


# Chapter 11

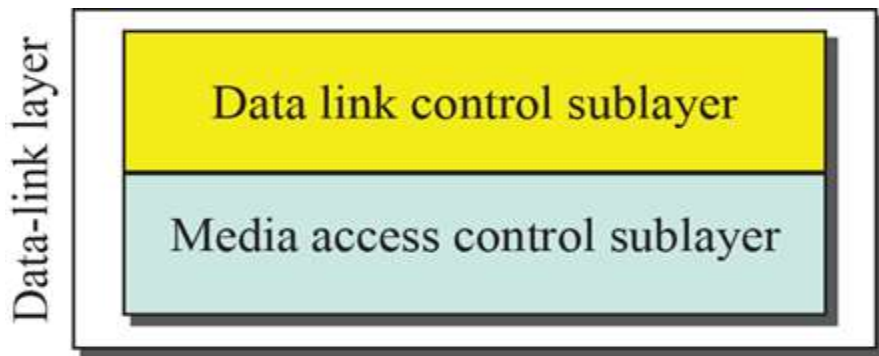
## 데이터링크 제어

### (DLC: Data Link Control)

The slide features two horizontal decorative bars at the bottom. The top bar is a solid dark green, and the bottom bar is a solid olive green.

# 데이터 링크 층

## ■ 데이터 링크 층의 가장 중요한 2가지 (부계층)



### ✓ Data Link Control (DLC) 부계층

← 이번 장에서 공부

- §1. 회선제어
- §2. 프레임 짜기(Framing)
- §3. 흐름 제어(Flow Control)
- §4. 오류 제어(Error Control)

### ✓ Media Access Control (MAC) 부계층

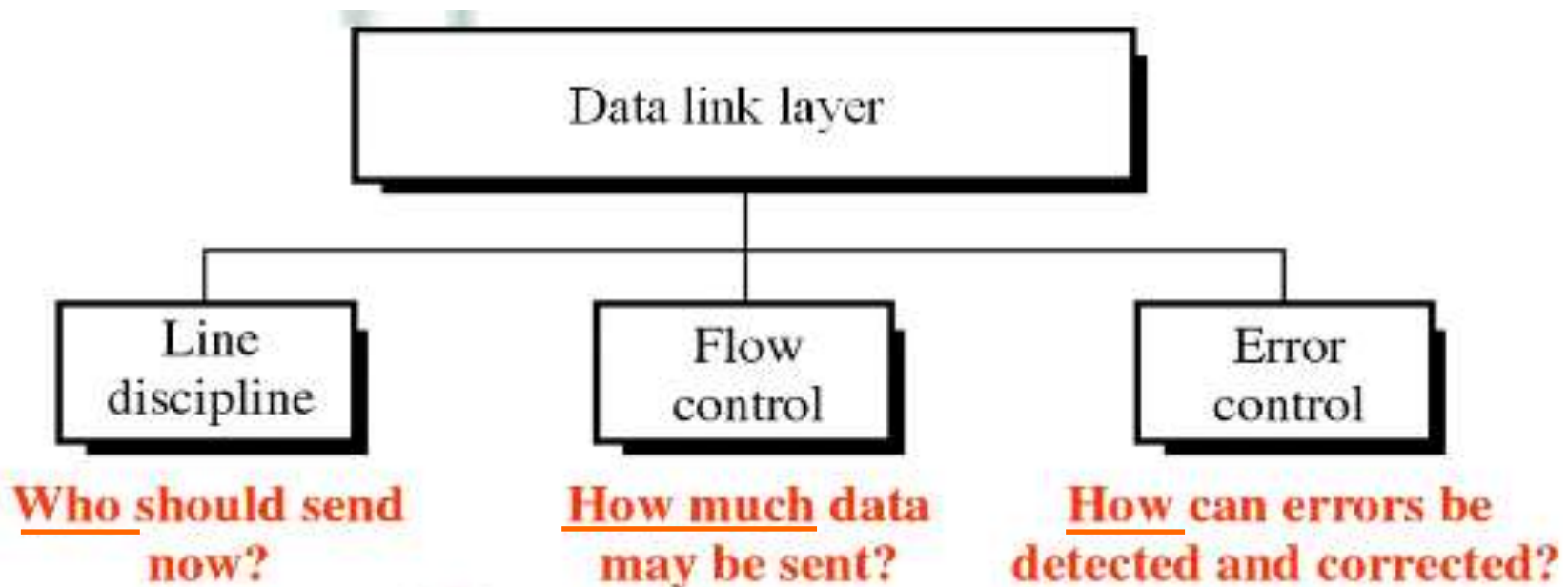
← 12장에서 공부

- 공유된 미디어로 접속하도록 하는 책임

# 데이터 링크 제어(Data Link Control)

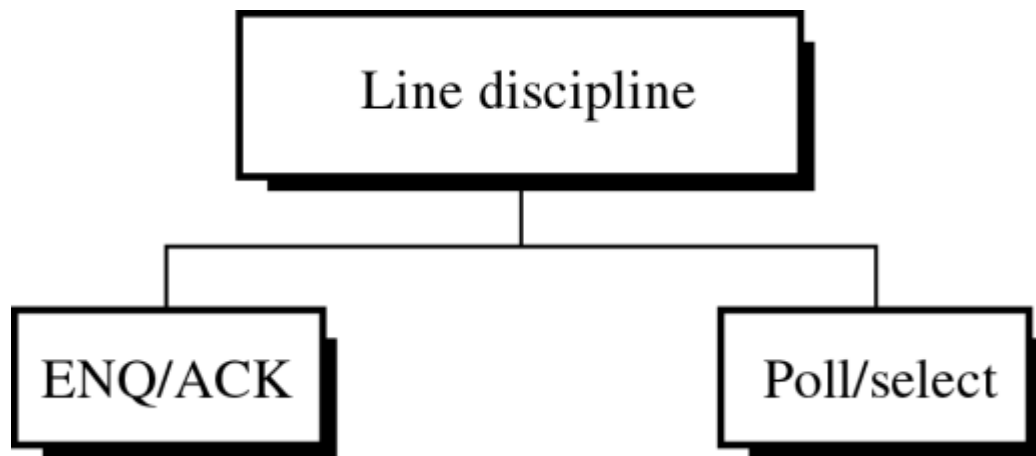
## ■ Data Link Control (DLC)

- ✓ 정보의 순서적 전송을 확실히 하기 위해 서로 연결된 컴퓨터와 단말기에 따른 제어
- ✓ 데이터 링크층에서 수행되는 기능
- ✓ 회선 제어(Line discipline), 프레임 짜기(Framing), 흐름 제어(Flow Control), 오류 제어(Error Control)



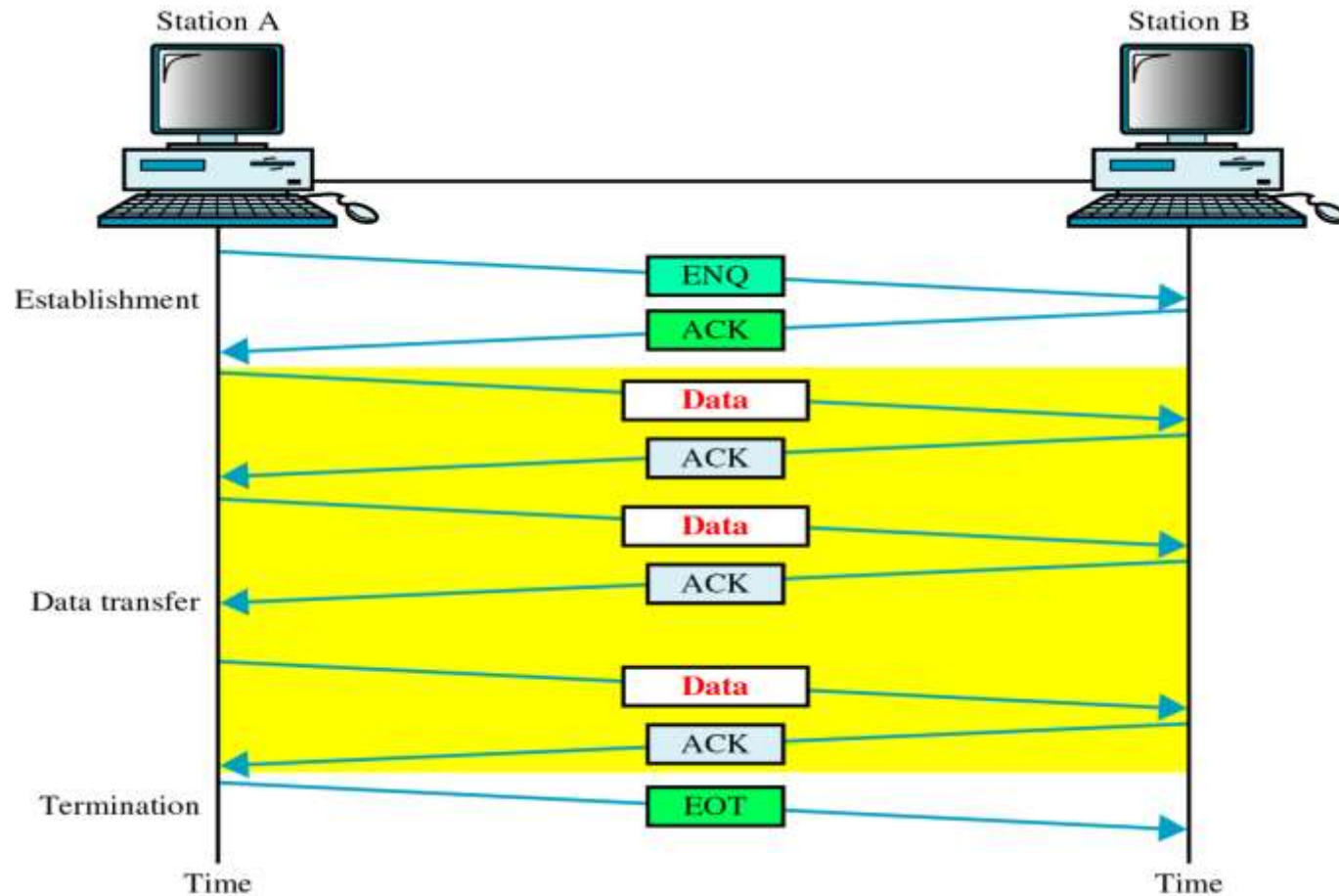
# § 1. 회선 제어

- 지금 누가 전송해야 하는가의 질문에 대한 응답
- 2가지 방법
  - ① ENQ/ACK (Enquiry/Acknowledgment): peer-to-peer 통신
  - ② poll/select : primary-secondary 통신



# ENQ/ACK

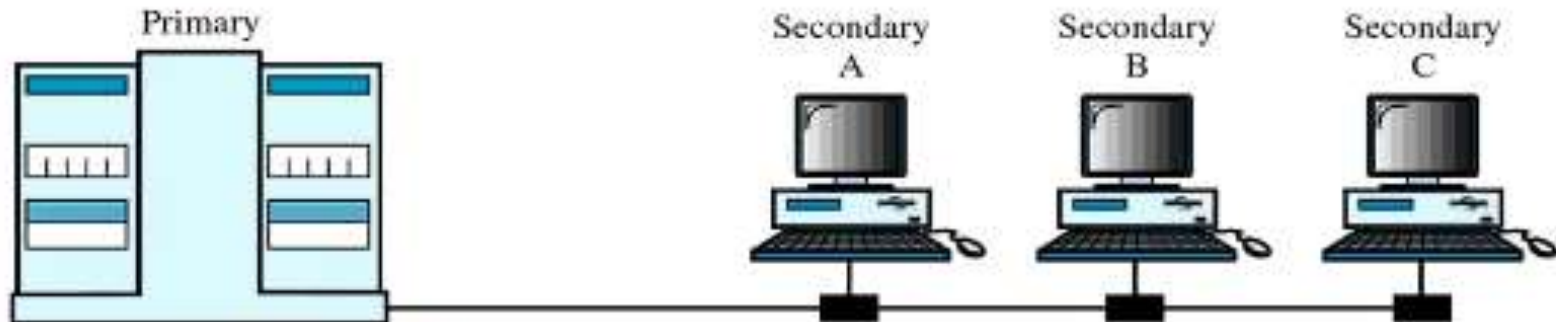
Who should start?  
How can one station be sure that  
the other is ready?



# Polling / Selection

- 회선 하나를 여러 개의 터미널이 공유하는 형태인 Multidrop(multipoint)에서는 한번에 하나의 터미널만이 메시지 보낼 수 있다.
  - 어떻게 하나만 선택하며 어떻게 해결하는가?  
Polling / Selection 방식 (현재 많이 사용 안함)

Who has the right to the channel?

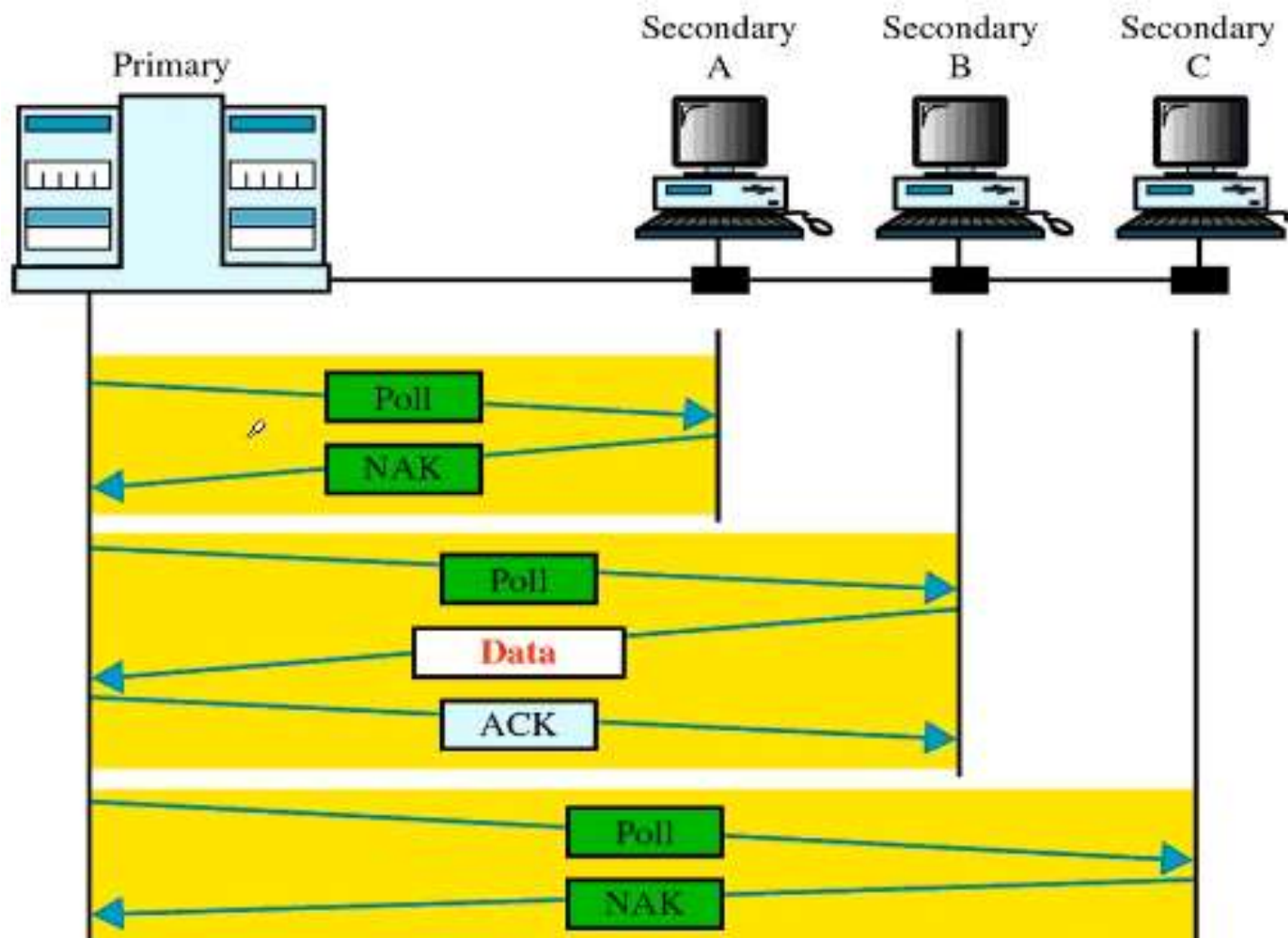


# Polling / Selection

---

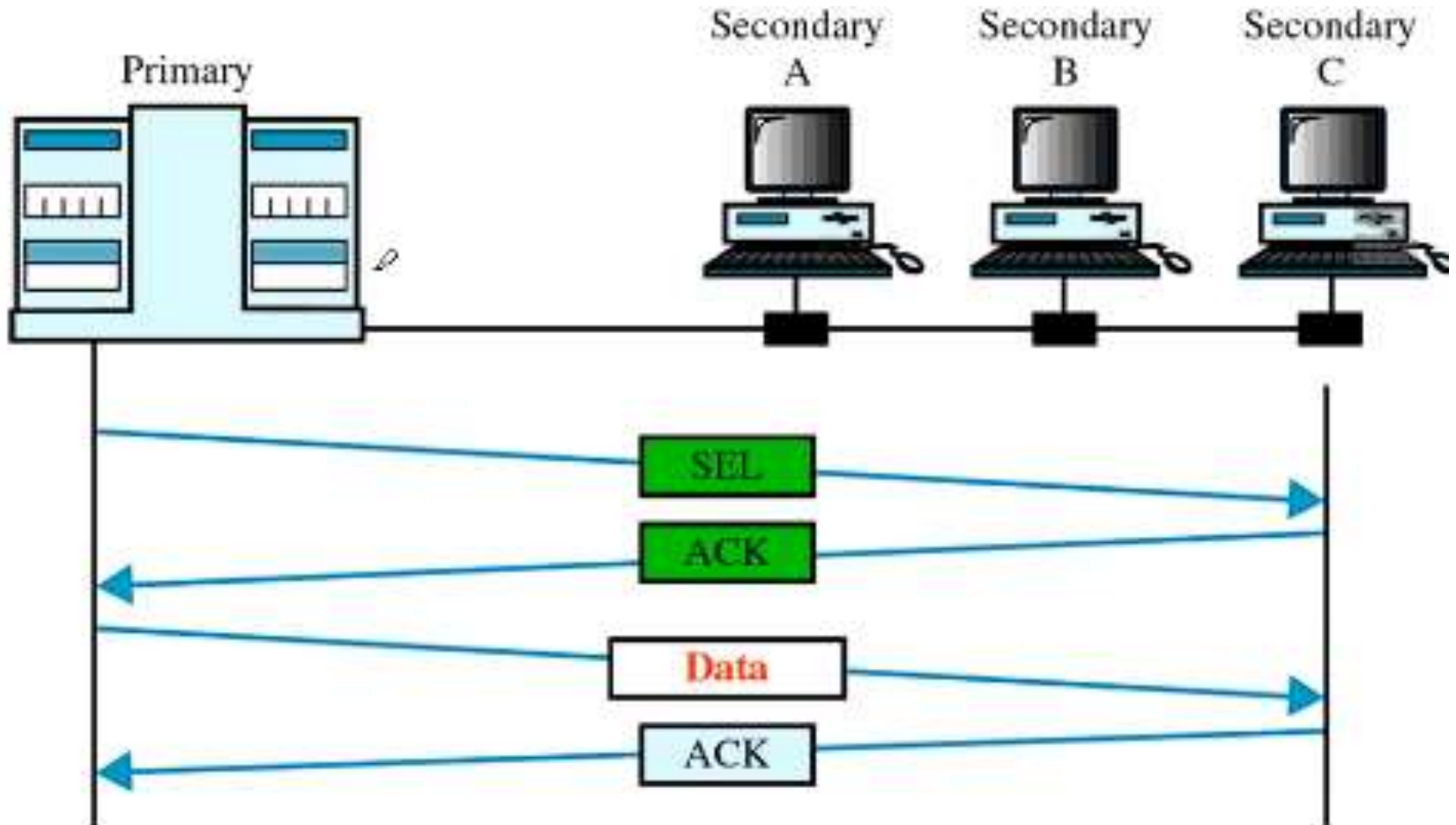
- **Polling** is the function of asking the secondaries if they have anything to send.
  - ✓ Roll-call polling
  - ✓ Hub polling
- **Select** is used whenever the primary device has something to send.

