

# Chương 4: Liên kết dữ liệu (data link)

Bùi Văn Hiếu

Email: [bvhieu@cse.hcmut.edu.vn](mailto:bvhieu@cse.hcmut.edu.vn)

# Vấn đề khi trao đổi dữ liệu

- Một số vấn đề khi hai thiết bị kết nối trực tiếp truyền nhận dữ liệu
  - Đồng bộ khung
  - Điều khiển tốc độ truyền dữ liệu
  - Xử lý lỗi gặp phải trên đường truyền
  - Định vị địa chỉ (trong cấu hình multipoint)
  - Phân biệt dữ liệu và thông tin điều khiển
  - Quản lý liên kết

# Nội dung

- Điều khiển dòng dữ liệu
- Điều khiển lỗi
- Một số nghi thức điều khiển liên kết dữ liệu

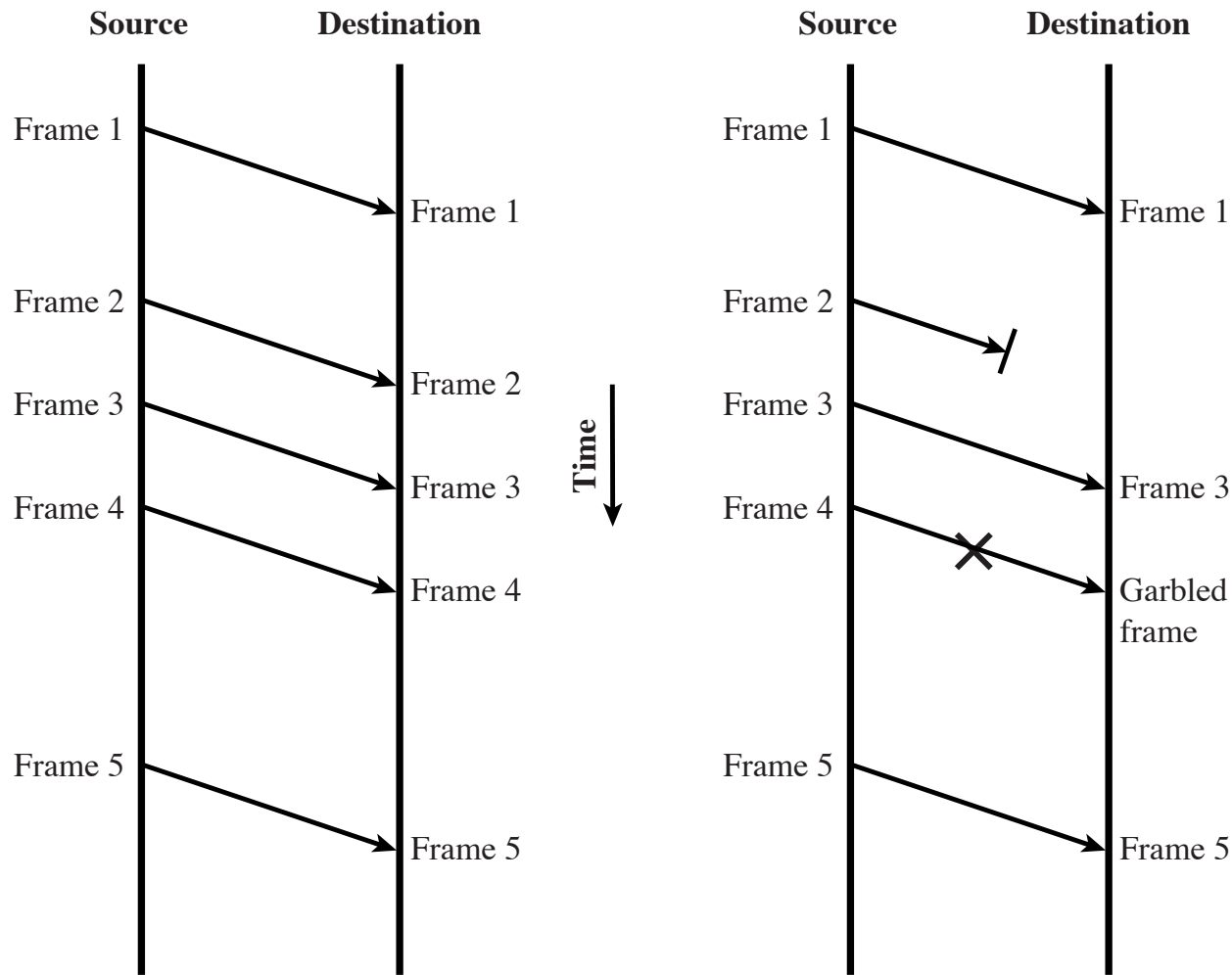
# Nội dung

- **Điều khiển dòng dữ liệu**
- Điều khiển lỗi
- Một số nghi thức điều khiển liên kết dữ liệu

# Điều khiển dòng dữ liệu

- Bên nhận thường có bộ đệm để nhận dữ liệu
- Khi dữ liệu đến, bên nhận thường thực hiện một số xử lý trước khi gửi lên lớp cao hơn
- Điều khiển dòng: đảm bảo bên phát không gửi dữ liệu quá nhanh
  - Ngăn ngừa việc tràn bộ đệm

# Mô hình truyền khung



Error-free transmission

Transmission with losses and errors

# Điều kiện giả định

- Tất cả frame đều đến đích
- Không có frame lỗi
- Các frame đến đúng thứ tự

# Nghi thức Idle RQ (Stop-and-Wait)

- Đặc điểm
  - Phương pháp đơn giản nhất
  - Được dùng chủ yếu trong các ứng dụng character-oriented.(byte-oriented)
  - Sử dụng kênh truyền hoạt động trong chế độ half-duplex
- Cơ chế hoạt động
  - Nguồn phát dữ liệu (dưới dạng các frame)
  - Đích nhận dữ liệu và trả lời bằng ACK
  - Nguồn phải đợi ACK trước khi phát tiếp dữ liệu
    - Đích có thể ngưng truyền dữ liệu bằng cách không gửi ACK



# Idle RQ – Hiệu suất

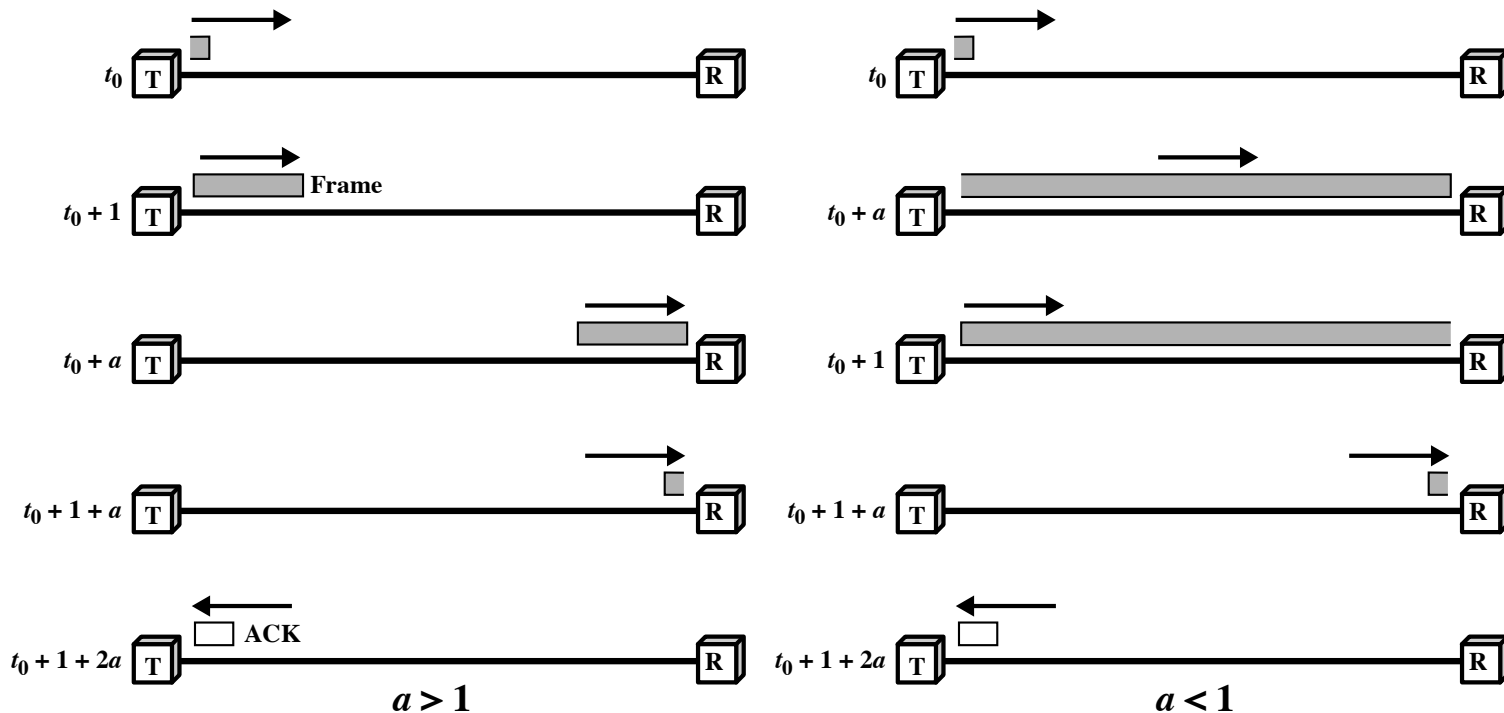
- Khái niệm

- Thời gian truyền ( $t_{\text{frame}}$ ): thời gian cần thiết để gửi tất cả các bit dữ liệu lên đường truyền
- Thời gian lan truyền ( $t_{\text{prop}}$ ): thời gian cần thiết để 1 bit đi từ nguồn đến đích

# Idle RQ – Hiệu suất

- Thời gian tổng cộng  $T_D = n(2t_{prop} + t_{frame})$
- Hiệu suất đường truyền

$$U = \frac{n \cdot t_{frame}}{T_D} = \frac{1}{1 + 2 \cdot \frac{t_{prop}}{t_{frame}}} = \frac{1}{1 + 2a}$$



Stop-and-Wait Link Utilization (transmission time = 1; propagation time =  $a$ )

# Vấn đề kích thước frame

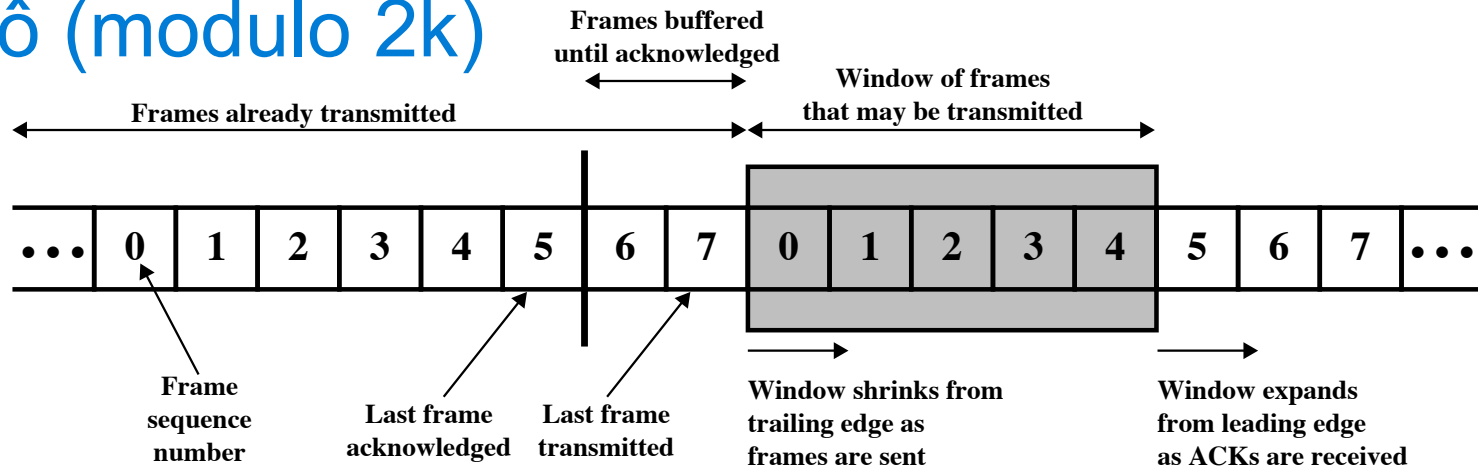
- Hiệu quả đường truyền cao nếu frame kích thước lớn
- Thực tế dữ liệu lớn được chia thành các frame có kích thước nhỏ
  - Kích thước bộ đệm có giới hạn
  - Frame kích thước nhỏ khó xảy ra lỗi
  - Lỗi được phát hiện sớm
  - Khi có lỗi, chỉ cần truyền lại frame nhỏ
  - Ngăn ngừa tình trạng 1 trạm làm việc chiếm đường truyền lâu

