

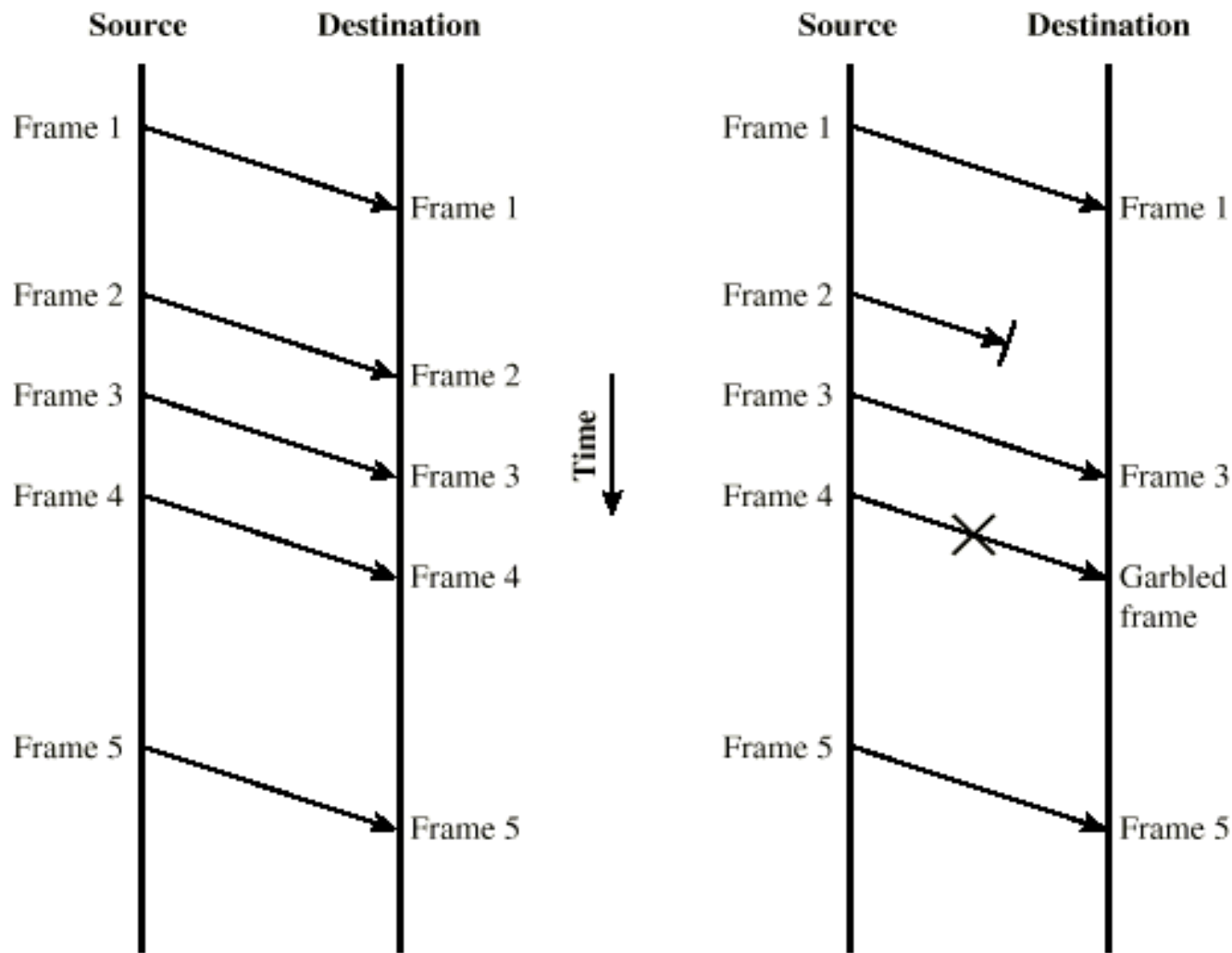
# CHƯƠNG 5

## CÁC GIAO THỨC ĐIỀU KHIỂN LIÊN KẾT DỮ LIỆU

# Điều khiển dòng (Flow Control)

- Bảo đảm cho máy phát không gửi dữ liệu quá nhanh
  - Ngăn ngừa việc tràn bộ đệm
- Thời gian truyền
  - Thời gian cần thiết để gửi tất cả các bit dữ liệu lên đường truyền.
- Thời gian lan truyền
  - Thời gian cần thiết để 1 bit đi từ nguồn đến đích.

# Mô hình truyền Frame



(a) Error-free transmission

(b) Transmission with losses and errors

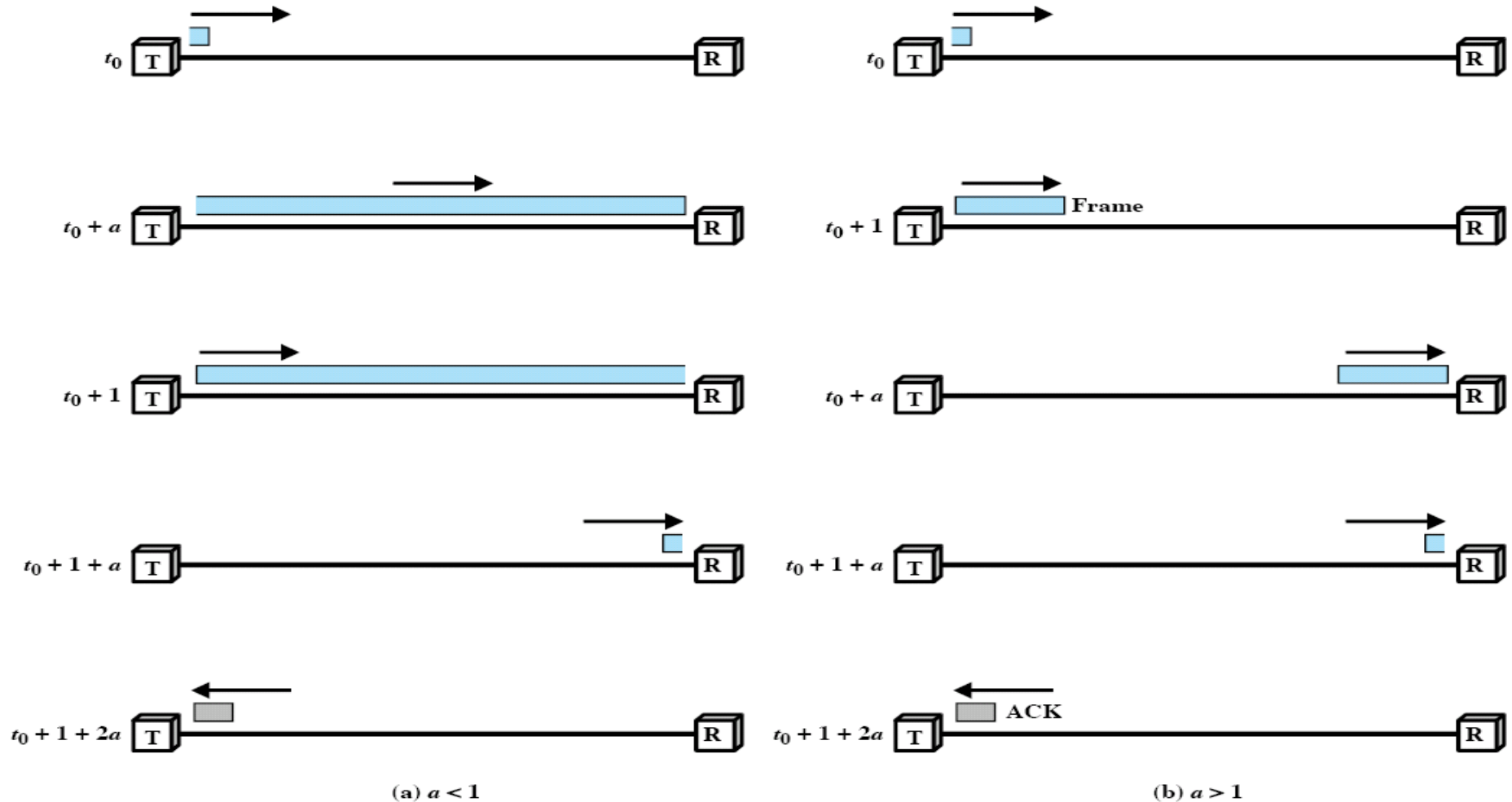
# Dừng lại và đợi (Stop-and-Wait)