# Polling 예





# Selection 예



## § 2. 프레임 짜기(Framing)

---

- 메시지를 작은 단위의 비트들로 나누어 프레임을 구성
  - ✓ 메시지에 대한 발신지와 목적지를 구분
    - 송신자와 수신자 주소를 포함
  - ✓ 목적지주소는 패킷이 가야할 곳을 규정하여 송신자는 수신자로 하여금 받았다는 것을 응답할 수 있도록 도움
  - ✓ 흐름제어와 오류제어에 있어 효율성 증대

# 프레임 크기

## ■ 일정 크기가 될 수도 가변 크기가 될 수도 있음

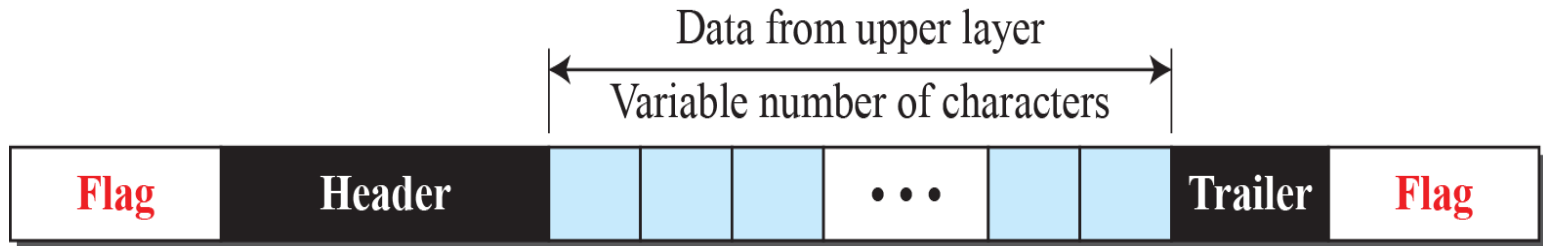
### ✓ 고정 크기 프레임(Fixed-size framing)

- 프레임 경계 필요 없이 크기 자체가 경계 역할을 함
  - 예: ATM 망에서는 cell 이라는 고정 크기의 프레임 사용

### ✓ 가변 크기 프레임(Variable-size framing)

- 프레임이 끝나는 곳과 다음 프레임이 시작하는 곳을 지정함
- 문자 중심 프로토콜 (다음 슬라이드)
  - 전달되는 데이터는 부호화 시스템의 8비트 문자
  - 시작과 마지막에 플래그(8bits) 추가
- 비트 중심 프로토콜 (슬라이드14)
  - 프레임의 데이터 부분을 전부 bit 열로 인식
  - 플래그: 프레임 경계로 비트열 0111 1110을 사용  
(프레임 내부의 같은 비트 패턴은 비트 채우기(bit stuffing)으로 처리하여 송신함)

# ① 문자중심 프로토콜의 프레임

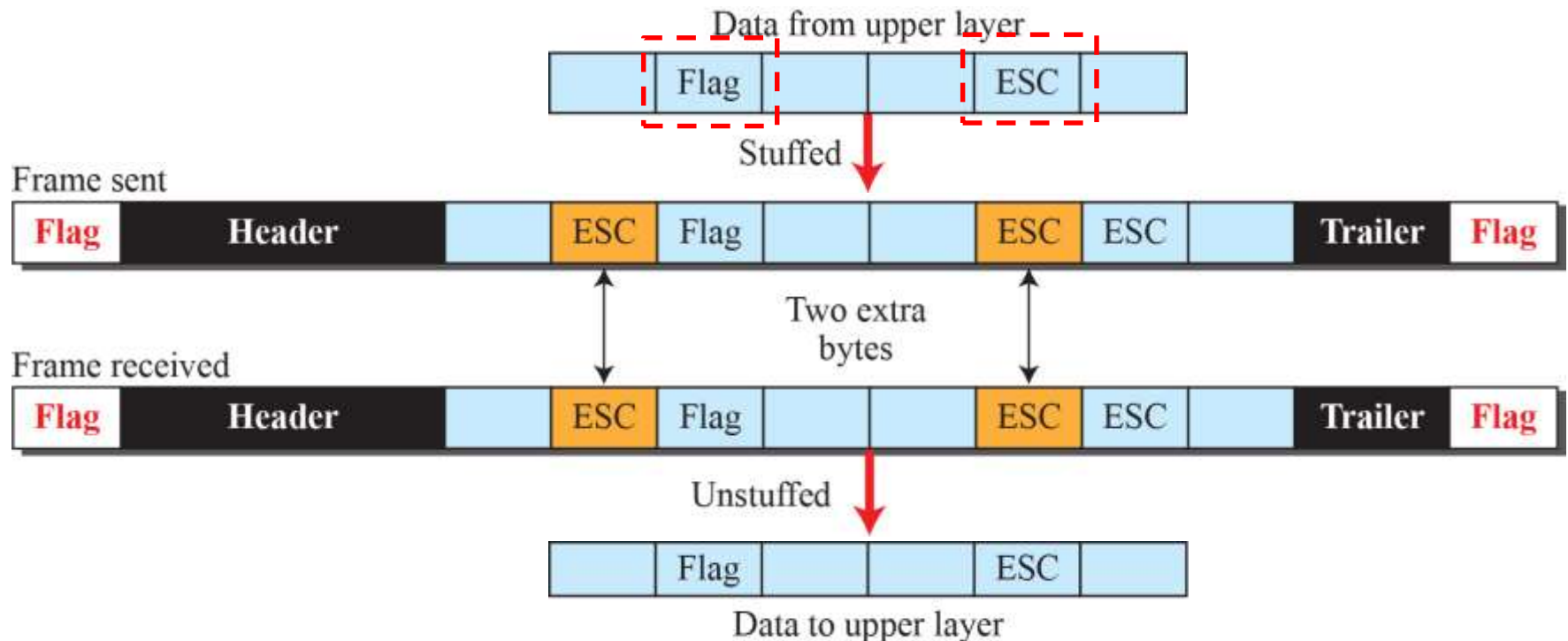


# Byte stuffing

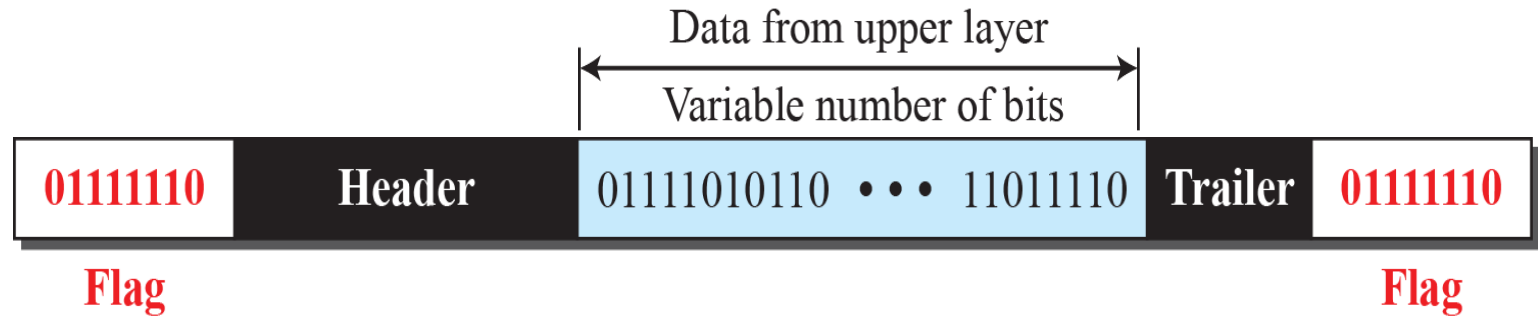
문자(바이트) 채우기

## ■ 바이트 채우기

- ✓ 데이터에 Flag 문자나 ESC 문자 있으면, 여분의 1문자(바이트) 즉, 추가문자(ESC문자)를 넣는다.



## ② 비트중심 프로토콜의 프레임

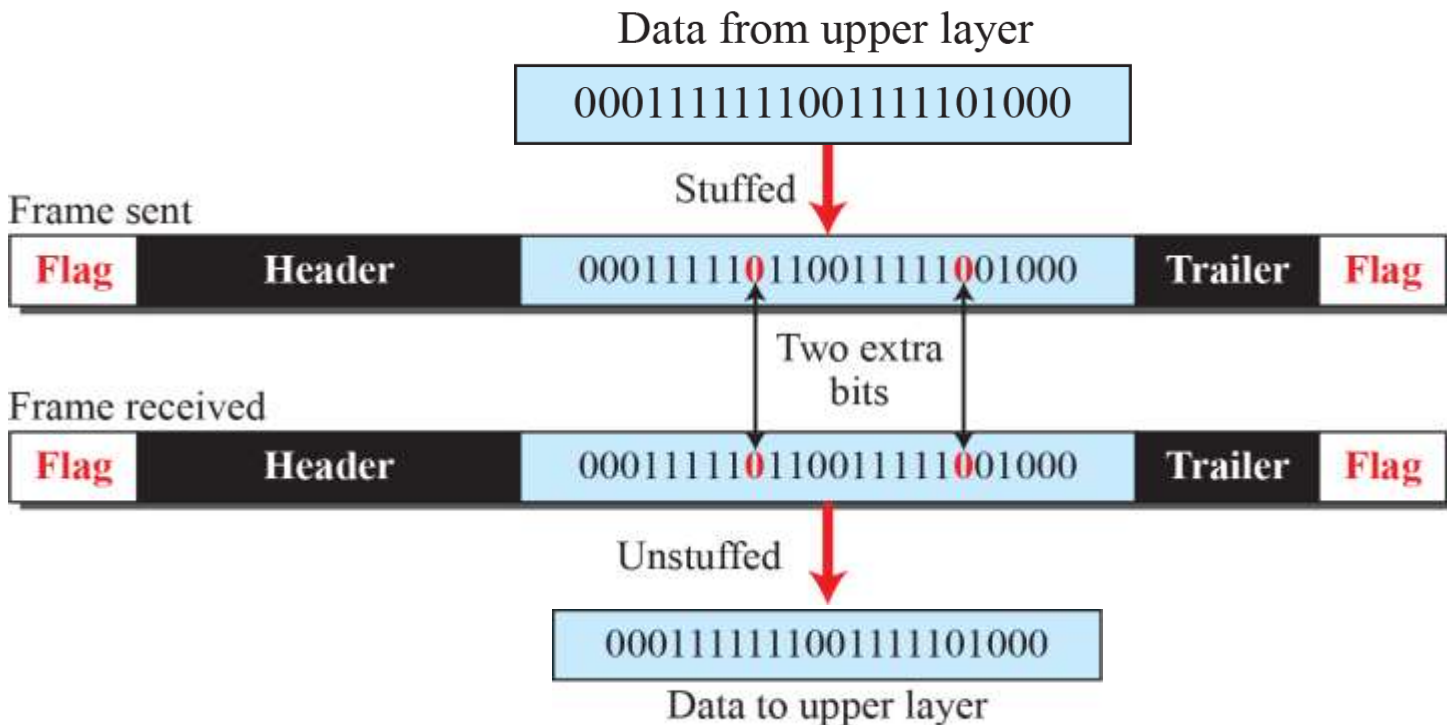


- 플래그 필드의 패턴: 0111 1110
- 앞의 문자 중심 프로토콜의 프레임과 같은 문제가 생길 수 있음. (즉, 플래그 패턴이 데이터 속에 나타나는 경우)  
→ 해결책? 비트 채우기

# Bit stuffing

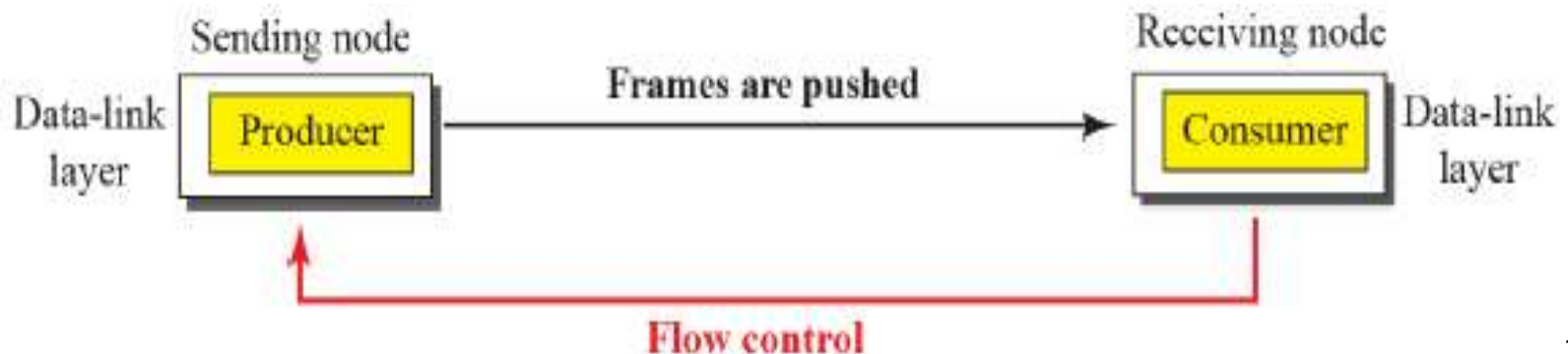
비트 채우기

- 데이터에 플래그 필드의 패턴 (즉, 0111 1110)과 같은 것이 포함되는 경우, 플래그 필드와 오인하지 않기 위해
- 데이터 중에서 1 이 연속으로 5번 있는 곳의 뒤에 0을 추가함



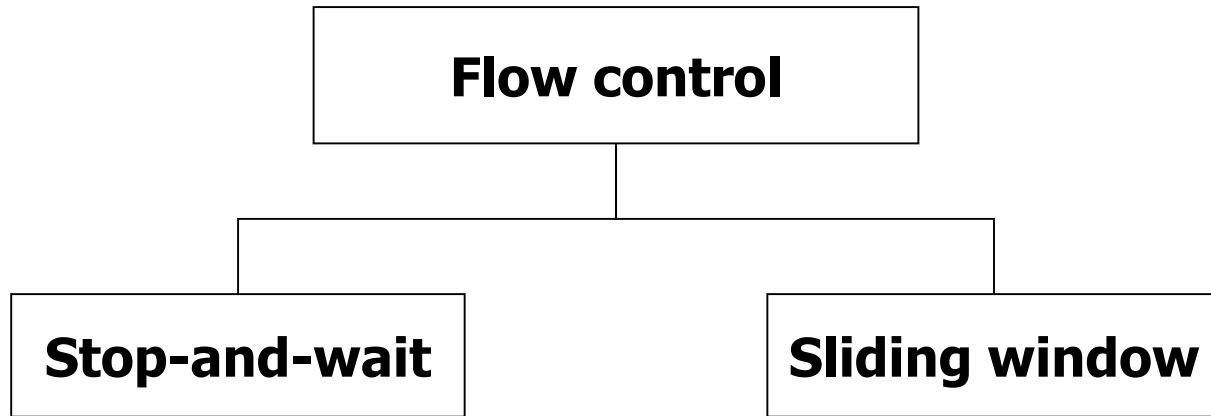
## § 3. 흐름제어(Flow Control)

- 송신기가 수신기로부터 확인응답(ACK)을 받기 전에 송신기가 보낼 수 있는 데이터 양을 제한하는 절차
  - ✓ ACK (Acknowledgment)
    - 수신기가 송신기에게 정상 수신 여부를 알리는 메시지
  - ✓ 수신기는 한계에 도달하기 전에 송신기에게 프레임을 적게 보내게 하거나 일시적으로 멈추도록 알려야 함
  - ✓ **처리율 < 전송율** 일때 → 버퍼에 저장하여 흐름제어