# Sliding windows

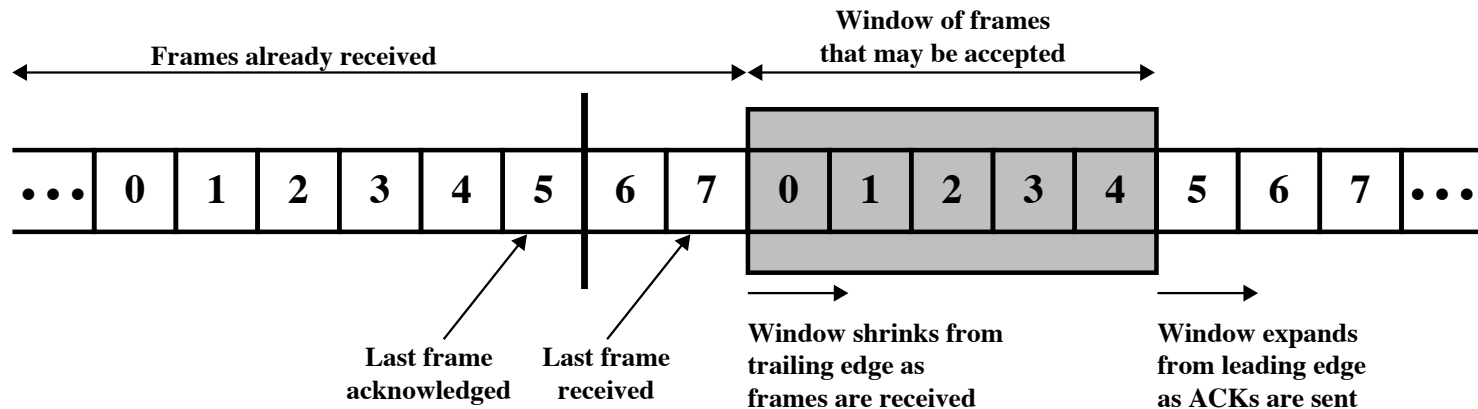
- Cơ chế hoạt động
  - Cho phép nhiều frame có thể truyền đồng thời
  - Bên thu có bộ đệm với kích thước  $W$  (có thể nhận  $W$  frame)
  - Bên phát có thể truyền tối đa  $W-1$  frame mà không cần đợi ACK
  - Đánh số thứ tự cho các frame
  - ACK có chứa số thứ tự của frame kế tiếp có thể truyền

# Sliding windows

- Số thứ tự được quay vòng bởi kích thước cửa sổ (modulo  $2k$ )



Sender's perspective

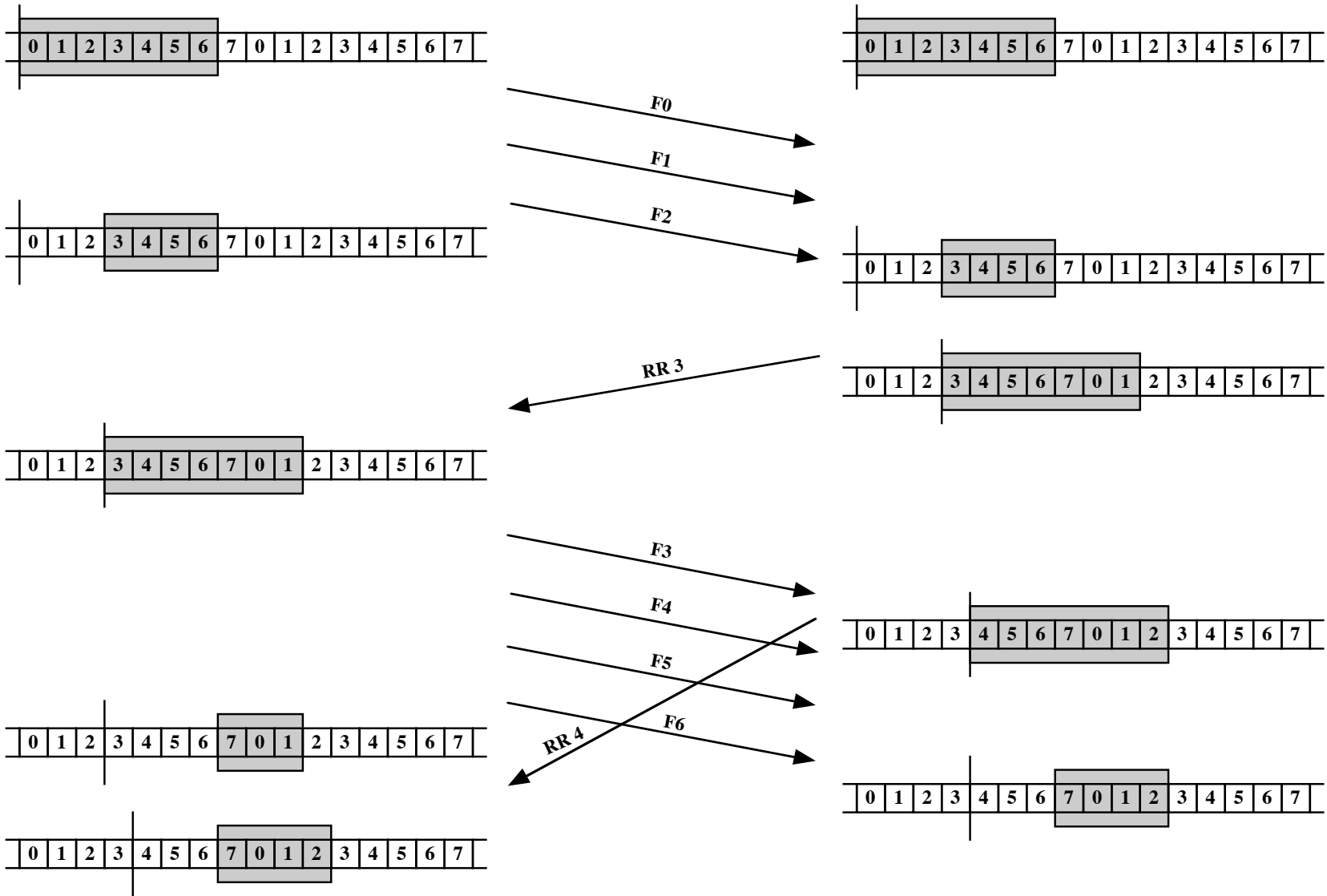


Receiver's perspective

# Sliding windows – Ví dụ (tt)

Source System A

Destination System B



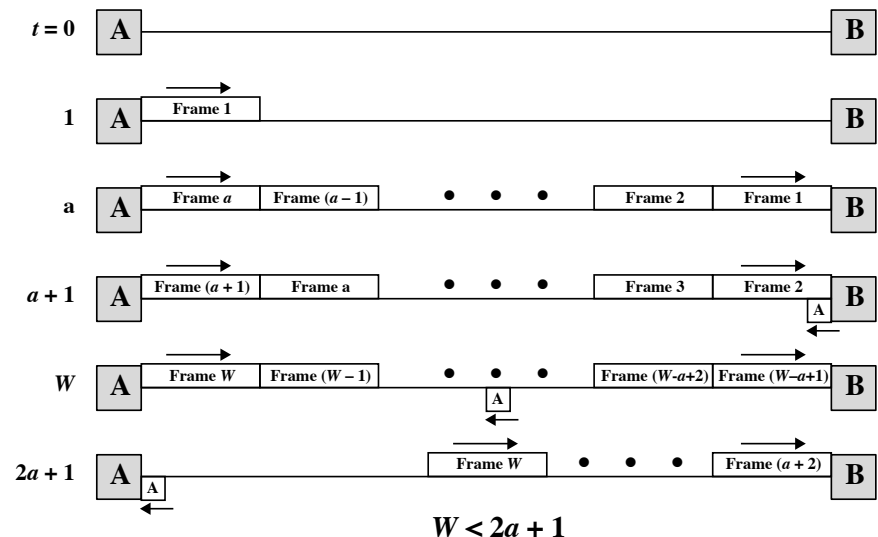
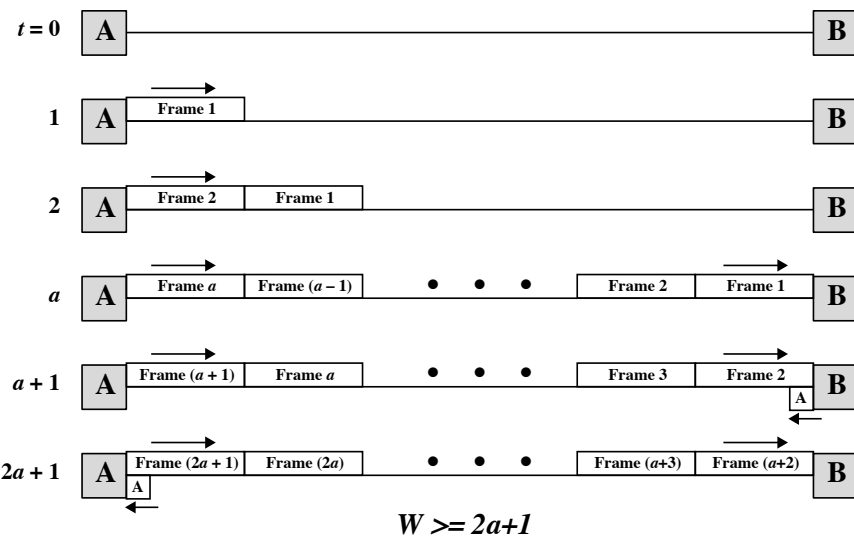
# Sliding windows

- Cải tiến
  - Bên thu có thể gửi ACK mà không cho phép bên phát gửi tiếp dữ liệu (Receive Not Ready)
    - Trong trường hợp này, bên thu phải gửi ACK để bình thường hóa việc truyền nhận dữ liệu khi nó sẵn sàng
  - Nếu đường truyền là full-duplex, dùng cơ chế “piggybacking”: tích hợp ACK vào frame dữ liệu
    - Nếu không có dữ liệu để truyền, dùng ACK frame
    - Nếu có dữ liệu để truyền nhưng không có ACK để truyền, gửi lại ACK cuối cùng, hoặc có cờ ACK hợp lệ (TCP)

# Sliding windows – Hiệu suất

- Hiệu suất
  - Full- Duplex

$$U = \begin{cases} 1 & N \geq 2a + 1 \\ \frac{N}{1 + 2a} & N < 2a + 1 \end{cases}$$





# Nội dung

- Điều khiển dòng dữ liệu
- **Điều khiển lỗi**
- Một số nghi thức điều khiển liên kết dữ liệu

# Điều khiển lỗi là gì ?

- Điều khiển lỗi là các kỹ thuật để **phát hiện** và **sửa lỗi** xảy ra trong quá trình truyền các frame
- Bảo đảm truyền nhận dữ liệu **chính xác**