- Máy phát truyền các frame dữ liệu
- Máy nhận nhận dữ liệu và trả lời bằng ACK
- Máy phát đợi ACK trước khi phát tiếp dữ liệu
- Máy nhận có thể ngưng bằng cách không gửi ACK
- Thích hợp khi chỉ có vài frame có kích thước lớn

# Chia nhỏ gói tin

- Dữ liệu lớn được chia thành các frame có kích thước nhỏ
  - Kích thước bộ đệm có giới hạn
  - Lỗi được phát hiện sớm (khi cả gói dữ liệu đã nhận được)
  - Khi có lỗi, chỉ cần truyền lại frame nhỏ
  - Ngăn ngừa tình trạng 1 trạm làm việc chiếm đường truyền lâu
- Stop and wait trở nên không thích hợp

# Sử dụng đường truyền của Stop and Wait

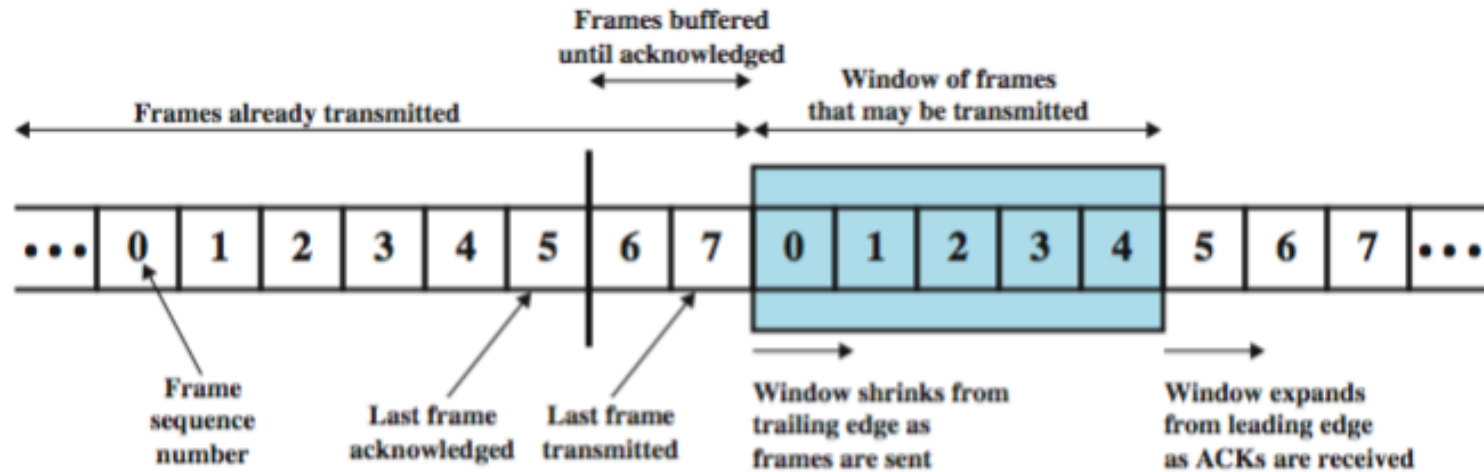


Stop-and-Wait Link Utilization (transmission time = 1; propagation time =  $a$ )

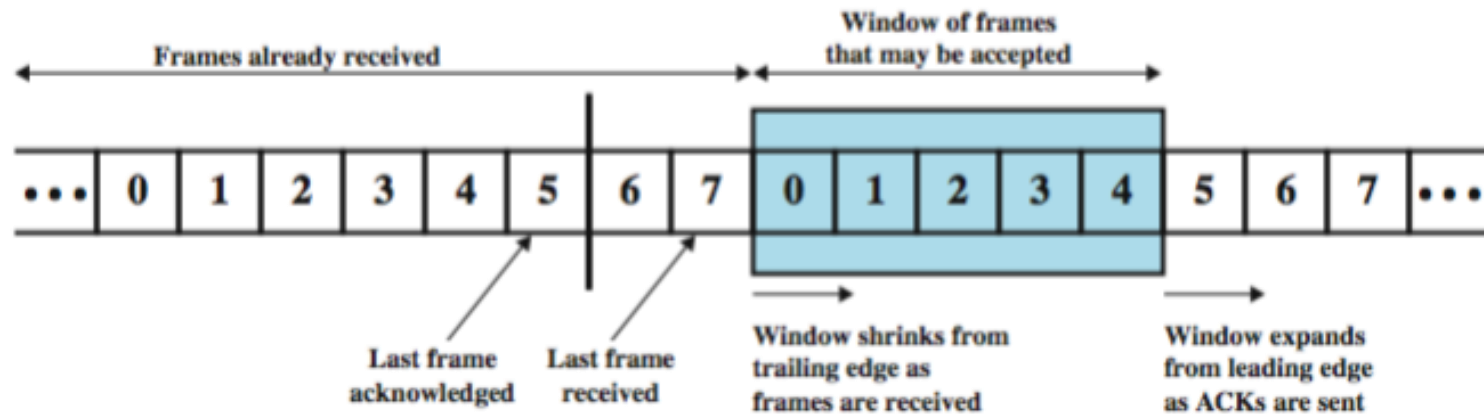
# Điều khiển dòng theo cửa sổ trượt (Sliding windows)

- Cho phép nhiều frame có thể truyền đồng thời
- Bên thu có bộ đệm với kích thước  $W$
- Bên phát có thể truyền tối đa  $W$  frame mà không cần đợi ACK
- Cơ chế đánh số thứ tự cho các frame
- ACK có chứa số của frame kế tiếp đang được mong đợi
- Số thứ tự được quay vòng bởi kích thước cửa sổ (modulo  $2^k$ )

