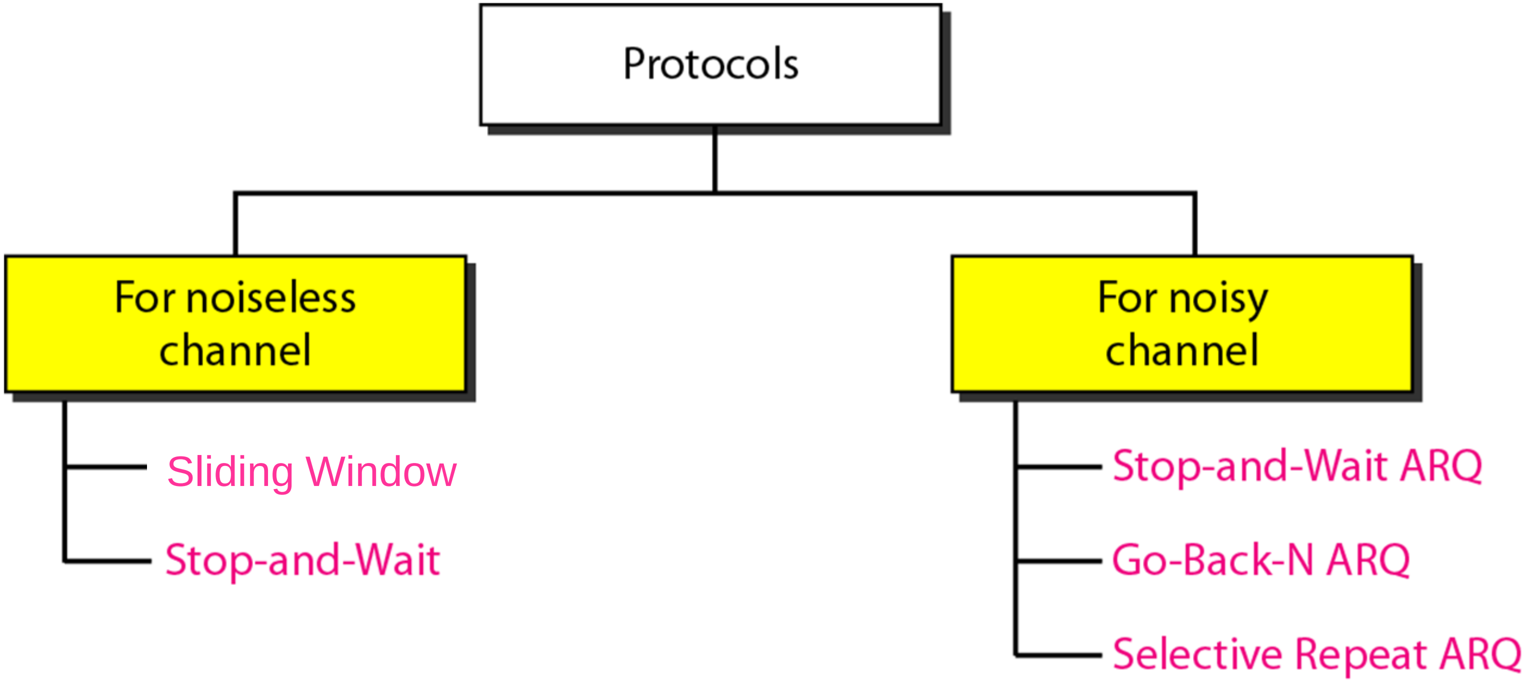
Content Outline (Unit-4)

