## ■ Nguyên tắc đánh số :

- Để nhận dạng các frame với nhau mỗi frame được gán một số thứ tự (Sequence number), theo nguyên tắc thì có thể gán đến vô cùng.
- Tuy nhiên để hạn chế, kết hợp với cửa sổ trượt, số tuần tự được giới hạn theo nguyên tắc đánh xoay vòng.
- Số lượng các số nhận dạng được hạn chế và quyết định bởi kích thước cửa sổ và nghi thức.



# KIỂM SOÁT LUỒNG

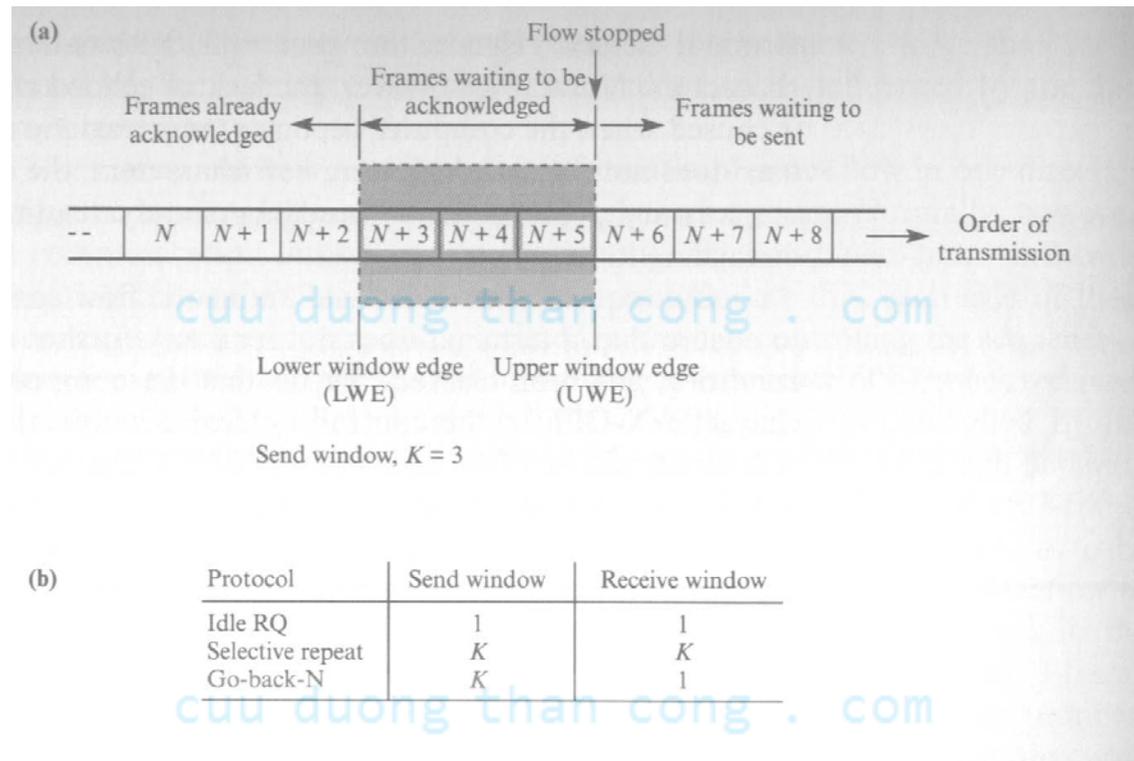
- Với Idle RQ số frame cần đánh số là 2
- Với Go-Back-N, số frame cần đánh số là  $K+1$
- Với Selective Repeat, số frame cần đánh số là  $2K+1$

cuu duong than cong . com

cuu duong than cong . com



# KIỂM SOÁT LƯỠNG

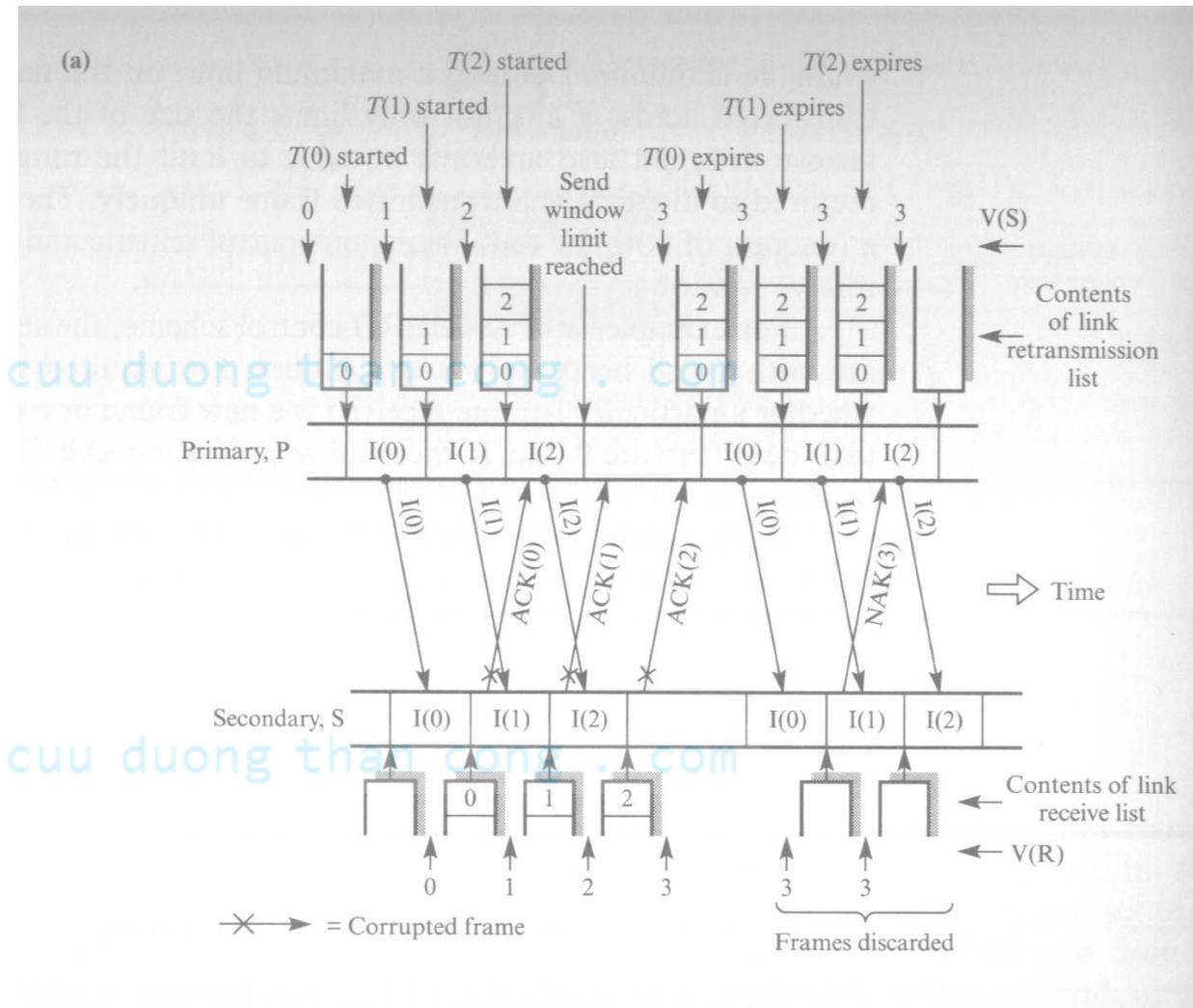




# KIỂM SOÁT LƯỒNG

**Ví dụ:**

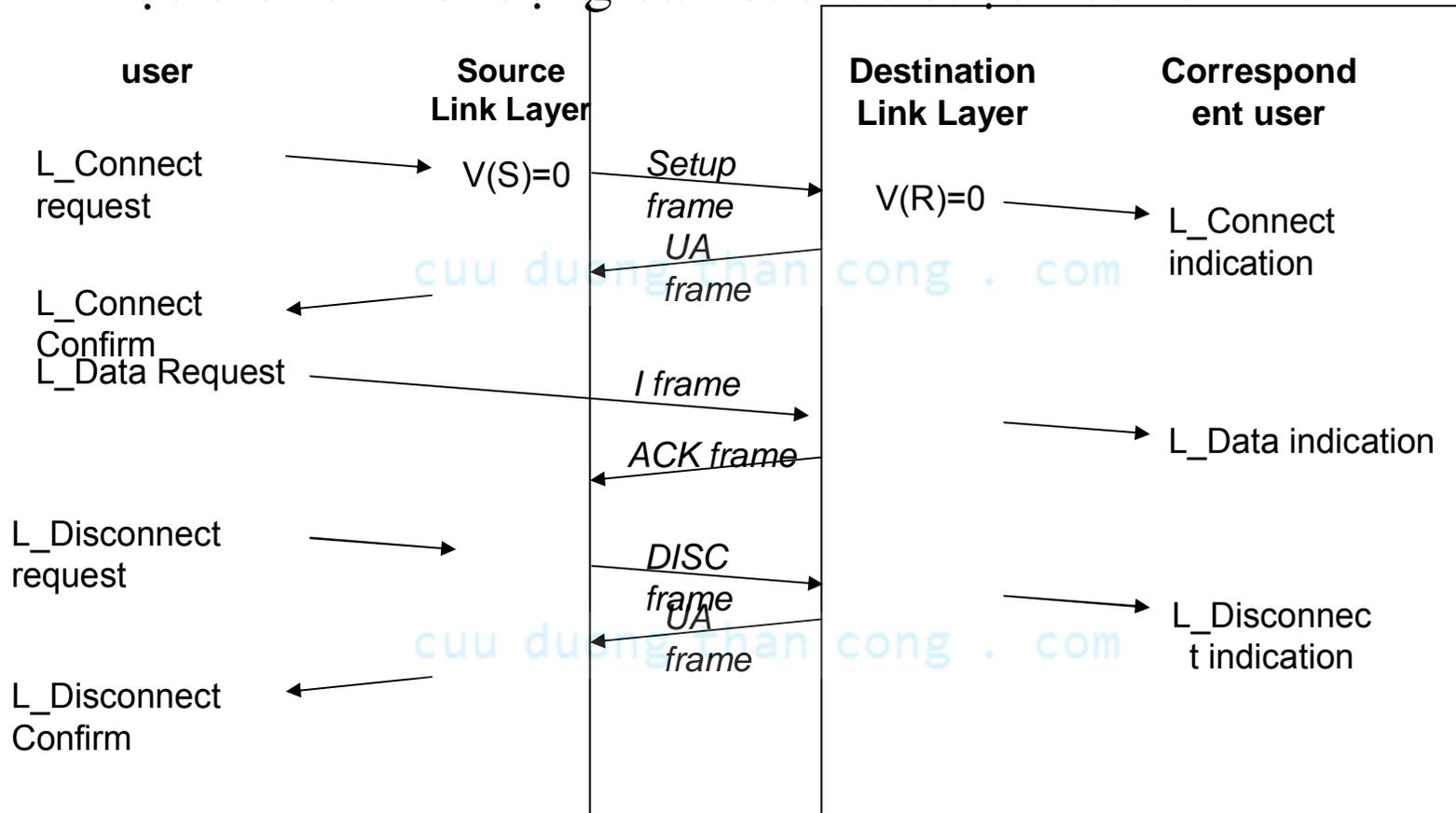
$k=2$





# QUẢN LÝ KẾT NỐI ( CONNECTION MANAGEMENT)

- Mục đích : Khởi động và kết thúc cuộc kết nối





# Nghi thức Lớp liên kết dữ liệu ( DATA LINK PROTOCOL)

- Chức năng lớp liên kết dữ liệu:
  - Lớp điều khiển liên kết dữ liệu (*data link control layer*) liên quan đến việc truyền dữ liệu qua một tuyến dữ liệu nối tiếp
  - Gồm các nghi thức truyền trong cấu hình điểm – điểm, hoặc đa điểm với các thành phần quan trọng như định dạng khung, kiểm soát lỗi, kiểm soát luồng và quản lý kết nối.
  - Tùy thuộc vào ứng dụng mà cung cấp 2 loại hình dịch vụ chính **connectionless**(kết nối không định hướng ) và **connection-oriented** (kết nối có định hướng) .