# Sơ đồ cửa sổ trượt



(a) Sender's perspective

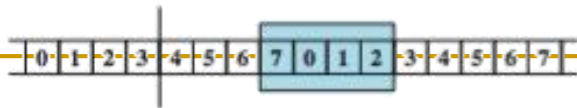
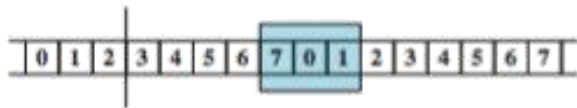
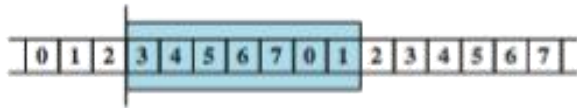
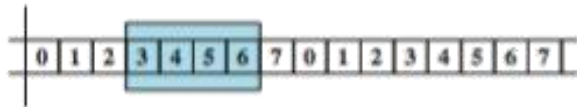
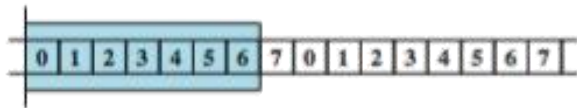


(b) Receiver's perspective

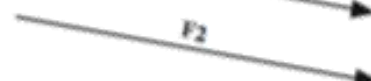
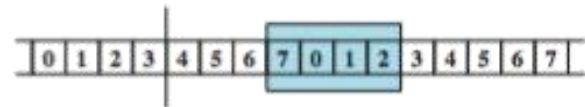
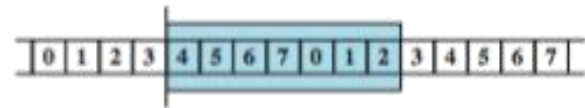
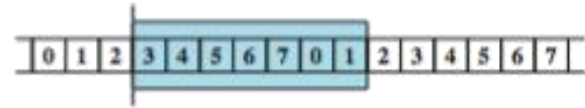
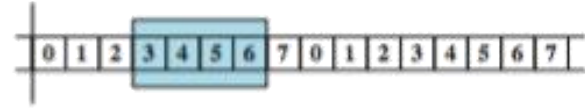
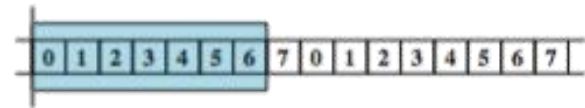


# Ví dụ cửa sổ trượt

Source System A



Destination System B



# Cửa sổ trượt cải tiến

- Máy nhận có thể công nhận các gói tin đồng thời không cho phép truyền tiếp (Receive Not Ready)
- Cần phải gửi ACK thông thường khi muốn tiếp tục
- Trong trường hợp song công sử dụng kiểu đánh khẳng
  - Nếu không có dữ liệu cần truyền gửi ACK
  - Nếu có dữ liệu mà không cần gửi ACK thì tiếp tục gửi số ACK cũ

---

# Phát hiện lỗi

- Thêm các bit để có thể phát hiện ra lỗi trên đường truyền
  - Sử dụng Parity
  - Sử dụng CRC
-

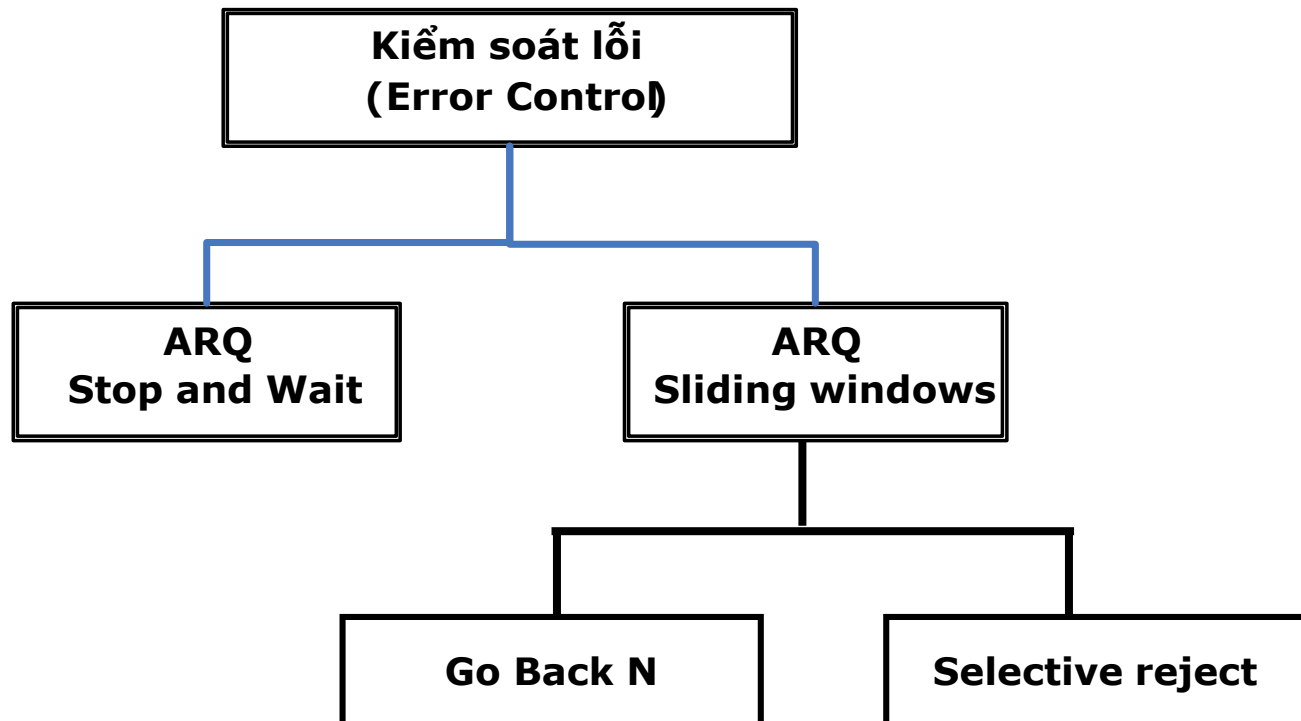
# Kiểm soát lỗi

- Bảo đảm dữ liệu nhận được đúng và chính xác
- Mất frame: frame không đến đích
- Frame sai: dữ liệu trong frame bị sai
- Cung cấp cơ chế cho việc truyền dữ liệu trong trường hợp dữ liệu bị mất hay sai sót trên đường truyền
  - Positive ACK – xác nhận các frame nhận được
  - Truyền lại sau một thời gian time-out
  - Negative ACK (NAK) và truyền lại – yêu cầu truyền lại (NAK) cho các frame bị hư