# Kỹ thuật điều khiển lỗi

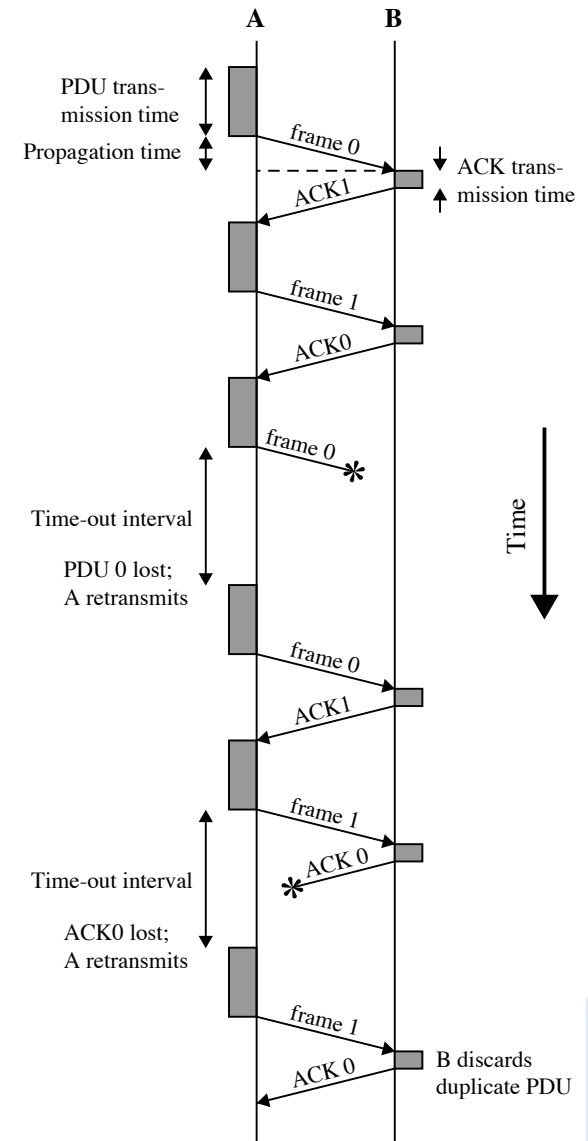
- Phân loại lỗi đối với frame
  - Mất frame: frame không đến đích hoặc đến nhưng thông tin điều khiển trên frame bị hư (bên nhận không thể xác định là frame nào)
  - Frame hư: thông tin điều khiển trên frame xác định được, nhưng dữ liệu trong frame bị lỗi
- Phát hiện lỗi (CRC, Parity, ...)
- Positive ACK – xác nhận các frame nhận được
- Negative ACK (NAK) – yêu cầu truyền lại cho các frame bị hư
- Truyền lại sau một thời gian time-out

# Cơ chế

- Dựa trên điều khiển dòng
- Kỹ thuật ARQ (Automatic Repeat Request)
  - Cho phép các nghi thức liên kết dữ liệu quản lý lỗi và yêu cầu **truyền lại**
  - Phân loại
    - Idle RQ (stop-and-wait)
      - Dùng với cơ chế điều khiển dòng stop-wait
    - Continuous RQ
      - Dùng với cơ chế điều khiển dòng sliding-window
      - Selective repeat
      - Go-back-N

# Idle RQ

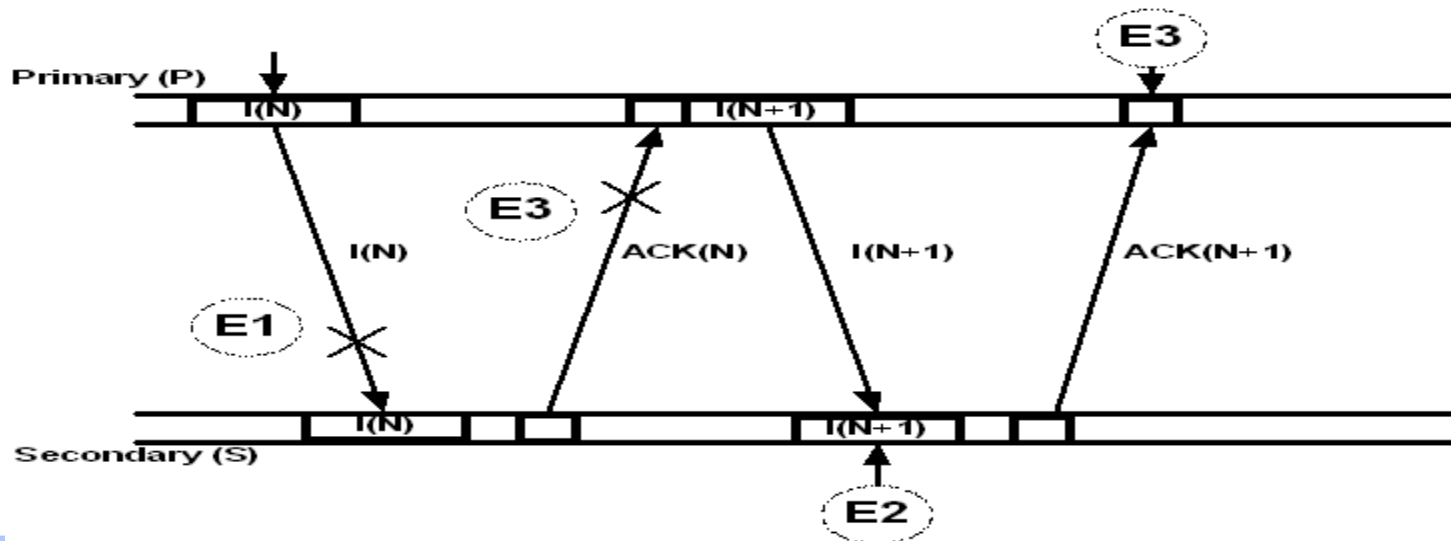
- Cơ chế hoạt động
  - A gửi một I-Frame (Information Frame) đến B
  - A đợi phản hồi từ B trước khi gửi tiếp frame
    - ACK-Frame – A gửi dữ liệu **mới**
    - NAK-Frame – A **gửi lại** dữ liệu
    - Không nhận được trả lời – A gửi lại sau thời gian **time-out**
- Ưu/khuyết điểm
  - Đơn giản
  - Độ hiệu quả đường truyền thấp



# Stop-and-Wait

## ■ Các loại lỗi

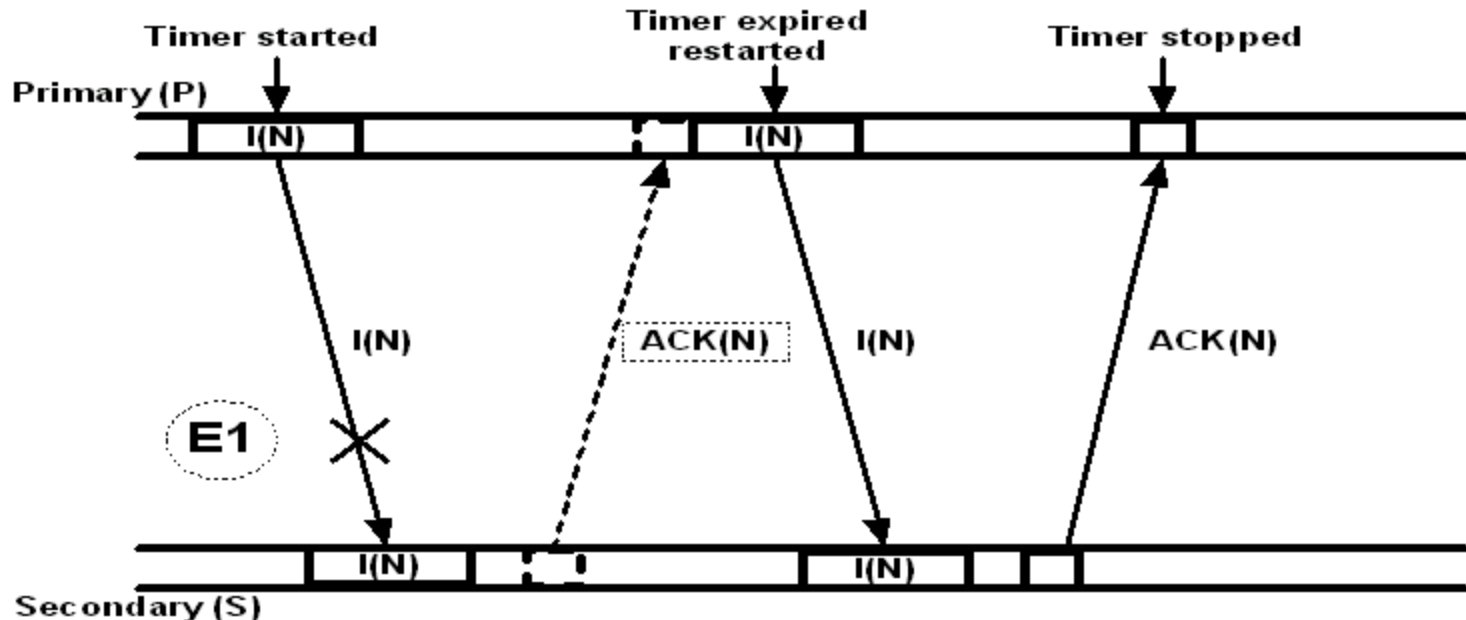
- (E1) I-Frame không đến được bên nhận
- (E2) I-Frame đến được bên nhận nhưng nội dung I-Frame bị sai
- (E3) ACK-Frame không đến được bên gửi hay ACK-Frame đến được bên gửi nhưng nội dung ACK-Frame bị sai



# Stop-and-Wait

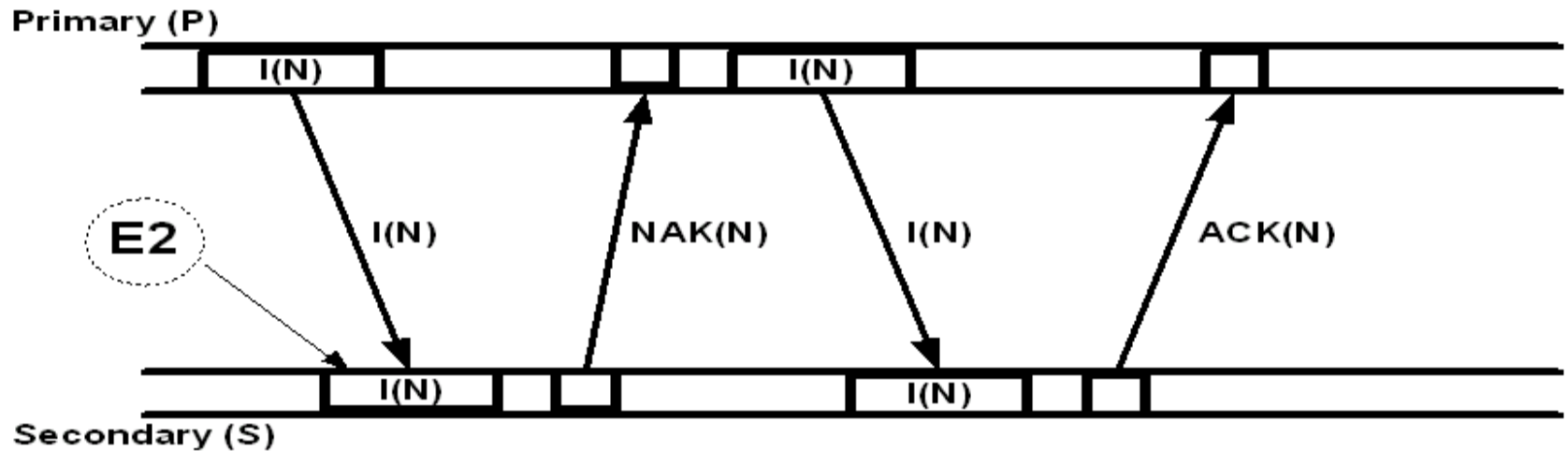
## ■ Sửa lỗi E1

- Sử dụng timer: bên gửi sau khi gửi đi một I-Frame thì khởi động một bộ đếm thời gian, sau khoảng thời gian đợi  $T$  mà chưa nhận được tín hiệu ACK/NAK báo về thì xem như I-Frame chưa tới và gửi lại frame này.