# Nghi thức Lớp liên kết dữ liệu

- Các kiểu liên kết :

- Điểm – điểm:

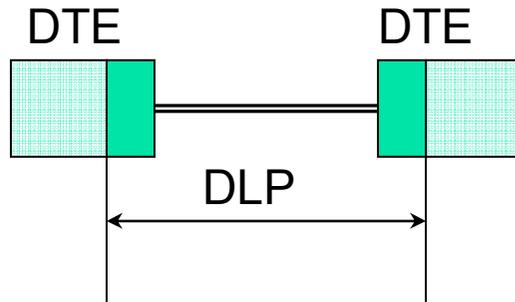
Sử dụng kết nối vật lý trực tiếp giữa 2 DTE (cáp xoắn, cáp đồng trục hay cáp quang) với khoảng cách gần. Hay sử dụng thông qua mạng PSTN, viba mặt đất hay vệ tinh với khoảng cách xa.

- Đa điểm:

Sử dụng một đường bus để liên kết tất cả các DTE. Giao thức liên kết dữ liệu phải đảm bảo việc truyền dữ liệu giữa các DTE hoạt động có sự điều khiển và việc truyền dữ liệu giữa các DTE không xảy ra đồng thời.

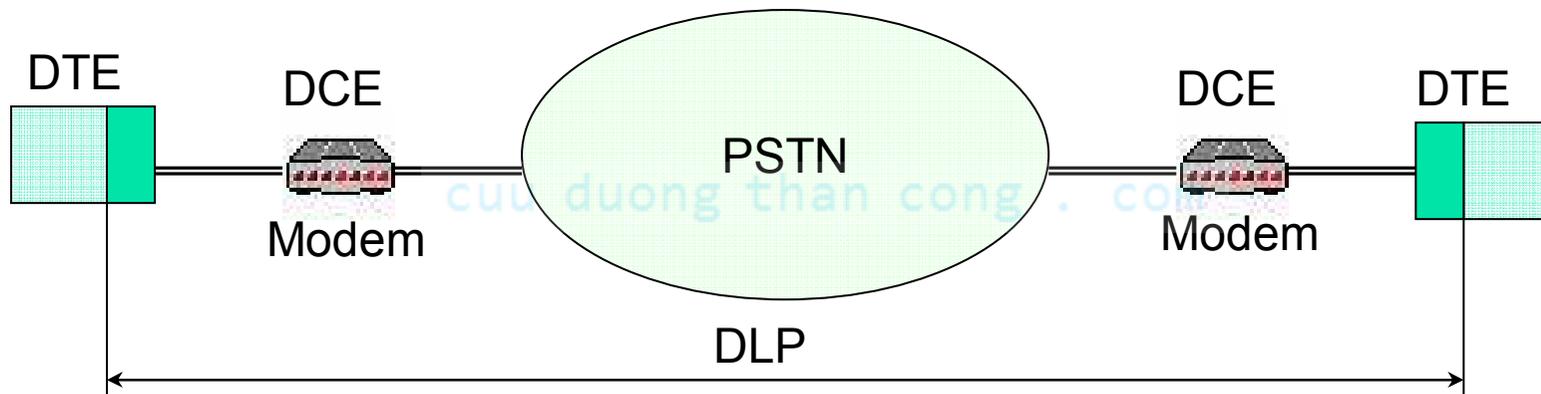


# Nghi thức Lớp liên kết dữ liệu



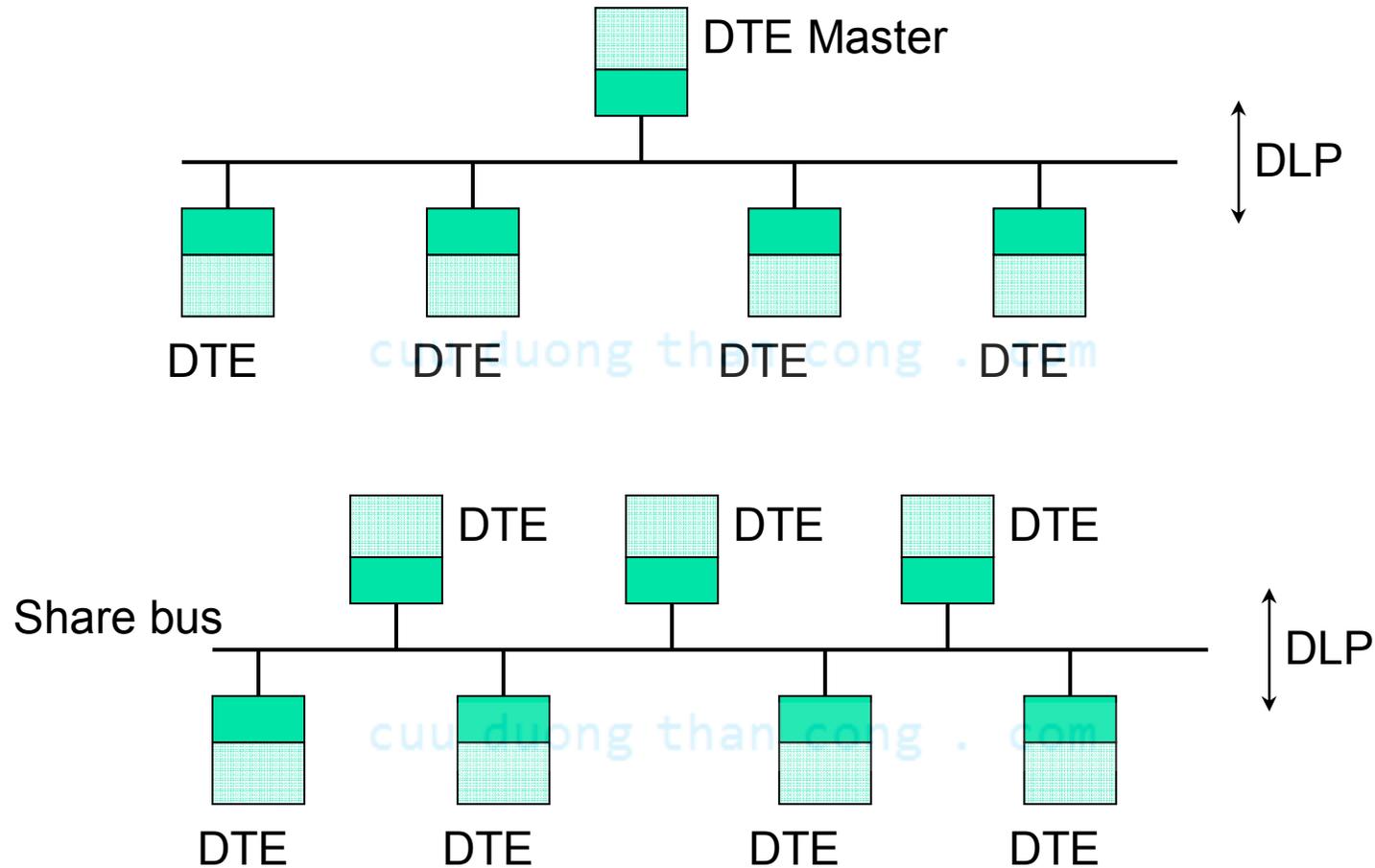
DTE: Data Terminal Equipment  
DLP: Data Link Protocol  
DCE: Data Circuit Terminating Equipment  
[Green Box] : Communication Subsystem

[cuuduongthancong.com](http://cuuduongthancong.com)



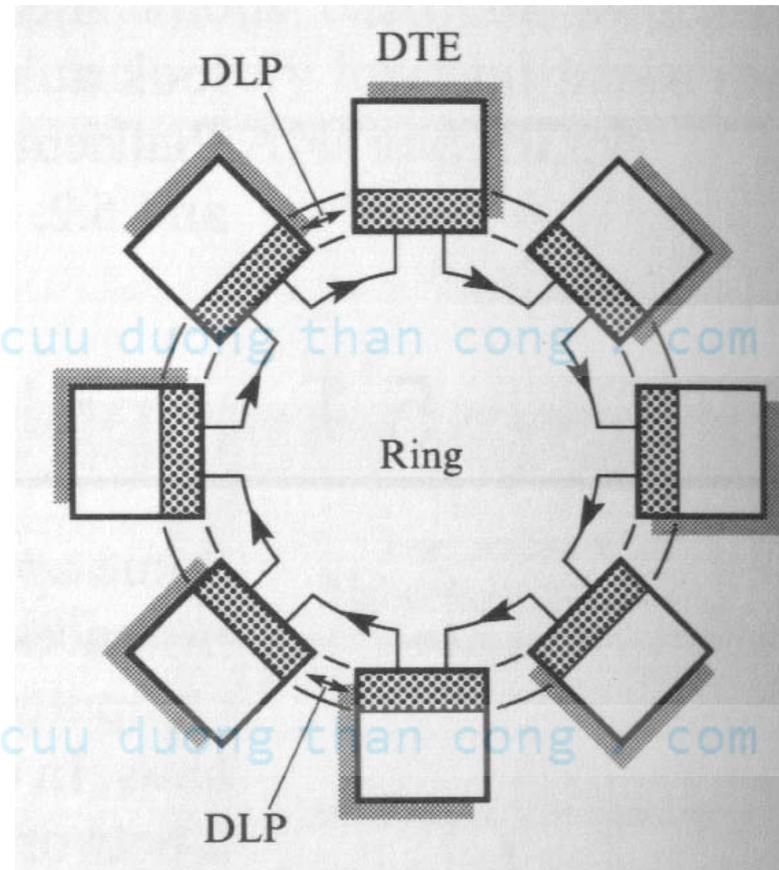


# Nghi thức Lớp liên kết dữ liệu





# Nghi thức Lớp liên kết dữ liệu





# Nghi thức Lớp liên kết dữ liệu

- Định hướng ký tự: Dùng các ký tự điều khiển để thực hiện các chức năng điều khiển việc truyền dữ liệu (*error control, start of frame, end of frame, data transparency*), sử dụng trong các cấu hình điểm điểm hoặc đa điểm.
  - Truyền bất đồng bộ
    - XMODEM
    - YMODEM
    - KERMIT
  - Truyền đồng bộ
    - BSC (Binary Synchronous Communication).
- Định hướng bit: Dùng các bit để thực hiện chức năng điều khiển việc truyền dữ liệu
  - HDLC

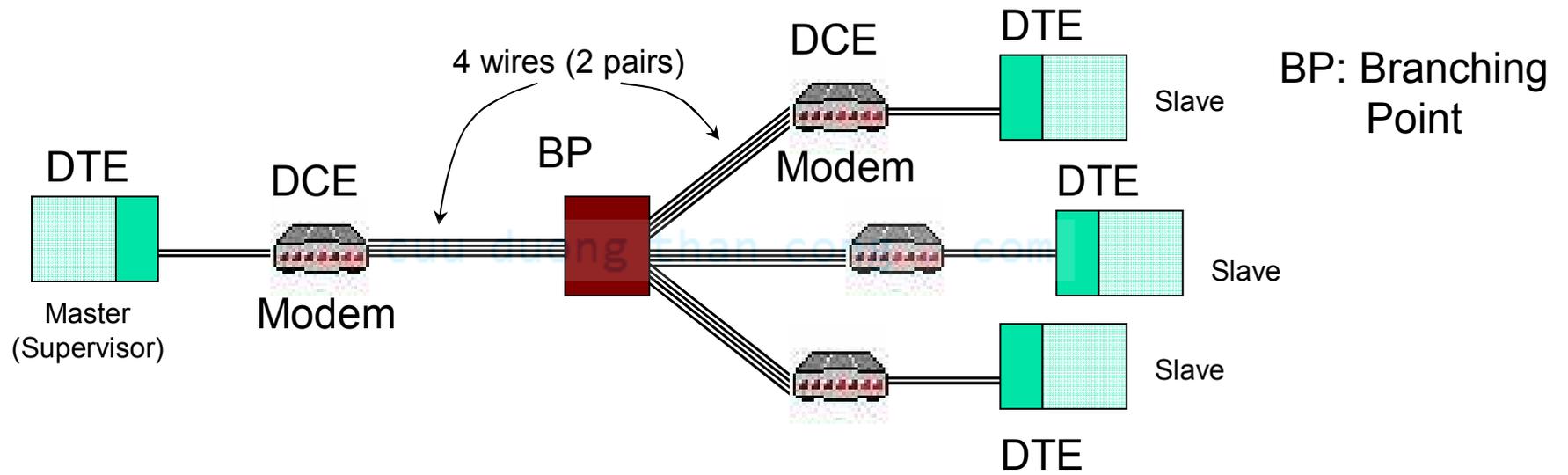


# BSC (Binary Synchronous Communication)

- Định hướng kí tự
- Nghi thức điều khiển lỗi là Idle RQ
- Thuộc nghi thức đồng bộ, định hướng kết nối (connection Oriented).
- Dữ liệu được truyền theo kiểu bán song công (Half-Duplex )
- Ứng dụng trong cấu hình điểm –điểm hoặc đa điểm.
- Trong cấu hình đa điểm, có một Master điều khiển việc truyền và nhận dữ liệu từ nhiều trạm Slave.