# Tự động thực hiện lại

## Automatic Repeat Request

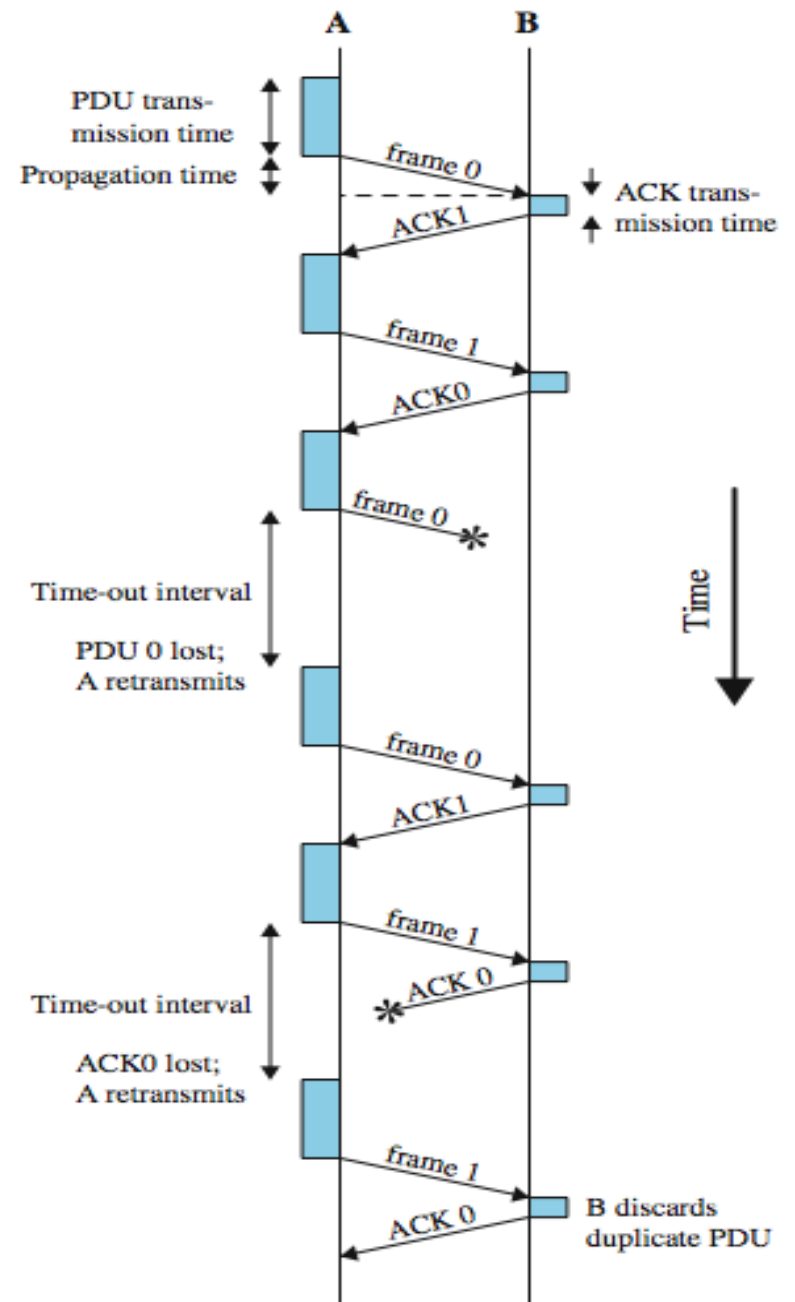
- Cơ chế cho phép các giao thức liên kết dữ liệu quản lý lỗi và yêu cầu truyền lại



# Stop and Wait

- Máy gửi gửi một gói tin đến máy nhận
- Máy gửi đợi trả lời
- Nếu gói tin bị hỏng thì sẽ gửi lại
  - Máy gửi có định thời gian
  - Không nhận được trả lời quá thời gian – Máy gửi gửi lại
- Nếu gói tin nhận được, nhưng ACK bị mất/hư?
  - Máy gửi gửi lại
  - Máy nhận sẽ nhận được 2 gói tin giống nhau
  - Sử dụng đánh số 0 và 1

# Stop and Wait



# Stop and Wait – ưu khuyết điểm

- Đơn giản
- Không hiệu quả



# Go-back-N

- Frame điều khiển
  - RR - receive ready = ACK - acknowledge
  - REJ - reply with rejection = NAK - negative acknowledge
- Dựa trên cơ chế sliding window
  - Máy gửi truyền liên tục các Frame đến máy nhận (trong khi cơ chế điều khiển dòng còn cho phép)
  - Máy nhận chỉ nhận Frame theo đúng chỉ số tuần tự (hoặc ) và gửi RR với số hiệu của Frame đang chờ nhận
- Khi có lỗi,
  - Máy nhận sẽ yêu cầu gửi lại và loại bỏ các frame tiếp theo đến khi nhận được sửa đổi
  - Máy gửi truyền lại tất cả các Frame sai kể từ Frame sai đầu tiên trở đi, bất kể các Frame sau là đúng hay sai

---

# Frame hỏng

- Máy nhận phát hiện lỗi trong Frame thứ  $I$
  - Máy nhận truyền Frame  $REJ(i)$
  - Máy gửi nhận được Frame  $REJ(i)$
  - Máy gửi truyền lại Frame thứ  $I$  và các Frame tiếp theo
-

# Frame mất (1)

- Frame  $i$  bị mất
- Máy gửi đã truyền Frame  $i+1$
- Máy nhận nhận được Frame  $i+1$  ngoài thứ tự
- Máy nhận truyền Frame REJ ( $i$ )
- Máy gửi quay trở lại Frame thứ  $i$  và gửi lại

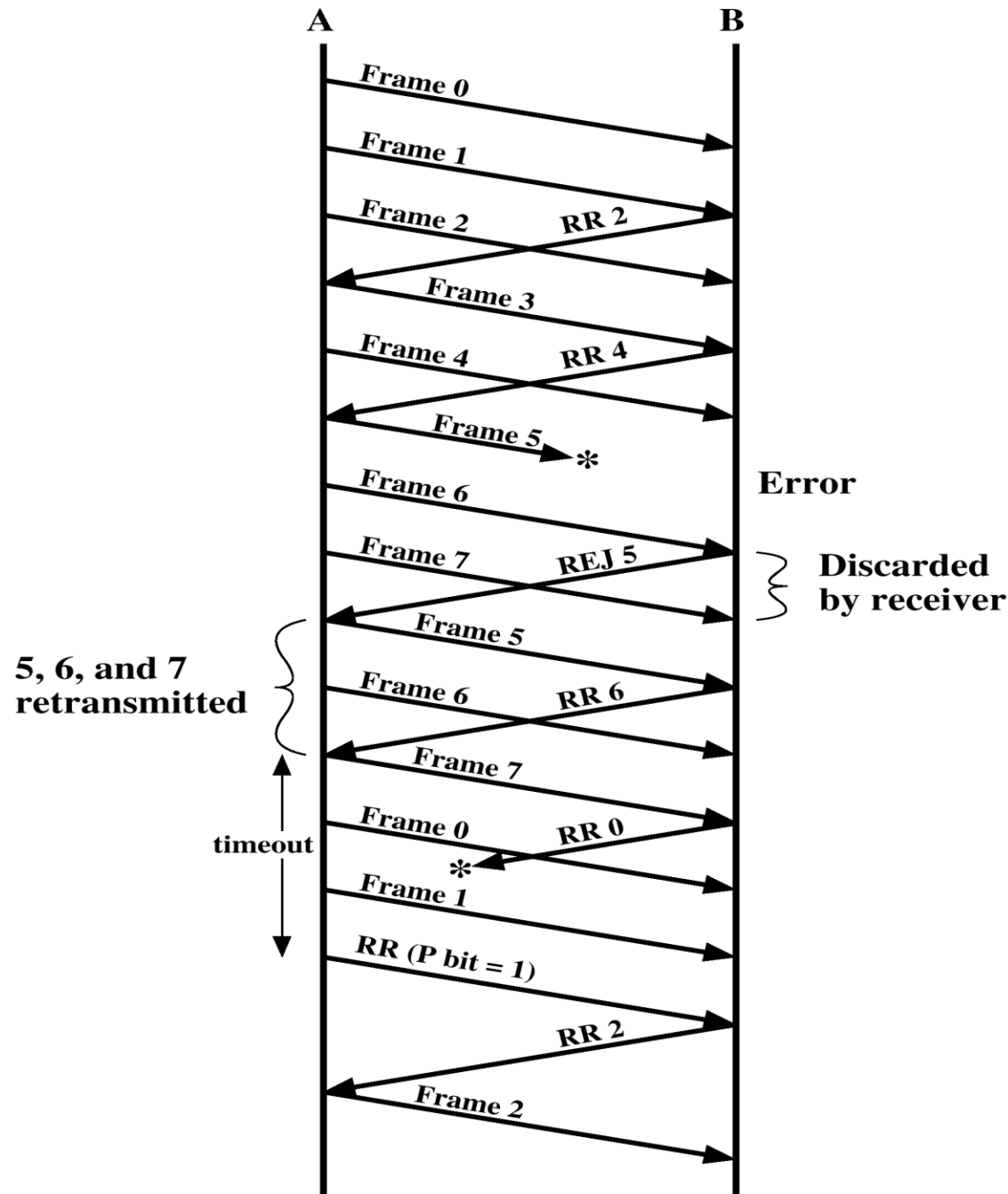
## Frame mất (2)