# Stop-and-Wait

- Sửa lỗi E2
  - Truyền lại Frame

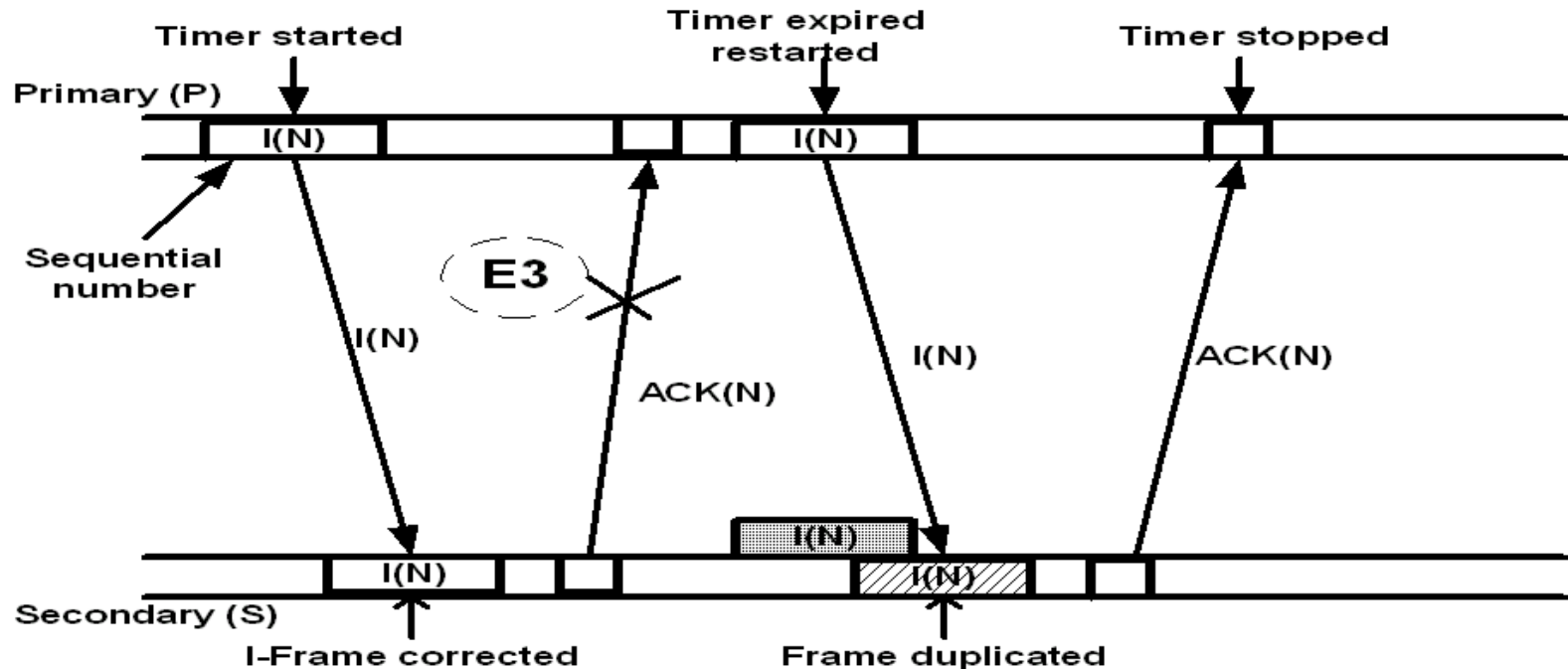




# Stop-and-Wait

## ■ Sửa lỗi E3

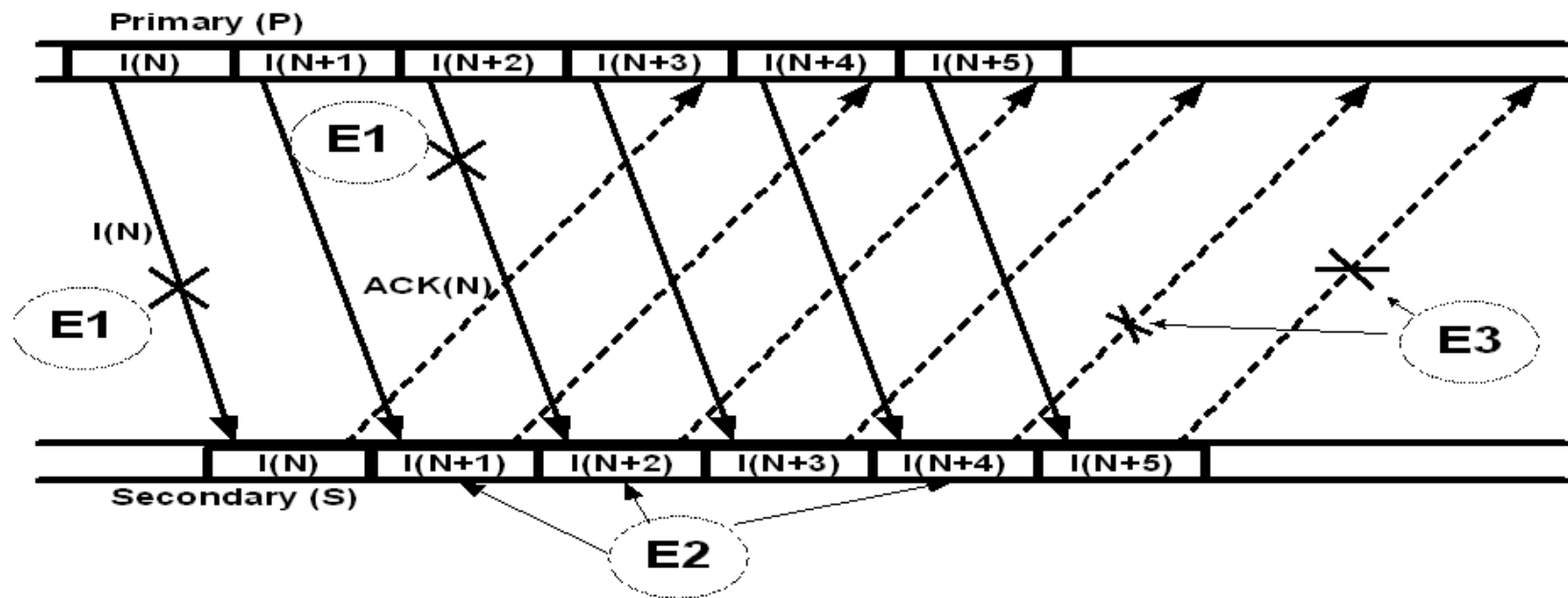
- Không nhận được ACK-Frame: truyền lại
- Lỗi lặp lại frame (duplicated frame): dùng chỉ số tuần tự frame (sequential number) để bên nhận có thể loại bỏ các frame trùng lặp



- Cơ chế hoạt động
  - Điều khiển
    - RR = receive ready = ACK = acknowledge
    - REJ = reject = NAK = negative acknowledge
  - Dựa trên cơ chế sliding window
    - A gửi liên tục các I-Frame đến B (trong khi cơ chế điều khiển dòng **còn cho phép**)
    - B chỉ nhận I-Frame theo **đúng chỉ số** tuần tự
    - Truyền lại **tất cả** các Frame kể từ Frame sai đầu tiên trở đi

# Go-back-N – Các trường hợp lỗi

- Các kiểu lỗi tương tự như trong Idle RQ (có thể xảy ra đồng thời trên nhiều frame)
  - (E1) I-Frame không đến được bên nhận
  - (E2) I-Frame đến được bên nhận, nội dung I-Frame sai
  - (E3) ACK-Frame không đến được bên gửi



## ▪ Sửa lỗi E1

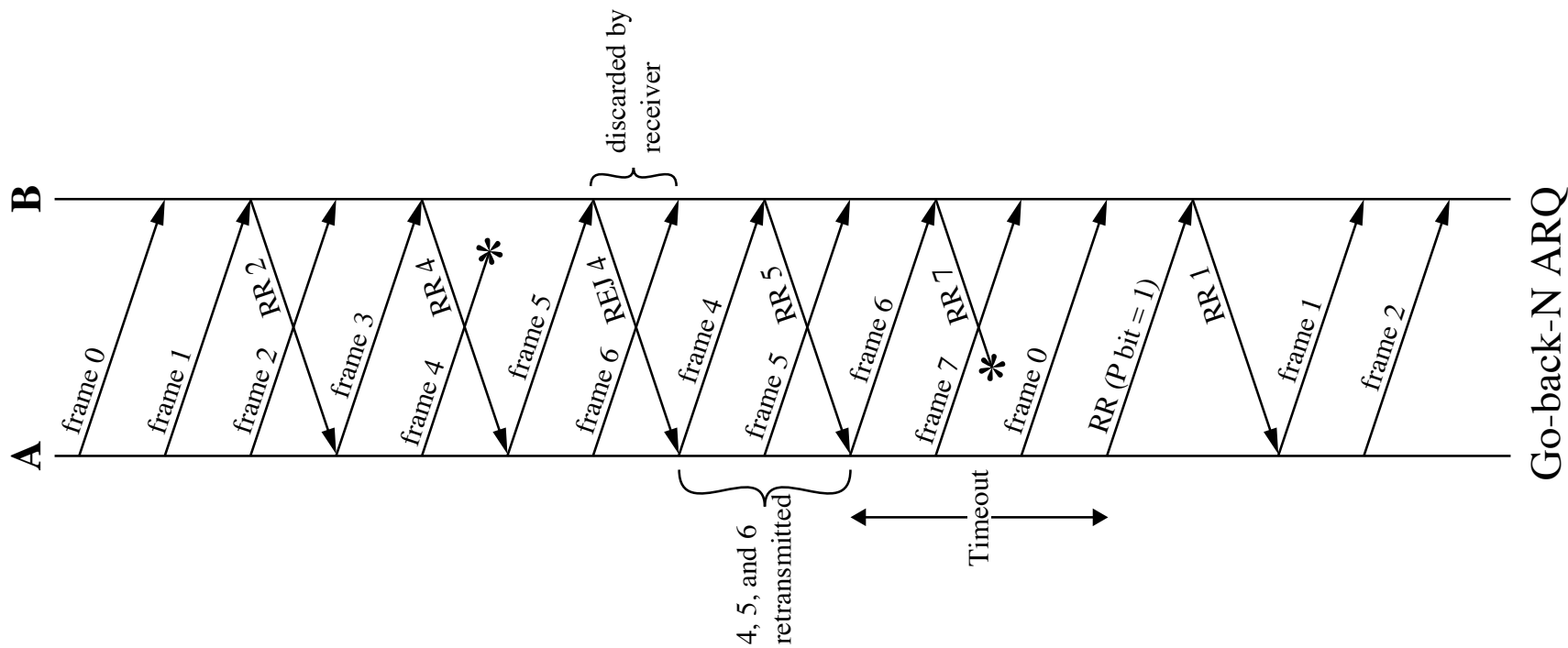
- Sử dụng danh sách truyền lại (Retransmission list) lưu các I-Frame gửi nhưng **chưa có ACK**
- Khi gửi mỗi I-Frame thì khởi động một **bộ đếm** thời gian tương ứng với I-Frame đó
- **Sau khoảng thời gian** đợi T mà chưa nhận được frame ACK thì xem như I-Frame này chưa tới và **gửi lại** các frame kể từ frame này
- Bên nhận chỉ nhận các frame đúng trình tự nên sẽ **bỏ những frame sau** frame đã không đến
  - Cải tiến bằng cách gửi NAK tương ứng (bên nhận không cần đợi timeout nên sẽ nhanh hơn)

# Go-back-N

- Sửa lỗi E2
  - Bên nhận **phát hiện lỗi** ở frame  $i$
  - Bên nhận báo cho bên gửi bằng **NAK<sub>i</sub>** và **loại bỏ** các frame sau  $i$
  - Bên gửi gửi lại các frame **từ frame<sub>i</sub>**
  - Thời gian đáp ứng nhanh hơn so với dùng timeout.