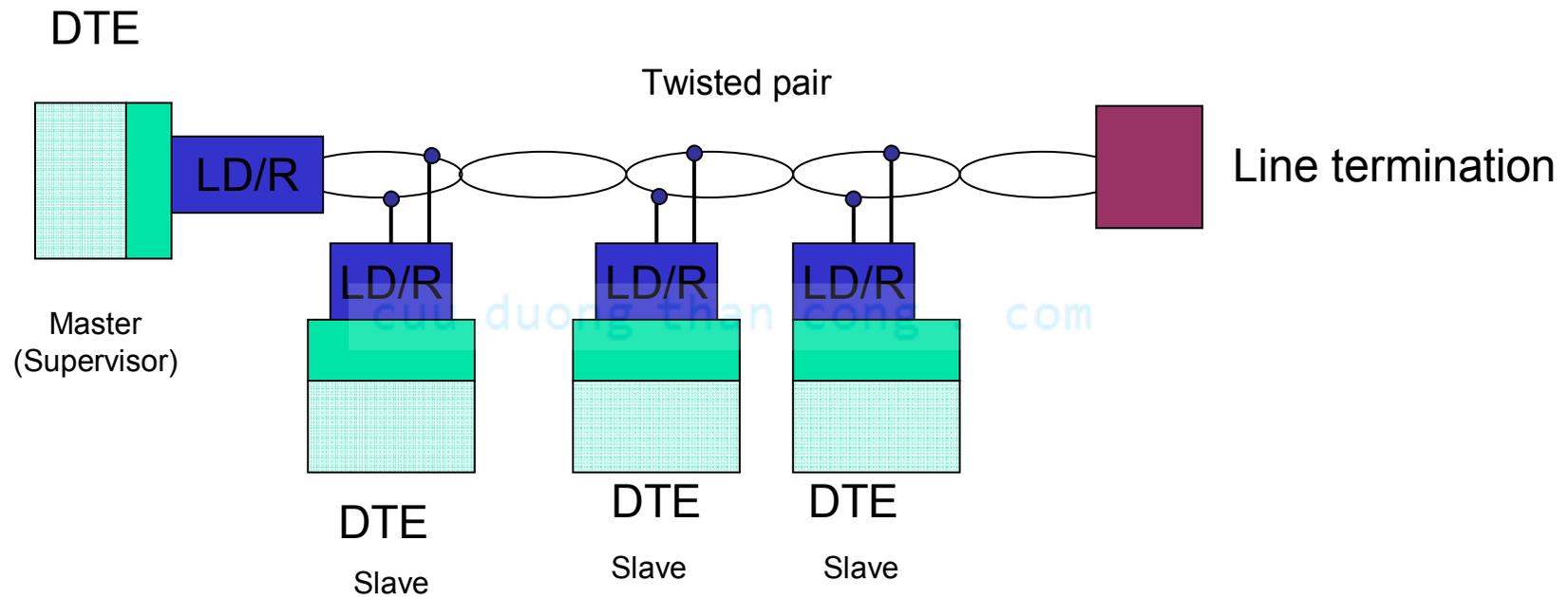
# BSC



cuu duong than cong . com



# BSC



LD/R: Line Driver/Receiver

cuu duong than cong . com



## ■ BSC

### ■ Cấu trúc khung (*Xem hình*) : có 2 loại

#### ■ Khung dữ liệu (Data)

##### ■ Đơn khối (Single Block Message)

##### ■ Đa khối (Multiblock Message): sử dụng [SOH,IBT] để phân biệt các khung

##### ■ Đa khung: khi 1 khung quá dài sẽ được cắt ra thành nhiều khung. Khung cuối cùng kết thúc bằng ETB (End of Transmission Block)

#### ■ Khung điều khiển (supervisory)

##### ■ Thiết lập kết nối và Điều khiển lỗi, luồng



# BSC

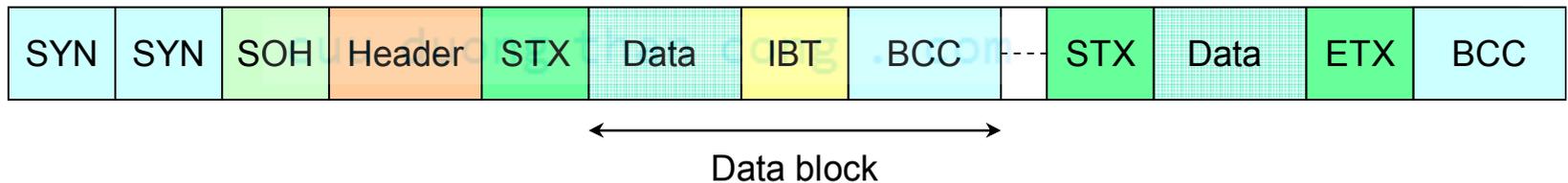
Cấu trúc một khung dữ liệu đơn giản



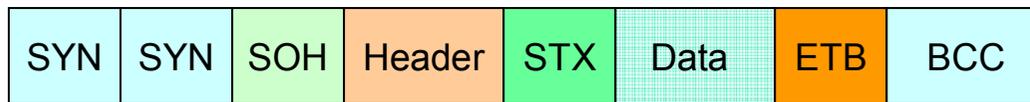
Cấu trúc một khung dữ liệu đơn giản với Header



Cấu trúc một khung đa khối



Cấu trúc một khung đa khung



Khung đầu tiên

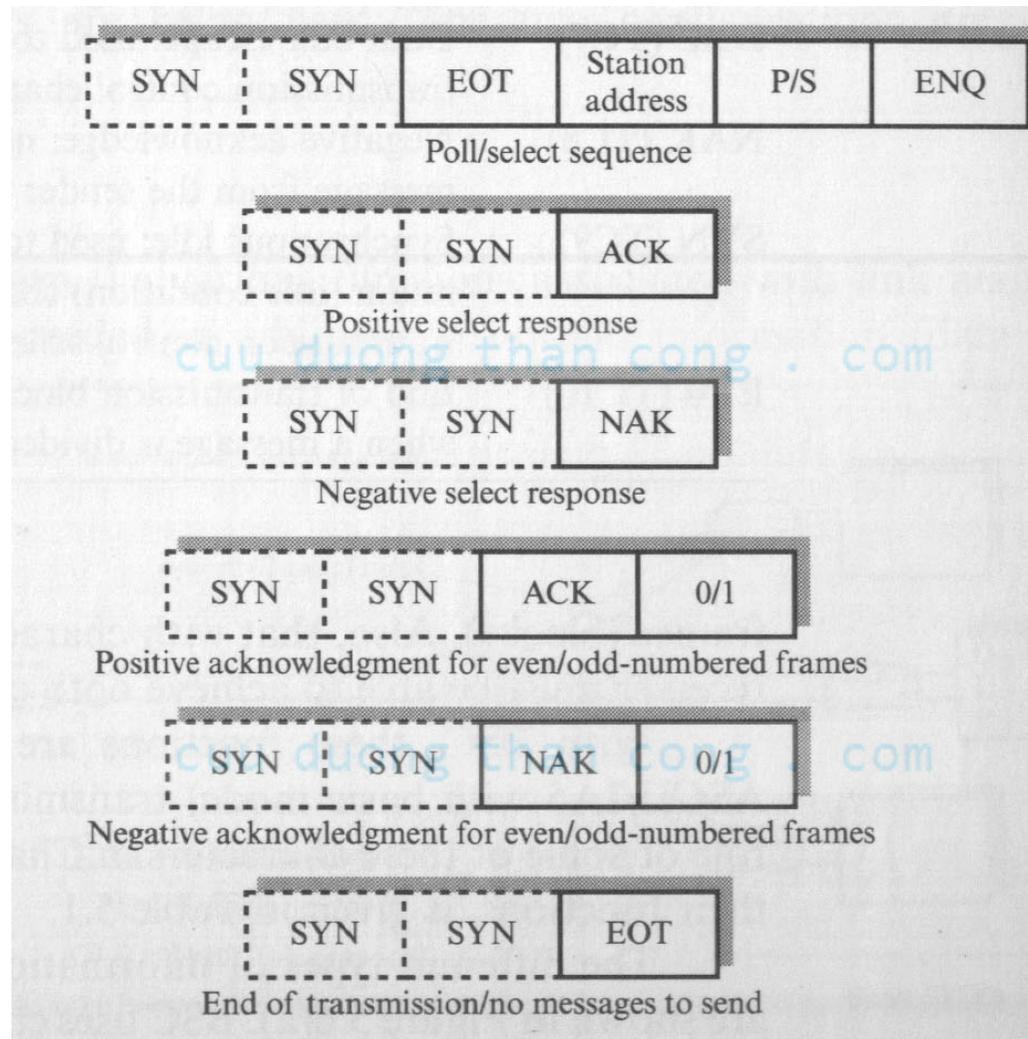


Khung cuối cùng

SOH – Start of Header	STX – Start of Text	ETX – End of Text
BCC – Block (sum) Check Character	ETB – End of Transmission Block	



# BSC

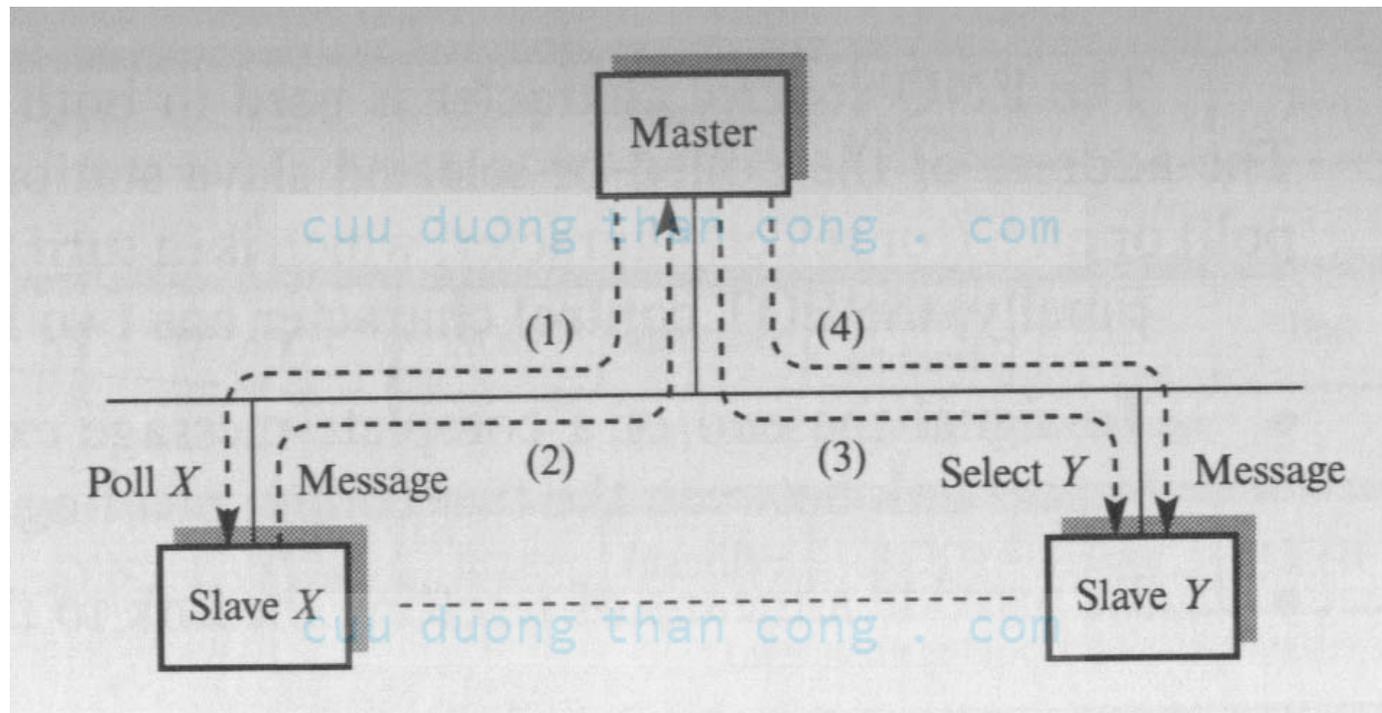




**Table 5.1** Transmission control characters used with BSC.

<i>Character</i>	<i>Function</i>
SOH (TC1)	Start of header: used to indicate the start of the header (if one is present) of an information frame (block)
STX (TC2)	Start of text: used both to terminate a header (if one is present) and to signal the start of a text string
ETX (TC3)	End of text: used to signal the end of a text string
EOT (TC4)	End of transmission: used to indicate the end of the transmission of one or more text (information) blocks and to terminate (clear) the connection
ENQ (TC5)	Enquiry: used as a request for a response from a remote station – the response may include the identity and/or status of the station
ACK (TC6)	Acknowledge: positive acknowledgment transmitted by a receiver in response to a message from the sender
DLE (TC7)	Data link escape: used to change the meaning of other selected transmission control characters
NAK (TC8)	Negative acknowledge: negative response transmitted by a receiver to a message from the sender
SYN (TC9)	Synchronous idle: used to provide the means for a receiver to achieve or retain (idle condition) character synchronization with a synchronous transmission control scheme
ETB (TC10)	End of transmission block: used to indicate the end of a block of data when a message is divided into a number of such blocks (frames)