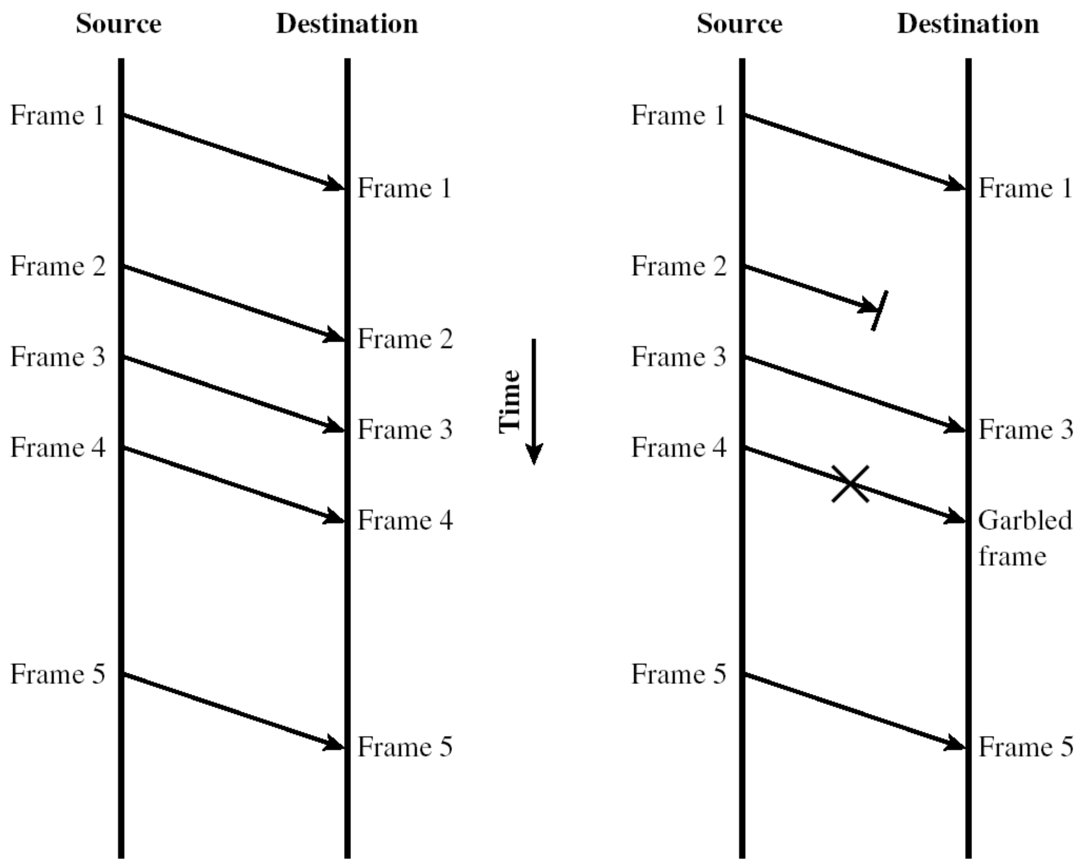
- **Flow Control**
  - Stop and Wait Flow Control
  - Sliding Window Flow Control

- **Error Control**
  - Automatic Repeat Request (ARQ)
    - Stop and wait ARQ
    - Go-back-N ARQ
    - Selective repeat ARQ

