

# 데이터통신

오류 제어: 재전송

충남대학교  
컴퓨터공학과  
이영석

# 오늘 공부할 것

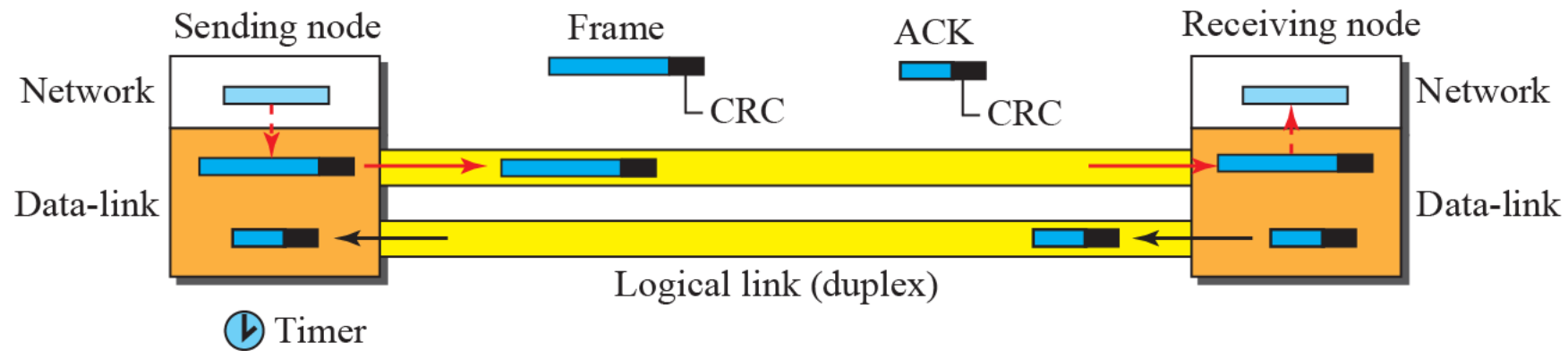
- 잘 들리나요?
  - 예
  - 아니오
    - 다시 이야기 하겠습니다.
- 너무 빨리 이야기했나요?
  - 예
  - 아니오
    - 조금 천천히 이야기 하겠습니다.

# 데이터 링크 계층

- 흐름제어
- 에러제어
- 예
  - Stop-N-Wait ARQ (Automatic Repeat Request)
  - Go-back-N ARQ
  - Selective-Repeat ARQ

# Stop-N-Wait

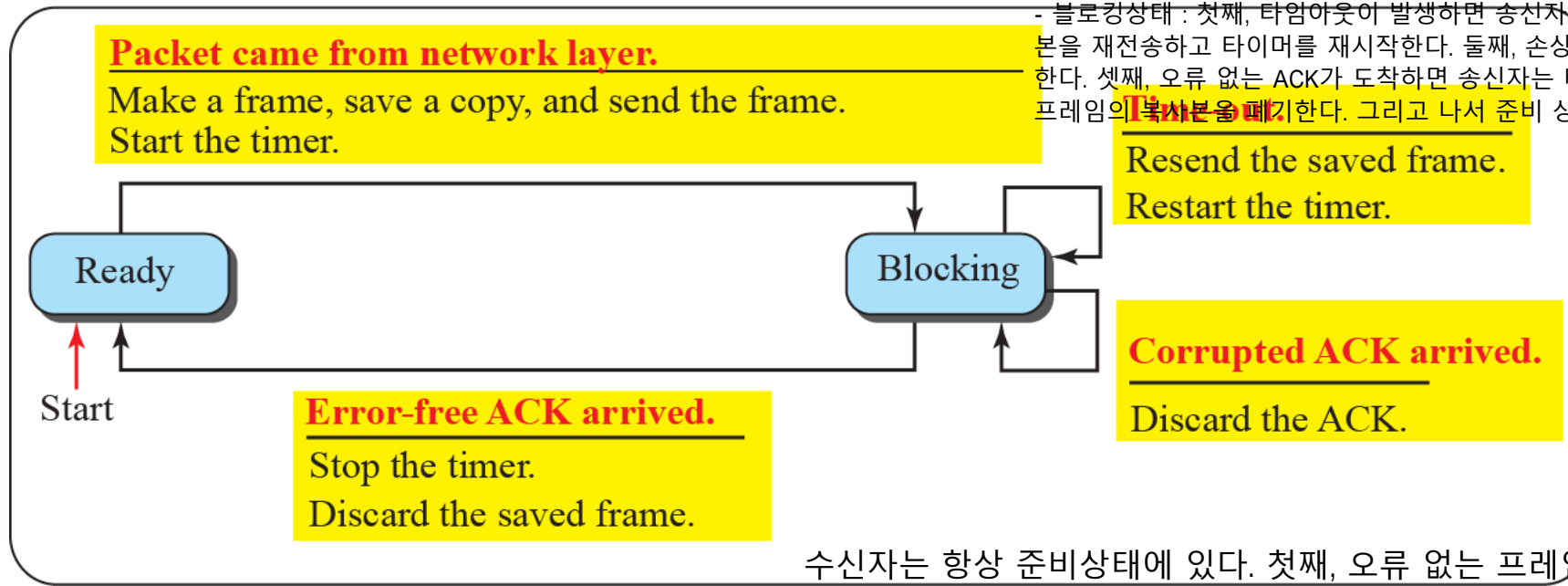
매 시간 채널에는 오직 하나의 프레임과 하나의 확인 응답만이 있을 수 있다.