# 흐름제어 방식



**Send one frame at a time**

**Send several frames at a time**

## 1) Stop and Wait(정지/대기) 방식

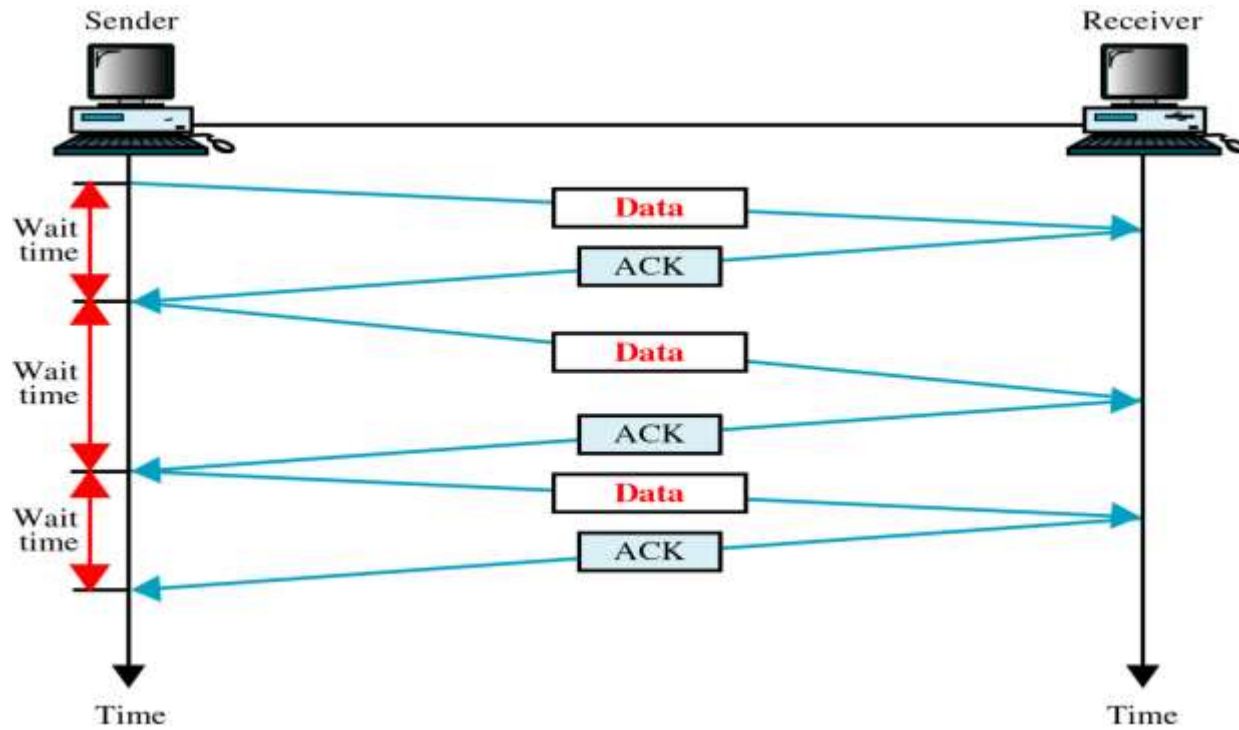
- ✓ 송신자는 하나의 프레임을 전송하고 다음 프레임을 전달하기 전에 확인응답을 기다림

## 2) Sliding Window (슬라이딩 윈도우) 방식

- ✓ 확인응답을 받기 전에 여러 개의 프레임을 전송할 수 있다

# 1) Stop and Wait 방식

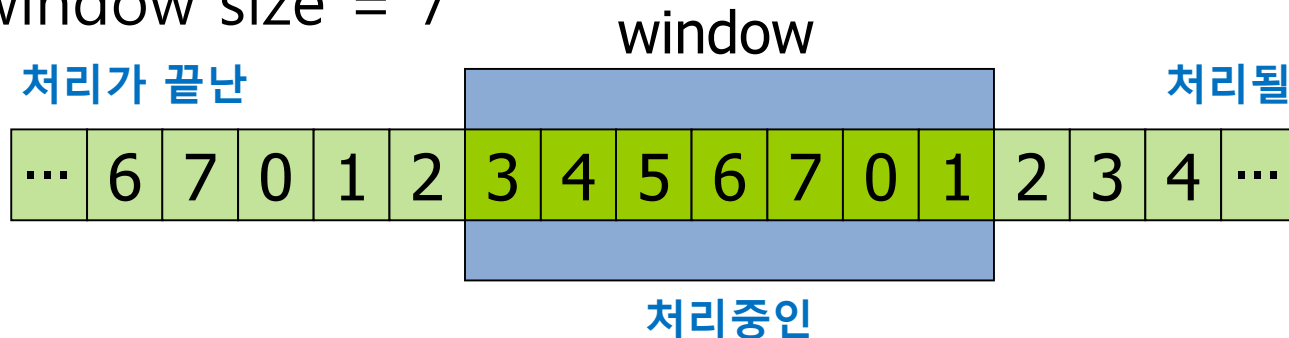
- 송신자는 하나의 프레임을 전송하고 다음 프레임을 전달하기 전에 확인응답을 기다림
- 특징
  - ✓ 장점 : 간단 (오류 프레임만 재전송)
  - ✓ 단점 : 느린 속도에 따른 비효율성 (프레임 전송마다 응답 기다려야 함) → 링크 이용률 저하



## 2) Sliding Window

- 확인응답을 받기 전에 여러 개의 프레임 전송 가능
  - ✓ 프레임 순서번호 = modulo  $2^m$ 
    - $2^m = n$ 이라 두면,  $0 \sim n-1$ 까지의 수
    - 여기서,  $m$ 은 순서번호를 표시하는데 사용되는 bit 수
- Window size = the number of outstanding frame (unacknowledged frame)
  - 즉, ACK 받기 전 연속해서 보낼수 있는 프레임 개수
  - 최대 크기는  $n-1$  (즉,  $2^m-1$ 개)

예) window size = 7



## § 4. 오류제어(Error Control)

### ■ 오류 검출 + 오류 정정

- ✓ 데이터 링크층에서의 오류제어 → 검출+재전송
  - 손상된 프레임
  - 손실된 프레임
  - 손실된 확인응답

### ■ 오류제어의 3가지 방법

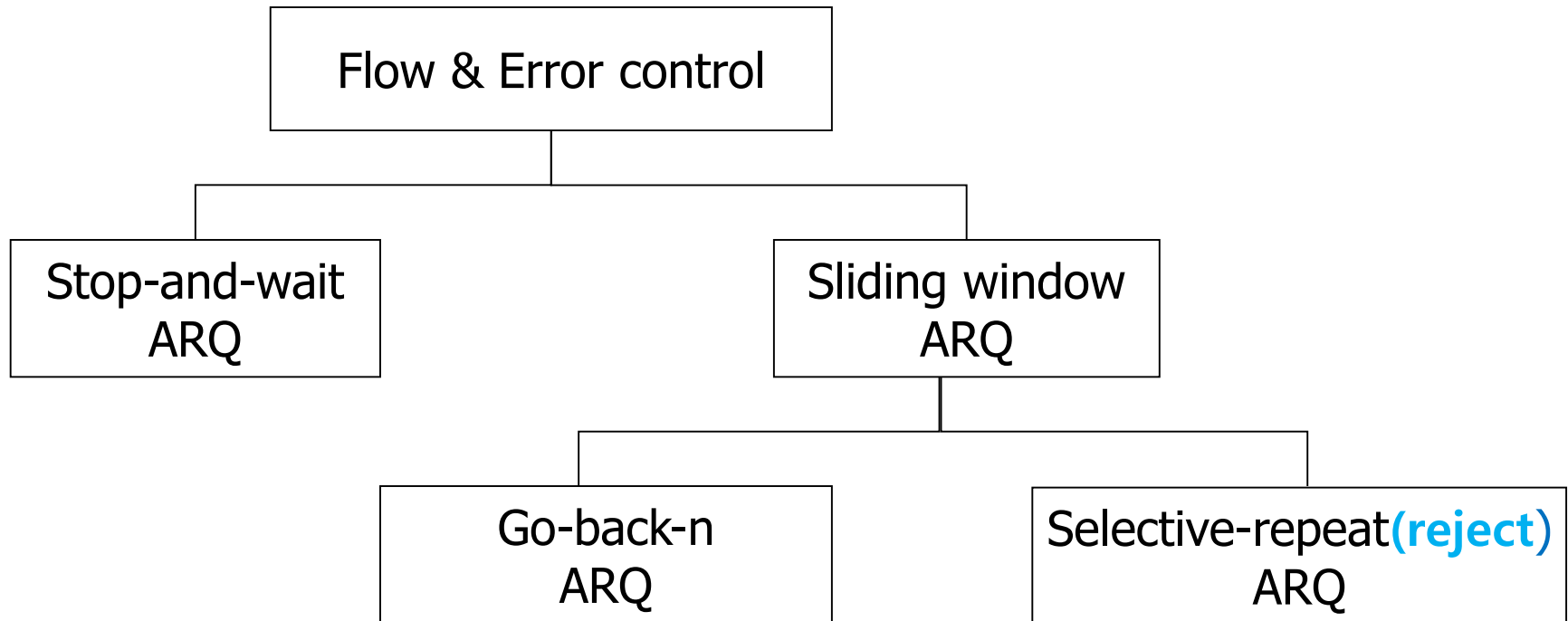
- 1) Discarding the errors
- 2) FEC(Forward Error Correction): 수신측에서 에러 정정함  
예) 해밍코드 등 사용하여 정정
- 3) ARQ (Automatic repeat request) : 재전송기법
  - 가장 많이 사용됨

# 오류제어 방법: ARQ

---

- 데이터링크 층에서의 오류제어는 ARQ를 기초로 함
- ARQ (Automatic Repeat Request)
  - ✓ 수신측에서 오류발생을 탐지하여 송신측에 통보, 송신측이 오류 발생한 프레임을 재전송하는 모든 절차들을 말함
    - ACK: 확인응답
    - NAK: 부정응답

# 흐름 및 오류 제어 장치의 분류

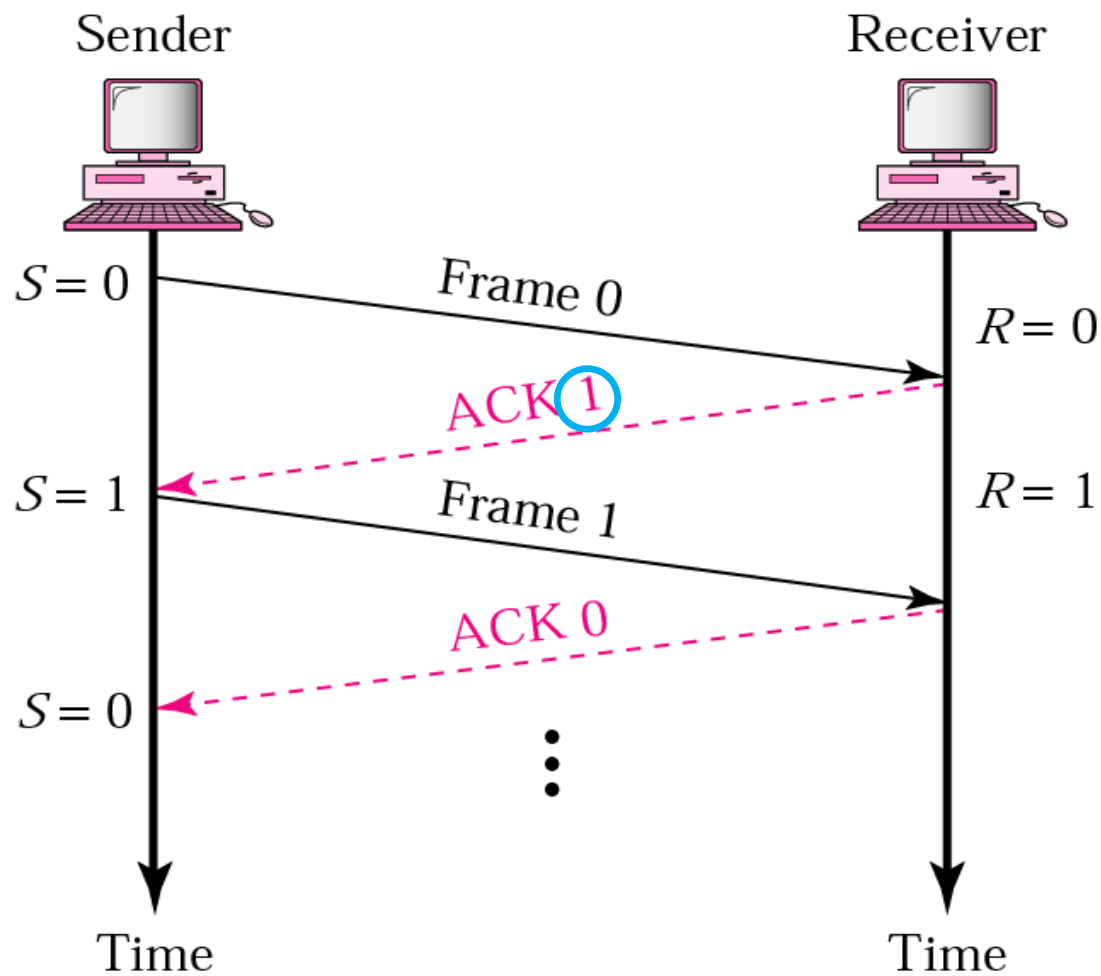


# 1) Stop and Wait ARQ

---

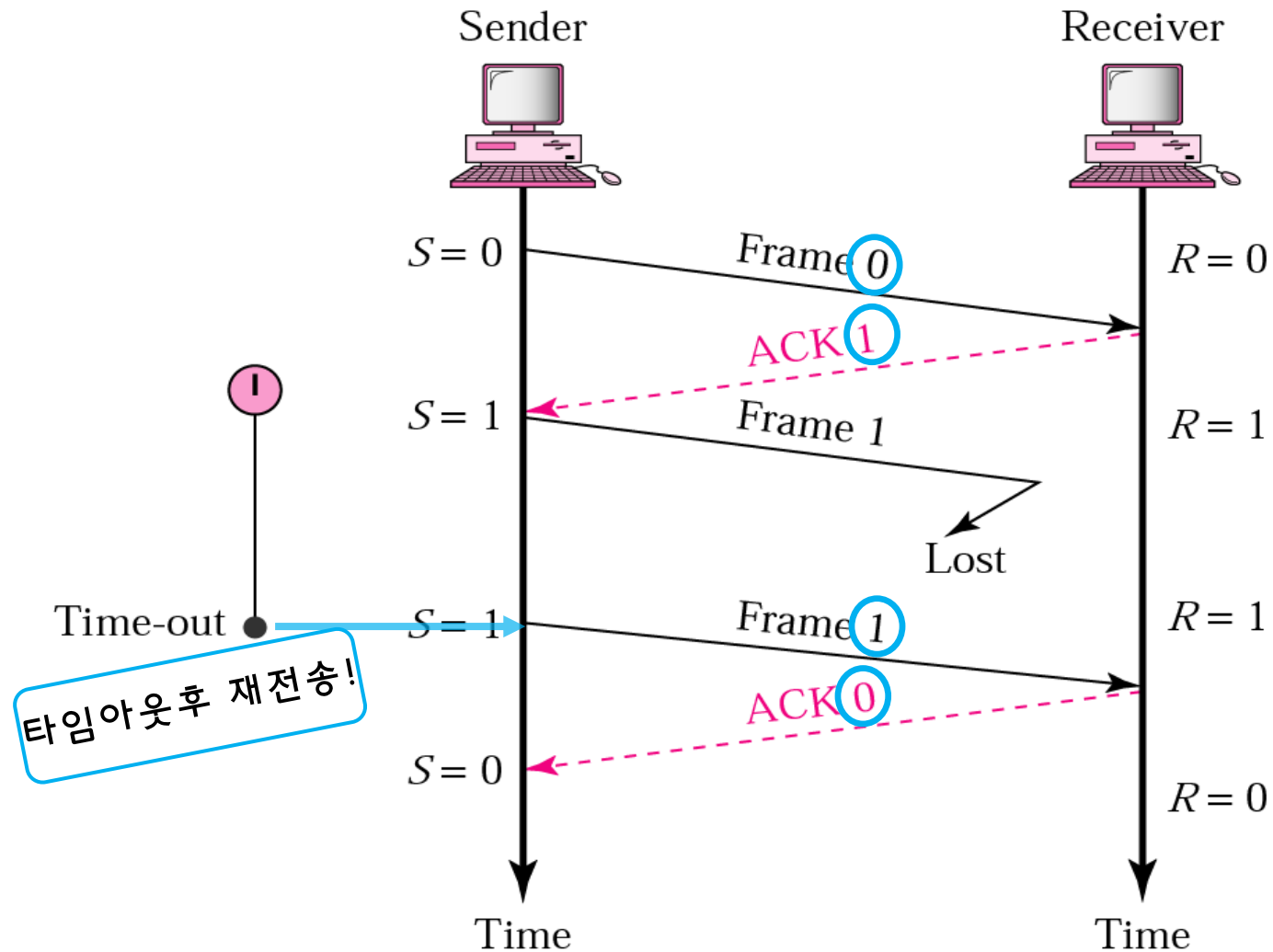
- 재전송을 위해 기본 흐름제어 메커니즘에 4가지 특성이 추가됨
  - ① 송신측은 ACK가 올 때까지 전송한 프레임의 사본을 가짐 → 분실시 재전송 위해
  - ② 데이터 프레임과 ACK 프레임에 번갈아 0과 1을 부여한다 → 중복 전송의 프레임 식별 위해
  - ③ NAK 프레임(번호가 없는)
  - ④ 타이머(송신측) → 프레임 손실 여부 파악