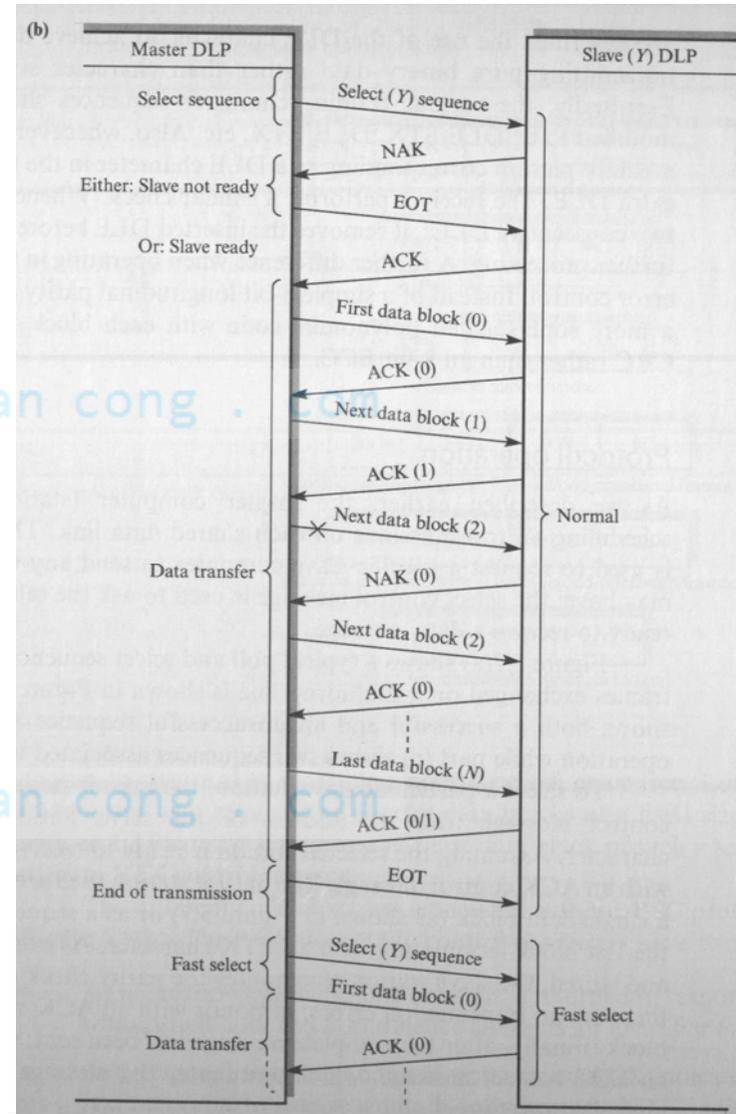
- Hoạt động của nghi thức





## **Ví dụ: Select**

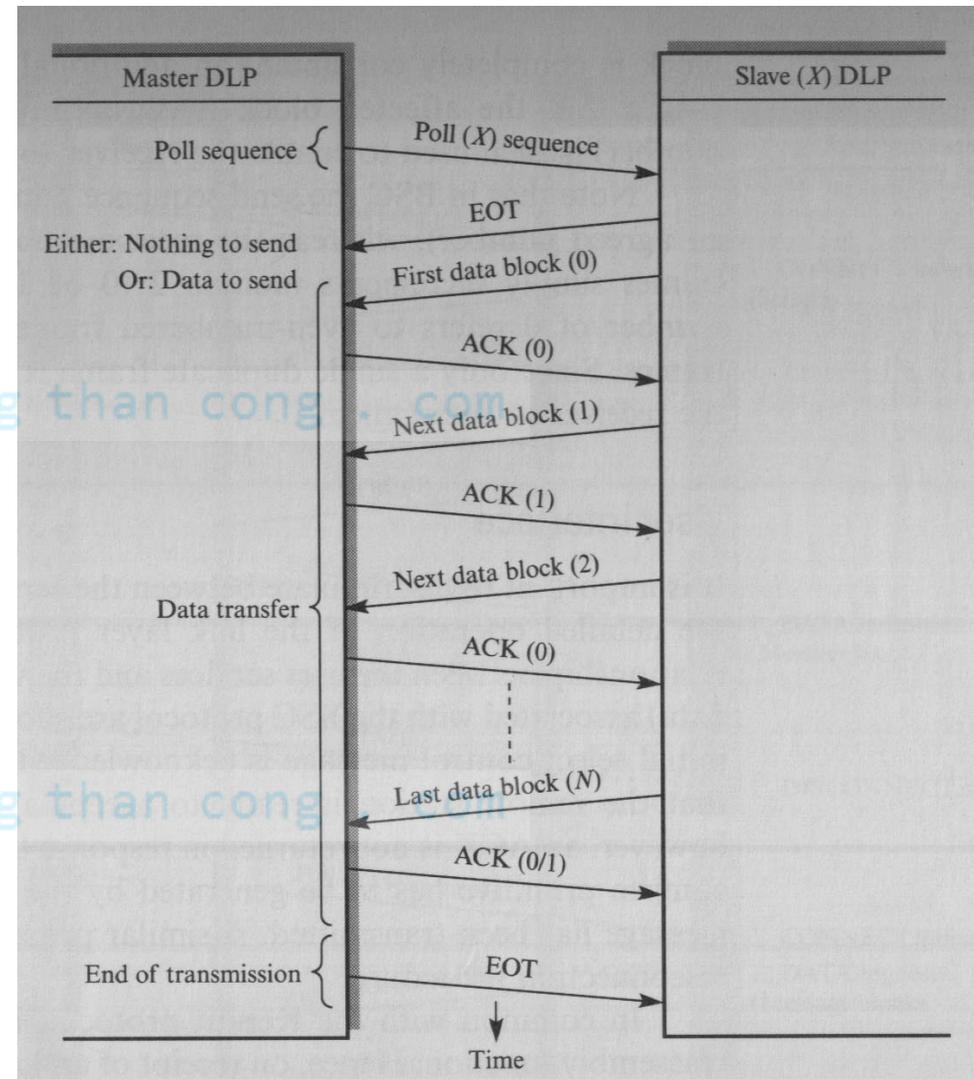
- NAK : Y không sẵn sàng nhận.
- ACK : Y sẵn sàng nhận.
- ACK(0) : Xác nhận khung dữ liệu chuẩn nhận tốt.
- ACK(1) : Xác nhận khung dữ liệu lỗi nhận tốt.
- NAK(0) : Xác nhận khung dữ liệu chuẩn bị lỗi
- NAK(1) : Xác nhận khung dữ liệu lỗi bị lỗi.
- EOT : Kết thúc truyền dữ liệu





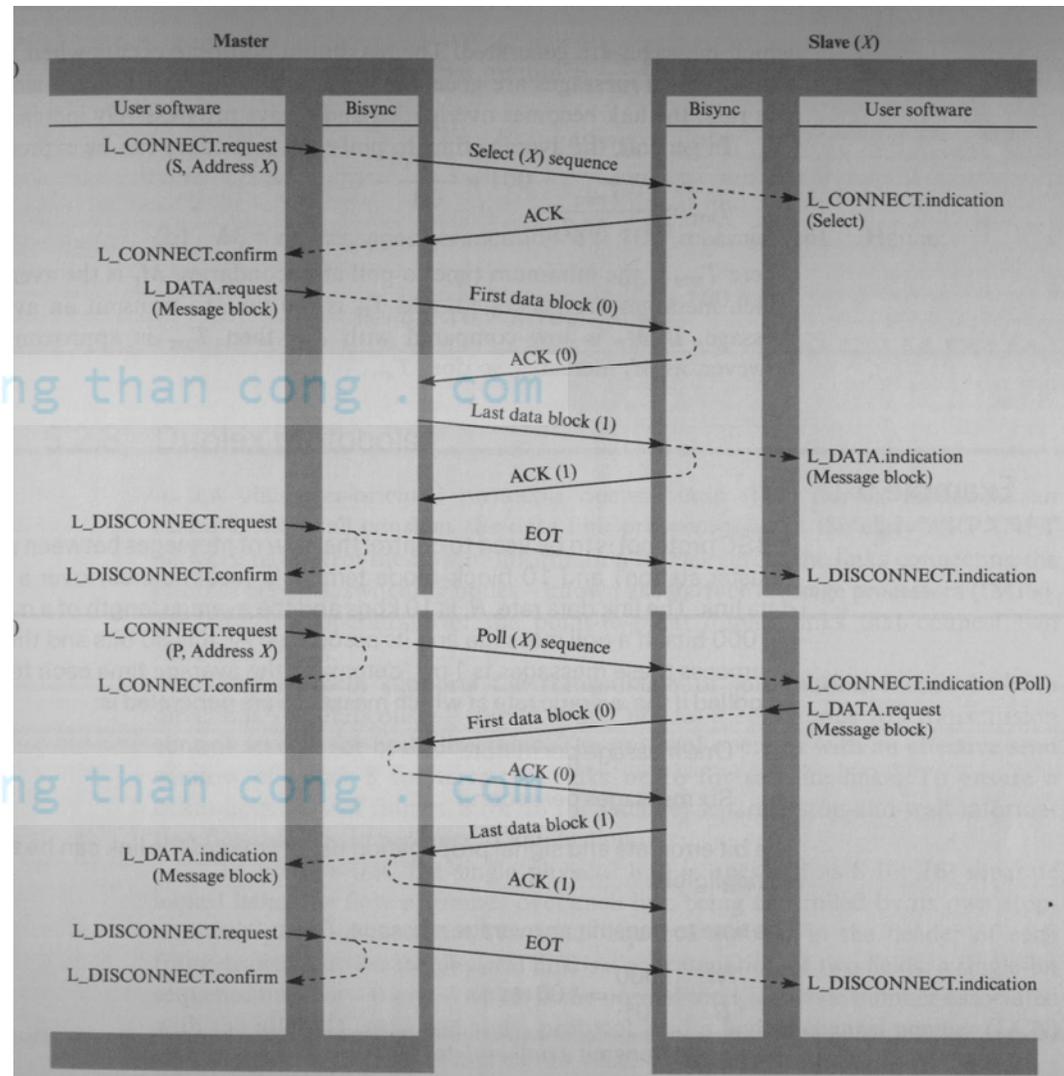
## **Ví dụ: Poll**

- X có dữ liệu thì gửi, nếu không thì trả về EOT
- ACK(0) : Xác nhận khung dữ liệu chấp nhận tốt.
- ACK(1) : Xác nhận khung dữ liệu lẻ nhận tốt.
- NAK(0) : Xác nhận khung dữ liệu chấp bị lỗi
- NAK(1) : Xác nhận khung dữ liệu lẻ bị lỗi.
- EOT : Kết thúc truyền dữ liệu





## Ví dụ: Select / Poll



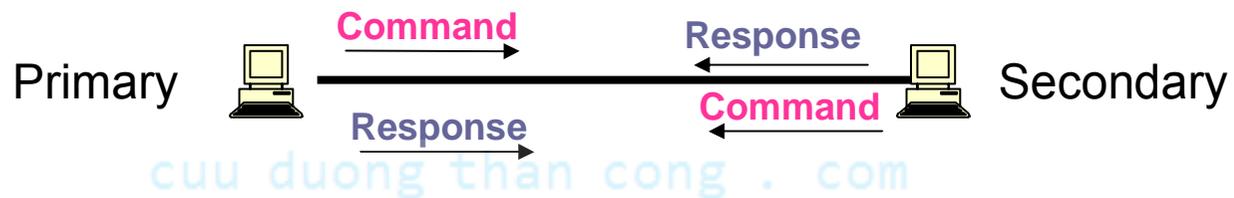
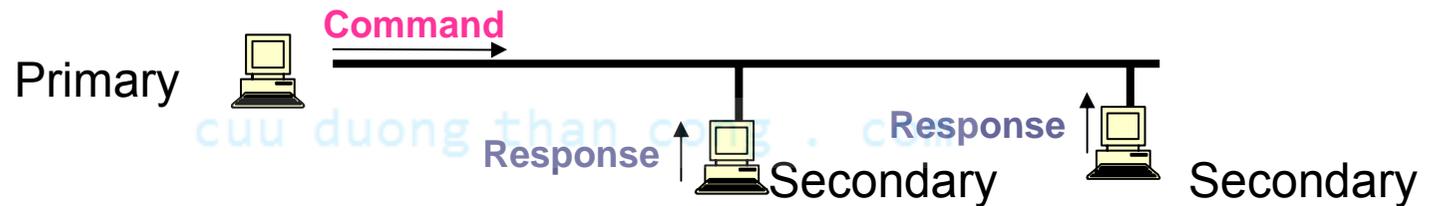
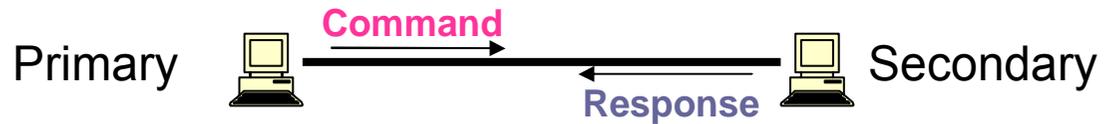