- Frame  $i$  mất và không có frame nào được gửi tiếp
- Máy nhận không nhận được gì và không trả lời
- Máy gửi đợi hết thời gian sẽ gửi ACK Frame với bit  $P$  cho bằng 1.
- Máy nhận truyền  $RR(i)$
- Máy gửi truyền lại Frame  $i$

# Frame ACK bị hư

- Máy nhận nhận được Frame  $i$ , máy nhận truyền Frame  $RR(i+1)$ , nhưng Frame này bị mất
- Máy gửi còn gửi các Frame tiếp theo như  $i+1, i+2..$ . Nên nếu máy gửi nhận được Frame  $RR(i+n)$  trước thời gian timeout thì được hiểu như bao gồm  $RR(i), RR(i+1) ..$
- Máy nhận đợi, sẽ gửi lại các Frame, kể từ Frame  $i$
- Máy nhận được Frame  $REJ(i)$  báo thiếu hay hay hư thì sẽ gửi lại các Frame, kể từ Frame  $i$

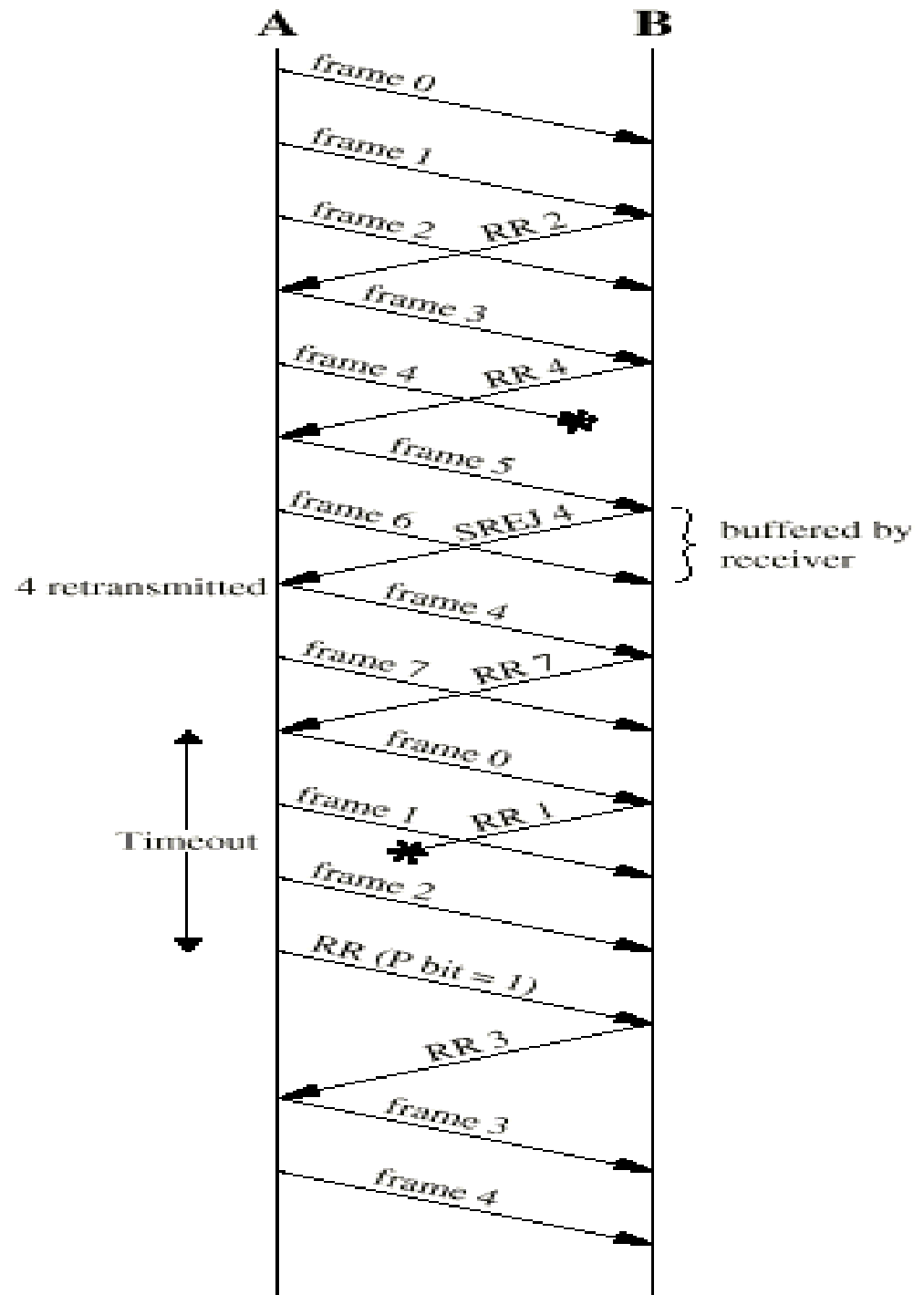
# Go-back-N



# Selective Reject

- Tương tự như Go-Back-N,
- Chỉ truyền lại các Frame bị hỏng hoặc time-out
- Máy nhận có thể nhận Frame không theo đúng tuần tự và máy nhận phải có buffer để lưu lại các Frame đến không theo đúng chỉ số tuần tự
- Giảm số lượng cần truyền lại
- Buffer cần phải đủ lớn
- Phức tạp hơn

# Selective Reject





# High-level Data Link Control – HDLC

- Giao thức điều khiển liên kết dữ liệu cấp cao (High-level Data Link Control – HDLC)
- Liên kết point-to-point hoặc multipoint
- Đặc điểm
  - Không phụ thuộc mã điều khiển
  - Khả năng thích ứng
  - Hiệu quả cao
  - Độ tin cậy cao