- Sửa lỗi E3
  - $ACK_i$  bị mất
  - Nếu sau đó có  $ACK_{i+1}, ACK_{i+2}...$  thì truyền bình thường
  - Nếu timeout bên gửi sẽ gửi lại I-Frame
  - Bên nhận phát hiện frame truyền lại này đã nhận nên sẽ báo lại ACK (hoặc NAK) tương ứng
  - Khi nhận ACK, bên phát không cần truyền lại tất cả các frame mà có thể truyền frame từ chỉ số trong ACK

# Go-back-N ví dụ



Go-back-N ARQ

# Selective Repeat

- Cơ chế hoạt động
  - Tương tự như Go-Back-N
  - Chỉ gửi lại các frame bị NAK hoặc time-out
  - Bên nhận có thể nhận I-frame không theo đúng chỉ số tuần tự
  - Bên nhận phải có buffer để lưu lại các frame đến không theo đúng chỉ số tuần tự



# Vấn đề kích thước cửa sổ

- Tình huống: window kích thước 7
  - A gửi các frame từ 0 đến 6 qua B
  - B xác nhận tất cả, nhưng tất cả ACK đều bị mất
  - A đợi bị timeout, nên gửi lại frame 0
  - B lúc này đã dịch cửa sổ nhận, có thể nhận các frame 7,0,1,...5. Nó tưởng frame 7 bị mất và 0 là frame mới, nên chấp nhận (trùng frame)
- Đây là vấn đề trùng lặp giữa cửa sổ gửi và cửa sổ nhận
- Do đó kích thước cửa sổ tối đa là  $\frac{1}{2}(N)$  (N là chỉ số tuần tự lớn nhất)

# Selective Repeat

- Sửa lỗi E1
  - Sử dụng danh sách truyền lại (Retransmission list) lưu các I-Frame gửi nhưng chưa có ACK
  - Khi gửi mỗi I-Frame thì khởi động một bộ đếm thời gian tương ứng với I-Frame đó
  - Sau khoảng thời gian đợi  $T$  mà chưa nhận được frame ACK thì gửi lại frame này

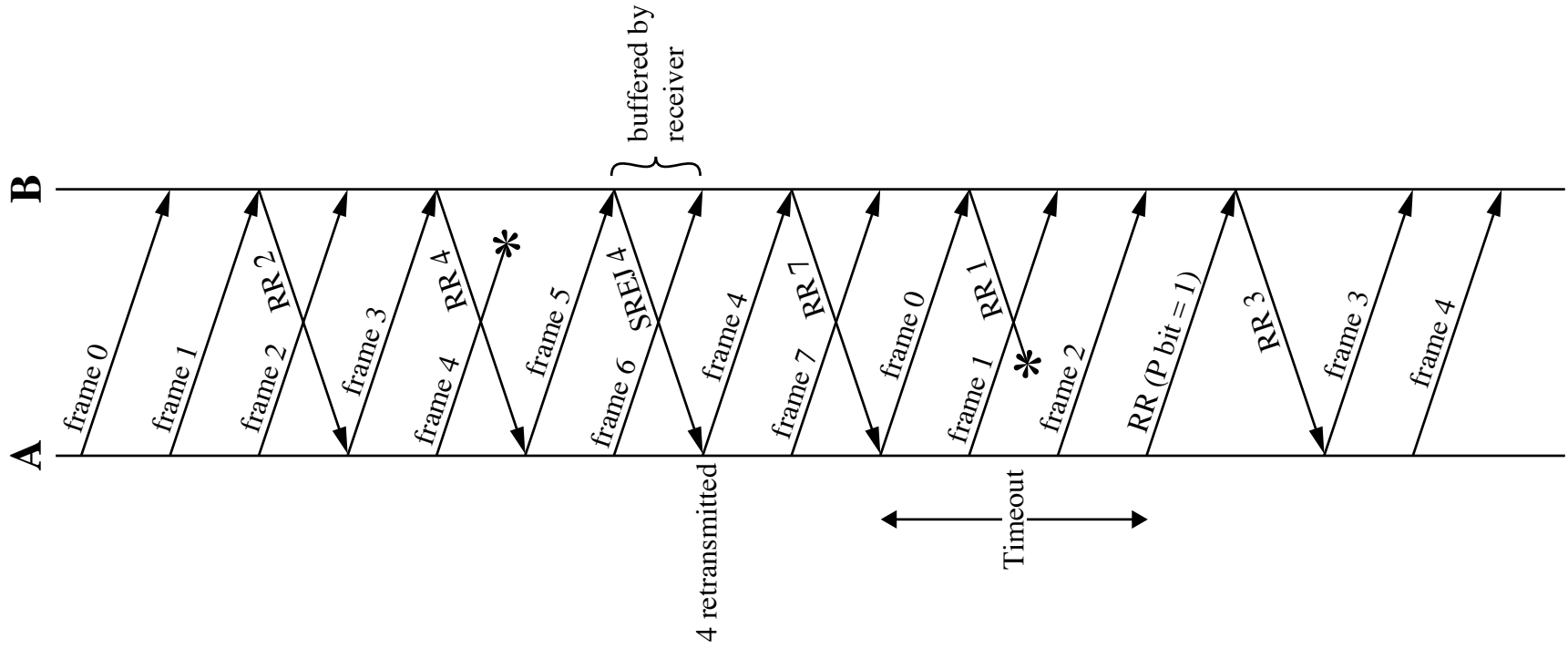
# Selective Repeat

- Sửa lỗi E2
  - I-Frame truyền đến bên nhận nhưng bị lỗi
  - Bên nhận báo cho bên gửi biết thông qua NAK-Frame
  - Bên nhận vẫn lấy các frame tiếp theo vào bộ đệm (nhưng chưa đưa lên lớp trên)
  - Bên gửi chỉ truyền lại I-Frame bị lỗi
  - Bên nhận nhận đúng các frame theo trình tự thì có thể đưa dữ liệu lên lớp trên

# Selective Repeat

- Sửa lỗi E3
  - Sử dụng chỉ số tuần tự frame (sequential number): Khi ACK-frame bị lỗi hay không đến được bên gửi, sau thời gian timeout bên gửi sẽ gửi lại I-Frame này

# Selective Repeat ví dụ



Selective-reject ARQ

# Điều khiển lỗi – Hiệu suất

- Stop-and-wait protocol

$$U = \frac{1-P}{1+2a}$$

- Go-back-N protocol

$$U = \begin{cases} \frac{1-P}{1+2aP} & N > 2a+1 \\ \frac{N(1-P)}{(2a+1)(1-P+NP)} & N < 2a+1 \end{cases}$$

- Selective repeat protocol

$$U = \begin{cases} 1-P & N > 2a+1 \\ \frac{N(1-P)}{1+2a} & N < 2a+1 \end{cases}$$

# Nội dung

- Điều khiển dòng dữ liệu
- Điều khiển lỗi
- **Một số nghi thức điều khiển liên kết dữ liệu**
  - Nghi thức Binary synchronous communication (BSC)
  - Nghi thức High-level data link control (HDLC)
  - Một số nghi thức tựa DLC khác

# Nội dung

- Điều khiển dòng dữ liệu
- Điều khiển lỗi
- Một số nghi thức điều khiển liên kết dữ liệu
  - **Nghi thức Binary synchronous communication (BSC)**
  - Nghi thức High-level data link control (HDLC)
  - Một số nghi thức tựa DLC khác

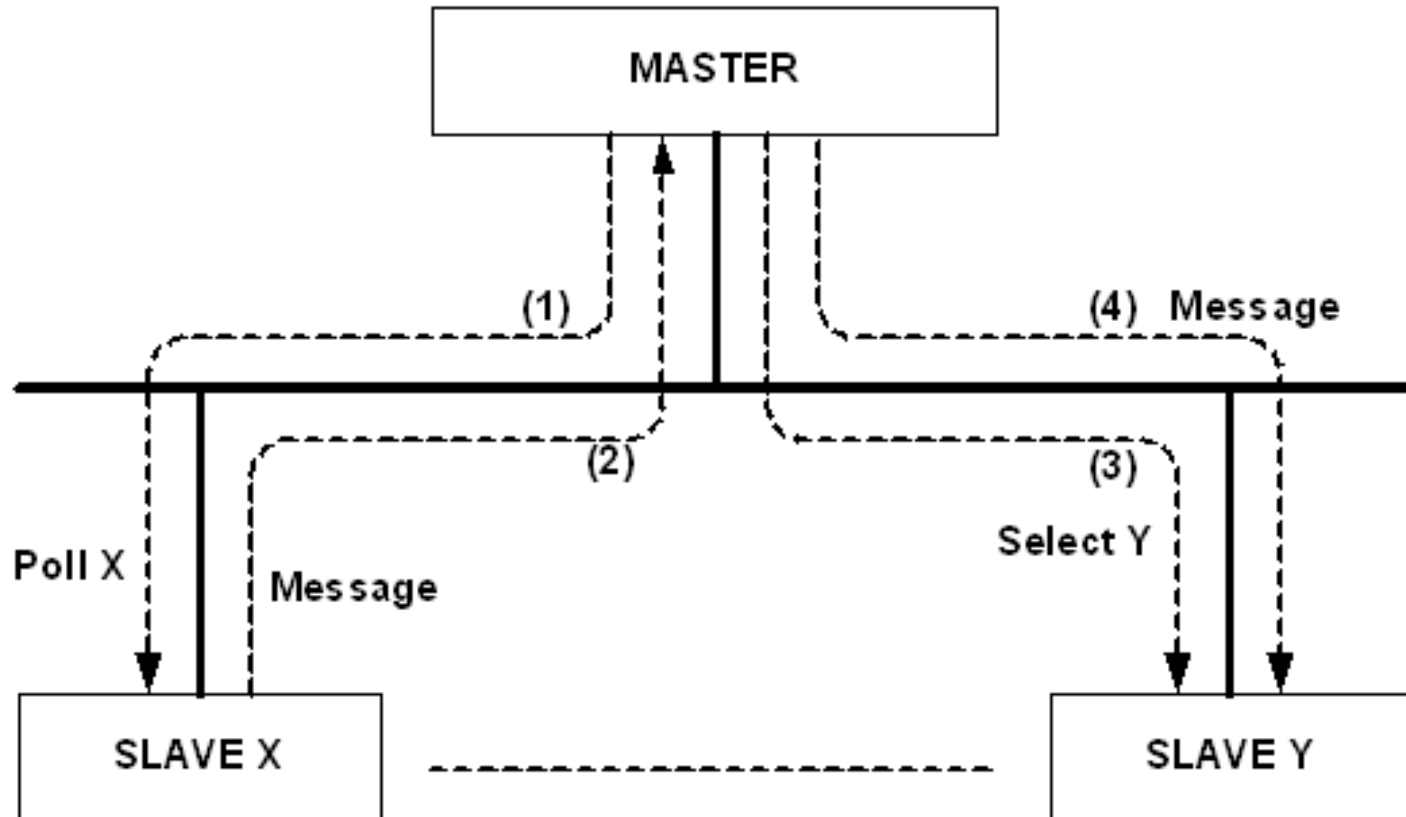


# Nghi thức BSC

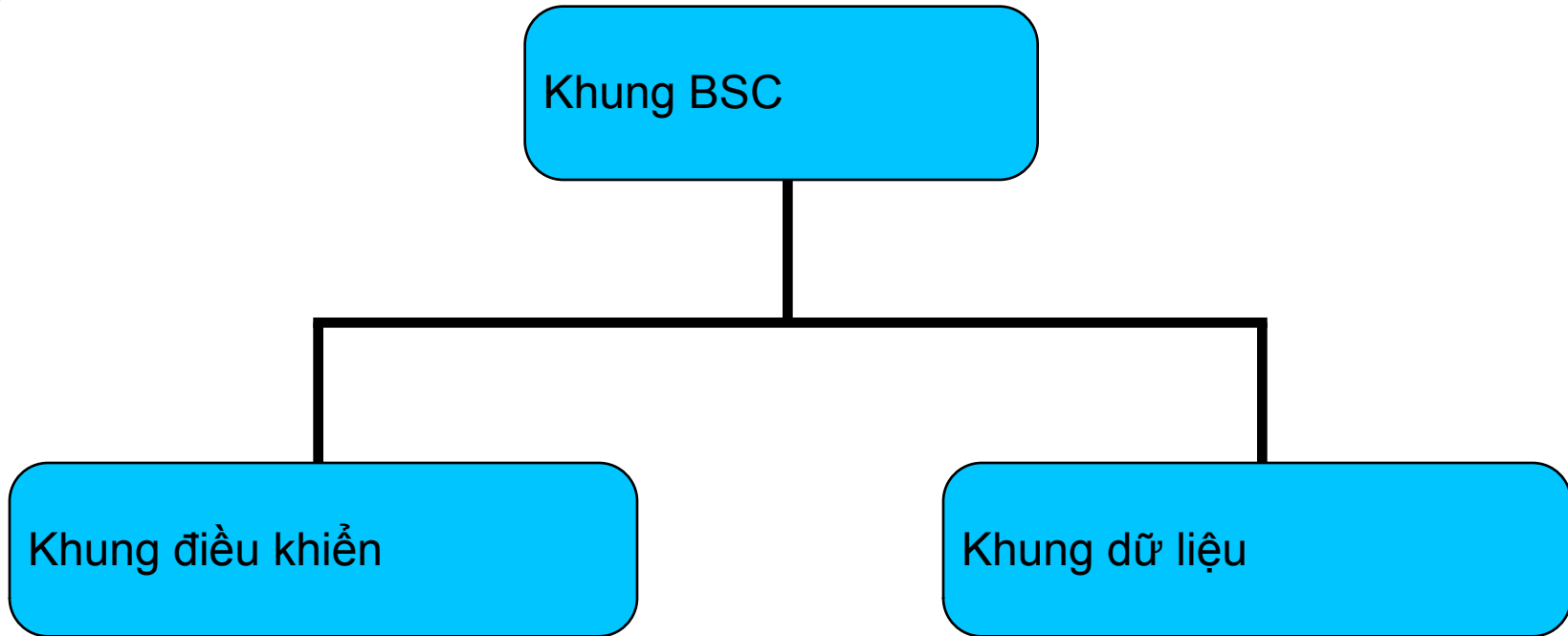
- Nghi thức hướng đến kí tự
- IBM
  - ISO 1745
  - ANSI X3.28
- Hỗ trợ truyền half-duplex
- Cấu hình point-to-point hoặc multipoint
- Truyền dẫn đồng bộ hoặc bất đồng bộ
- Điều khiển lỗi “Idle RQ”