# HDLC (High Level Data Link Control)

- **HDLC(High Level Data Link Control ):** 1979, ISO
  - LAPB, LAPD...: 1981, ITU-T
  - Frame Relay, PPP : ITU-T,ANSI
- **Các loại trạm trong HDLC**
  - Trạm sơ cấp
    - Điều khiển hoạt động của đường kết nối
    - Các khung được phát đi được gọi là các lệnh (Command)
  - Trạm thứ cấp
    - Chịu sự điều khiển của trạm sơ cấp
    - Các khung được phát đi được gọi là đáp ứng (Responses)
  - Trạm kết hợp
    - Có thể phát lệnh hoặc đáp ứng



# HDLC





- **Các chế độ truyền HDLC**
  - Chế độ đáp ứng thường  
(NRM – Normal Response Mode)
    - Có cấu hình không cân bằng
    - Trạm sơ cấp khởi tạo quá trình truyền tới trạm thứ cấp
    - Trạm thứ cấp có thể chỉ phát dữ liệu để đáp ứng lệnh từ trạm sơ cấp
    - Được sử dụng trên đường truyền có nhiều điểm rẽ
    - Máy chủ là trạm sơ cấp
    - Các đầu cuối là trạm thứ cấp



- Chế độ đáp ứng bất đồng bộ  
(ARM – Asynchronous Response Mode)
  - Có cấu hình không cân bằng
  - Trạm thứ cấp có thể khởi tạo quá trình truyền mà không cần sự cho phép của trạm sơ cấp
  - Trạm sơ cấp chịu trách nhiệm có đường truyền
  - Ít được sử dụng

cuu duong than cong . com

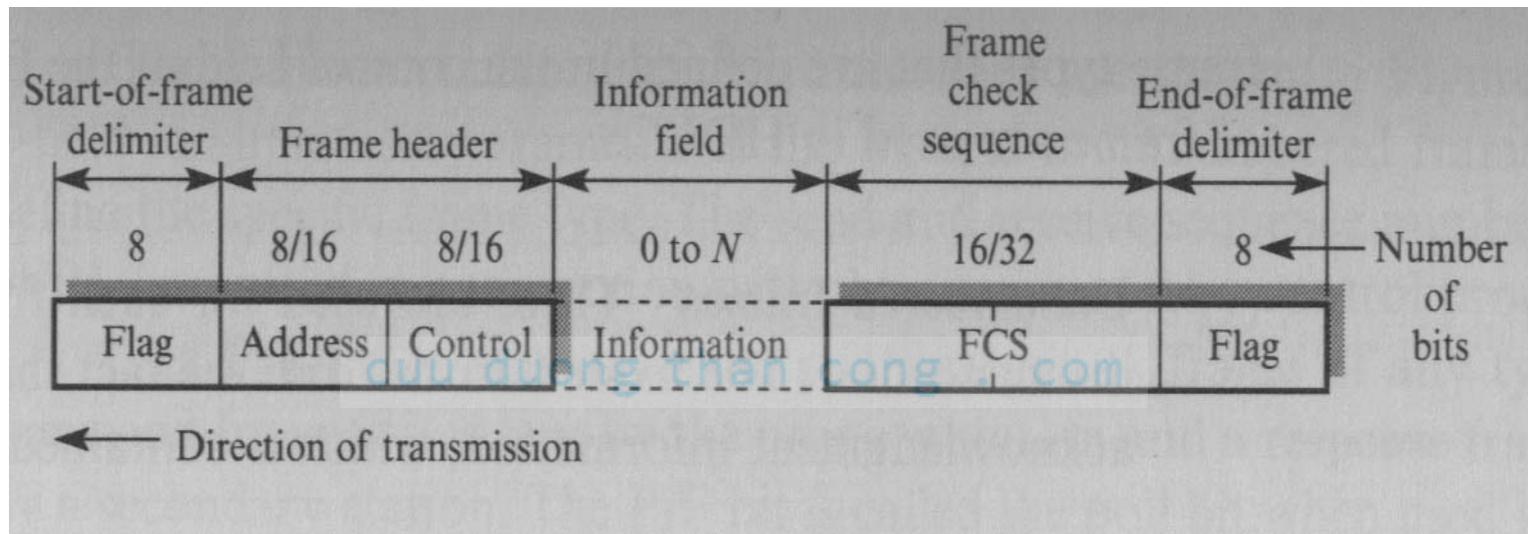


- Chế độ cân bằng bất đồng bộ  
(ABM – Asynchronous Balanced Mode)
  - Có cấu hình cân bằng
  - Cả hai trạm có thể khởi tạo quá trình truyền mà không cần nhận lệnh cho phép
  - Được sử dụng rộng rãi nhất
  - Không cần hỏi vòng

cuu duong than cong . com



- **Cấu trúc khung**
  - Truyền đồng bộ
  - Tất cả truyền dẫn dạng khung
  - Dạng khung đơn cho trao đổi dữ liệu và điều khiển





- Trường cờ
  - Xác định vị trí bắt đầu và kết thúc của khung
  - 01111110
  - Có thể đóng 1 khung và mở khung tiếp theo
  - Máy thu tìm chuỗi cờ để đồng bộ
  - Kỹ thuật chèn bit được sử dụng để tránh sự nhầm lẫn với dữ liệu chứa chuỗi 01111110
    - 0 được chèn sau mỗi chuỗi 5 bit 1
    - Nếu máy thu phát hiện 5 bit 1 thì kiểm tra bit tiếp theo
      - Nếu bit tiếp theo là 0 thì nó được xóa bỏ
      - Nếu bit tiếp theo là 1 và bit thứ 7 là 0 thì đó là cờ
      - Nếu bit tiếp theo là 1 và bit thứ 7 là 1, thì nó tiếp tục đếm số bit 1
        - Nếu số bit 1 < 15 : máy phát chỉ sự kết thúc
        - Nếu số bit 1 ≥ 15 : Máy phát chỉ kênh rỗi.



# HDLC

Original Pattern:

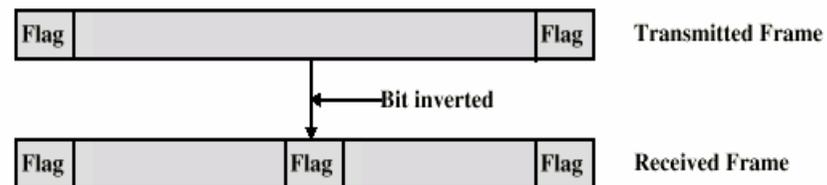
111111111111011111101111110

After bit-stuffing

1111101111101101111101011111010

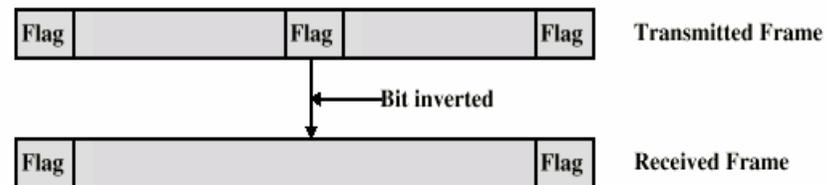
(a) Example

cuu duong than cong . com



(b) An inverted bit splits a frame in two

cuu duong than cong . com



(c) An inverted bit merges two frames



# HDLC

## Trường địa chỉ

- Phụ thuộc vào chế độ hoạt động.
- Trong mode NRM, cấu hình đa điểm ( multidrop line), mỗi trạm thứ cấp có một địa chỉ. Khi trạm sơ cấp liên lạc với trạm thứ cấp nào thì trường địa chỉ chứa địa chỉ của trạm thứ cấp đó.
- Không sử dụng trong cấu hình ABM, (point – to - point).Thay vào đó, được sử dụng để chỉ hướng lệnh và đáp ứng.
- Thông thường dài 8 bit
- Có thể được mở rộng ra bội số của 7 bit
  - LSB của mỗi octet chỉ rằng đây là octet cuối (1) hay không (0)
- Tất cả là bit 1 chỉ khung quảng bá



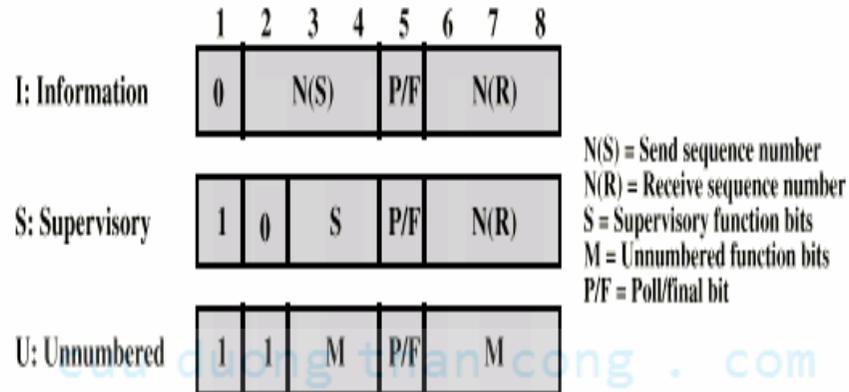
## (b) Extended Address Field



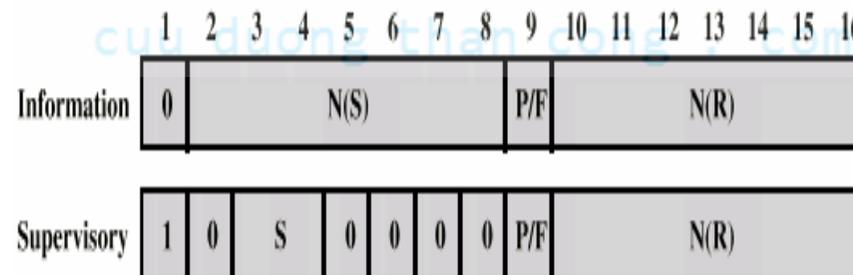
- Trường điều khiển
  - Có cấu trúc khác nhau ứng với những loại khung khác nhau, có 3 loại khung trong HDLC
    - **Thông tin (I-frame)** : Mang dữ liệu cần gửi
    - **Giám sát (S-frame)** : điều khiển lỗi và luồng, chứa số thứ tự khung gửi và nhận
    - **Không đánh số (U-frame)** : Thiết lập và kết thúc kết nối.
  - Độ dài có thể 1 hay 2 byte



# HDLC



(c) 8-bit control field format



(d) 16-bit control field format



- Trường điều khiển
  - 1 hay 2 bit đầu tiên của trường điều khiển chỉ ra loại khung
  - $N(S)$  : Số thứ tự frame gửi
  - $N(R)$  : Số thứ tự frame nhận
  - P/F : Có 2 chức năng Poll hoặc Final phụ thuộc hoàn cảnh sử dụng
    - Khung lệnh
      - Bit P
        - 1 để yêu cầu đáp ứng từ đối phương
    - Khung đáp ứng
      - Bit F
        - 1 chỉ rằng đây là trả lời cho lệnh



# HDLC

## HDLC - HIGH LEVEL DATA LINK CONTROL

- Trường điều khiển
  - S gồm 2 bit xác định loại khung giám sát. Có 4 loại khung giám sát