**Figure 11.10: Stop-and-wait Protocol**

**Figure 11.11: FSM for the stop-and-wait protocol**

Sending node

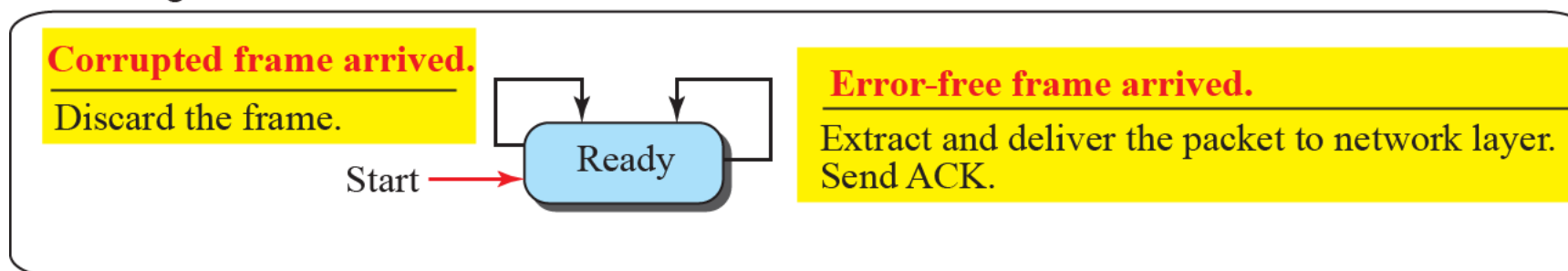


송신자는 처음에 준비 상태에 있다. 이것은 준비 상태와 블로킹 상태 사이로 이동한다.

- 준비상태 : 송신자는 네트워크층으로부터의 패킷을 기다린다. 패킷이 오면 프레임 생성하고 프레임의 복사본을 저장하며, 타이머를 시작하고 프레임을 보낸다. 그리고 블로킹상태로 이동한다.
- 블로킹상태 : 첫째, 타임아웃이 발생하면 송신자는 저장된 프레임의 복사본을 재전송하고 타이머를 재시작한다. 둘째, 손상된 ACK가 도착하면 폐기한다. 셋째, 오류 없는 ACK가 도착하면 송신자는 타이머를 멈추고 저장된 프레임의 복사본을 폐기한다. 그리고 나서 준비 상태로 이동한다.

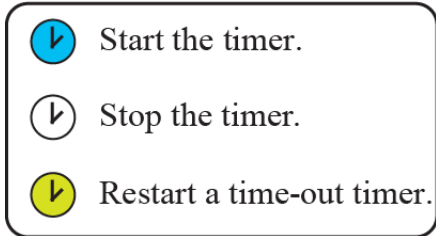
수신자는 항상 준비상태에 있다. 첫째, 오류 없는 프레임이 도착하면 프레임의 메시지는 네트워크층으로 전달되고 ACK가 보내진다. 둘째, 손상된 프레임이 도착하면 프레임은 폐기된다.