# Nghi thức BSC

- Cơ chế hoạt động Poll-Select



# Khung BSC



# Khung giám sát (điều khiển)

## Poll/Select sequence



## Positive ack for even/odd-numbered frame



## Negative ack for even/odd-numbered frame



## Positive select response



## Negative select response



## End of Transmission/ no messages to send



# Định dạng khung BSC

## Single Block Message

SYN	SYN	SOH	Id	Station Adc	STX	Text	ETX	BCC
-----	-----	-----	----	-------------	-----	------	-----	-----

## First Block of Multiblock message

SYN	SYN	SOH	Id	Station Adc	STX	Text	ETB	BCC
-----	-----	-----	----	-------------	-----	------	-----	-----

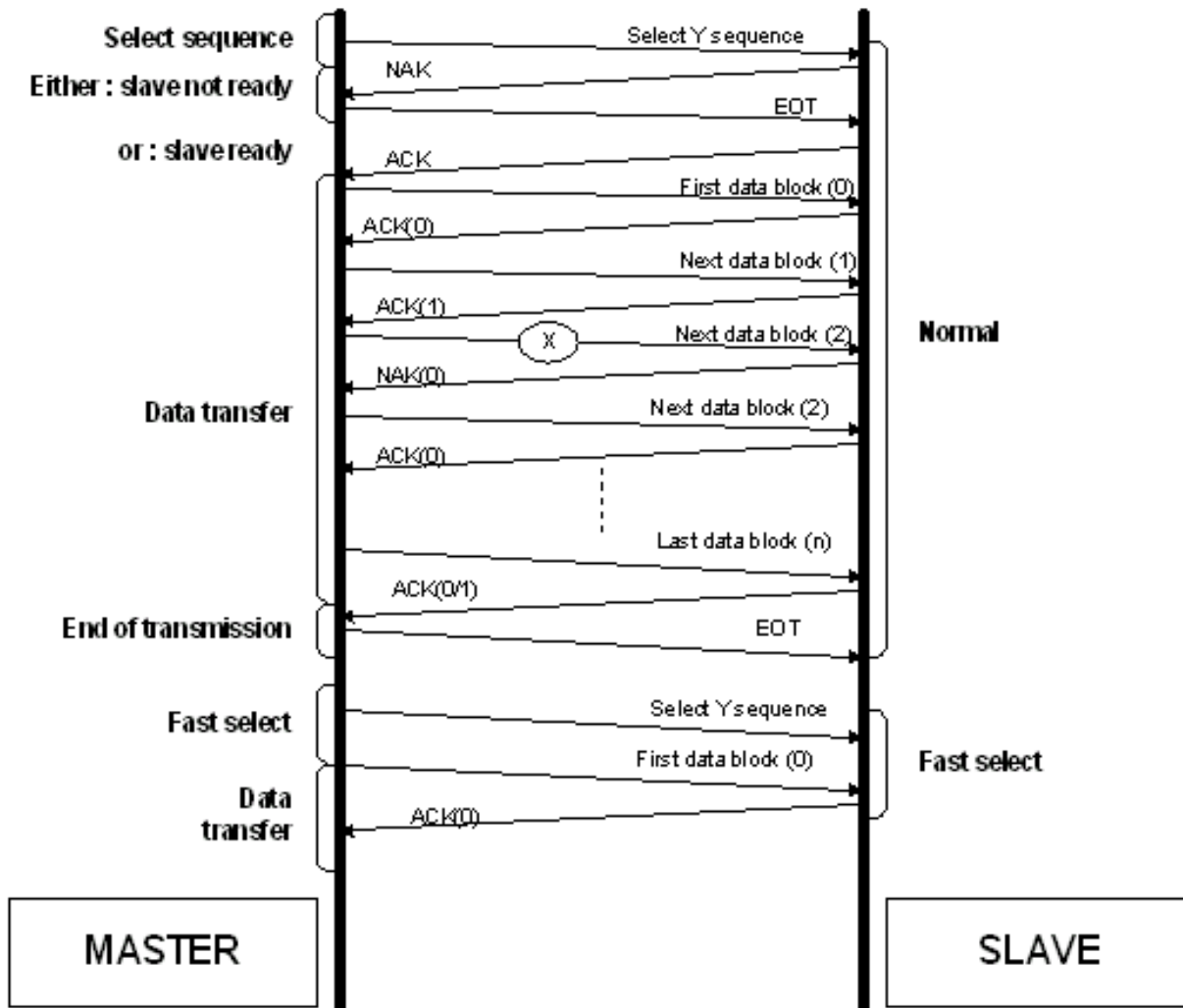
## Intermediate block(s) of Multiblock message

SYN	SYN	SOH	Id	STX	Text	ETB	BCC
-----	-----	-----	----	-----	------	-----	-----

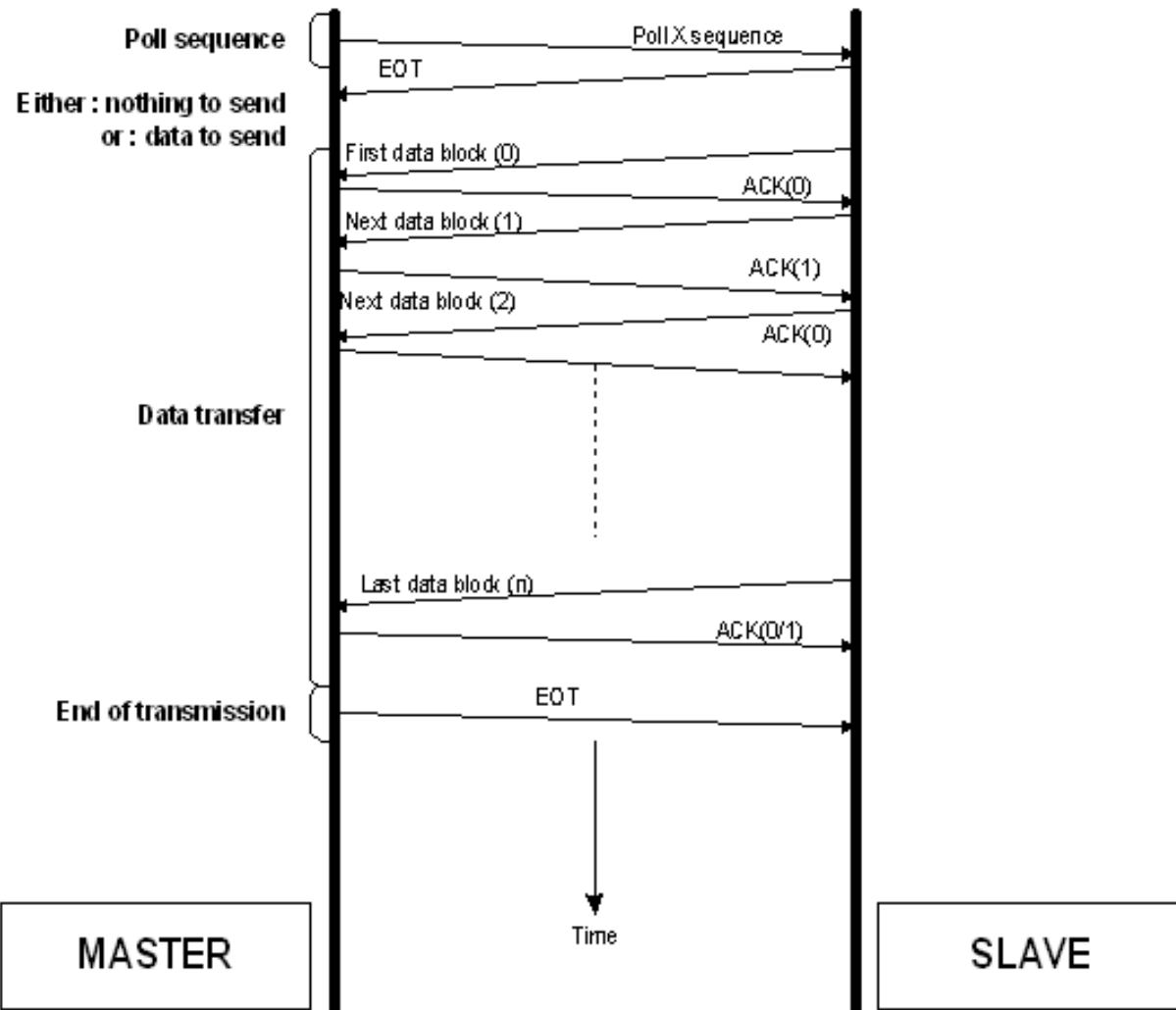
## Last block of Multiblock message

SYN	SYN	SOH	Id	STX	Text	ETX	BCC
-----	-----	-----	----	-----	------	-----	-----

# Chuỗi khung trong chế độ Select



# Chuỗi khung trong chế độ Poll



# Đánh giá giao thức BSC

- Hạn chế
  - Không đánh số thứ tự (idle ARQ)
  - Chỉ truyền half-duplex
  - Phụ thuộc mã điều khiển (code-dependent)
  - Không điều khiển lỗi cho các trường điều khiển
  - Không có khả năng mở rộng
  - Tình trạng độc chiếm đường truyền của máy phát
  - Thủ tục khó sử dụng đối với văn bản



# Nội dung

- Điều khiển dòng dữ liệu
- Điều khiển lỗi
- Một số nghi thức điều khiển liên kết dữ liệu
  - Nghi thức Binary synchronous communication (BSC)
  - **Nghi thức High-level data link control (HDLC)**
  - Một số nghi thức tựa DLC khác

# High-level Data Link Control (HDLC)

- Nghi thức liên kết dữ liệu quan trọng nhất
- Chuẩn hoá: ISO 3009, ISO 4335
- Nhiều nghi thức liên kết dữ liệu khác tương tự (hoặc dựa trên) nghi thức này

# High-level Data Link Control – HDLC

- Nghi thức hướng đến bit
- Đặc điểm
  - Hoạt động ở chế độ full-duplex
  - Liên kết point-to-point hoặc multipoint
  - Truyền dẫn đồng bộ
  - Điều khiển lỗi “Continuous RQ”
  - Có thể dùng cho các liên kết với giá trị lớn và nhỏ của  $a$

# Nghi thức HDLC

- Mục tiêu
  - Không phụ thuộc mã điều khiển
  - Khả năng thích ứng
  - Hiệu quả cao
  - Độ tin cậy cao

# Các đặc tính cơ bản

- Phân loại các trạm
  - Trạm chính (primary station)
    - **Điều khiển** hoạt động của liên kết
    - Các khung (frame) phát ra được gọi là lệnh (command)
  - Trạm phụ (secondary station)
    - Hoạt động **dưới sự điều khiển** của trạm chính
    - Các khung phát ra được gọi là đáp ứng (response)
    - Trạm chính duy trì các liên kết **luận lý** riêng cho các trạm phụ
  - Trạm tổ hợp (combined station)
    - **Kết hợp** đặc điểm của cả trạm chính và trạm phụ
    - Có thể phát ra các lệnh và đáp ứng