S	Lệnh
00	<p><b>RR</b> : Receive Ready -&gt; Dừng 4 cách</p> <ul style="list-style-type: none"><li>➢ ACK – RR: dừng như một xác nhận dương của 1 khung thông tin đã nhận khi bộ thu không có dữ liệu để truyền (tức không thể dùng piggyback).</li><li>➢ P - RR : <b>Poll</b> -&gt; yêu cầu trạm thứ cấp xem có dữ liệu để gửi không ?</li><li>➢ F - RR : đáp ứng cho <b>Poll</b> -&gt; Trạm thứ cấp trả lời cho sơ cấp là không có dữ liệu gửi (xác nhận âm). Nếu có thì sẽ đáp ứng bằng I-frame</li><li>➢ F –RR : <i>đáp ứng cho <b>Select</b> -&gt; Trạm thứ cấp trả lời cho sơ cấp là có khả năng nhận dữ liệu (xác nhận dương)</i></li></ul>
01	<p><b>REJ</b>: Reject -&gt; là xác nhận âm (NAK) được trả về trong hệ thống Go-back –n khi bộ thu không có dữ liệu gửi (tức không thể truyền theo piggyback)</p>
10	<p><b>RNR</b> : Receive not Ready Dừng 3 cách</p> <ul style="list-style-type: none"><li>➢ ACK – RNR : Yêu cầu trạm gửi ngưng không gửi thêm nữa cho đến khi 1 RR được phát</li><li>➢ P – RNR : <b>Select</b> -&gt; Khi trạm sơ cấp muốn truyền dữ liệu cho trạm thứ cấp nào đó.</li><li>➢ F – RNR : <i>đáp ứng cho <b>Select</b> -&gt; Khi một thứ cấp được chọn mà không thể nhận dữ liệu (xác nhận âm).</i></li></ul>
11	<p><b>SREJ</b>: Selective Reject -&gt; là xác nhận âm (NAK) được trả về trong hệ thống Selective Repeat khi bộ thu không có dữ liệu gửi (tức không thể truyền theo piggyback)</p>



- Trường điều khiển
  - M gồm 5 bit xác định các loại U-frame khác nhau

M	Lệnh	Đáp ứng
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
00 101	DISC	RD
10 000	SIM	RIM
00 100	UP	
11 001	RSET	
11 101	XID	XID
10 001		FRMR

*Unnumbered commands*

- Set Asynchronous Response Mode (SARM)
- Set Asynchronous Response Mode Extended (SARME)
- Set Normal Response Mode (SNRM)
- Set Normal Response Mode Extended (SNRME)
- Set Asynchronous Balanced Mode (SABM)
- Set Asynchronous Balanced Mode Extended (SABME)
- Reset (RSET)
- Set Initialization Mode (SIM)**
- Disconnect (DISC)
- Unnumbered Poll (UP)

*Unnumbered responses*

- Unnumbered Acknowledge (UA)
- Request Disconnect (RD)**
- Frame Reject (FRMR)
- Disconnect Mode (DM)
- Request Information Mode (RIM)**

Unnumbered Information (UI)  
 Exchange ID (XID)



# HDLC

- Trường thông tin
  - Chỉ có trong khung thông tin và vài khung không đánh số
  - Phải có một số nguyên lần octets
  - Chiều dài thay đổi được

cuu duong than cong . com

cuu duong than cong . com



# HDLC

---

- Trường kiểm tra
  - Phát hiện sai
  - 16 bit CRC
  - Tùy chọn 32 bit CRC

[cuu duong than cong . com](http://cuuduongthancong.com)

[cuu duong than cong . com](http://cuuduongthancong.com)



## ■ Hoạt động của HDLC

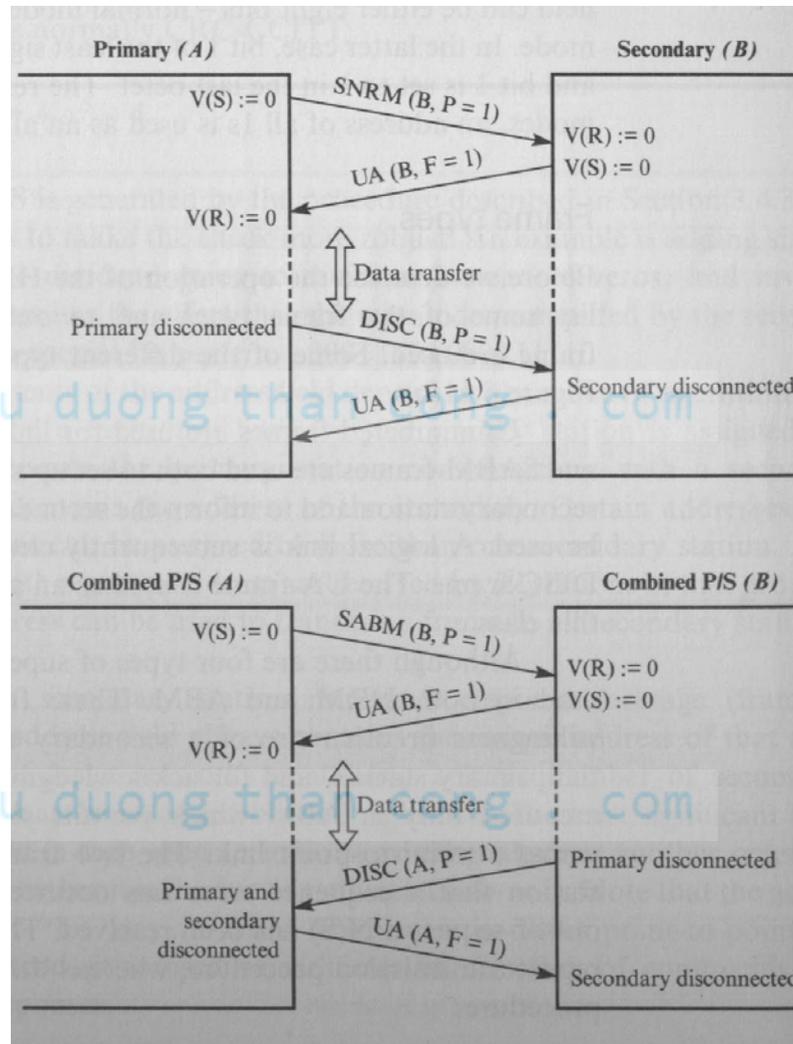
- Điều khiển kết nối : Thiết lập và giải phóng kết nối logical giữa 2 bên phát và nhận
- Trao đổi dữ liệu : Trao đổi dữ liệu giữa 2 bên. Trong quá trình này điều khiển lỗi và điều khiển luồng được ứng dụng.

**Ví dụ:** Về hoạt động của HDLC như sau :

- V(S) chỉ số tuần tự truyền kế tiếp N(S).
- V(R) chỉ số tuần tự của I-frame mà phía thu đang mong đợi nhận.
- Tại phía thu nếu  $N(S) = V(R)$  thì xem như thu đúng vì đúng thứ tự, ngược lại nếu  $N(S) \neq V(R)$  thì xem như thu sai vì không đúng thứ tự.