# Các loại trạm làm việc HDLC

- Trạm cấp 1
  - Điều khiển hoạt động của liên kết
  - Frame phát ra gọi là các lệnh
  - Duy trì liên kết logic riêng cho các trạm cấp 2
- Trạm cấp 2
  - Hoạt động dưới sự điều khiển của trạm cấp 1
  - Frame phát ra gọi là các trả lời
- Trạm tổ hợp
  - Có thể phát ra các lệnh và trả lời

# Cấu hình liên kết

- Không cân bằng
  - Một trạm cấp 1 và một hoặc nhiều trạm cấp 2
  - Cho phép full duplex và half duplex
- Cân bằng
  - 2 trạm tổ hợp
  - Hỗ trợ full duplex và half duplex

# Giao thức HDLC – chế độ truyền (1)

- Normal Response Mode (NRM)
  - ❑ Cấu hình không cân bằng
  - ❑ Trạm cấp 1 khởi động việc truyền đến trạm cấp 2
  - ❑ Trạm cấp 2 chỉ có thể truyền dữ liệu để đáp ứng lại lệnh từ trạm cấp 1
  - ❑ Dòng đường truyền dạng multi-drop
  - ❑ Máy chủ hoạt động như là trạm cấp 1
  - ❑ Các terminal hoạt động như là trạm cấp 2

# Giao thức HDLC – chế độ truyền (2)

- Asynchronous Balanced Mode (ABM)
    - Cấu hình cân bằng
    - Trạm nào cũng có thể bắt đầu truyền mà không cần sự cho phép
    - Được dùng rộng rãi
    - Không tốn chi phí cho việc polling
-

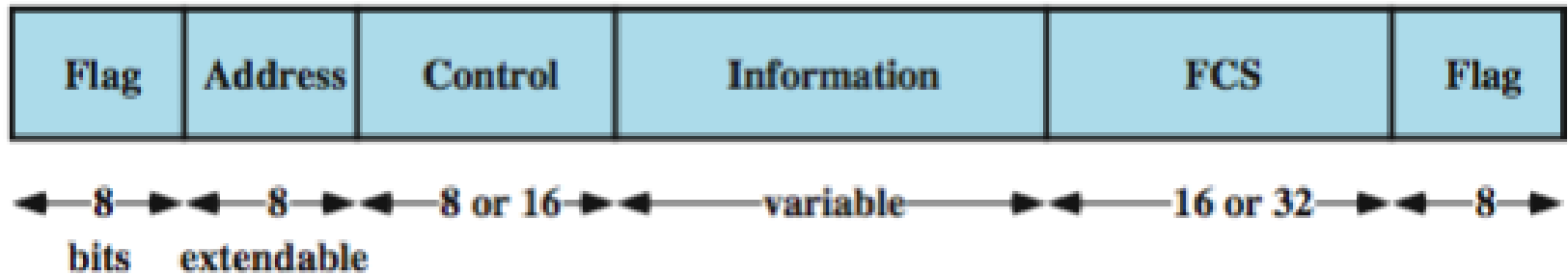
# Giao thức HDLC – chế độ truyền (3)

- Asynchronous Response Mode (ARM)
  - Cấu hình không cân bằng
  - Trạm cấp 2 có thể bắt đầu truyền mà không cần sự cho phép của trạm cấp 1
  - Trạm cấp 1 chịu trách nhiệm cho đường truyền
  - Ít dùng

# Cấu trúc Frame của HDLC

- Truyền đồng bộ
- Truyền theo Frame
- Định dạng Frame chung cho việc trao đổi dữ liệu và điều khiển

# Cấu trúc Frame



(a) Frame format



# Nghi thức HDLC – cờ điều khiển

- Dùng để phân định Frame cả ở 2 đầu
- 01111110
- Có thể dùng để kết thúc Frame này và bắt đầu Frame khác
- Máy nhận quét tìm cờ để đồng bộ
- Chèn thêm bit (bit stuffing) được dùng để tránh lẫn lộn dữ liệu chứa 01111110
  - 0 được chèn thêm vào mỗi khi chuỗi 5 số 1 liên tiếp xuất hiện
  - Nếu máy nhận phát hiện 5 số 1, nó kiểm tra bit kế tiếp
  - Nếu bit đó là 0, nó xóa bit 0 đó
  - Nếu bit là 1 và bit thứ 7 là 0, nó biết đây là cờ
  - Nếu bit thứ 6 và 7 đều là 1, bộ phát ra lệnh hủy bỏ

# Chèn thêm bit

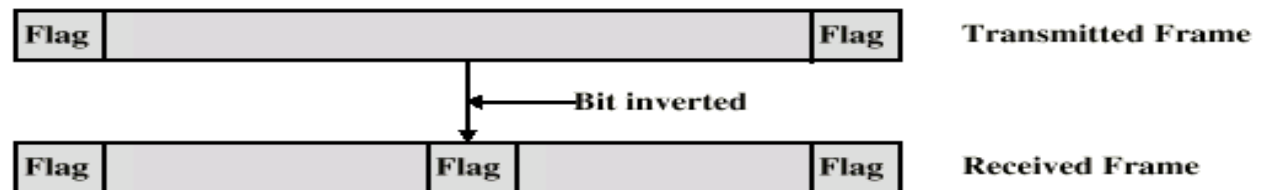
**Original Pattern:**

1111111111111011111101111110

**After bit-stuffing**

1111101111101101111101011111010

(a) Example



(b) An inverted bit splits a frame in two

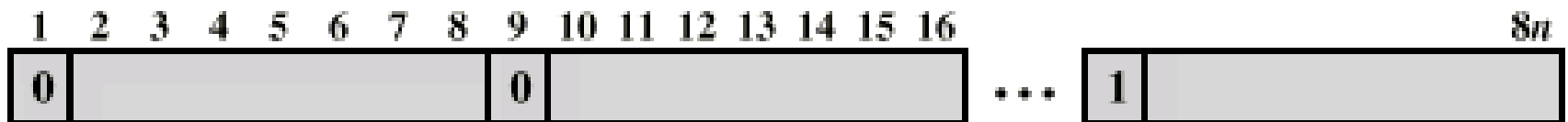


(c) An inverted bit merges two frames

- Ví dụ các lỗi có thể

# Trường địa chỉ

- Dùng để nhận diện trạm cấp 2 đã gửi hoặc sẽ nhận Frame
- Thường dài 8 bit
- Có thể mở rộng thành bội số của 7 bit
  - LSB của mỗi octet chỉ thị đây là octet cuối cùng (1) hay chưa (0)
- Địa chỉ toàn 1 (11111111) là địa chỉ broadcast