# 정상 동작

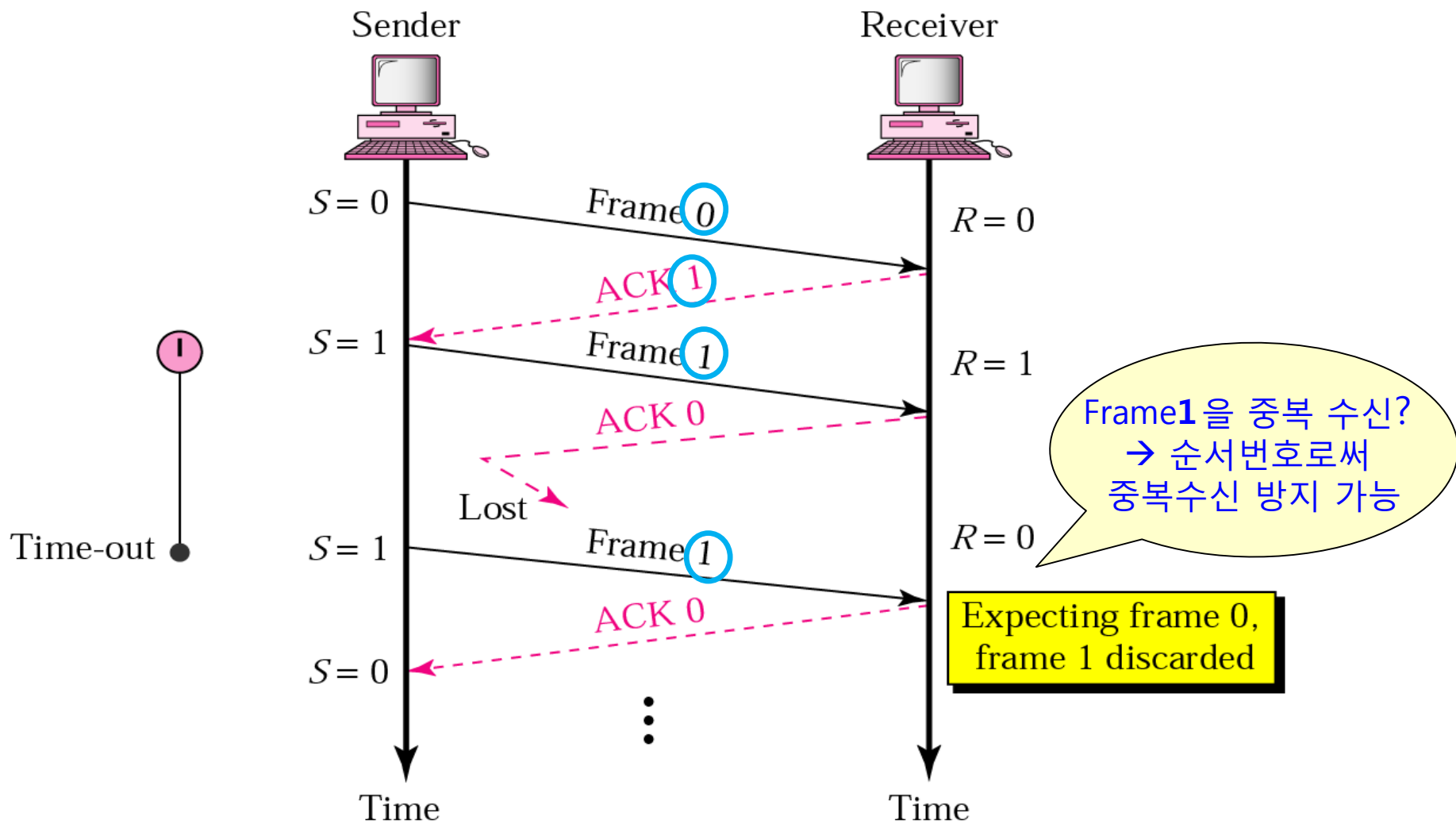




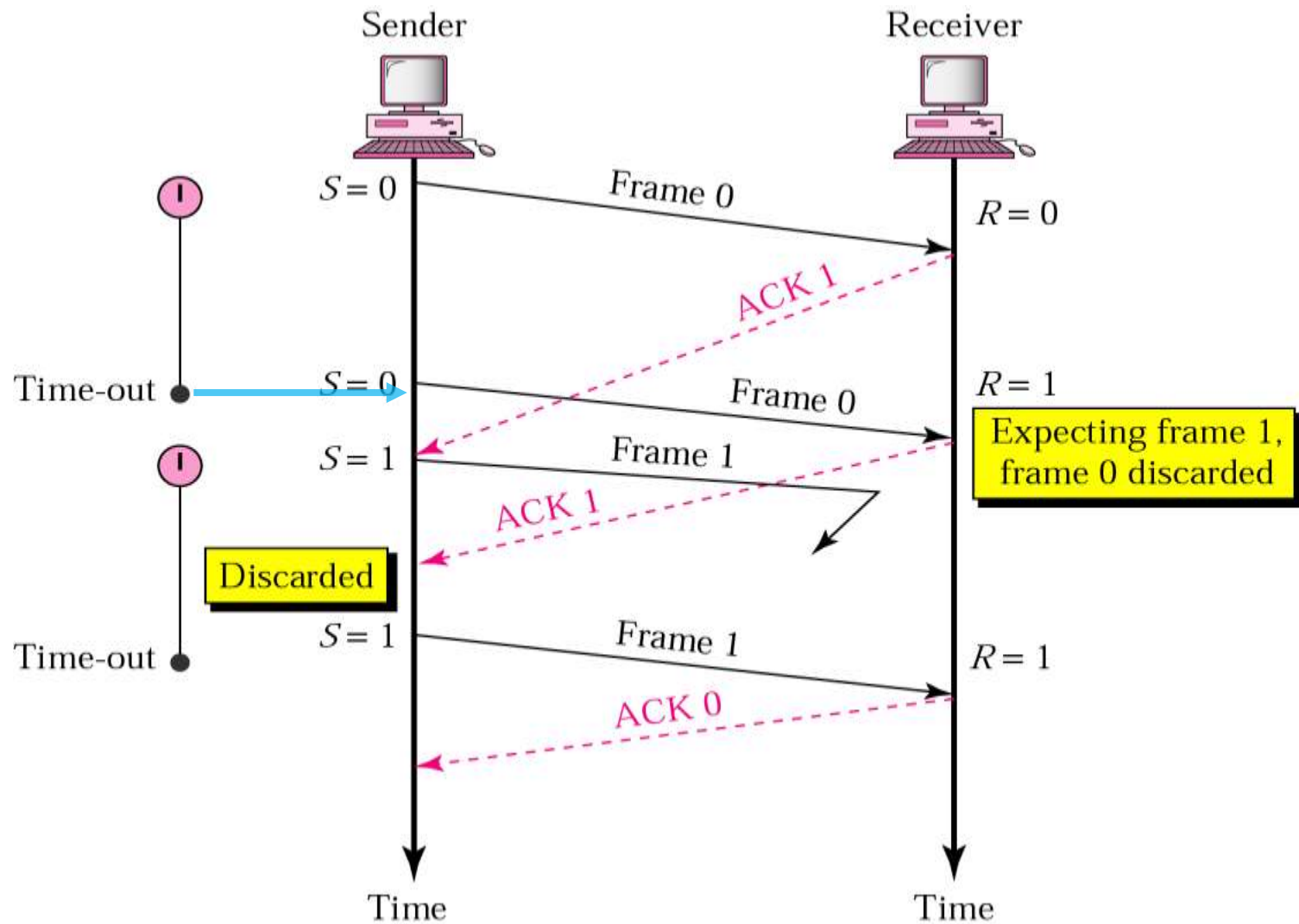
# 잃어버린 프레임



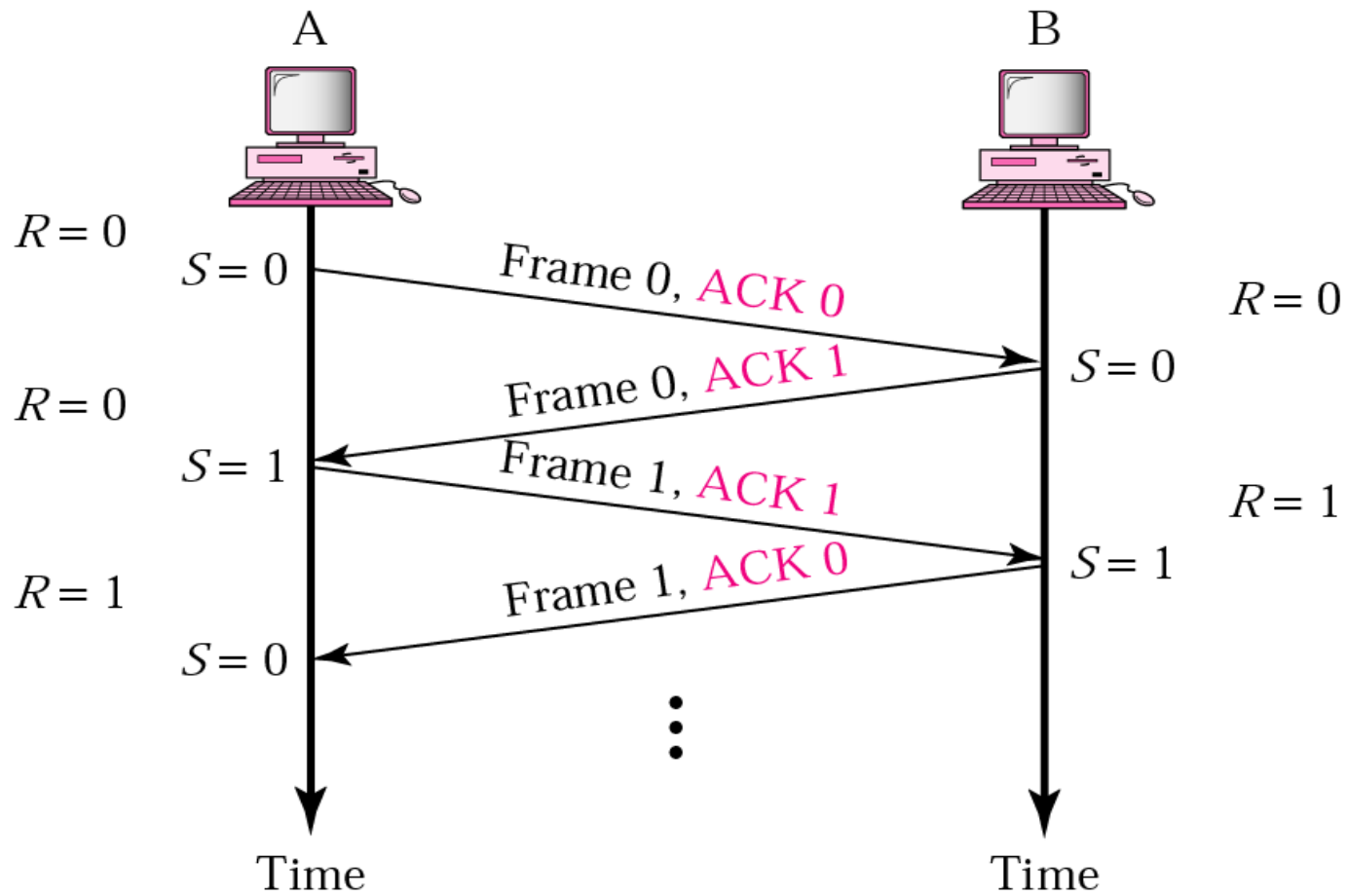
# 잃어버린 ACK 프레임



# 지연된 ACK



# Piggybacking(끼워넣기)

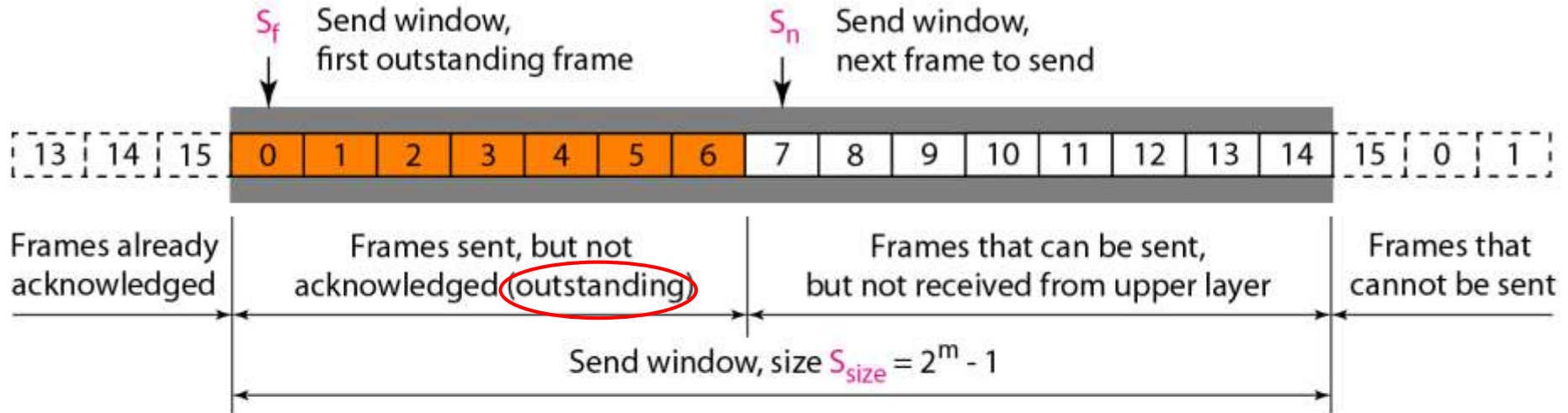


## 2) Sliding Window ARQ

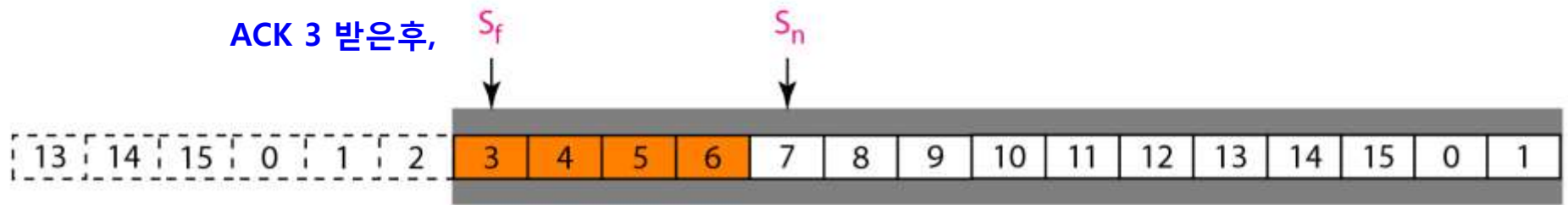
---

- 슬라이딩 윈도우 흐름제어에 손실 혹은 손상된 프레임  
임을 재전송할 수 있도록 확장한 것
- 슬라이딩 윈도우 ARQ 종류
  - ① go-back-n ARQ
  - ② selective-repeat(reject) ARQ
- 기본 흐름 제어 메커니즘에 추가된 특성
  - ✓ 송신측은 ACK가 올 때까지 전송한 모든 프레임의  
사본을 가짐
  - ✓ 확인/부정 응답
    - ACK: 수신이 예상되는 다음 프레임 번호를 전달
    - NAK: 손상된 프레임 번호를 전달
  - ✓ 분실된 확인 응답을 처리하기 위해 타이머 설치

# 송신기의 슬라이딩 윈도우



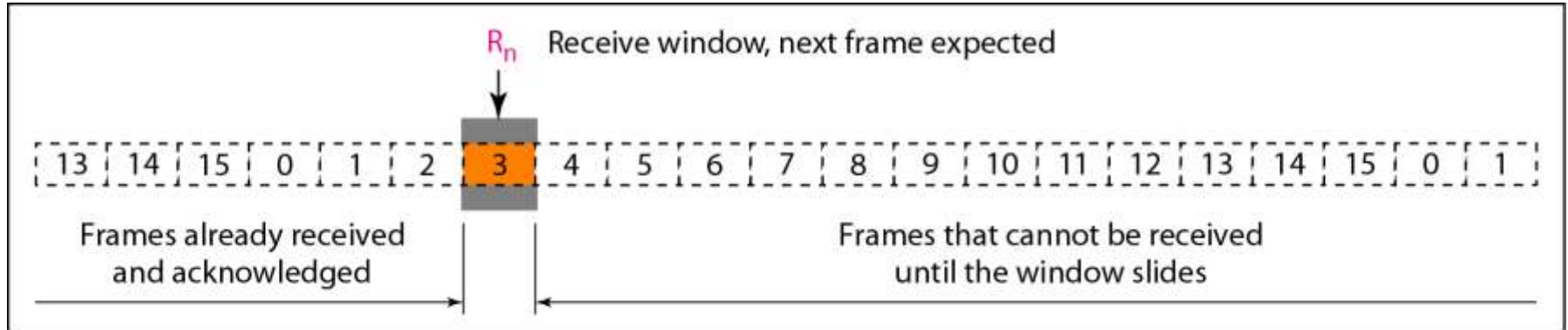
a. Send window before sliding



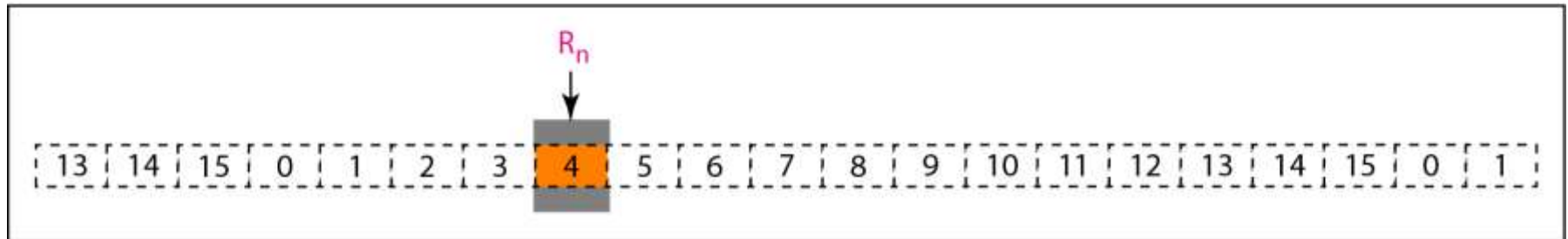
b. Send window after sliding

- The 'send window' can slide one or more slots when a valid acknowledgment arrives.