Receiving node



**Figure 11.12:** Flow diagram ( 데이터 또는 ACK 손실시? Stop-n-Wait )

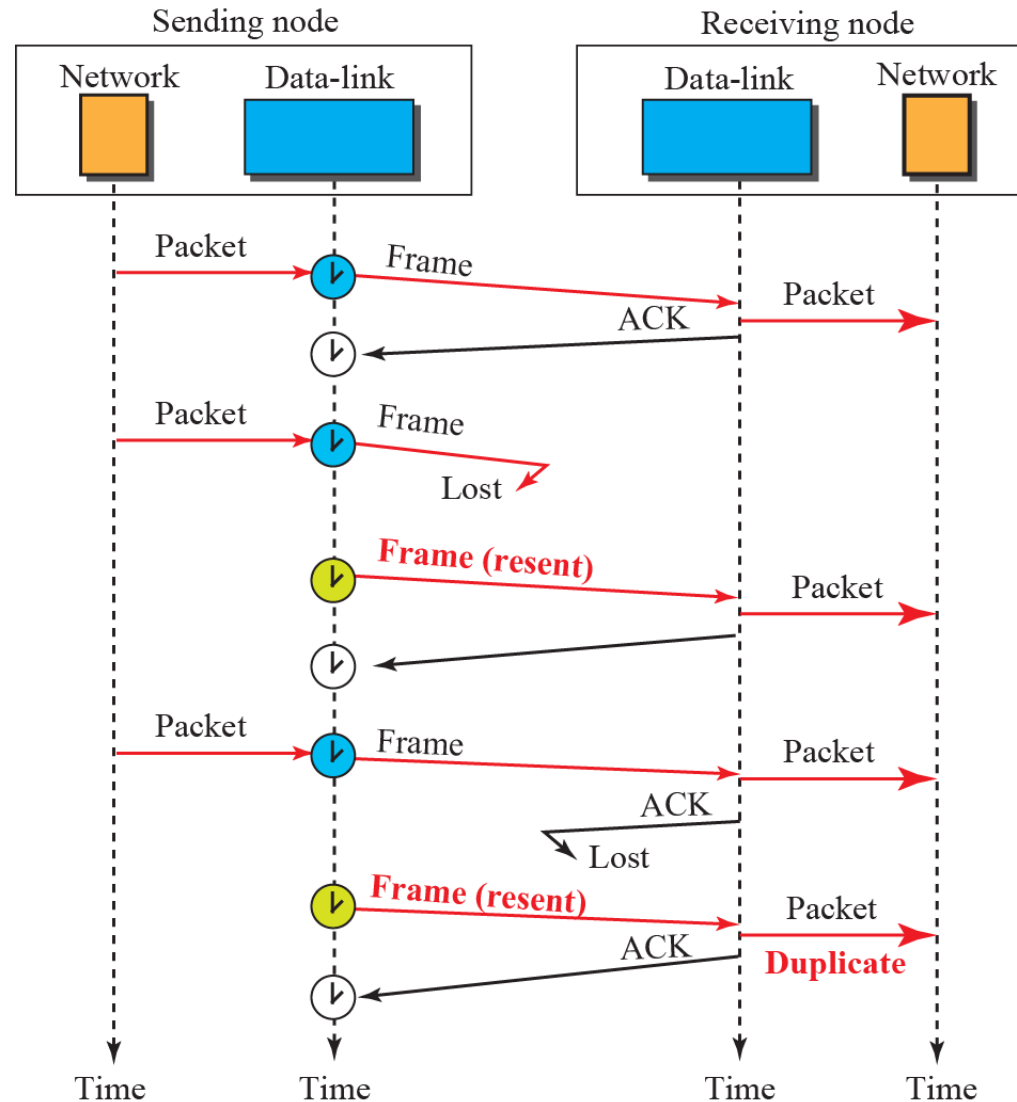
**Legend**



**Notes:**




A lost frame means either lost or corrupted.

A lost ACK means either lost or corrupted.



**Figure 11.13: Flow diagram: 중복 메시지 해결 방법, Stop-n-Wait**

**Legend**

-  Start the timer.
-  Stop the timer.
-  Restart a time-out timer.