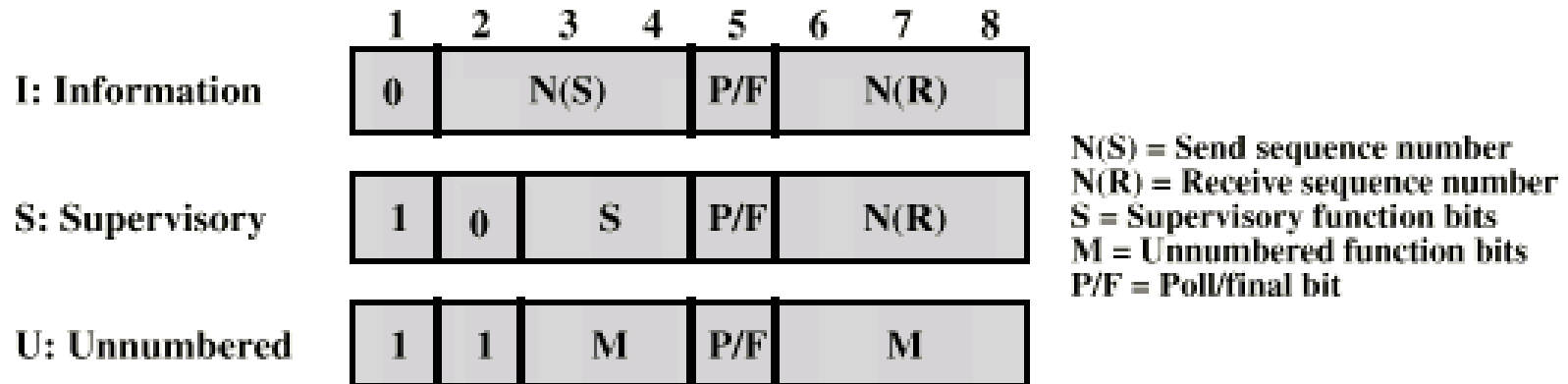


(b) Extended Address Field

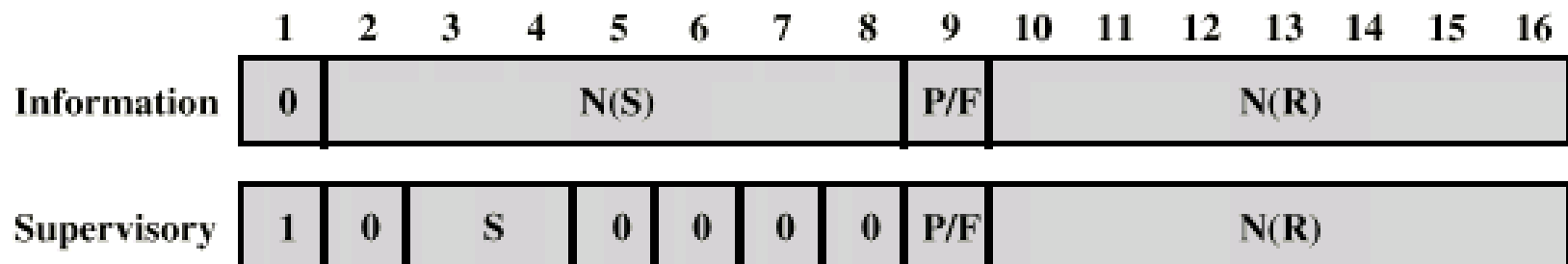
# Trường điều khiển

- Khác nhau tùy thuộc vào loại Frame
  - Thông tin – dữ liệu cần truyền đến người dùng (lớp trên)
    - Điều khiển dòng và điều khiển lỗi được gửi kèm (piggybacked) trong các khung thông tin
  - Giám sát – dùng ARQ khi piggyback không được dùng
  - Không số – hỗ trợ cho việc điều khiển liên kết
- 1 hoặc 2 bit đầu tiên của trường điều khiển dùng để nhận dạng loại Frame

# Trường điều khiển



(c) 8-bit control field format



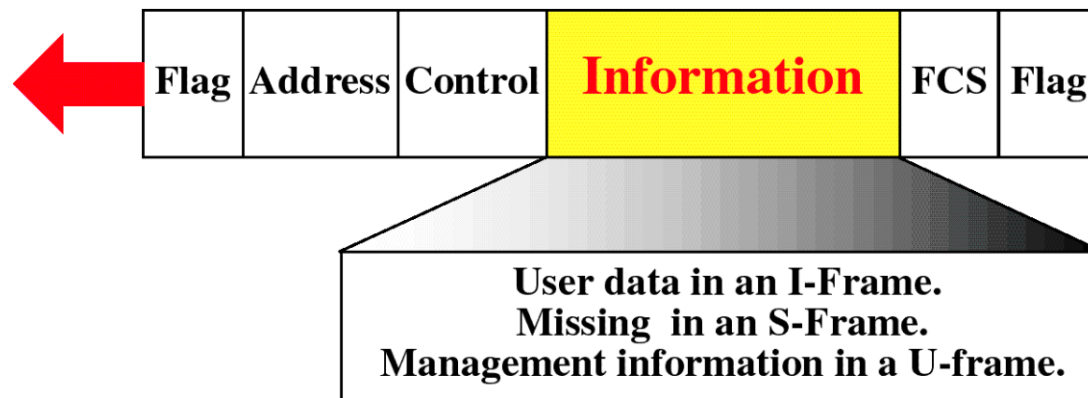
(d) 16-bit control field format

# Bit Poll/Final

- Dừng tùy theo ngữ cảnh
- Frame lệnh
  - Bit P
  - 1 để mời gọi (poll) đáp ứng của các trạm ngang hàng
- Frame trả lời
  - Bit F
  - 1 để chỉ thị đáp ứng đối với lệnh mời gọi

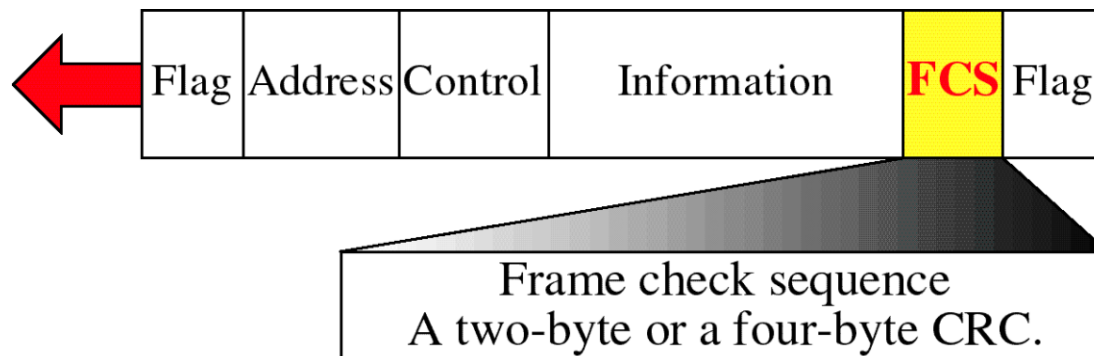
# Trường thông tin

- Chỉ có trong Frame thông tin và một số Frame không số
- Cần bao gồm một nhóm các octet
- Kích thước thay đổi