# 수신기의 슬라이딩 윈도우



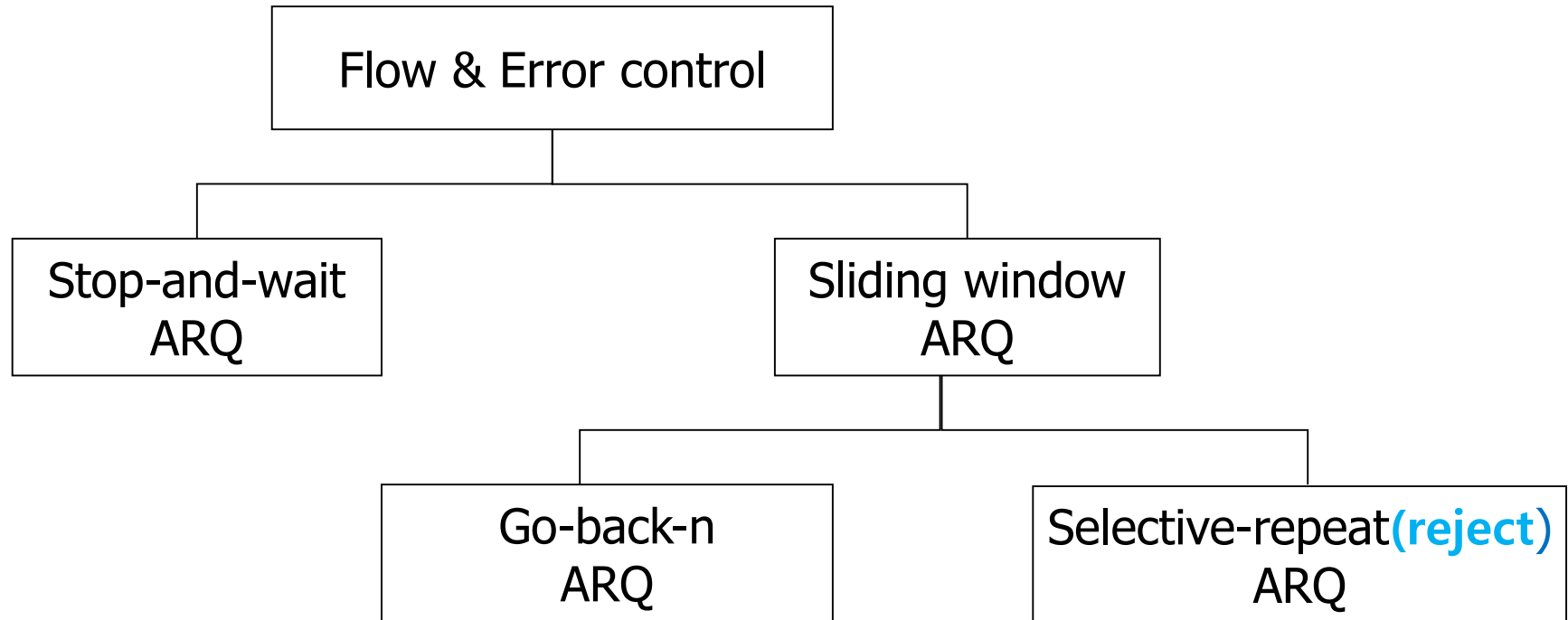
a. Receive window



b. Window after sliding

- The window slides when a correct frame has arrived; sliding occurs one slot at a time.

# [Review]





# ① Go-Back-N (n 프레임 후퇴) ARQ

## ■ Go-Back-N ARQ에서는

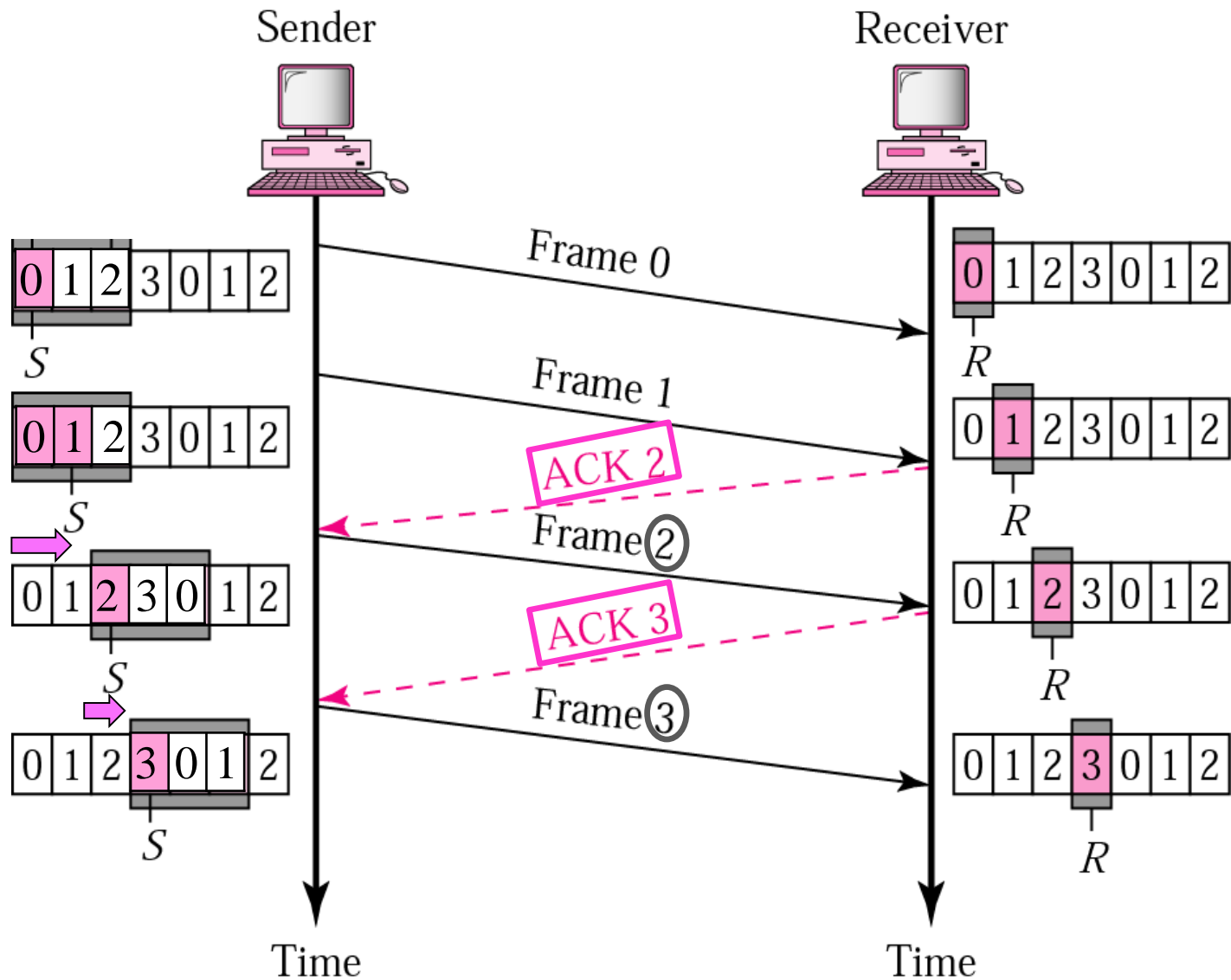
- ✓ 송신기의 윈도우 크기는  $2^m$ 보다 작아야 함.  
최대  $2^m - 1$  (보통  $2^3 - 1 = 7$ 개)
- ✓ 수신기의 윈도우 크기는 항상 1이다.
- ✓ 전송 오류 없으면 연속적으로 송신 가능

## ■ 프레임의 분실, 손상시

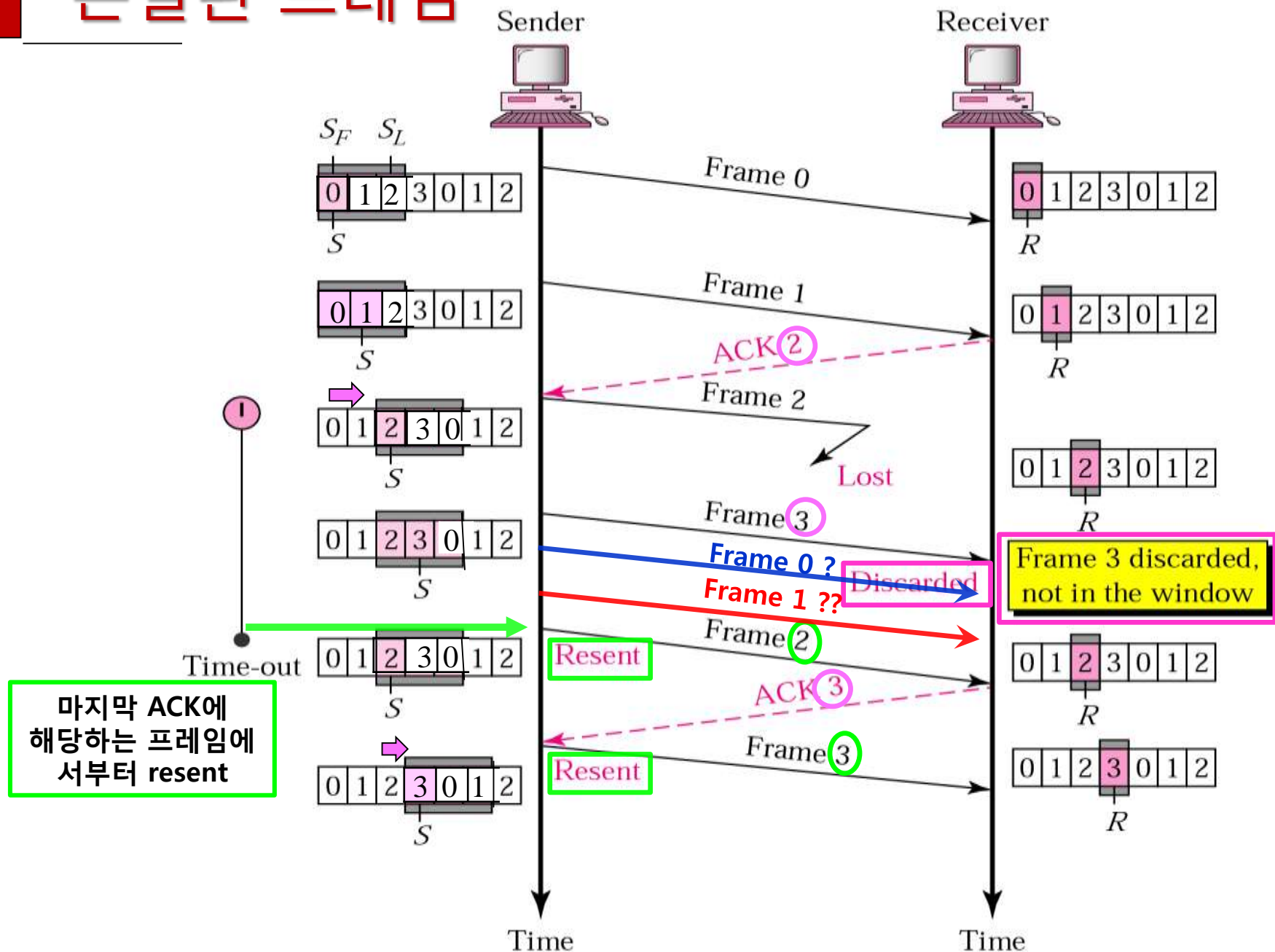
- ✓ 여러 프레임을 연속 전송하는데, 프레임이 분실되거나 손상되면 수신측에서 NAK 보냄
- ✓ 송신측에서 NAK 받으면, 오류 발생한 이후의 모든 프레임을 재전송함

## ■ 단점: 오류 발생된 이후의 프레임을 모두 재전송하므로 중복 전송

# 정상 동작



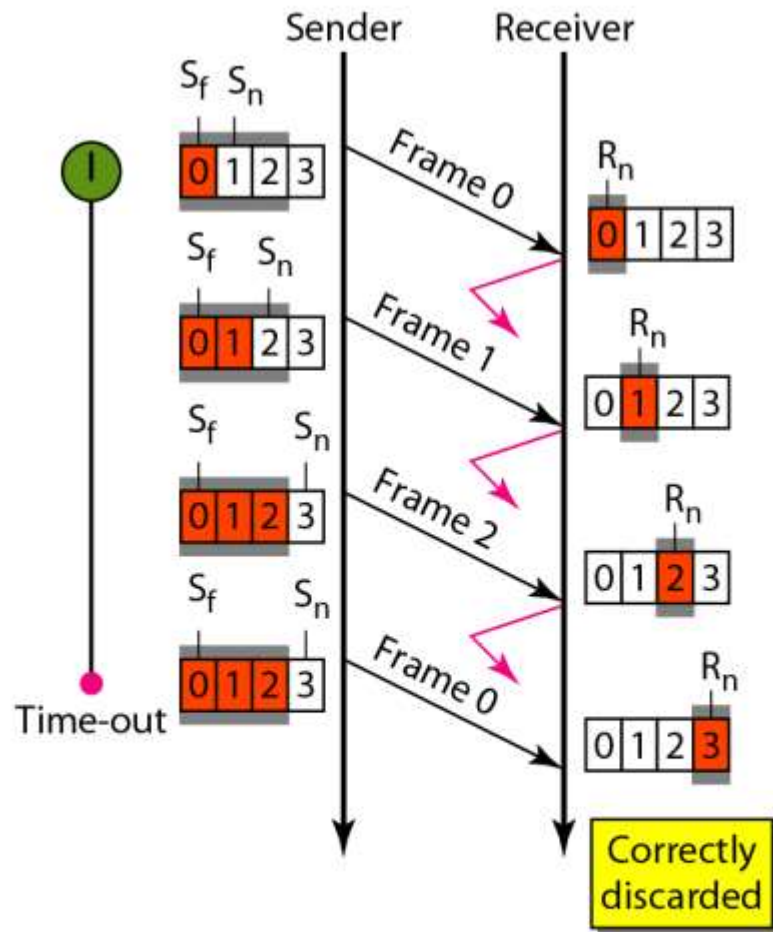
# 손실된 프레임



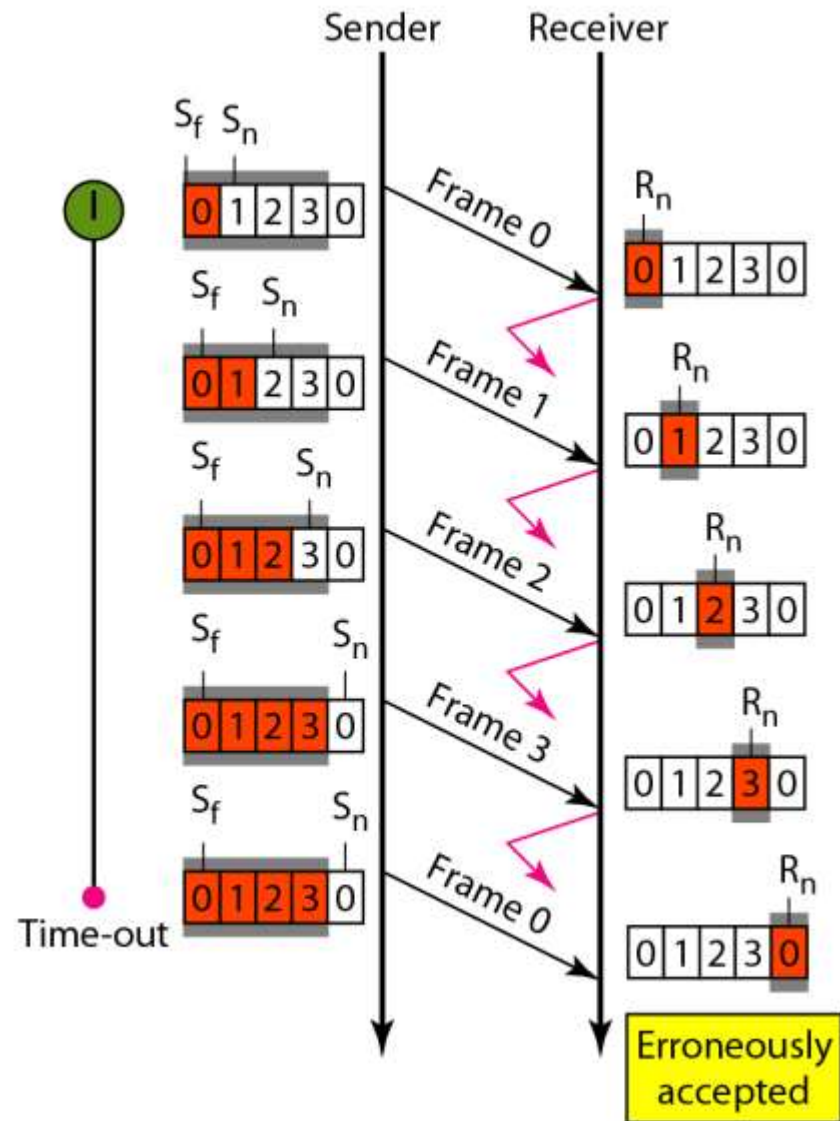
# 송신기 윈도우 크기

a. Window size  $< 2^m$

$m=2$ 일때



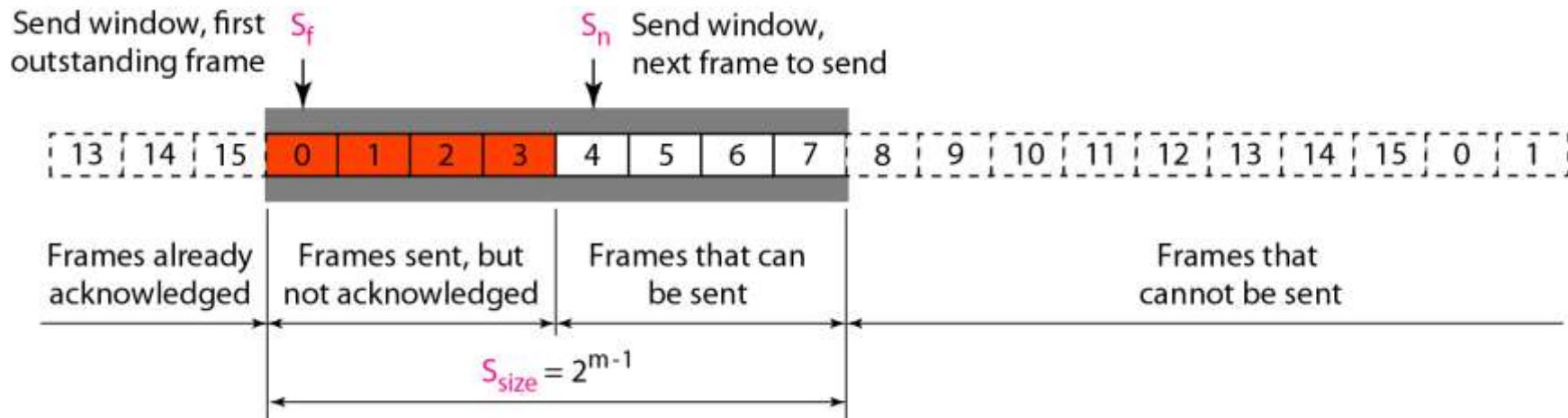
b. Window size  $= 2^m$



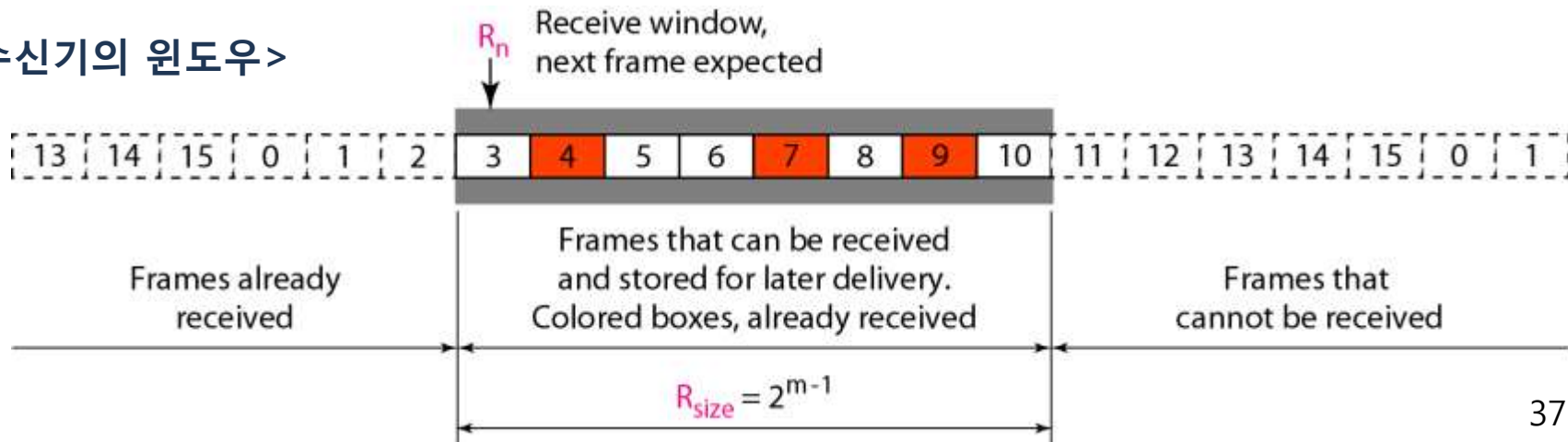
## ② Selective-Repeat ARQ(선택적 반복ARQ)