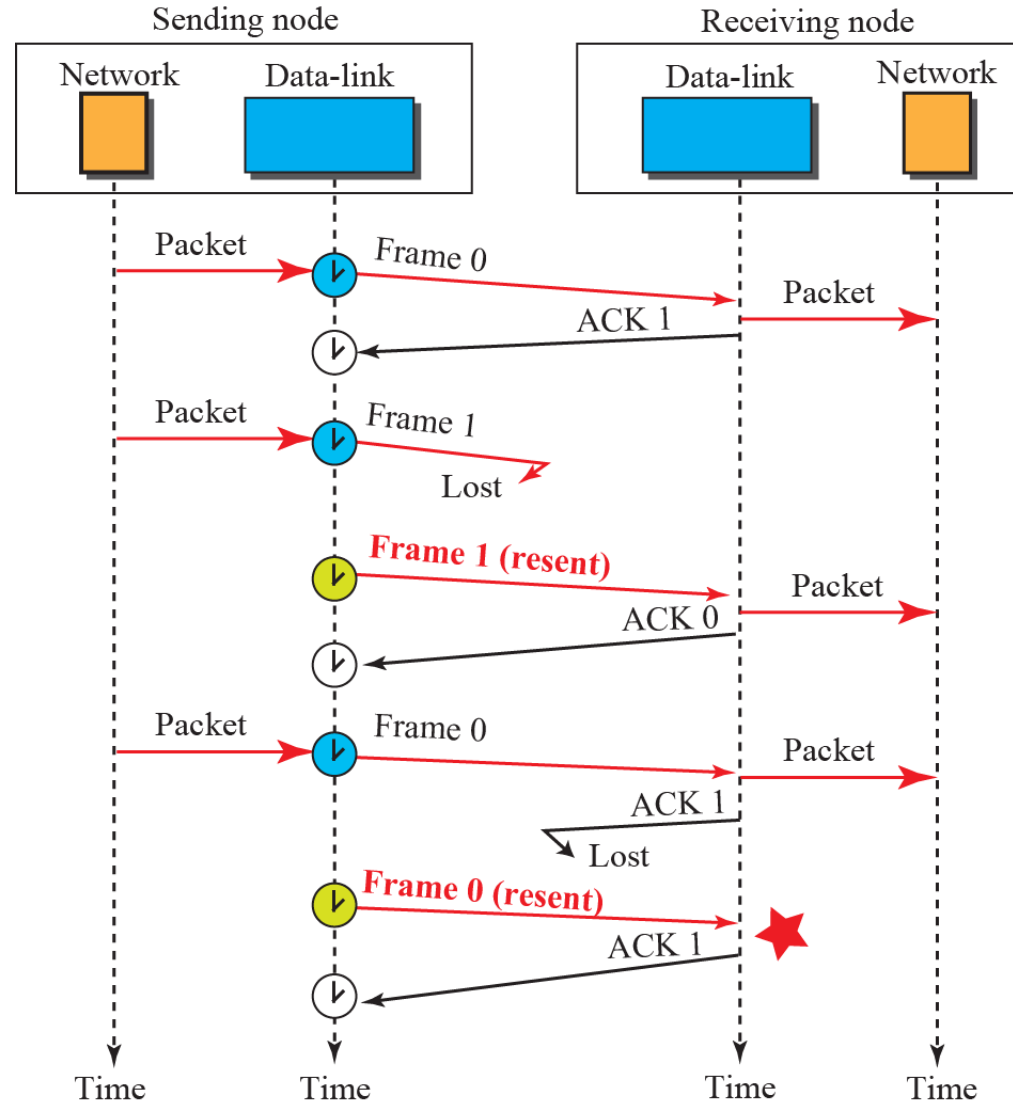
**Notes:**

A lost frame means either lost or corrupted.

A lost ACK means either lost or corrupted.



Frame 0 is discarded because the receiver expects frame 1.



# Stop-N-Wait ARQ 분석

- 동작은 하지만, 성능은 어떤가요?
  - 예: 1 Gbps link, 15 ms prop. delay, 8000 bit packet: Stop-N-Wait 적용시 처리율은 얼마인가요?

$$d_{trans} = \frac{L}{R} = \frac{8000\text{bits}}{10^9\text{bps}} = 8\text{microseconds}$$

$U_{\text{sender}}$ : **utilization** - fraction of time sender busy sending

$$U_{\text{sender}} = \frac{L / R}{RTT + L / R} = \frac{.008}{30.008} = 0.00027$$

- 즉, RTT=30 msec, 1KB pkt every 30 msec -> 33kB/sec thruput over 1 Gbps link
- 물리자원은 충분한데, 네트워크 프로토콜(소프트웨어)로 인하여 성능 저하!



# 좀더 좋은 성능을 내기 위해서는?

- 한번에 여러 개를 보내자.
- Go-back-N ARQ
  - 송신자는 ACK을 수신받지 않은 N개 패킷을 전송
  - 수신자는 누적(cumulative) ACK 전송
  - 가장 오래된 ACK 미수신 타이머를 유지: 타이머 종료시 ACK 미수신 패킷 모두 재전송
- Selective-Repeat ARQ
  - 수신자는 매 패킷마다 ACK
  - 송신자는 ACK 미수신 패킷 타이머 유지: 타이머 종료지 ACK 미수신 패킷만 재전송

# 송신자, 수신자 윈도우 크기와 순서번호

- 순서번호
  - m 비트로 프레임에 표시됨
- 송신자 윈도우 크기
  - SW: 0 or 1
  - GN: 0, 1, 2, ... ,  $(2^m - 1)$
  - SR: 0, 1, 2, ... ,  $(2^{m-1})$
- 수신자 윈도우 크기?