# Các đặc tính cơ bản (tt)

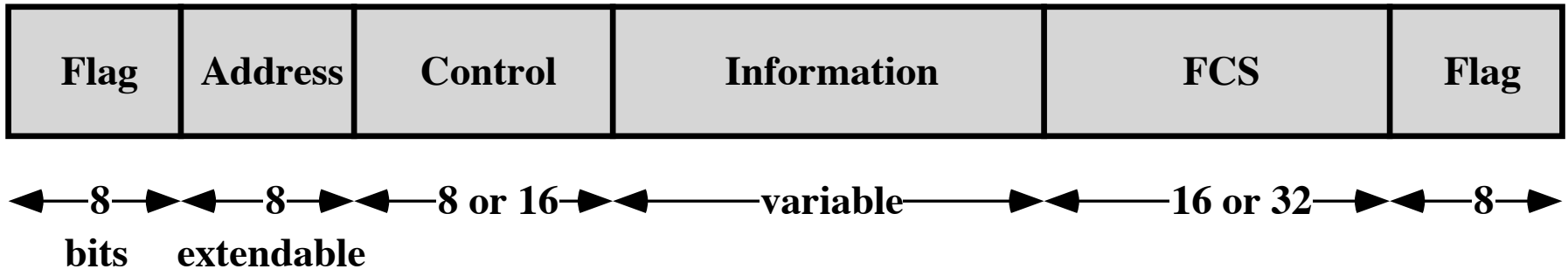
- Cấu hình liên kết (link configuration)
  - Không cân bằng (unbalanced)
    - Bao gồm **một** trạm chính và **một hoặc nhiều** trạm phụ
    - Hỗ trợ truyền half duplex và full duplex
  - Cân bằng (balanced)
    - Bao gồm **hai** trạm tổ hợp
    - Hỗ trợ truyền half duplex và full duplex

# Các đặc tính cơ bản (tt)

- Chế độ truyền (transfer mode)
  - Normal Response Mode (NRM)
    - Được sử dụng trong cấu hình **không cân bằng**
    - **Trạm chính** khởi động việc trao đổi dữ liệu
    - Trạm phụ chỉ có thể truyền dữ liệu **đáp ứng** với lệnh từ trạm chính
    - Được dùng trong đường multidrop
  - Asynchronous Balanced Mode (ABM)
    - Được sử dụng trong cấu hình **cân bằng**
    - **Một trong hai** trạm có thể khởi động việc trao đổi dữ liệu
    - Được sử dụng phổ biến nhất
  - Asynchronous Response Mode (ARM)
    - Được sử dụng trong cấu hình **không cân bằng**
    - Trạm phụ **có thể khởi động** quá trình truyền dữ liệu
    - Trạm chính **vẫn chịu trách nhiệm** cho đường truyền (khởi động, điều khiển lỗi, ngắt kết nối ... )
    - Ít được sử dụng

# Cấu trúc khung (frame structure)

- HDLC sử dụng truyền dẫn đồng bộ
- Truyền dẫn theo khung
- Sử dụng một cấu trúc khung **duy nhất** cho tất cả trao đổi dữ liệu và điều khiển





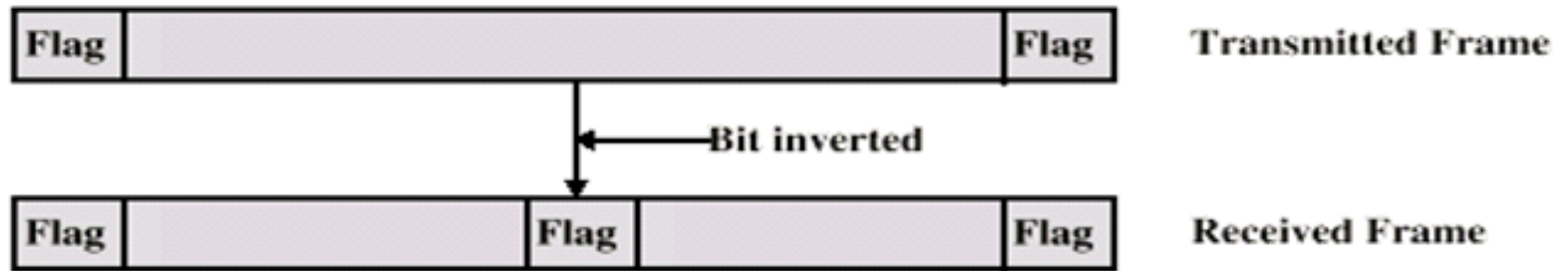
# Cờ điều khiển (flag)

- Dùng để **phân cách** khung (đầu và cuối)
  - Giá trị: 01111110
- Có thể dùng **vừa là kết thúc** khung này **vừa là bắt đầu** khung khác
- Sử dụng kỹ thuật **chèn bit** (bit stuffing) để tránh xuất hiện cờ trong dữ liệu
  - Bit stuffing: 0 được chèn thêm mỗi khi xuất hiện năm số 1 liên tiếp trong phần dữ liệu

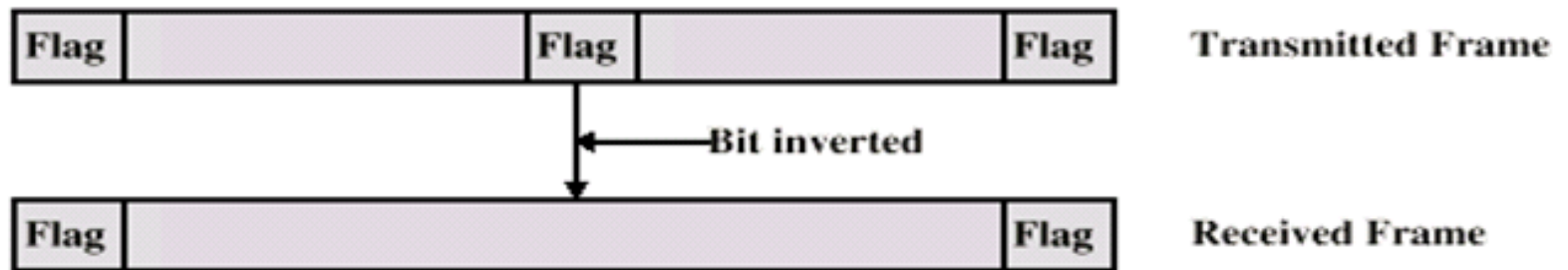
11111111111111011111101111110

1111101111101101111101011111010

# Ảnh hưởng nếu lỗi xảy ra với cờ



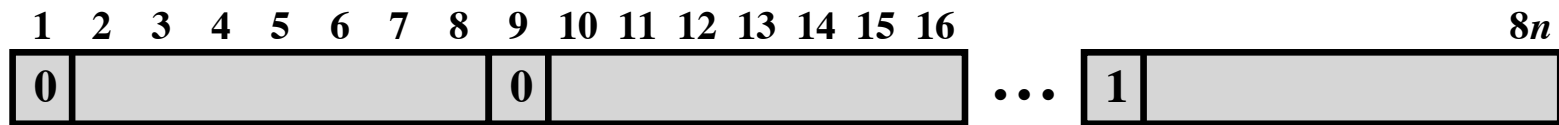
(b) An inverted bit splits a frame in two



(c) An inverted bit merges two frames

# Trường địa chỉ

- Dùng để **xác định** trạm phụ đã gửi hoặc sẽ nhận khung
- Không cần thiết trong kết nối điểm-điểm nhưng **vẫn được giữ** để nhất quán)
- Thường dài 8 bit
- Giá trị “1111111” là địa chỉ broadcast (gửi đến tất cả)
- **Có thể mở rộng** thành bội số của 7 bit
  - LSB của mỗi octet báo đây là octet cuối cùng (1) hay chưa (0)



**Extended Address Field**

# Trường điều khiển

- HDLC định nghĩa **ba** loại khung, tương ứng có ba loại trường điều khiển
  - Khung thông tin (I-frame) chứa dữ liệu cần truyền
    - Điều khiển dòng và điều khiển lỗi được gửi kèm trong các khung thông tin (piggybacked)
  - Khung giám sát (supervisor frame, S-frame) dùng cho ARQ khi piggybacking không được dùng
  - Khung không số (unnumbered frame, U-frame) bổ sung các chức năng điều khiển liên kết

	1	2	3	4	5	6	7	8
<b>I: Information</b>	0	N(S)		P/F	N(R)			
<b>S: Supervisory</b>	1	0	S	P/F	N(R)			
<b>U: Unnumbered</b>	1	1	M	P/F	M			

N(S) = Send sequence number  
N(R) = Receive sequence number  
S = Supervisory function bits  
M = Unnumbered function bits  
P/F = Poll/final bit

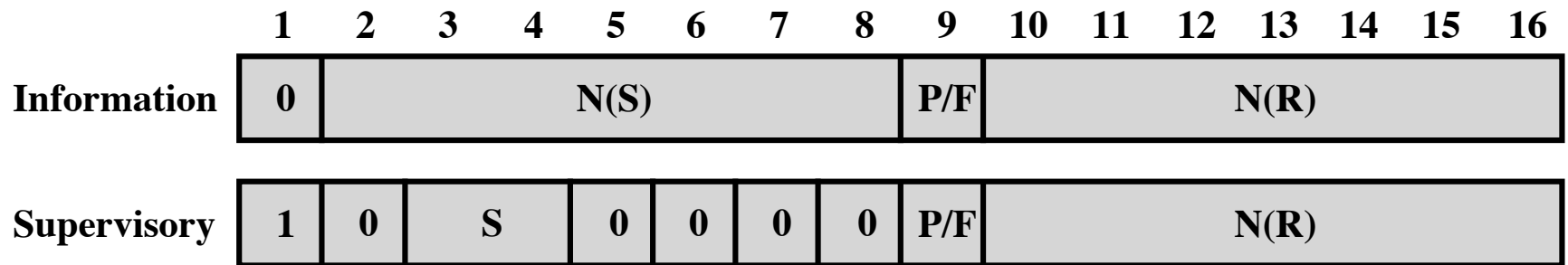
**8-bit control field format**

# Trường điều khiển (tt)

- Bit Poll/Final: ý nghĩa tùy theo ngữ cảnh
  - Trong khung lệnh (command frame)
    - Ý nghĩa là bit P
    - 1 để mời gọi (poll) khung đáp ứng của các trạm ngang hàng
  - Trong khung đáp ứng (response frame)
    - Ý nghĩa là bit F
    - 1 để chỉ thị khung đáp ứng là kết quả của lệnh mời gọi

# Trường điều khiển (tt)

- Khung thông tin và khung giám sát có thể mở rộng 16 bit
- Sử dụng chỉ số tuần tự là 7 bit



**16-bit control field format**

# Trường thông tin

- **Chỉ có** trong các khung thông tin (I-frame) và một số khung không số (U-frame)
- Phải là một **số nguyên** các octet (8 bits)
- Chiều dài **thay đổi**, giới hạn **tùy** hệ thống

# Trường FCS

- Dùng để phát hiện lỗi
- Được tính dựa trên **các bit còn lại** của khung
- CRC 16 bit
- Có thể dùng CRC 32 bit



# Tổng kết Frame format

Flag	Address	Control	Information	FCS	Flag
8 bit	8/16 bit	8/16 bit	0 to N bit	16/32 bit	8 bit
Start of frame	Frame header		Information field	Frame check sequence	End of frame

0	N(S)				P/F	N(R)		
1	2	3	4	5	6	7	8	

Information

N(S) : send sequence number  
N(R) : receive sequence number  
P/F : Poll/final number

1	0	S	P/F	N(R)
---	---	---	-----	------

Supervisory

Receiver ready - RR  
Receiver not ready - RNR  
Reject - REJ  
Selective reject - SREJ

S

1	1	M	P/F	M
---	---	---	-----	---

Unnumbered

CONTROL FIELD - 8 bit

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

Information

1	0	S	-					P/F	N(R)							
---	---	---	---	--	--	--	--	-----	------	--	--	--	--	--	--	--

Supervisory

1	1	M	P/F	M	P/F	-									
---	---	---	-----	---	-----	---	--	--	--	--	--	--	--	--	--

Unnumbered

CONTROL FIELD - 16 bit

Set Asynchronous Response Mode (SARM)  
Set Normal Response Mode (SNRM)  
.....