### ■ 손상되거나 분실된 프레임만 재전송

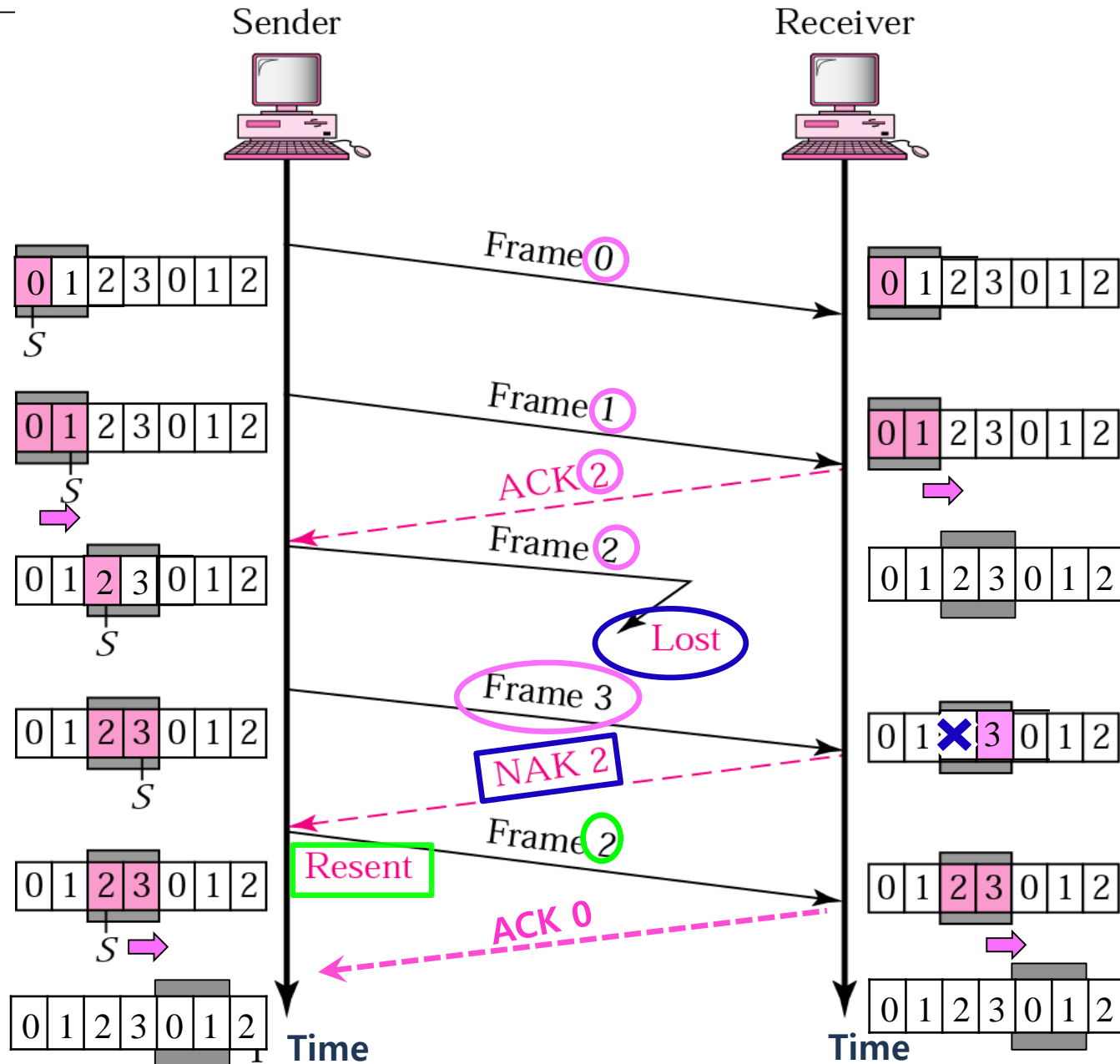
#### <송신기의 윈도우>



#### <수신기의 윈도우>



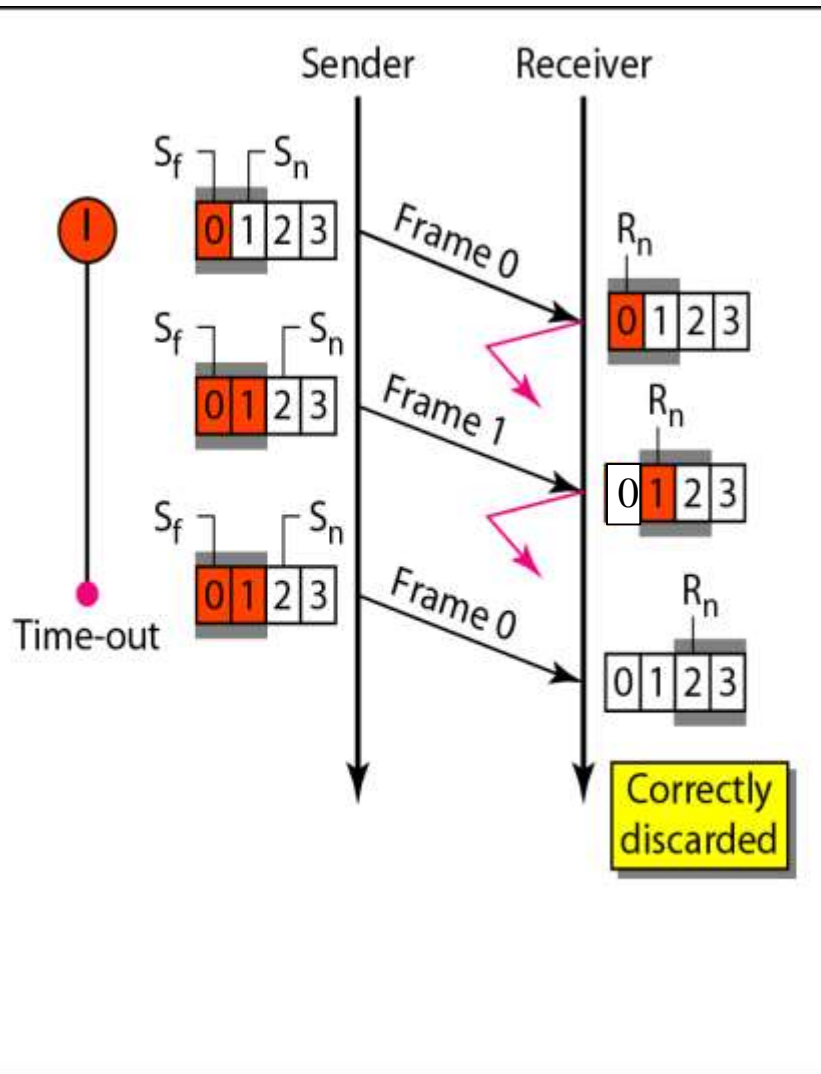
# 손실된 프레임



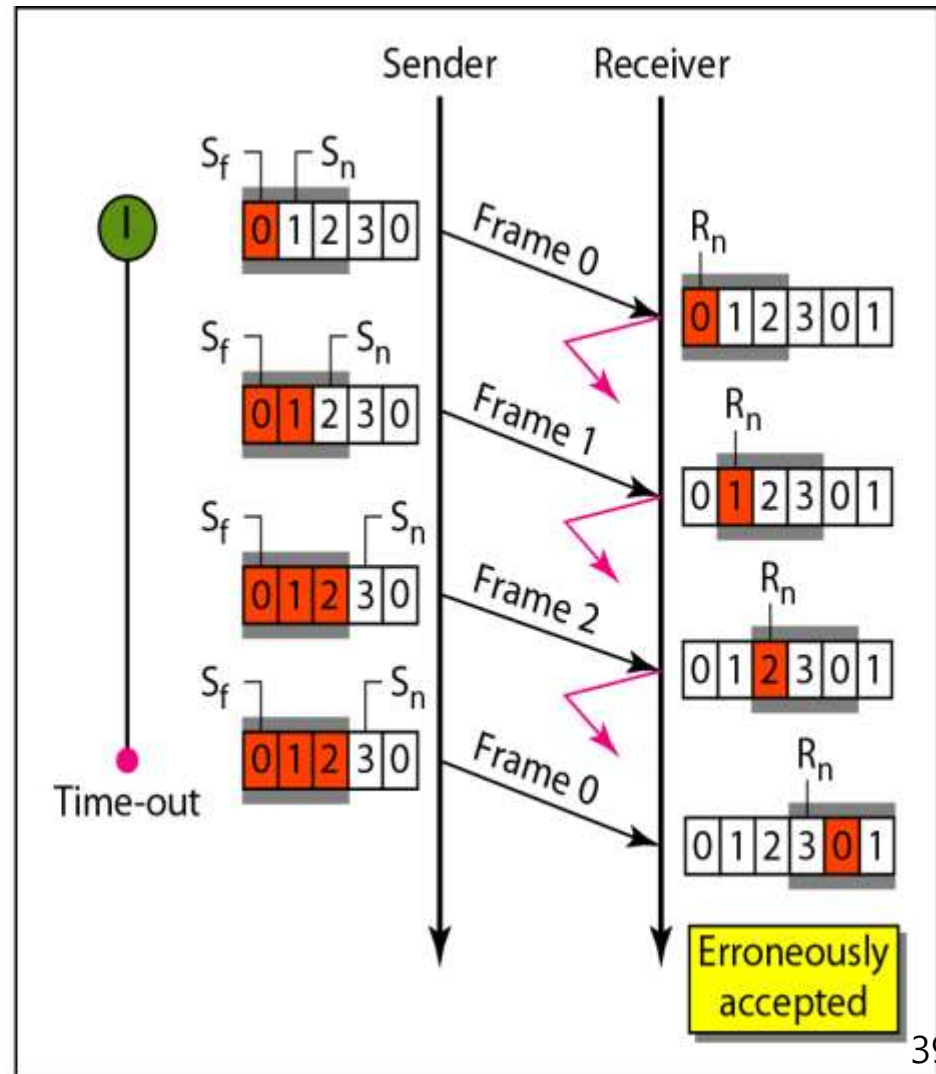
# Selective-Repeat ARQ에서 윈도우 크기

$m=2$ 일때

a. Window size =  $2^{m-1}$



b. Window size  $> 2^{m-1}$



## [비교] 3가지 ARQ

---

### ■ Stop and Wait ARQ

- ✓ 구조가 간단하여 구현이 용이하나, 비효율적이어서 활용도가 낮음

### ■ Go-Back-N ARQ

- ✓ 구조가 비교적 간단하고, 효율성이 향상되어 가장 널리 사용

### ■ Selective-Repeat ARQ

- ✓ 가장 효율적이거나 구조가 복잡해서 유지 관리 비용 증가, 필요시 사용



## § 5. HDLC(High-level Data Link Control)

---

- HDLC는 점-대-점과 다중점 구성에서 반이중과 전이중 모드를 지원하도록 설계된 실제 프로토콜
  - ✓ 앞서 소개한 ARQ 장치의 구현
  - ✓ 오늘날 PPP와 이더넷에서 사용되는 공통 데이터링크 프로토콜의 기초가 되는 프로토콜임
  - ✓ 비트중심 프로토콜
- HDLC는 지국의 형태, 구성, 응답 모드에 따라 구분

# Station Types

지국형태

---

## ■ 주국(primary station)

- ✓ 다중점 회선구성에서 링크의 모든 제어권을 가짐
- ✓ 종국에게 명령을 전송

## ■ 종국(secondary station)

- ✓ 주국의 명령에 대한 응답을 전송

## ■ 혼합국(combined station)

- ✓ 명령과 응답을 전송
- ✓ 주국,종국 기능 모두 가짐

- 통신을 하고 있는 두 장치 간의 관계를 나타냄
- 누가 링크를 제어하는가를 표시
- HDLC는 두 가지 전송모드를 제공
  - 1) NRM(Normal Response Mode) 정규 응답 모드
  - 2) ABM(Asynchronous Balanced Mode)비동기 균형 모드

# NRM(Normal Response Mode)

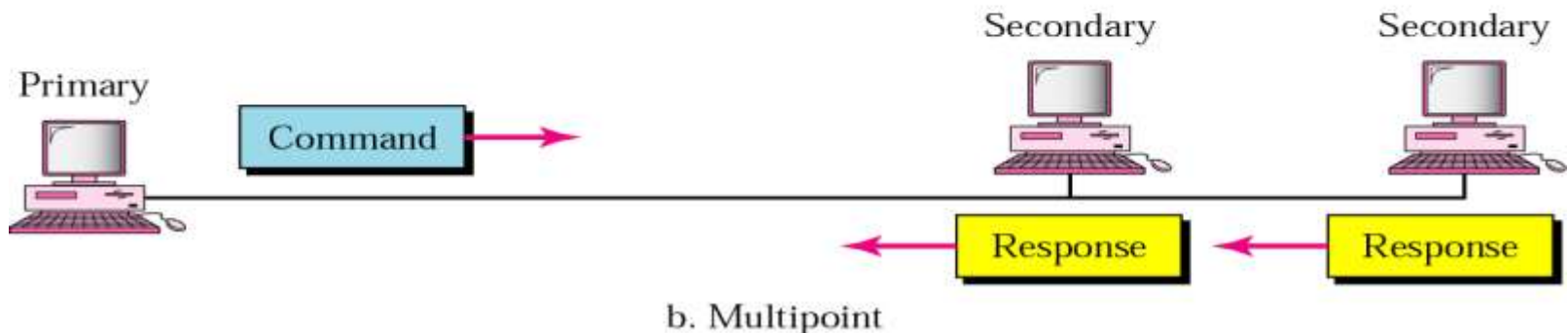
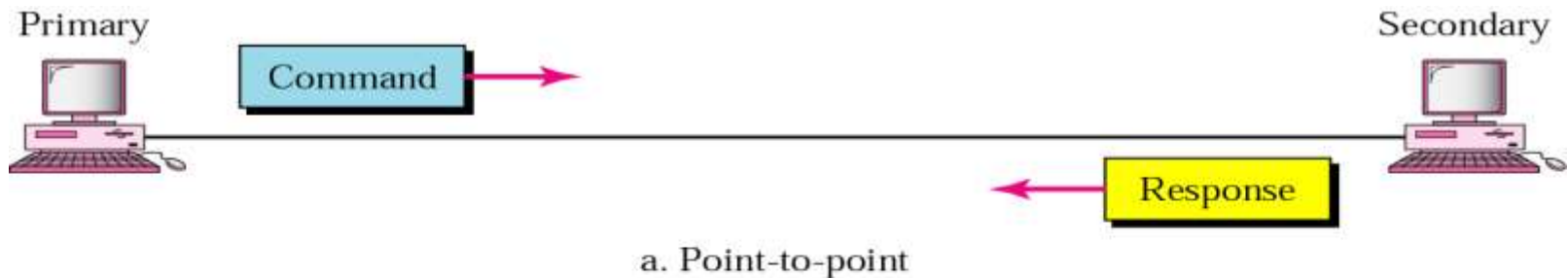
정규 응답모드

## ■ 표준의 주국-종국 관계에 적용

- ✓ 하나의 주국, 다수의 종국
- ✓ 점-대-점과 다중점 링크에 모두 사용

## ■ 주국의 명령, 종국의 응답

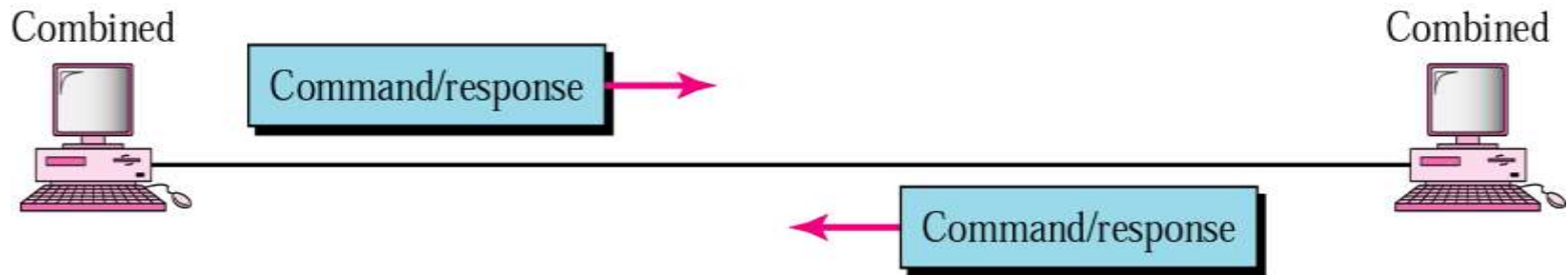
- ✓ 종국은 전송 전에 주국의 허가를 받아야 하며
- ✓ 이후 하나 이상의 프레임의 응답전송 시작