Figure 11.14 *Design of Go-Back-N ARQ*

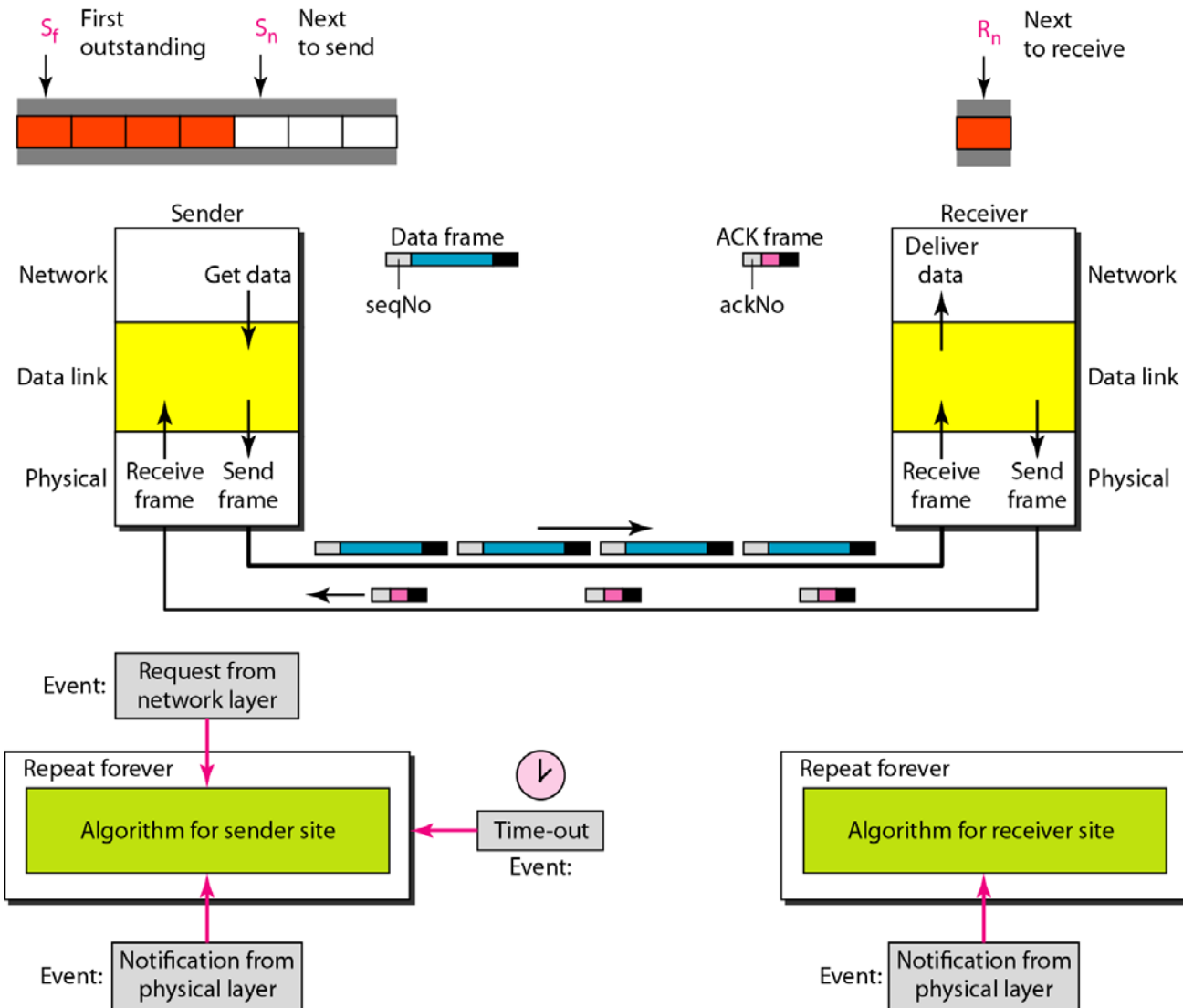