# Trường FCS - Frame Check Sequence Field

- Dùng để phát hiện lỗi
- CRC 16 bit
- Có thể dùng CRC 32 bit

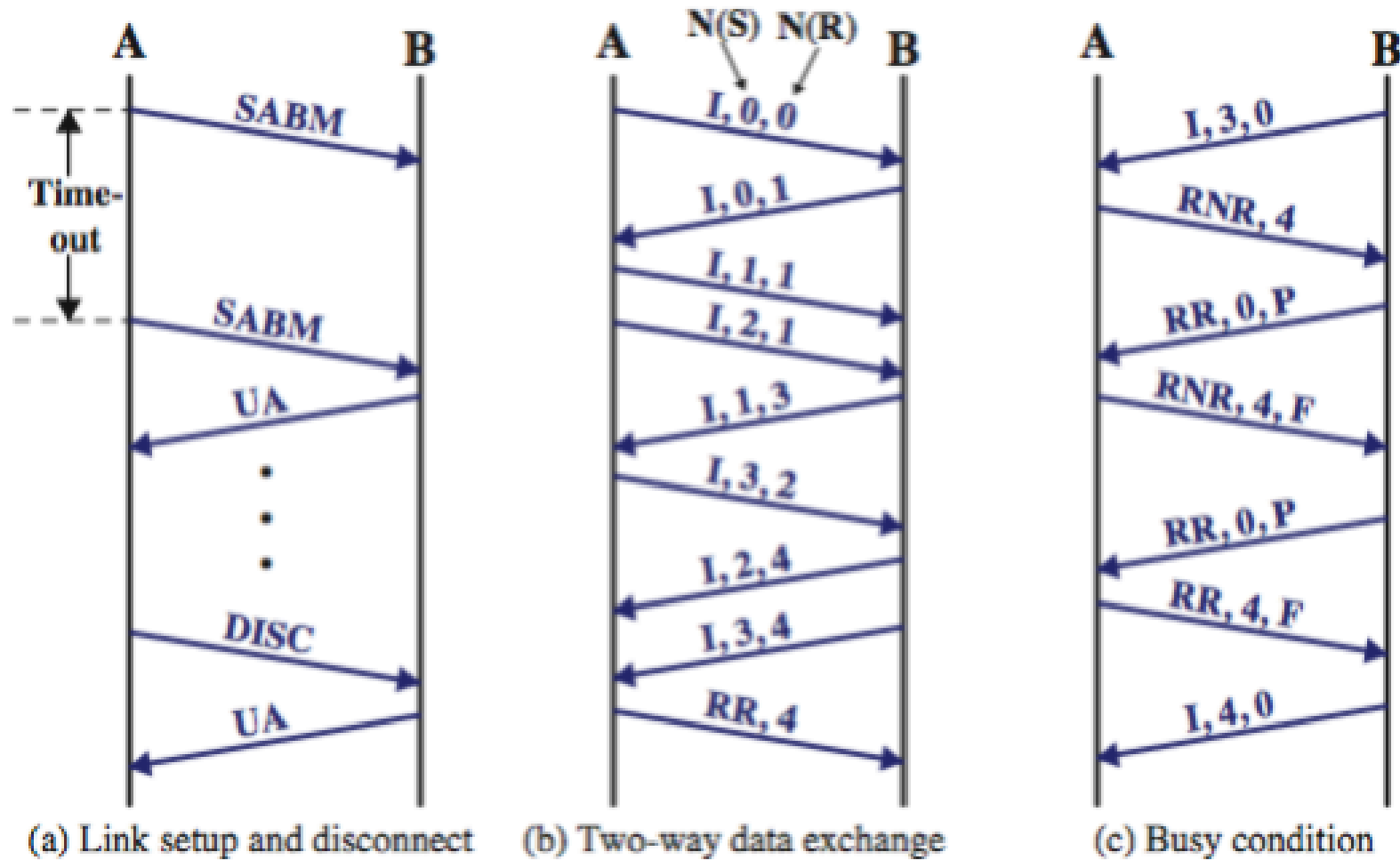




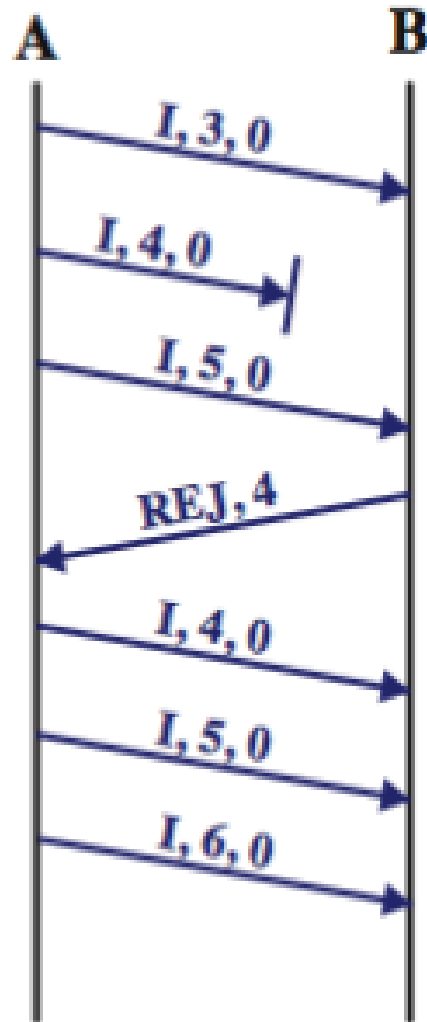
# Hoạt động của giao thức HDLC

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

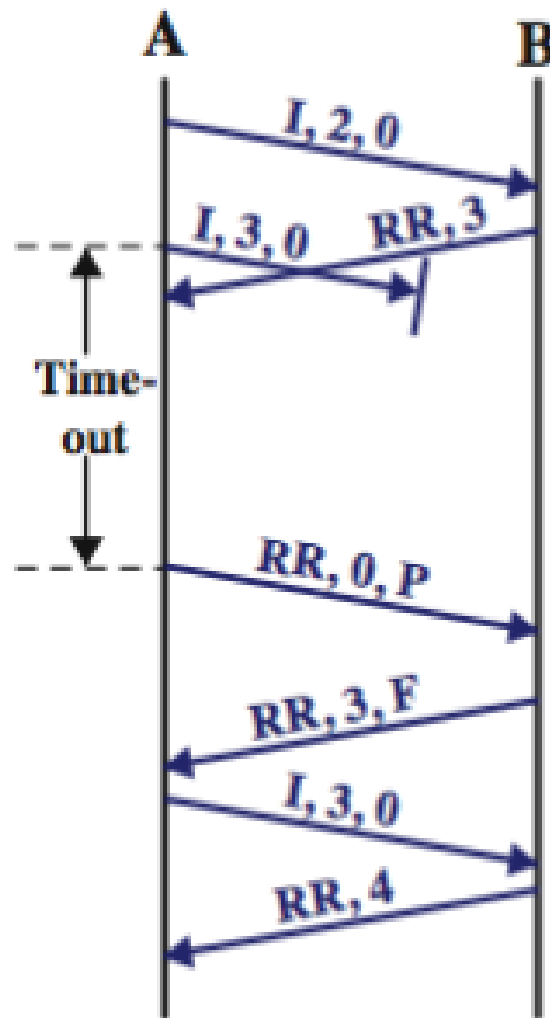
# Ví dụ hoạt động



## Ví dụ hoạt động (2)



(d) Reject recovery



(e) Timeout recovery

---

HẾT CHƯƠNG 5

---