# ABM(Asynchronous Balanced Mode)

비동기 균형모드

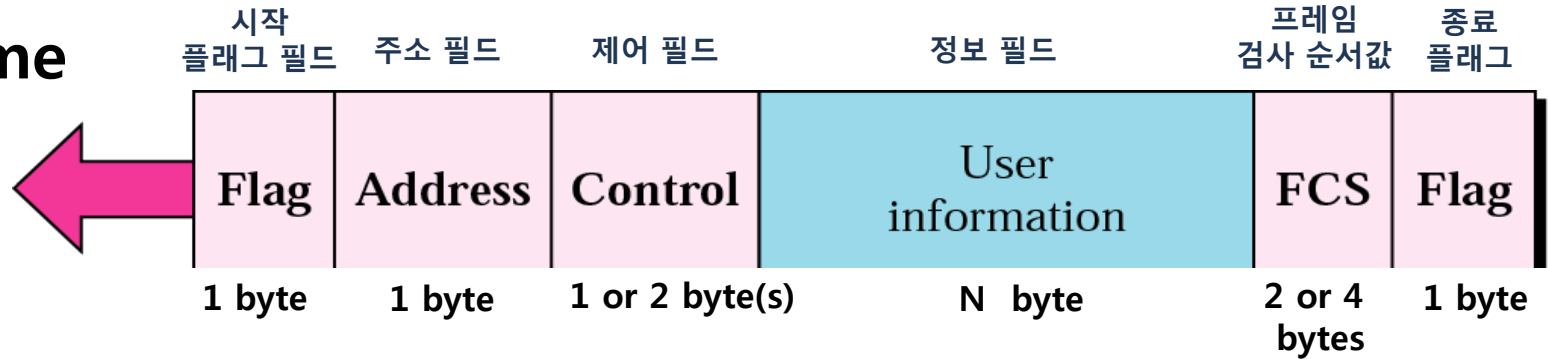
- 모든 지국은 동등
  - ✓ 점-대-점 링크에서만 사용
  - ✓ 각 지국은 주국이자 종국으로 기능
- 각 지국은 허가 없이도 다른 지국과 전송을 시작
- 요즘 보통 사용하는 모드



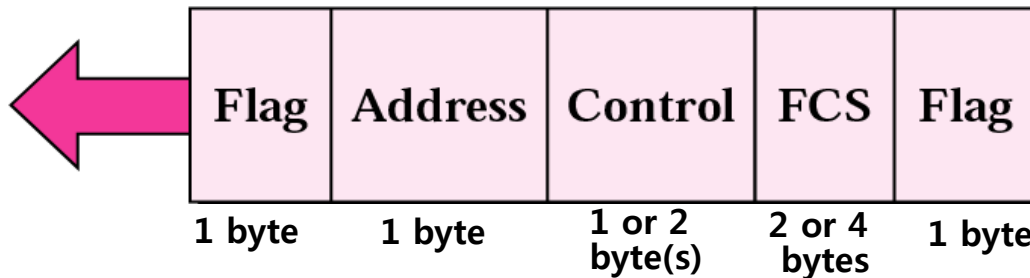
- 여러 통신모드와 선택사항을 지원하기 위해 3종류의 프레임을 정의
  - ✓ I (**I**nformation, 정보) 프레임
    - 사용자 데이터와 이와 관계된 제어정보 전송에 사용
  - ✓ S (**S**upervisory, 감시, 감독) 프레임
    - 데이터링크층의 흐름제어, 오류제어 등과 같은 제어정보 전송에 사용
  - ✓ U (**U**nnumbered, 무번호) 프레임
    - 링크 자체를 관리할 목적으로 사용
    - 연결된 단말기 사이의 세션 관리, 제어정보 교환에 사용

# HDLC frame 유형

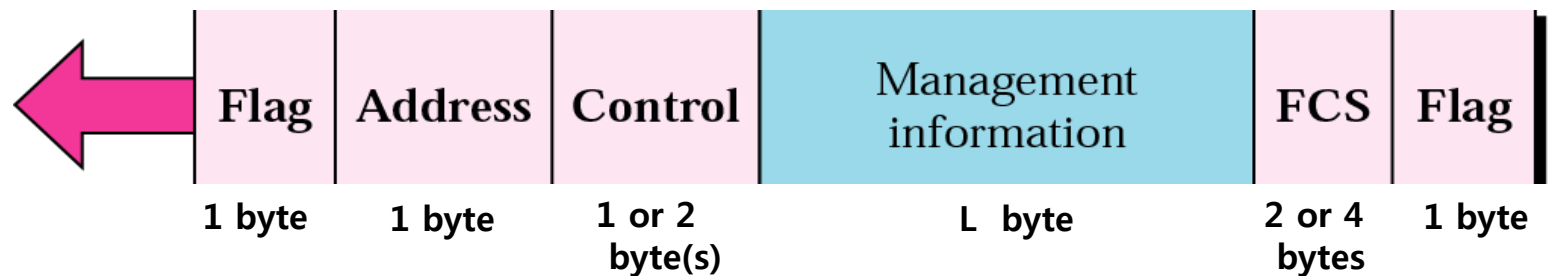
## I frame



## S frame

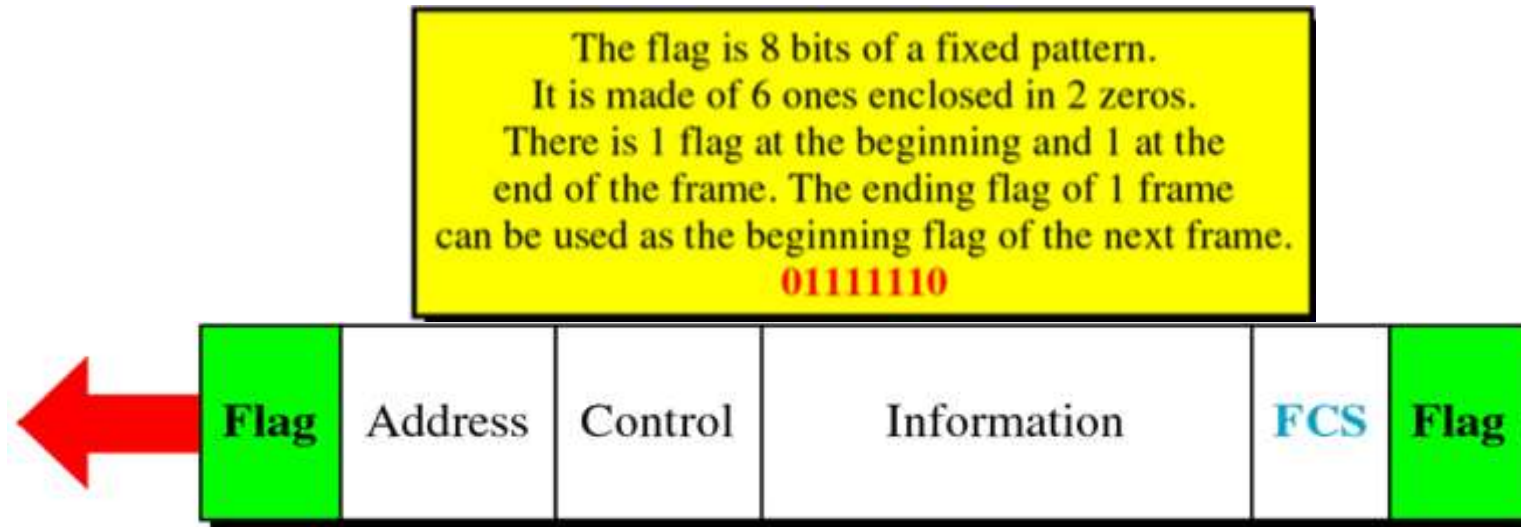


## U frame



## ① Flag 필드

- 프레임의 시작과 끝을 인식
- 수신자를 위한 동기화 패턴으로 제공 (즉, 0111 1110)
- 다중 프레임 전송에서는 종료 flag는 다음 프레임의 시작 flag를 겸할 수 있다



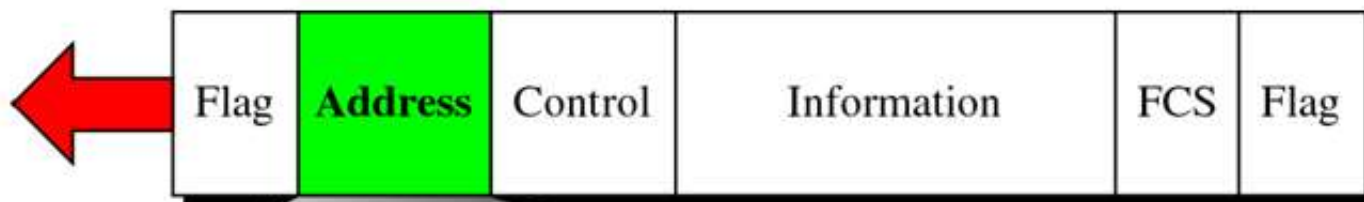


## ② 주소 필드

### ■ 프레임의 발신지 주소나 목적지 주소를 가짐

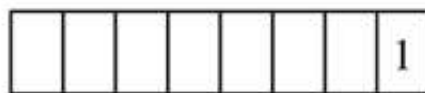
- ✓ 주국이 프레임 생성했으면 목적지(종국) 주소를 가지고, 종국이 프레임 생성했으면 발신지(종국자신) 주소를 가짐.

• 종국의 주소



The address is one byte (8 bits) or a multiple of bytes. 주로 1byte

네트워크  
필요에 따라  
선택적 사용



One-byte address

:  $2^7$  개 지국 구별



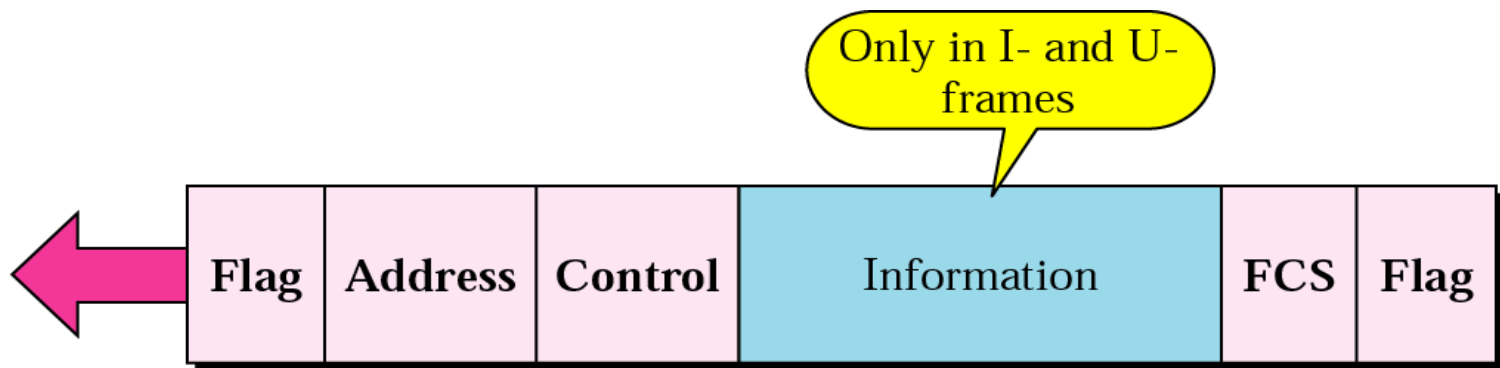
Multibyte address

→  $2^{21}$

다음 byte까지 주소가  
이어짐을 나타냄

### ③ 정보 필드

- I frame: 사용자 데이터
- U frame: 네트워크 관리정보
- Piggybacking
  - ✓ 전송할 데이터와 수신된 데이터의 응답을 한 프레임에 조합하여 사용하는 방법

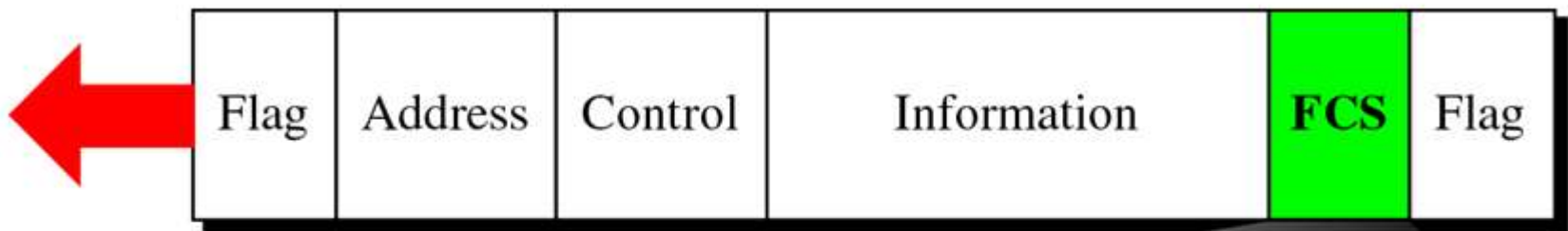


## ④ FCS 필드

Frame Check Sequence/프레임 검사 순서값

### ■ HDLC의 오류검출 필드

✓ 2 or 4바이트 CRC

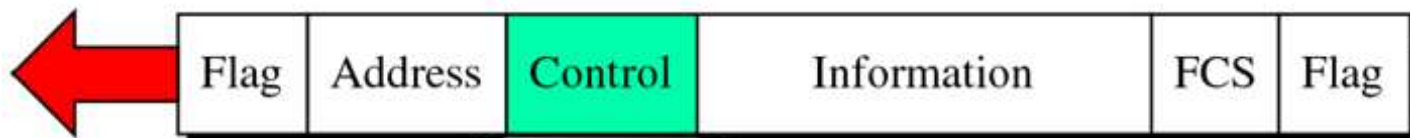


Frame check sequence is the error detection field.  
It can be a two-byte or a four-byte CRC.

## ⑤ 제어 필드

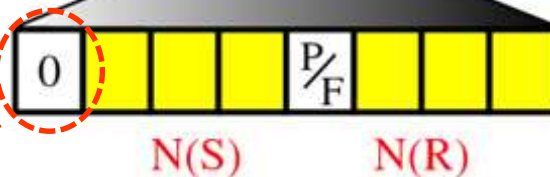
### ■ 흐름관리에 사용 (보통 1 혹은 2 byte)

#### ✓ 1 byte 제어 필드의 경우



•정보전송  
(+확인응답)

I-Frame



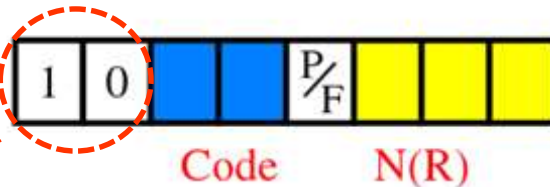
PF : Poll/Final bit

N(S) : Sequence number of frame sent (프레임 순서 번호 0~7)

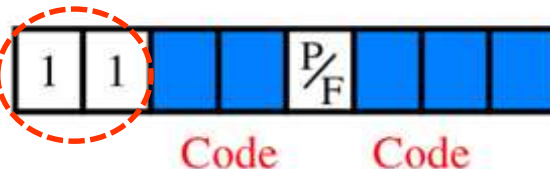
N(R) : Sequence number of next frame expected (ACK, NAK)

•수신기에서 보낼  
data가 없을때 ACK  
•흐름/오류  
정보를 코드화

S-Frame

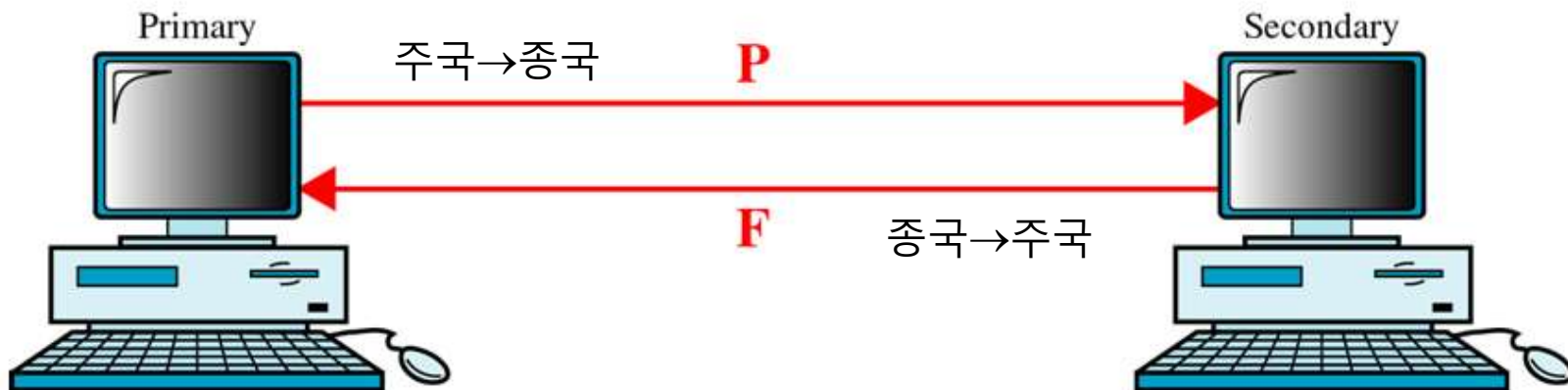


U-Frame



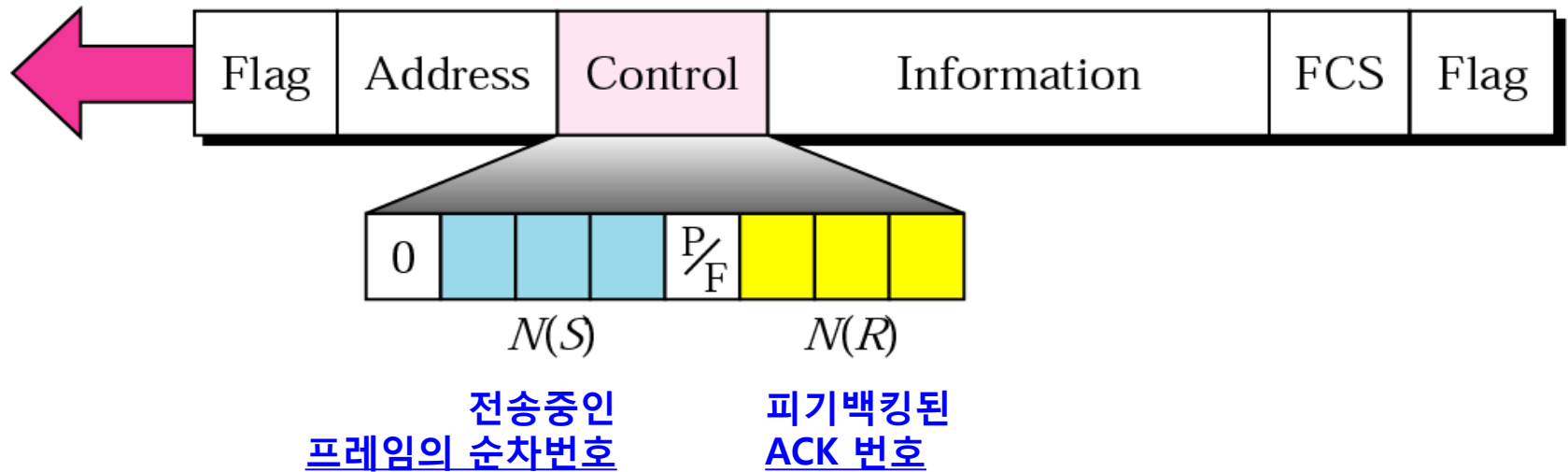
Code : Code for supervisory or unnumbered frame  
(5비트로 32가지 U프레임)