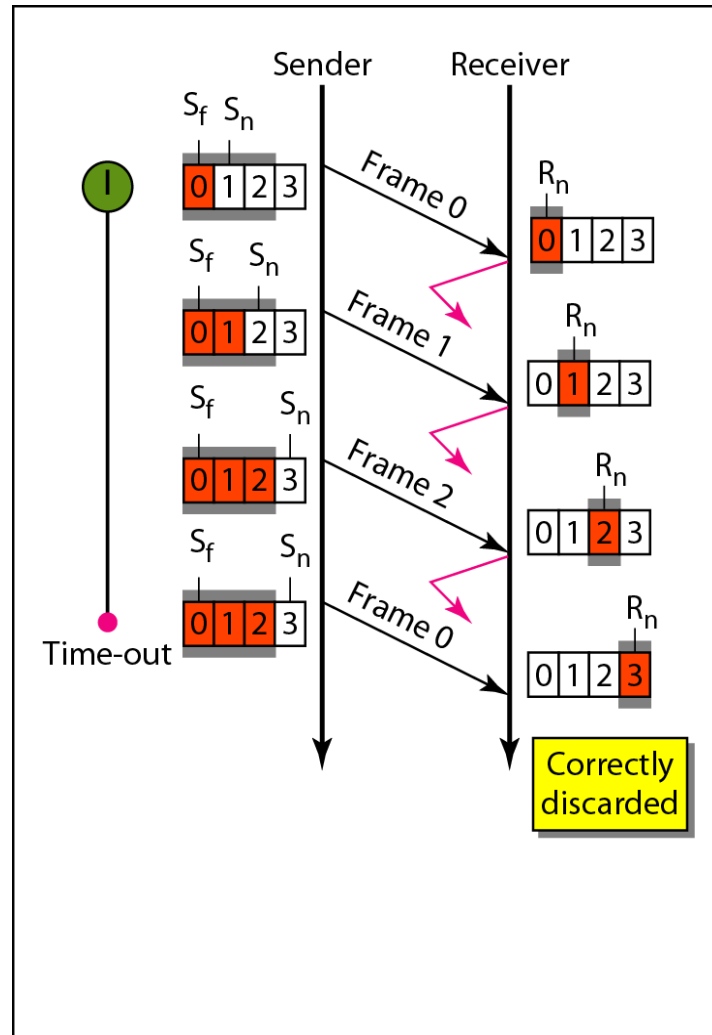
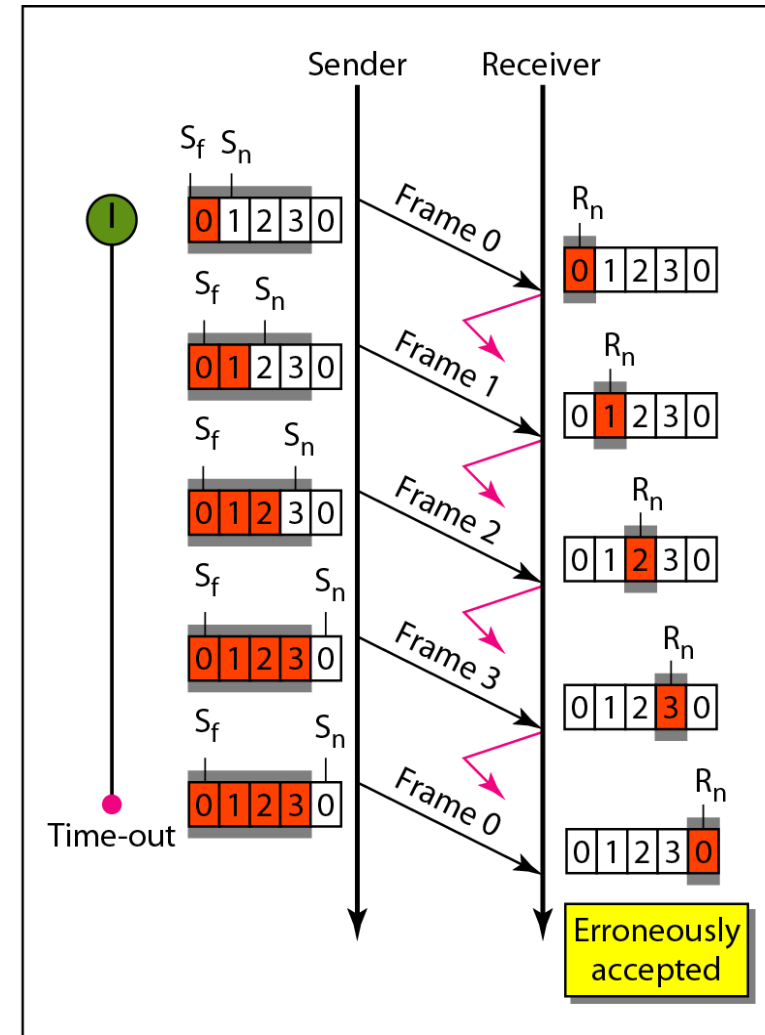