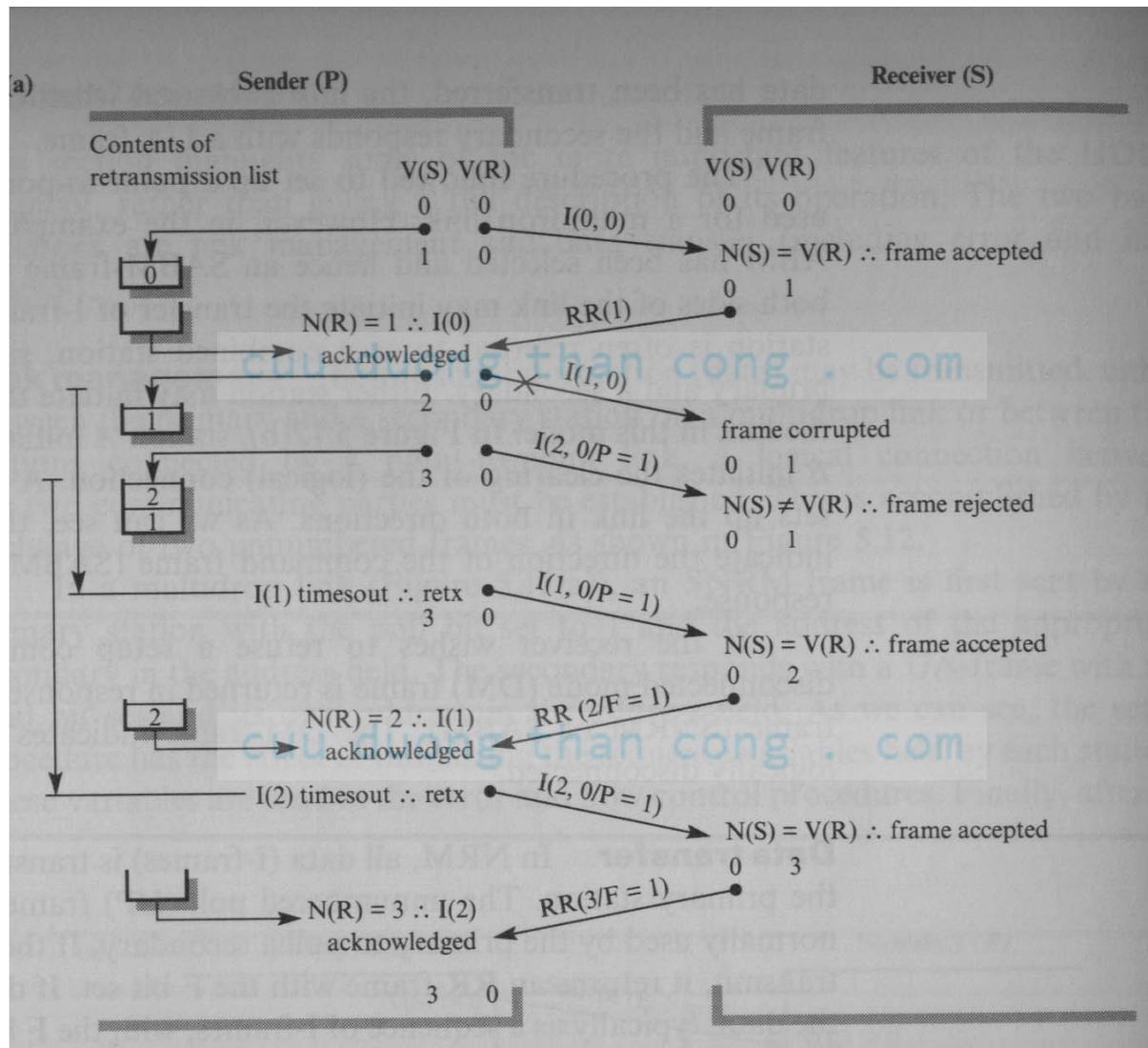


# HDLC



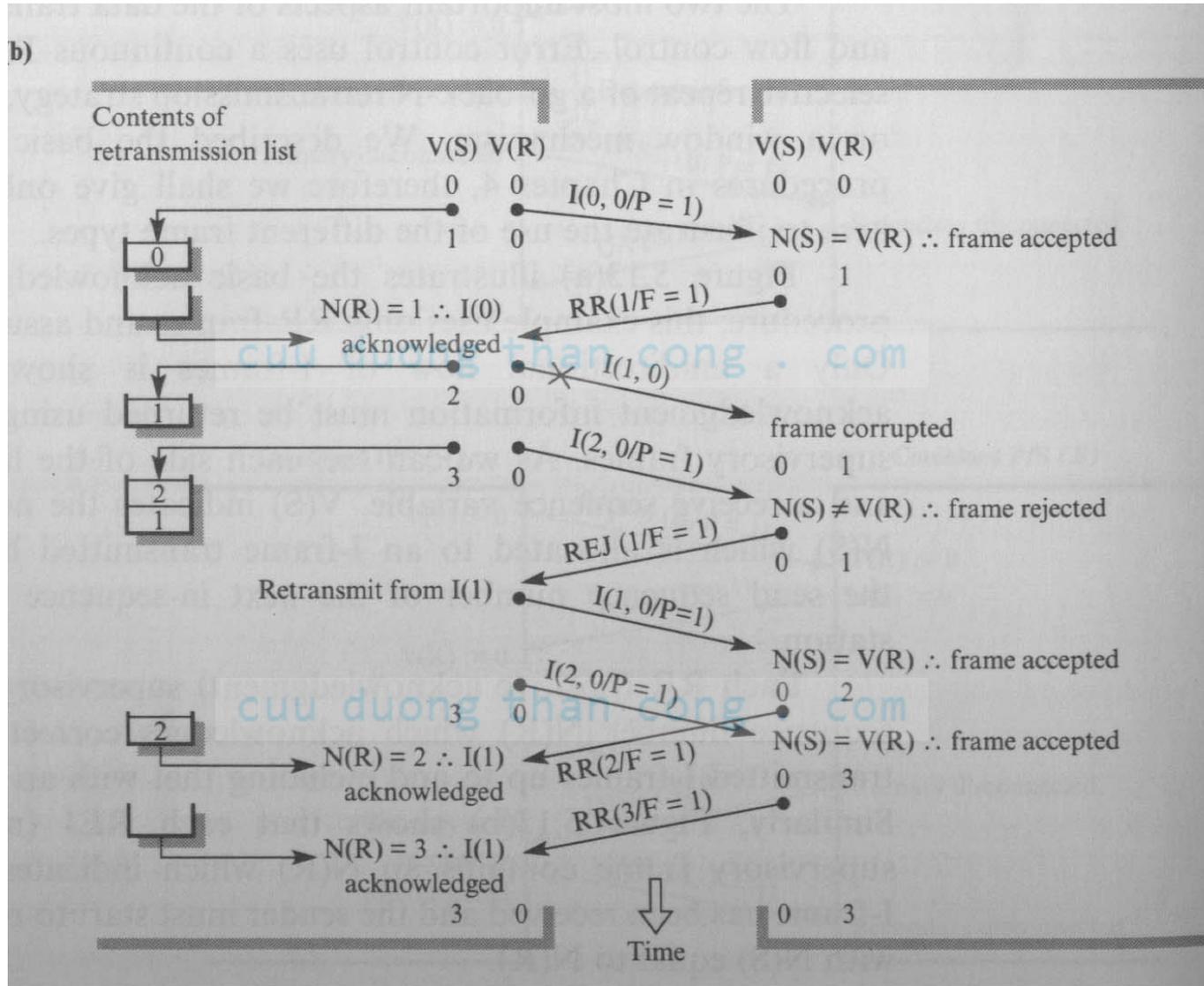


# HDLC



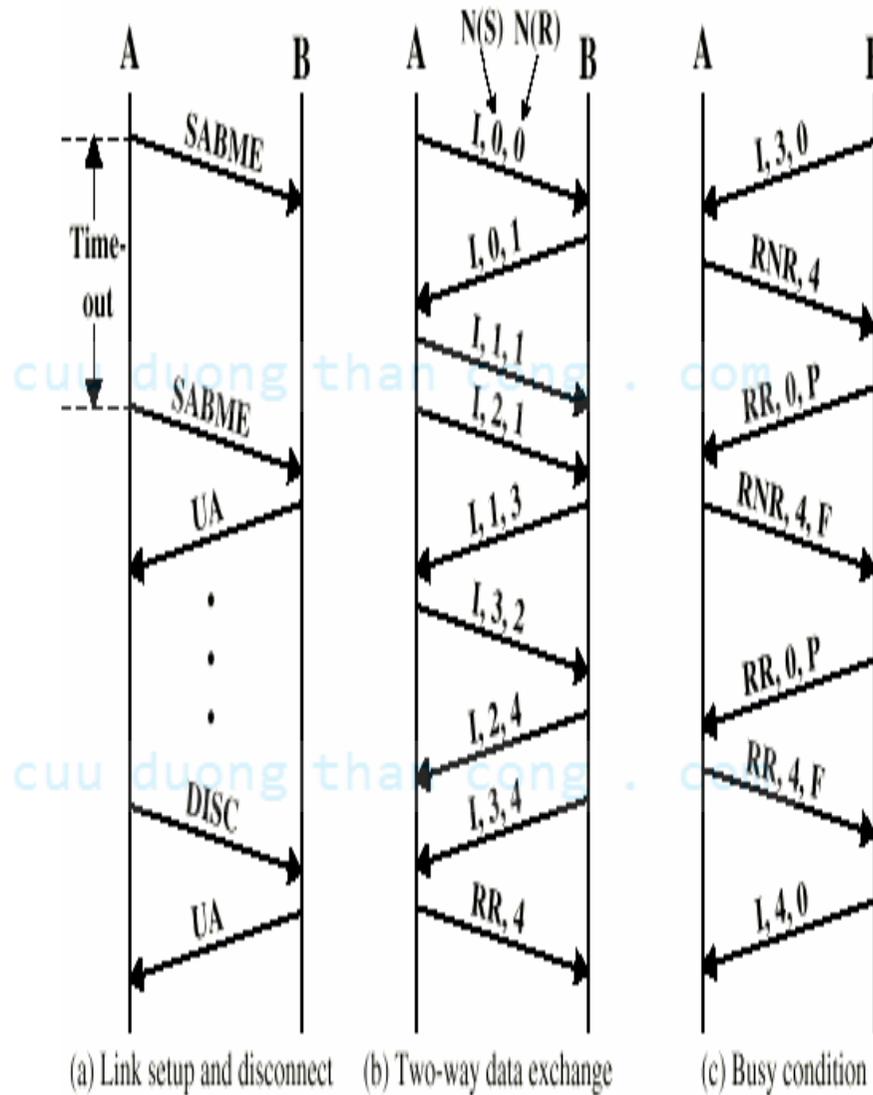


# HDLC



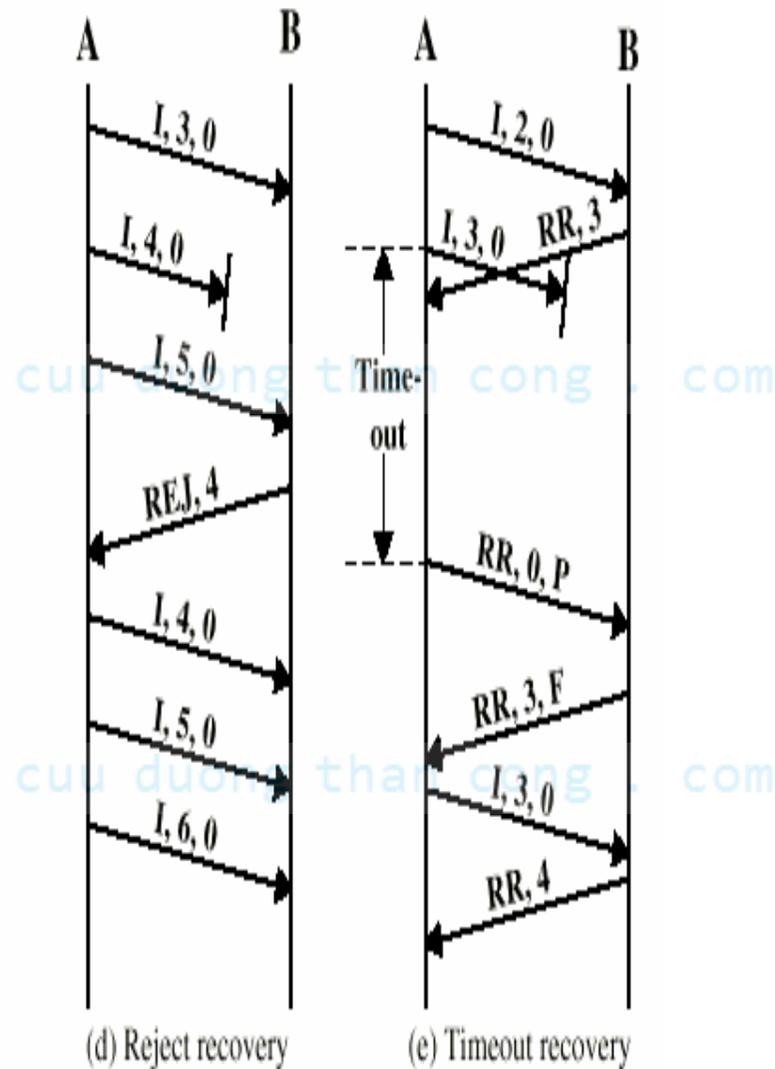


# HDLC



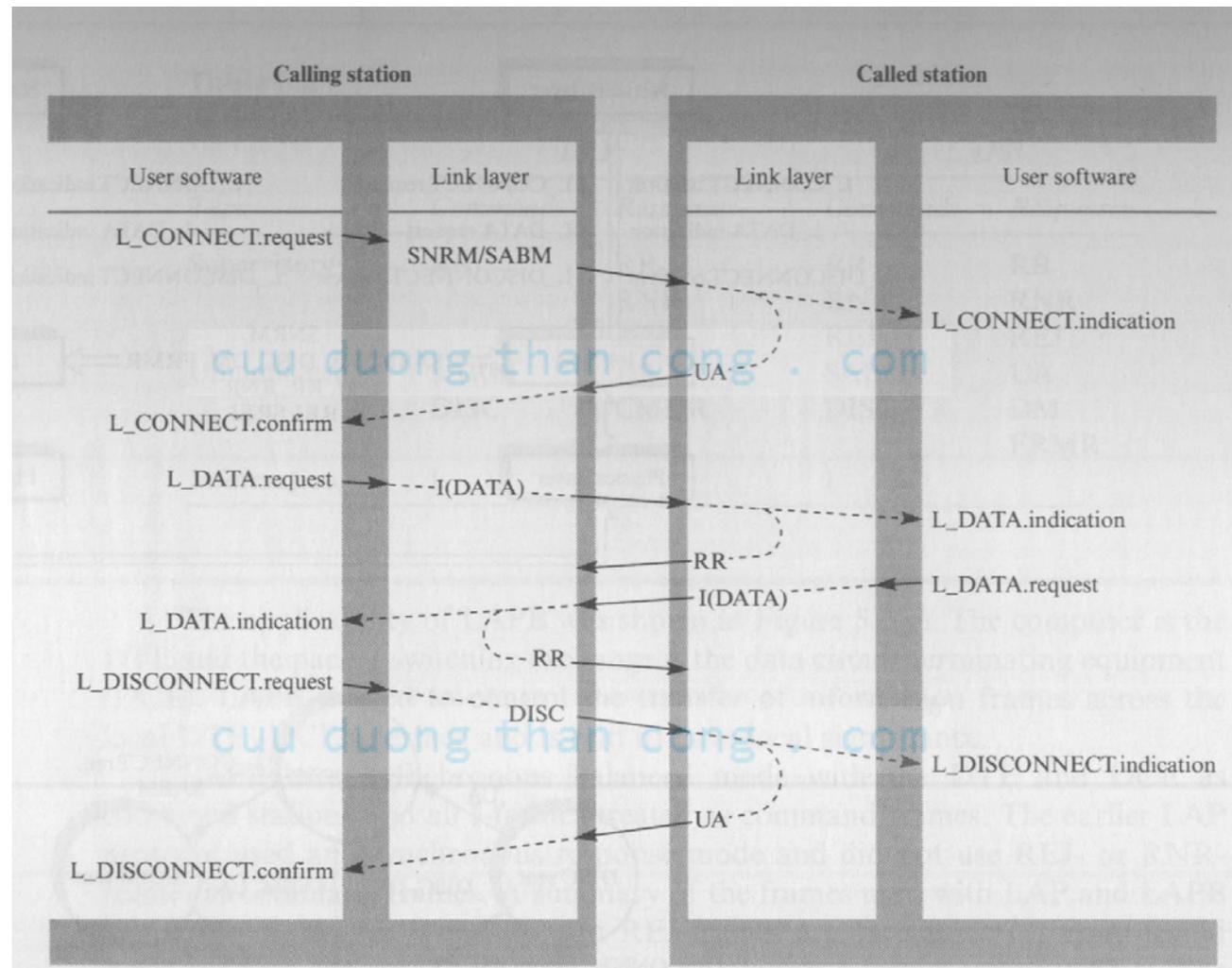


# HDLC



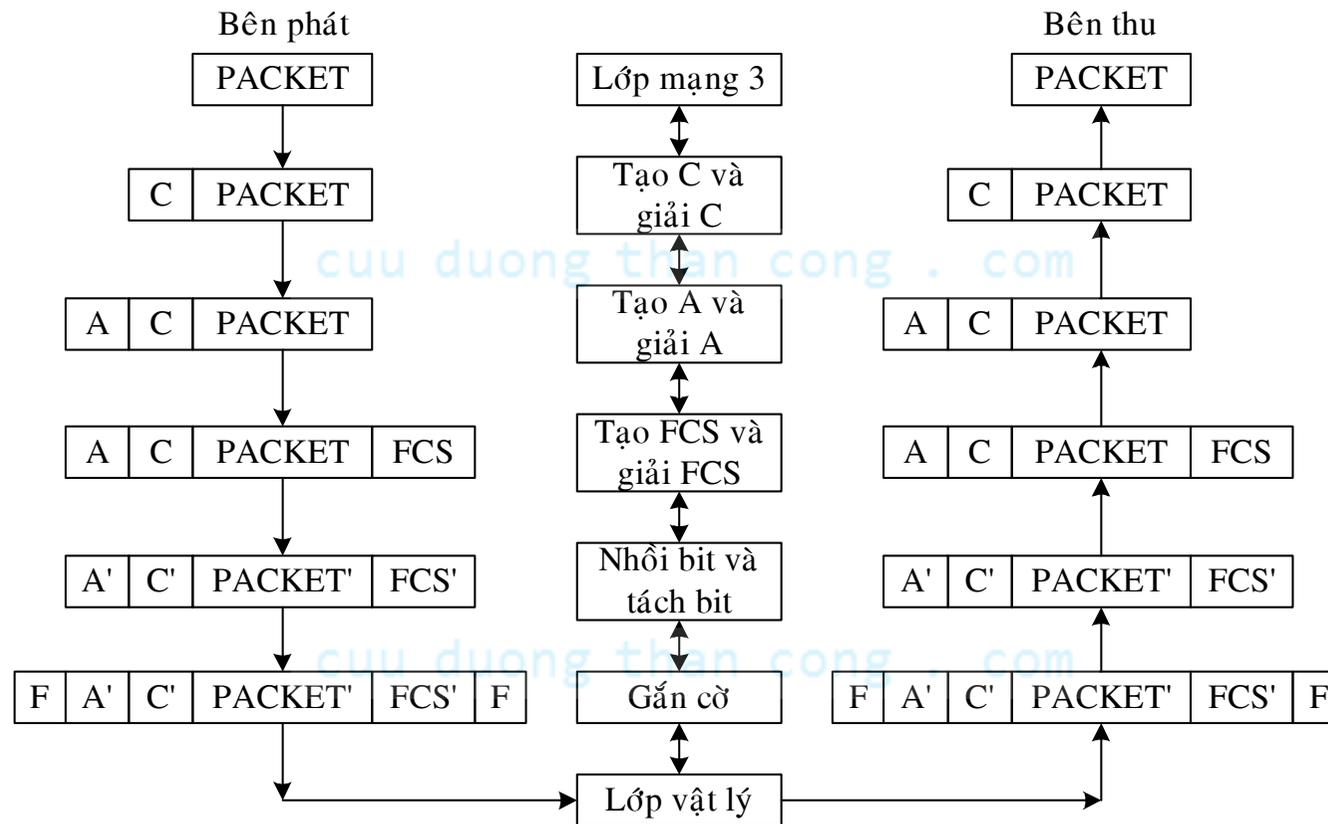


# HDLC





## Quá trình đóng khung trong HDLC



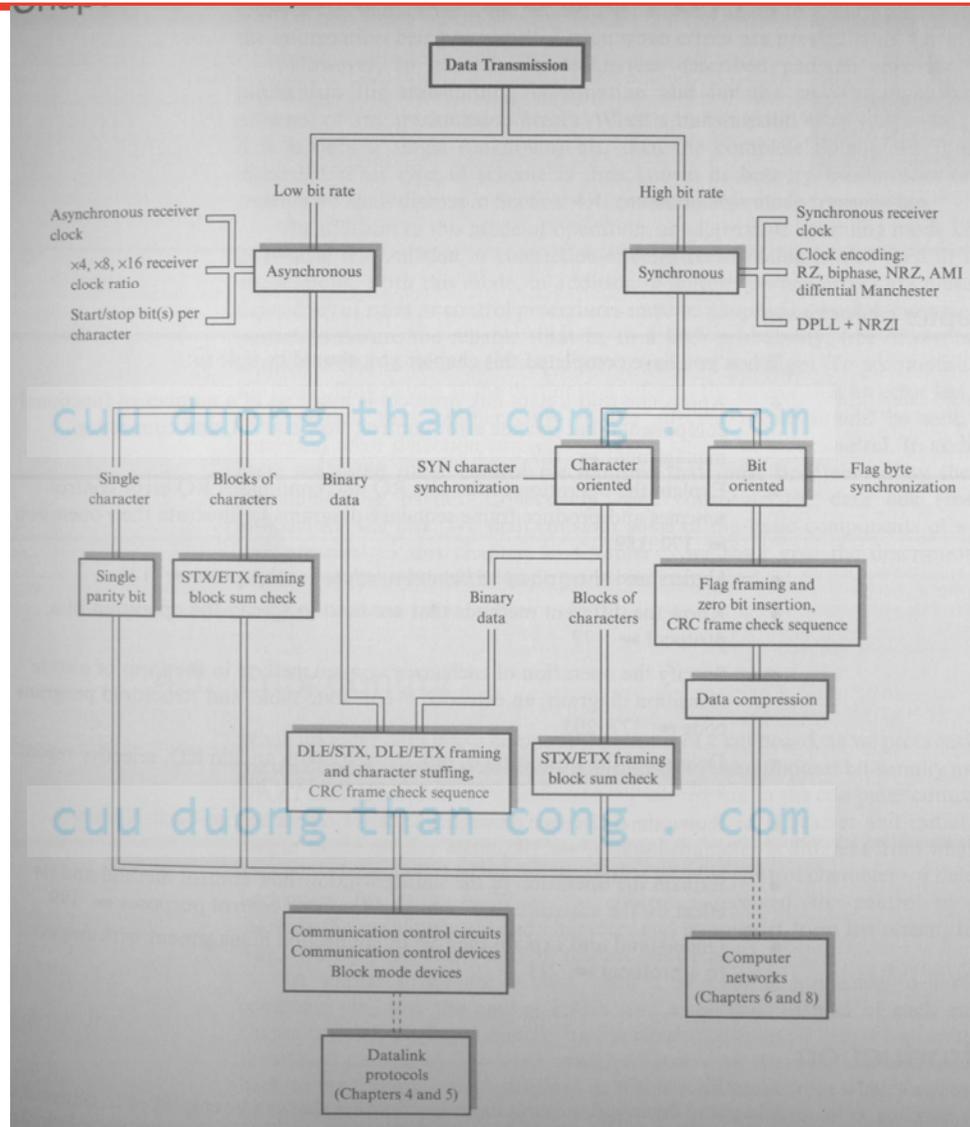


# HDLC

- LAPB – Link Access Procedure, Balanced
  - Là 1 phần của mạng X.25 (ITU-T)
  - Là 1 tập con của HDLC – ABM
  - Sử dụng cho đường điểm – điểm giữa hệ thống và các nút của mạng chuyển mạch gói
- LAPD – Link Access Procedure, D-Channel
  - ISDN (ITU-D)
  - ABM
  - Luôn là chuỗi số 7 bit
  - Trường địa chỉ 16bit chứa 2 địa chỉ con
    - 1 cho thiết bị và 1 cho người sử dụng
- LLC – Logical Link Control
  - IEEE 802
  - Dạng khung khác
  - Điều khiển kết nối tách biệt giữa MAC (Medium Access Layer) và LLC (lớp trên cùng của MAC)
  - Không có trạm sơ cấp và thứ cấp
  - cần 2 địa chỉ: người gửi và người nhận
  - Phát hiện sai ở lớp MAC: 32 bit CRC
  - Các điểm truy xuất nguồn và đích (DSAP, SSAP)



# Thảo luận





# Chương 3\_Bài tập :

## Các Nghi Thức Lớp Liên Kết

### Dư Liệu

cuu duong than cong . com



# Bài 1

- Vẽ tiến trình trao đổi khung (có ghi chú thông tin cần thiết) của nghi thức Idle- RQ loại tường minh giữa DTE A và DTE B thoả mãn các giả sử sau: bắt đầu truyền khung thứ N và khung dữ liệu bị sai một lần, khung dữ liệu thứ N+1 truyền ngay lần đầu tiên không bị lỗi nhưng hai khung xác nhận liên tiếp bị lỗi và khung xác nhận lần ba là tốt

cuu duong than cong . com



## Bài 2

- DTE A truyền cho DTE B 7 khung dữ liệu theo nghi thức Idle- RQ loại không tường minh , thoả mãn các giả sử sau: khung dữ liệu thứ 1 bị sai một lần, khung dữ liệu thứ 3 truyền ngay lần đầu tiên bị lỗi và hai khung xác nhận liên tiếp bị lỗi và khung xác nhận lần ba là tốt. Vẽ tiến trình trao đổi khung (có ghi chú thông tin cần thiết)

cuu duong than cong . com



## Bài 3

- DTE A truyền DTE B 6 khung dữ liệu dùng nghi thức điều khiển lỗi Go back N, kích thước cửa sổ  $k = 3$ . Vẽ tiến trình trao đổi khung với các giả sử sau:
  - Khung dữ liệu thứ 2 bị lỗi 1 lần
  - Khung trả lời của khung thứ 3 bị lỗi 1 lần.
  - Khung thứ 5 và khung trả lời của khung 5 bị lỗi 1 lần