M

# Hoạt động

- Trao đổi khung thông tin, khung giám sát và khung không số
- 3 giai đoạn
  - Khởi tạo
  - Trao đổi dữ liệu
  - Ngắt kết nối

# Các lệnh và đáp ứng

Name	Command/ Response	Description
<b>Information (I)</b>	C/R	Exchange user data
<b>Supervisory (S)</b>		
Receive ready (RR)	C/R	Positive acknowledgment; ready to receive I-frame
Receive not ready (RNR)	C/R	Positive acknowledgment; not ready to receive
Reject (REJ)	C/R	Negative acknowledgment; go back N
Selective reject (SREJ)	C/R	Negative acknowledgment; selective reject
<b>Unnumbered (U)</b>		
Set normal response/extended mode (SNRM/SNRME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous response/extended mode (SARM/SARME)	C	Set mode; extended = 7-bit sequence numbers
Set asynchronous balanced/extended mode (SABM, SABME)	C	Set mode; extended = 7-bit sequence numbers
Set initialization mode (SIM)	C	Initialize link control functions in addressed station
Disconnect (DISC)	C	Terminate logical link connection
Unnumbered Acknowledgment (UA)	R	Acknowledge acceptance of one of the set-mode commands
Disconnected mode (DM)	R	Responder is in disconnected mode
Request disconnect (RD)	R	Request for DISC command
Request initialization mode (RIM)	R	Initialization needed; request for SIM command
Unnumbered information (UI)	C/R	Used to exchange control information
Unnumbered poll (UP)	C	Used to solicit control information
Reset (RSET)	C	Used for recovery; resets N(R), N(S)
Exchange identification (XID)	C/R	Used to request/report status
Test (TEST)	C/R	Exchange identical information fields for testing
Frame reject (FRMR)	R	Report receipt of unacceptable frame

## HDLC Commands and Responses

# Khởi tạo

- Gửi U-frame khởi tạo **1 trong 6** chế độ
  - SNRM / SNRME
  - SARM / SARME
  - SABM / SABME
- Nếu đồng ý kết nối gửi lại U-frame **UA**  
(unnumbered acknowledged)
- Nếu không đồng ý kết nối gửi lại U-frame **DM**  
(disconnected mode)

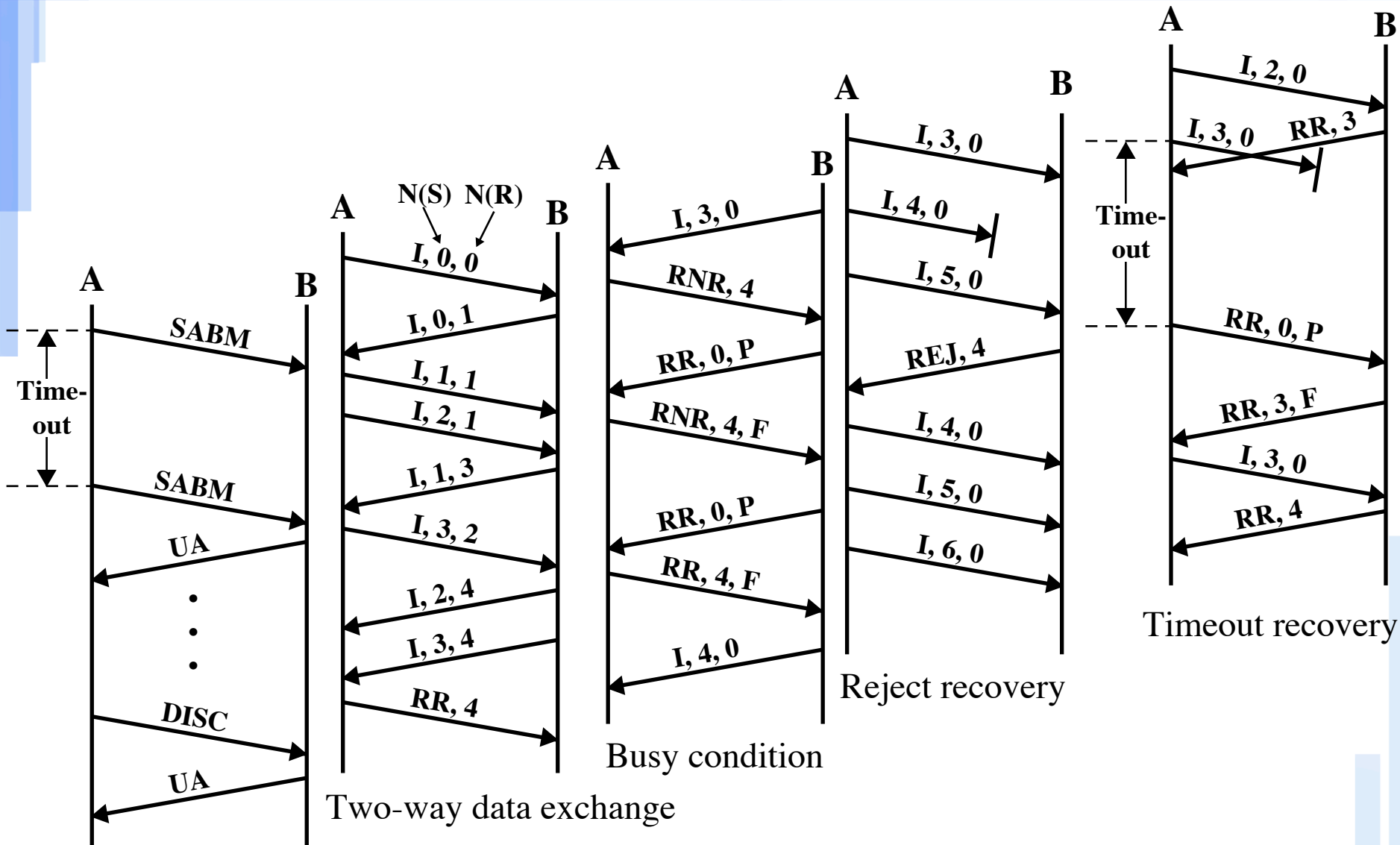
# Trao đổi dữ liệu

- Sau khi đã kết nối
- Cả hai bên đều có thể gửi I-frame (chỉ số tuần tự bắt đầu từ 0)
- Các S-frame có thể được dùng để điều khiển dòng và điều khiển lỗi nếu
  - RR : ACK
  - RNR : bên nhận bận, sau đó phải phát RR để tiếp tục nhận dữ liệu
  - REJ: NACK (go-back-N)
  - SREJ: NACK (selective repeat)

# Ngắt kết nối

- **Một trong hai** bên ngắt kết nối bằng cách gửi U-frame DISC (disconnect)
- Bên kia **phải** chấp nhận ngắt kết nối, **gửi lại** U-frame UA(unnumbered acknowledgment)
- Các khung quá độ có thể bị mất (việc phục hồi phải do các lớp trên)

# Ví dụ quá trình hoạt động



Link setup and disconnect

# BSC vs. HDLC

Feature	BSC	HDLC
Transmission technique	Async/Sync	Sync
Transmission mode	Half-duplex	Half/Full duplex
Framing		
Start	SYN SYN	Flag
Stop	Characters	Flag
Frame formats	Numerous	1 (3 types)
Control information	Header, control frames	Header (1-2 bytes)
Addressing	Contention or polling	Header
Error detection	LRC, CRC-16, CRC-12	CRC-CCITT
Error checking	Transmission block only	Entire frame
Error control	Stop-and-wait	Go-back-N or selective repeat
Flow control	WACK	Sliding window
Window size	1	7 or 127
Transparency	DLE	Bit stuffing



# Nội dung

- Điều khiển dòng dữ liệu
- Điều khiển lỗi
- Một số nghi thức điều khiển liên kết dữ liệu
  - Nghi thức Binary synchronous communication (BSC)
  - Nghi thức High-level data link control (HDLC)
  - **Một số nghi thức DLC khác**

# Link Access Procedure, Balanced (LAPB)

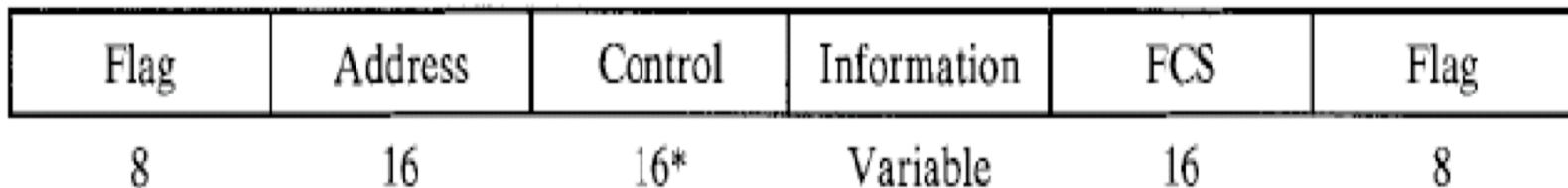
- Một phần của chuẩn X.25 (ITU-T)
- Tập con của HDLC, chỉ cung cấp chế độ ABM
- Dùng trong liên kết point-to-point giữa hệ thống của người dùng và node của mạng chuyển mạch gói

Flag	Address	Control	Information	FCS	Flag
8	8n	8 or 16	Variable	16 or 32	8

(a) HDLC, LAPB

# Link Access Procedure, D-Channel(LAPD)

- Dùng trong mạng ISDN (ITU-D)
- Chỉ dùng chế độ truyền ABM
- Luôn dùng chỉ số trình tự 7 bit (không dùng 3 bit)
- Trường FCS luôn 16 bit
- Trường địa chỉ 16 bit chứa 2 địa chỉ phụ
  - Một cho thiết bị và một cho người dùng (lớp trên)



(b) LAPD

# Logical Link Control (LLC)

- Một phần của chuẩn IEEE 802 (điều khiển mạng LAN)
- Định dạng khung khác HDLC
- Tất cả các trạm ngang hàng
- Điều khiển liên kết tách thành lớp MAC (medium access layer) và lớp LLC (phía trên lớp MAC)

# LCC (tt)

- MAC

- Cung cấp địa chỉ thiết bị gửi và thiết bị nhận
- Phát hiện lỗi (CRC - 32)

- LLC

- Cung cấp điểm truy cập nguồn và đích (Destination and source access points – DSAP, SSAP)
- Trường điều khiển giống HDLC

MAC control	Dest. MAC address	Source MAC address	DSAP	SSAP	LLC control	Information	FCS
Variable	16 or 48	16 or 48	8	8	16*	Variable	32

(c) LLC/MAC

# Frame relay

- Khả năng truyền dòng dữ liệu tốc độ cao qua mạng chuyển mạch
- Được dùng thay thế cho X.25
- Điều khiển liên kết dữ liệu dùng LAPF (Link Access Procedure for Frame-Mode Bearer Services)
- Thực sự gồm 2 nghi thức
  - Điều khiển (control protocol) tương tự như HDLC
  - Nhân (core protocol) tập con của nghi thức điều khiển
- Chỉ hỗ trợ chế độ truyền ABM

# Frame relay (tt)

- Chỉ số trình tự 7 bit, CRC-16
- Trường địa chỉ dài 2, 3 hay 4 octet
- LAPF core không có trường điều khiển (không điều khiển dòng, điều khiển lỗi)

Flag	Address	Control	Information	FCS	Flag
8	16 or 32	16*	Variable	16 or 32	8

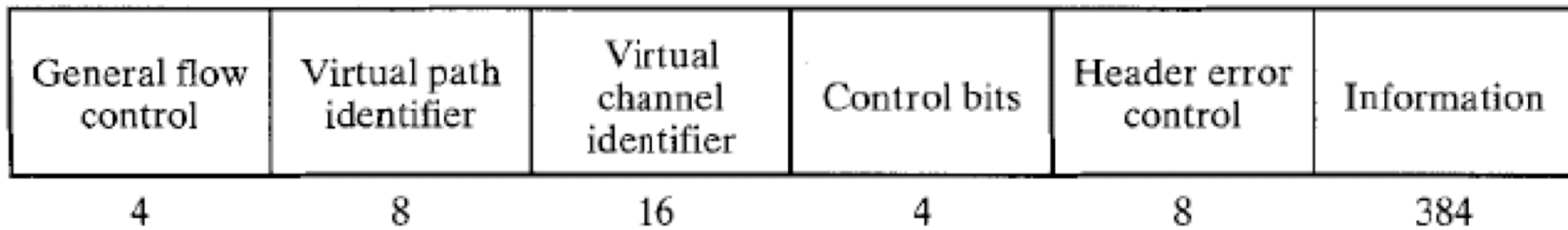
(d) LAPF (control)

Flag	Address	Information	FCS	Flag
8	16 to 32	Variable	16	8

(e) LAPF (core)

# Asynchronous Transfer Mode (ATM)

- Khả năng truyền dòng dữ liệu qua mạng tốc độ cao
- Không dựa trên HDLC
- Định dạng khung gọi là “cell”
- Cell có kích thước cố định 53 octet (424 bit)



(f) ATM