Figure 11.15 Window size for Go-Back-N ARQ



a. Window size  $< 2^m$



b. Window size  $= 2^m$

수신자는 새로운 0번을 기다리고 있는데 송신자는 기존의 0번을 보냄(오류)  
window size가  $2^m$ 개이기 때문.

Figure 11.17 Flow diagram: 데이터 손실, Go-back-N ARQ

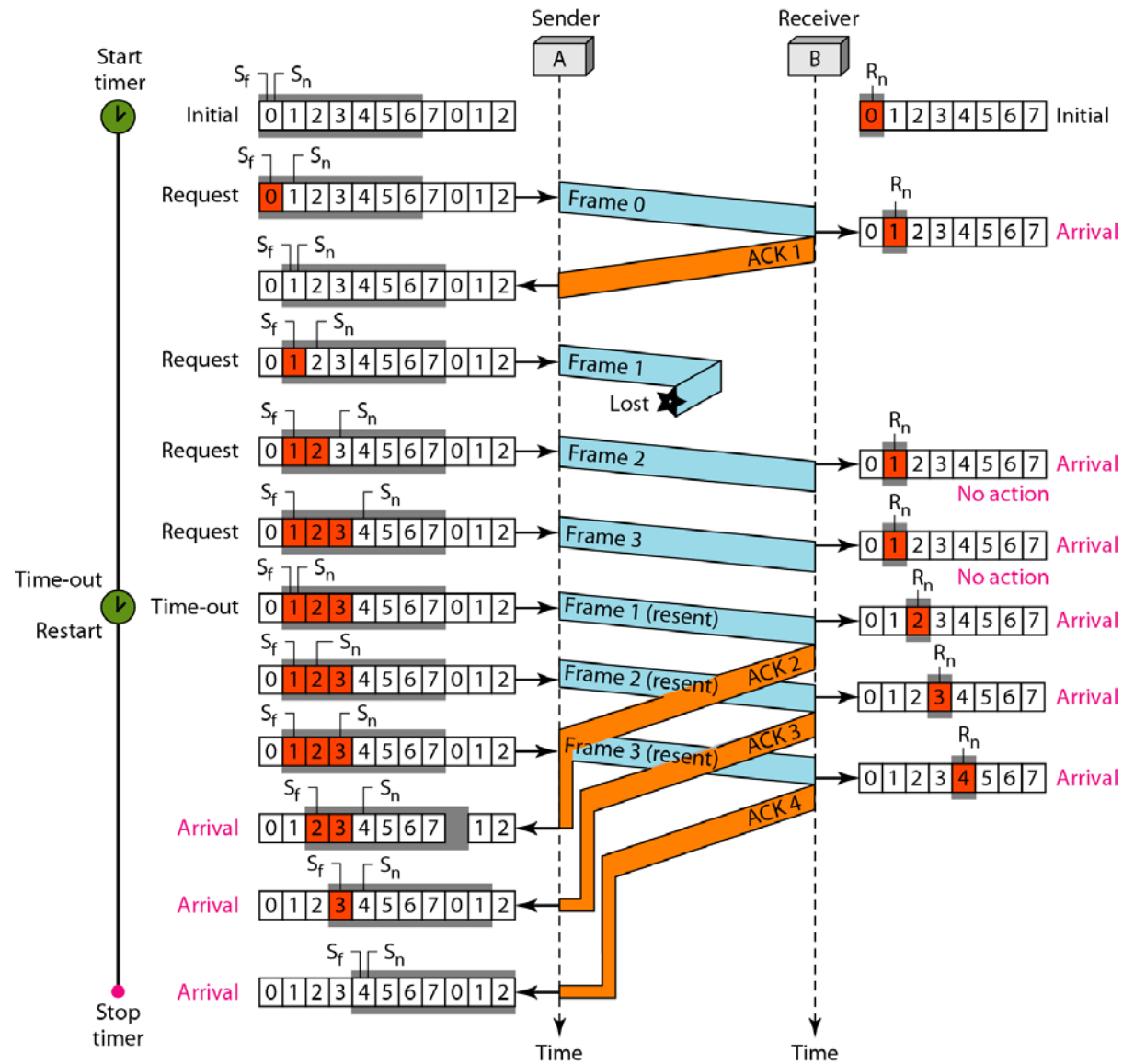


Figure 11.20 *Design of Selective Repeat ARQ*

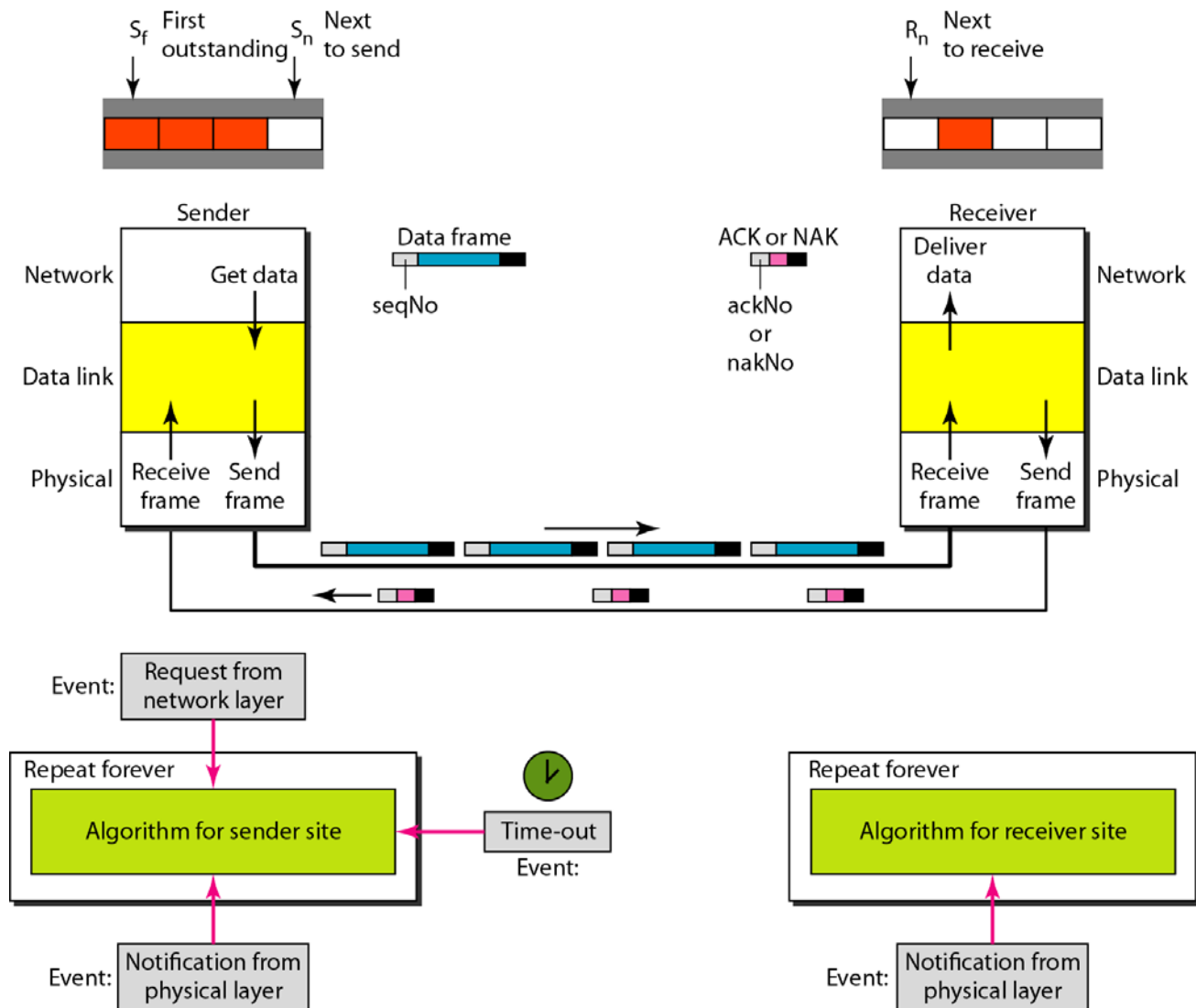