cuu duong than cong . com



## Bài 4

- DTE A truyền DTE B 6 khung dữ liệu dùng nghi thức điều khiển lỗi Selective Repeat dạng tường minh, kích thước cửa sổ  $k = 3$ . Vẽ tiến trình trao đổi khung với các giả sử sau:
  - Khung dữ liệu thứ 2 bị lỗi 1 lần
  - Khung trả lời của khung thứ 3 bị lỗi 1 lần.
  - Khung thứ 5 và khung trả lời của khung 5 bị lỗi 1 lần

cuu duong than cong . com



## Bài 5

- DTE A truyền DTE B 6 khung dữ liệu dùng nghi thức điều khiển lỗi Selective Repeat dạng không tường minh, kích thước cửa sổ  $k = 3$ . Vẽ tiến trình trao đổi khung với các giả sử sau:
  - Khung dữ liệu thứ 2 bị lỗi 1 lần
  - Khung trả lời của khung thứ 3 bị lỗi 1 lần.
  - Khung thứ 5 và khung trả lời của khung 5 bị lỗi 1 lần

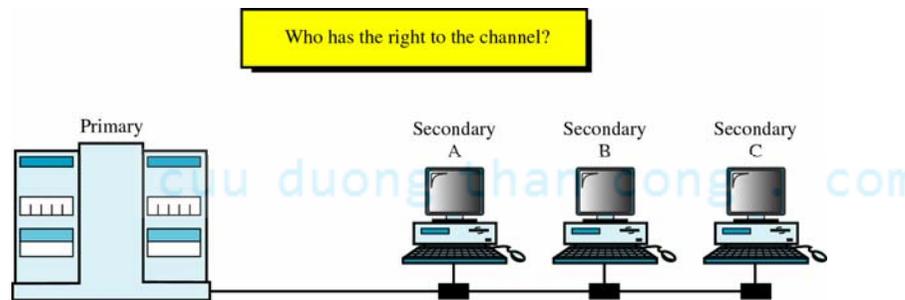
cuu duong than cong . com



## Bài 6

- Một chuỗi các khung dữ liệu có chiều dài  $L = 1000$  bits được phát trên đường truyền có chiều dài  $S = 4000$  Km, tốc độ bit là  $R = 2$  Mbps, tỉ lệ lỗi bit là  $BER = 10^{-4}$ . Tính hiệu suất liên kết khi sử dụng các giao thức sau (bỏ qua thời gian của gói ACK/NAK và thời gian xử lý, Tốc độ truyền sóng là  $C = 3 \times 10^8$  m/s):
  - Stop & Wait
  - Selective Repeat với kích thước cửa sổ là  $N=7$
  - Go-back-N với kích thước cửa sổ là  $N=12$
  - Go-back-N với kích thước cửa sổ là  $N=127$

Cho mô hình truyền số liệu như sau



Biết rằng Slave A truyền cho Slave C 4 khung dữ liệu bằng nghi thức BSC, sử dụng nghi thức điều khiển lỗi Idle RQ (hỏi đáp có nghi) dạng tường minh. Vẽ tiến trình trao đổi khung dữ liệu, với các giả sử sau :

- Khi Slave A truyền Master thì khung dữ liệu thứ 2 bị lỗi 1 lần
- Khi Master truyền dữ liệu cho Slave C khung dữ liệu thứ 3 bị lỗi 1 lần .



## Bài 9

Xác định frame HDLC để phát đi từ sơ cấp với giả sử sau:

- Data : phát chữ “vo” theo mã ASCII 7 bits, kiểm tra chẵn
- Số thứ tự của frame phát là 7, sử dụng Stop and Wait ARQ.
- Dữ liệu được gửi đến mọi terminal trong mạng.
- Bỏ qua FCS

cuu duong than cong . com



# Bài 10

- Vẽ quá trình trao đổi các frames. Biết rằng sử dụng giao thức HDLC ở mode NRM với các giả sử sau:
  - Primary gửi 3 frame đến secondary và I-frame thứ 2 bị lỗi.
  - Kích thước cửa sổ  $k = 2$ , dùng giao thức Go-Back N

cuu duong than cong . com

cuu duong than cong . com