- P/F 비트가 1인 경우만 유효하며, Poll/Final을 의미함
  - ✓ Poll : 프레임이 주국에서 종국으로 보내질때, 즉, 주소필드가 수신기(목적지) 주소를 가질때
  - ✓ Final : 프레임이 종국에서 주국으로 보내질때 즉, 주소필드가 발신지 주소를 가질때



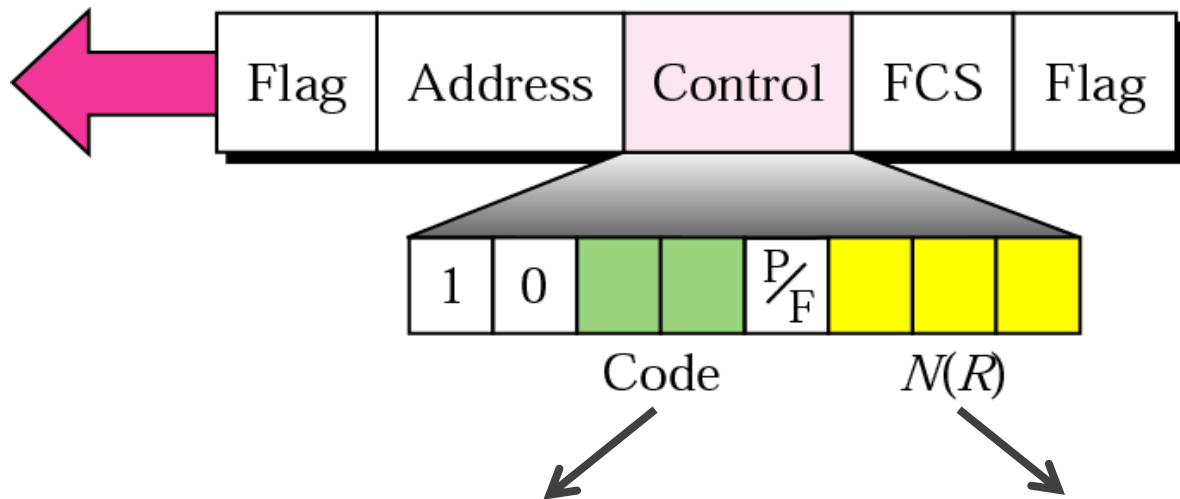
# I-frame의 제어 필드

- 사용자의 데이터 전송
- 확인응답을 포함할 수 있음(끼워 보내기)



# S-frame의 제어필드

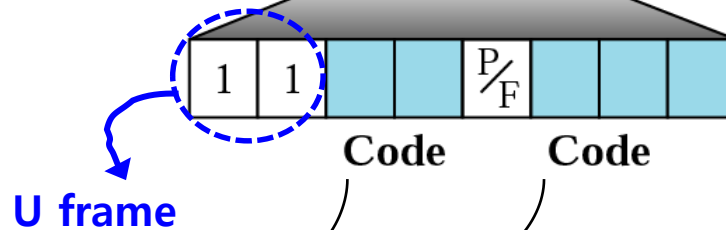
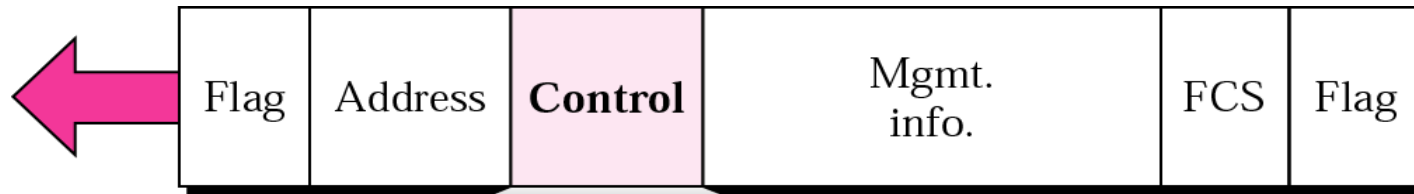
- 지국이 송신하려는 데이터가 없을 때 확인응답 전송 위해
- 흐름제어와 오류 제어를 전송 위해



00 : RR 수신준비; 온전히 전송된 frame의 ACK  
10 : RNR 수신불가; ACK 그러나 바빠서 더 이상 수신불가  
01 : REJ 수신거부; Go-Back-N ARQ에서 사용; N(R)값은 NAK번호  
11 : selective reject ARQ의 NAK; N(R)값은 NAK번호

ACK 번호

# U-frame의 제어필드

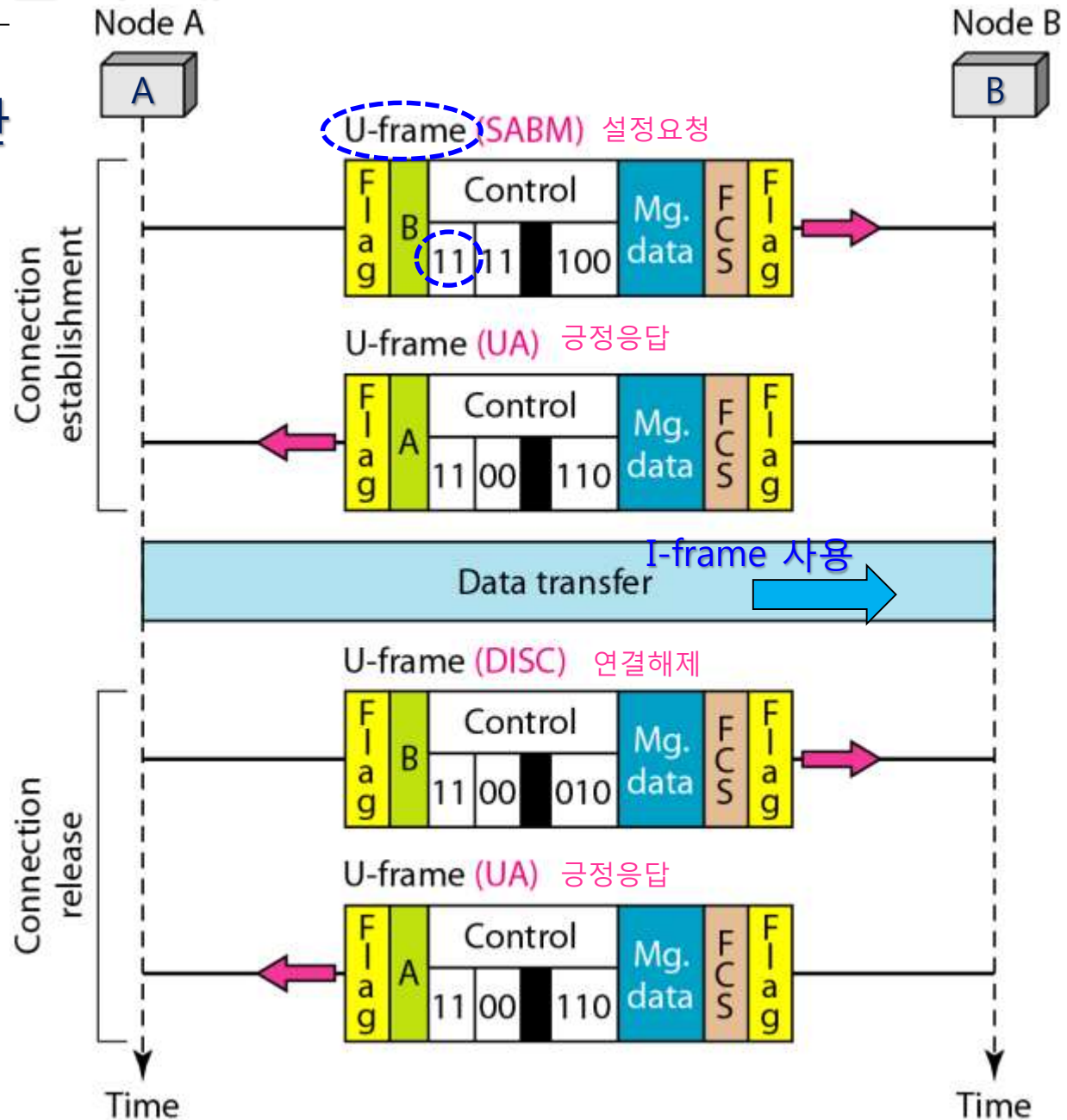


Code		Command	Response
00	001	SNRM	
11	011	SNRME	
11	100	SABM (설정요청)	DM
11	110	SABME	
00	000	UI	UI
00	110		UA (긍정응답)
00	010	DISC (연결해제)	RD
10	000	SIM	RIM
00	100	UP	
11	001	RSET	
11	101	XID	XID
10	001		FRMR



# 연결 및 종결의 예

- U-frame 사용한 연결 및 해제



# 오류없는 피기백킹 예

- I-frame 사용한 데이터 전송

I-frame의 control field



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

1 3 1 3

byte

S-frame의 control field



10 code P/F N(R)

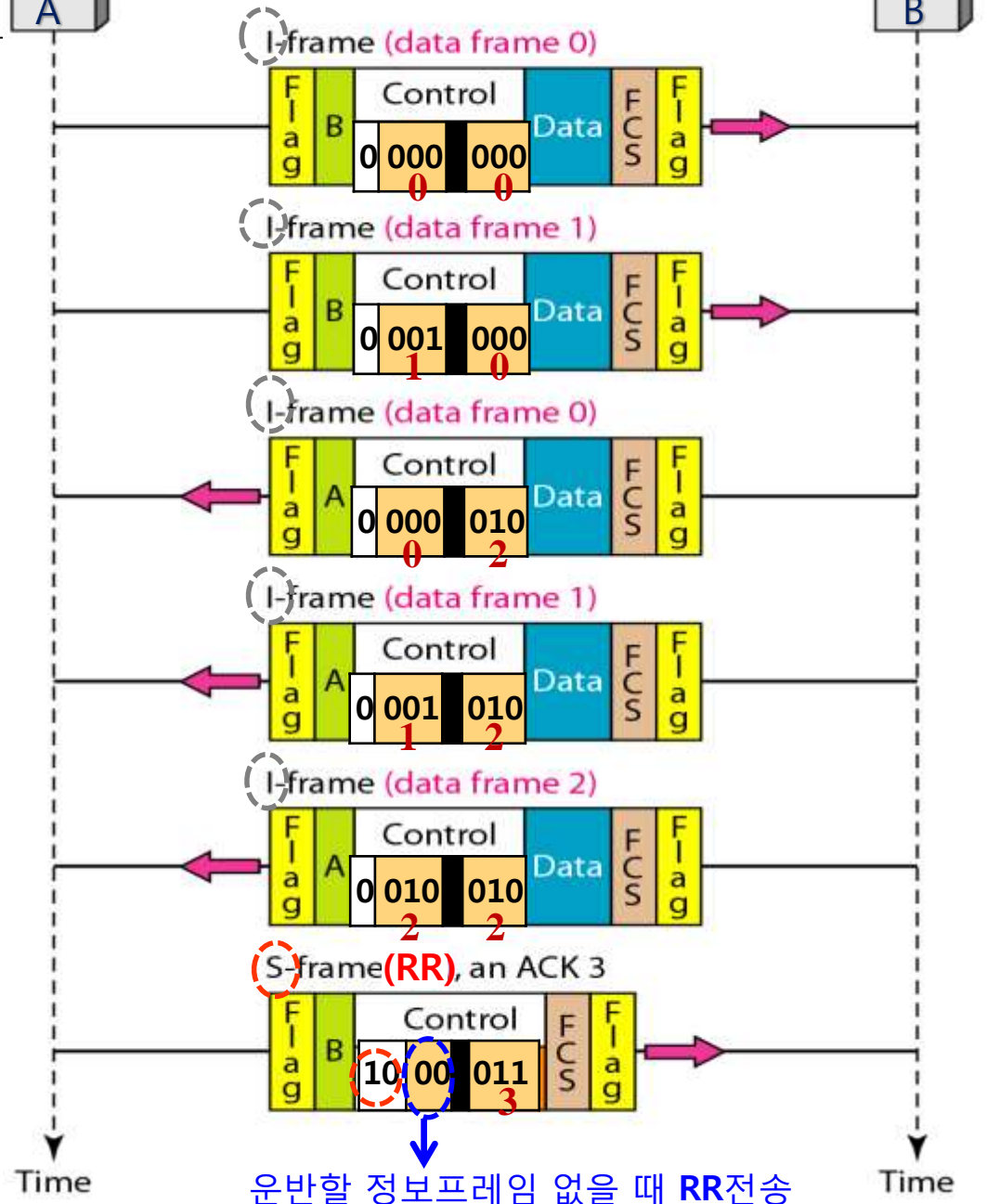
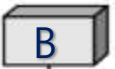
2 2 1 3

byte

Node A

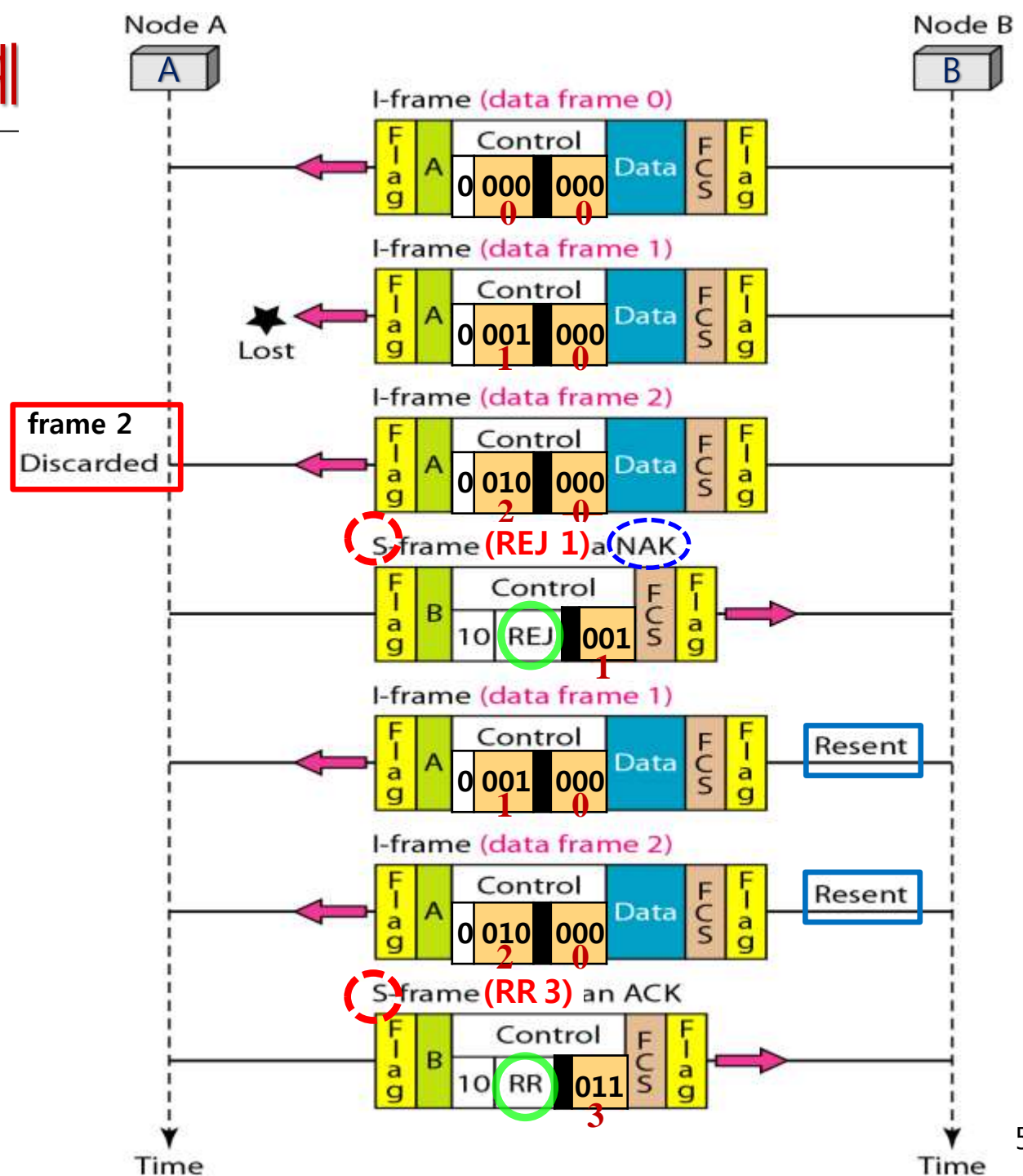


Node B



# 오류발생의 예

- Go-back N ARQ  
가정



# 11장 - 끝 -