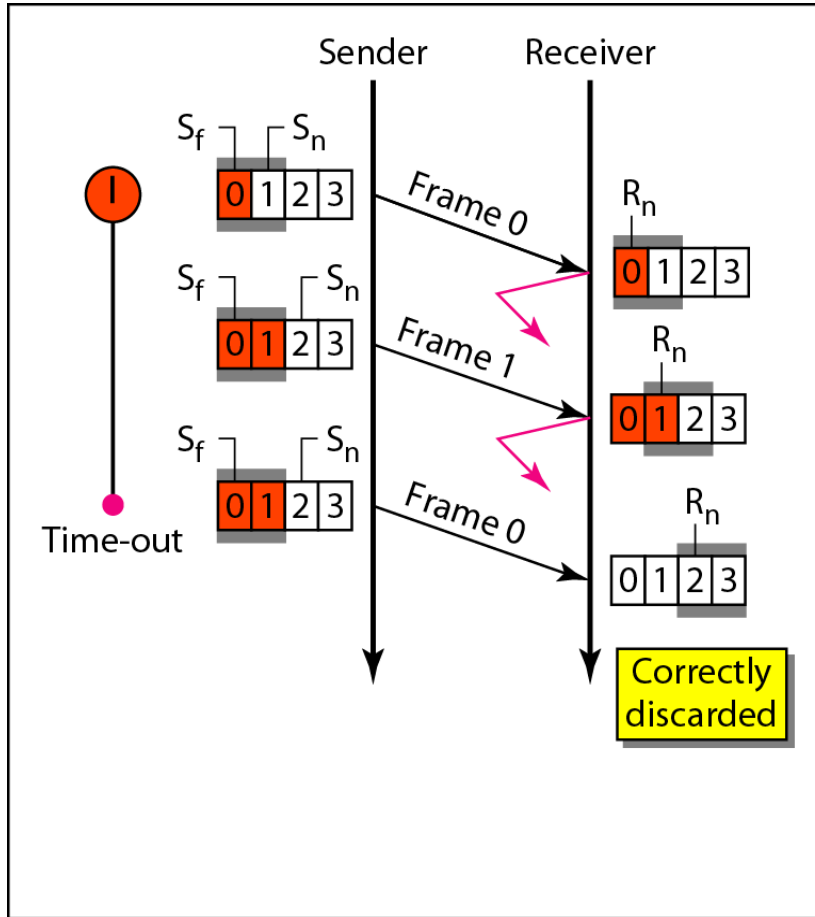
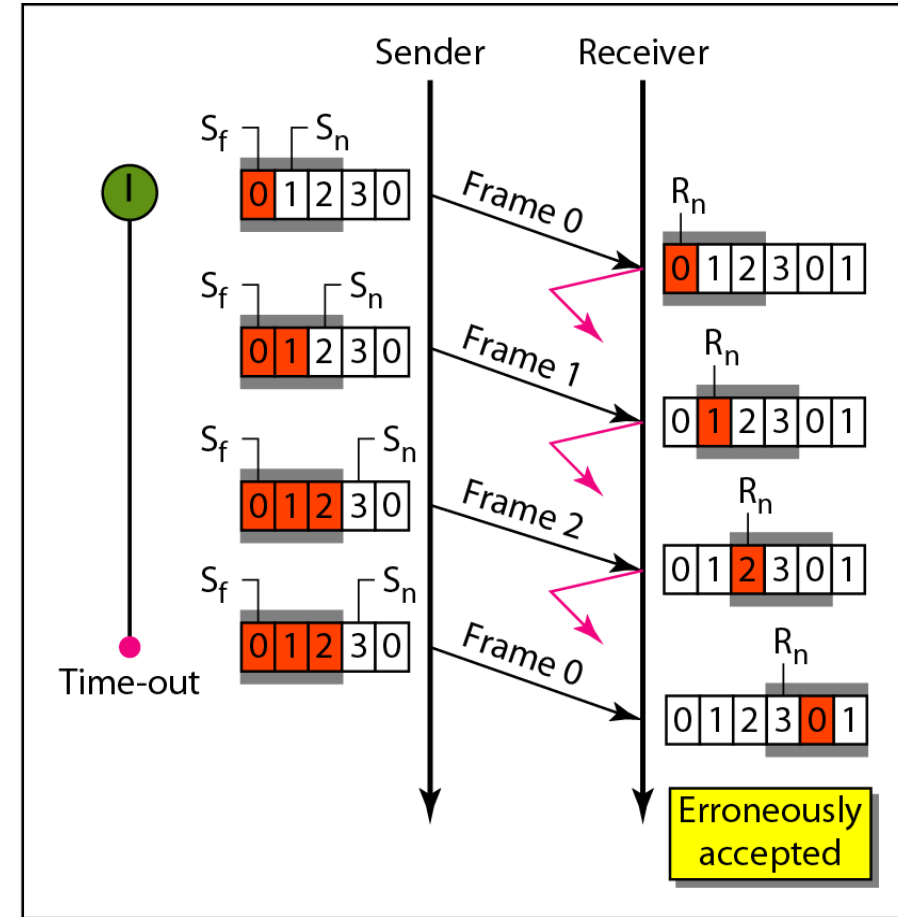


Figure 11.21 *Selective Repeat ARQ, window size*



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



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

수신자는 새로운 0번을 기다리지만 송신자는 기존의 0번을 보냄.  
window size가  $2^m - 1$ 보다 크기 때문.

Figure 11.23 Flow diagram: 데이터 손실, Selective-Repeat ARQ