Flow Control and Error Control



Model of Frame Transmission



(a) Error-free transmission

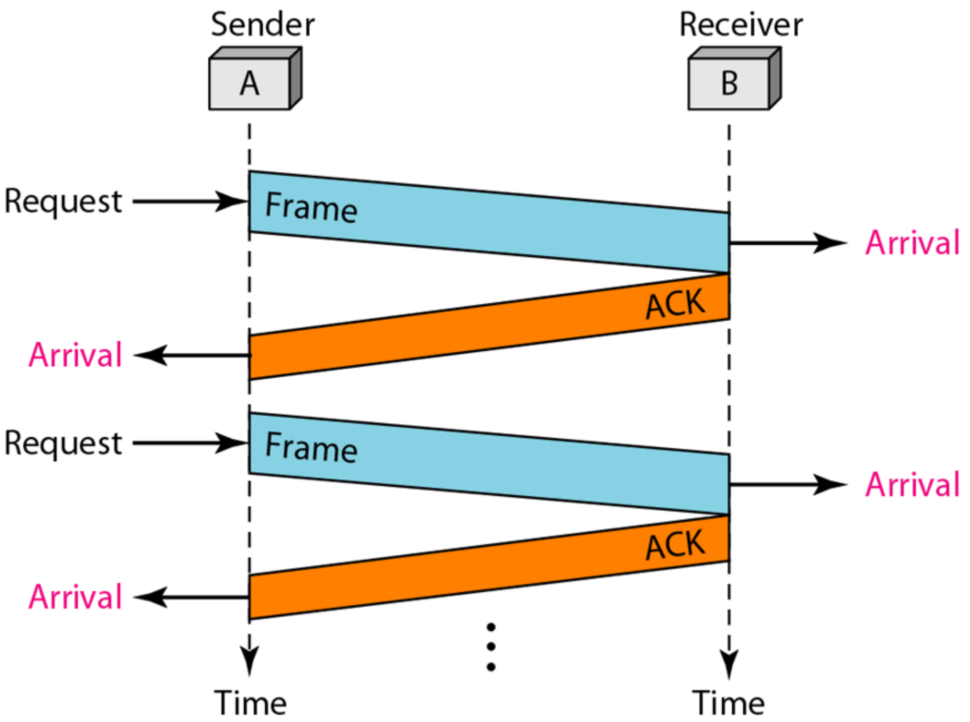
(b) Transmission with losses and errors

Flow Control

- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.
- Ensuring the sending entity does not overwhelm the receiving entity
  - Preventing buffer overflow
- Influenced by:
  - *Transmission time*
    - Time taken to emit all bits into medium
  - *Propagation time*
    - Time for a bit to traverse the link

Stop and Wait Flow Control

- Source transmits frame
- Destination receives frame and replies with *acknowledgement (ACK)*
- Source waits for *ACK* before sending next frame
- Destination can stop flow by not sending *ACK*
- Works well for a few large frames



Stop and Wait Link Utilization

- ***Link Utilization (Normalized Throughput)***
  - Definition: - the fraction of time the channel is used to transmit useful data bits
  - Mathematically if T_D seconds is the amount of time the channel transmits useful data bits out of a total of T seconds examined

$$U = \frac{T_D}{T}$$

Stop and Wait Link Utilization

- For Stop and Wait flow control:
  - $T_D = t_{\text{frame}}$ (frame transmission time)
  - $T = 2\,t_{\text{prop}} + t_{\text{frame}}$
- Link Utilization:

$$U = \frac{T_D}{T} = \frac{t_{frame}}{t_{frame} + 2t_{prop}}$$

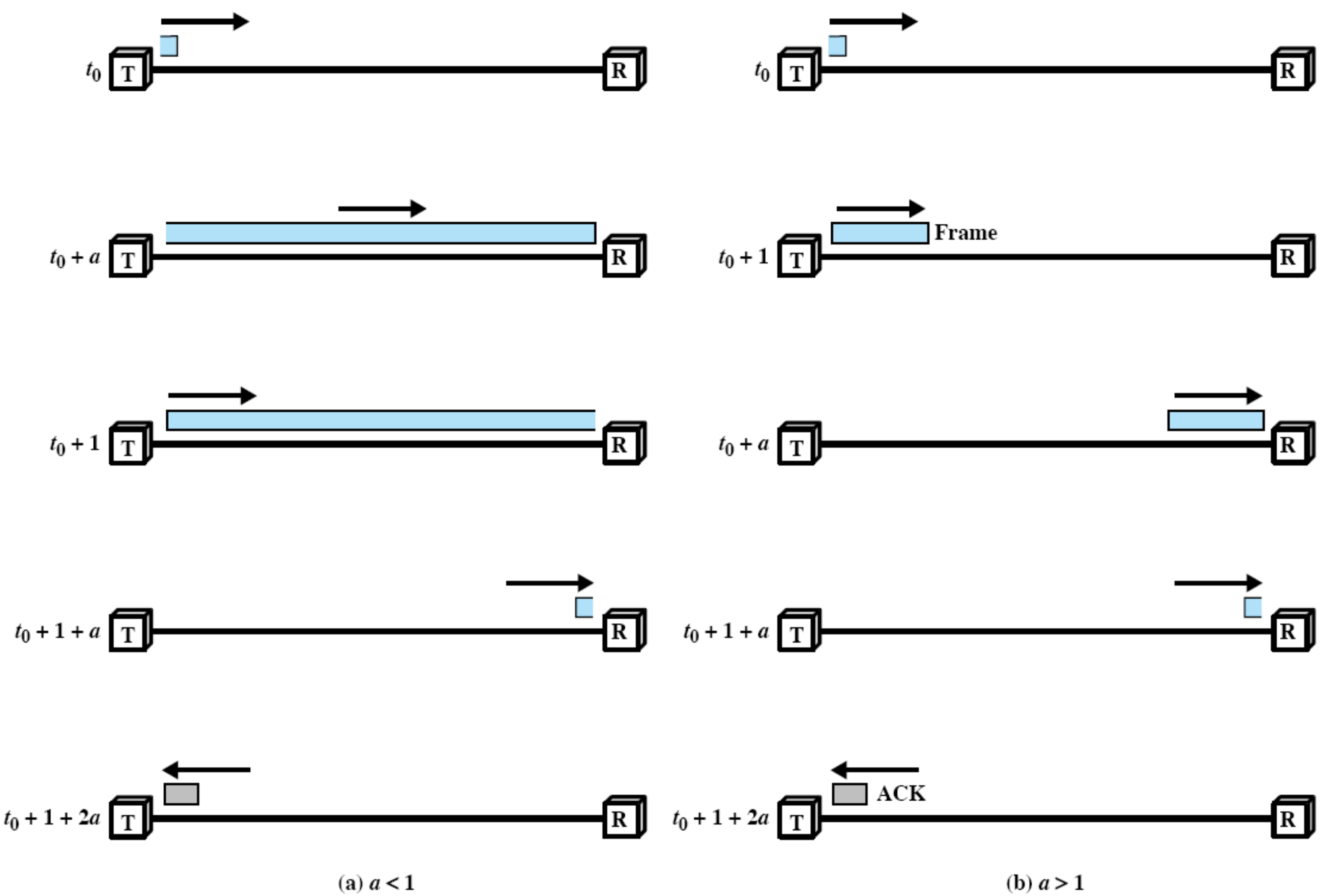
$$= \frac{1}{1+2a}$$

Assume ACK is short and hence the transmission time of ACK is neglected

where

$$a = \frac{\text{Propagation Time}}{\text{Transmission Time}} = \frac{t_{prop}}{t_{frame}}$$

Stop and Wait Link Utilization



(Transmission time for ACK is neglected)

(transmission time = 1; propagation time = a)

Sliding Window Flow Control

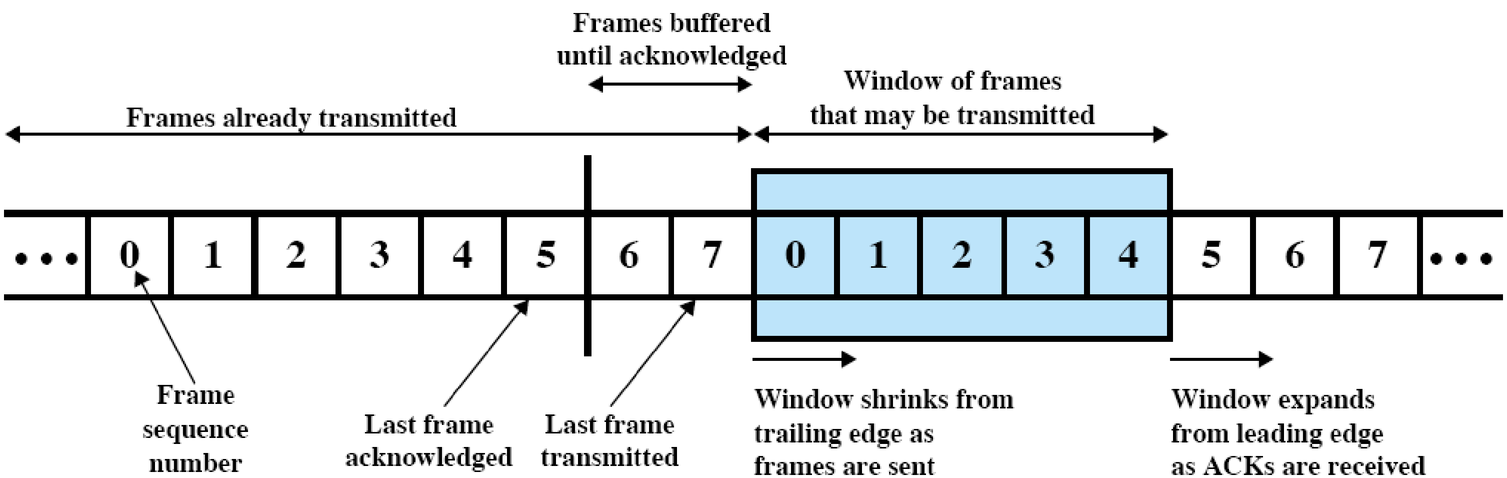
- ***Key ideas***:

- Allow multiple frames to be in transit
- Receiver has a buffer of storing W frames
- Transmitter can send up to W frames without *ACK*
- Each frame is numbered
- *ACK* includes the sequence number of the next frame expected
- Sequence number bounded by size of field (k)
  - Frames are numbered modulo 2^k

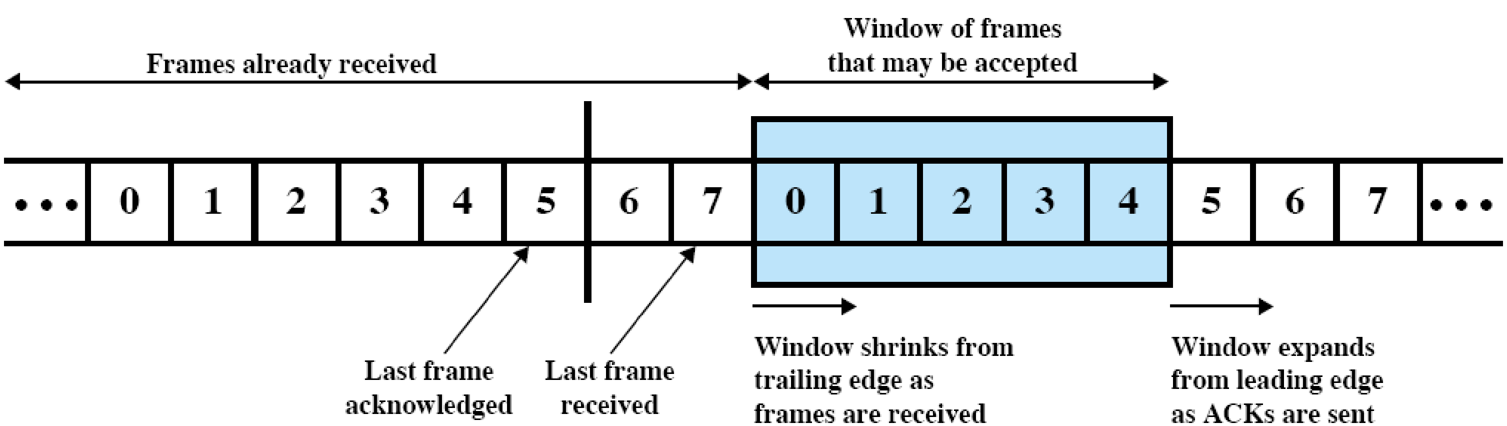
Sliding Window Flow Control

- For full-duplex transmission, when a data frame arrives, receiver DLL (Data Link Layer) waits until its NL (Network Layer) passes it the next packet.
- At any instant, sender maintains a list of consecutive *Sequence Numbers (SNs)* corresponding to frames that are allowed to send. These frames are said to fall within the sending window.
- Receiver maintains receiving window corresponding to frames it is allowed to accept.

Sliding Window Diagram

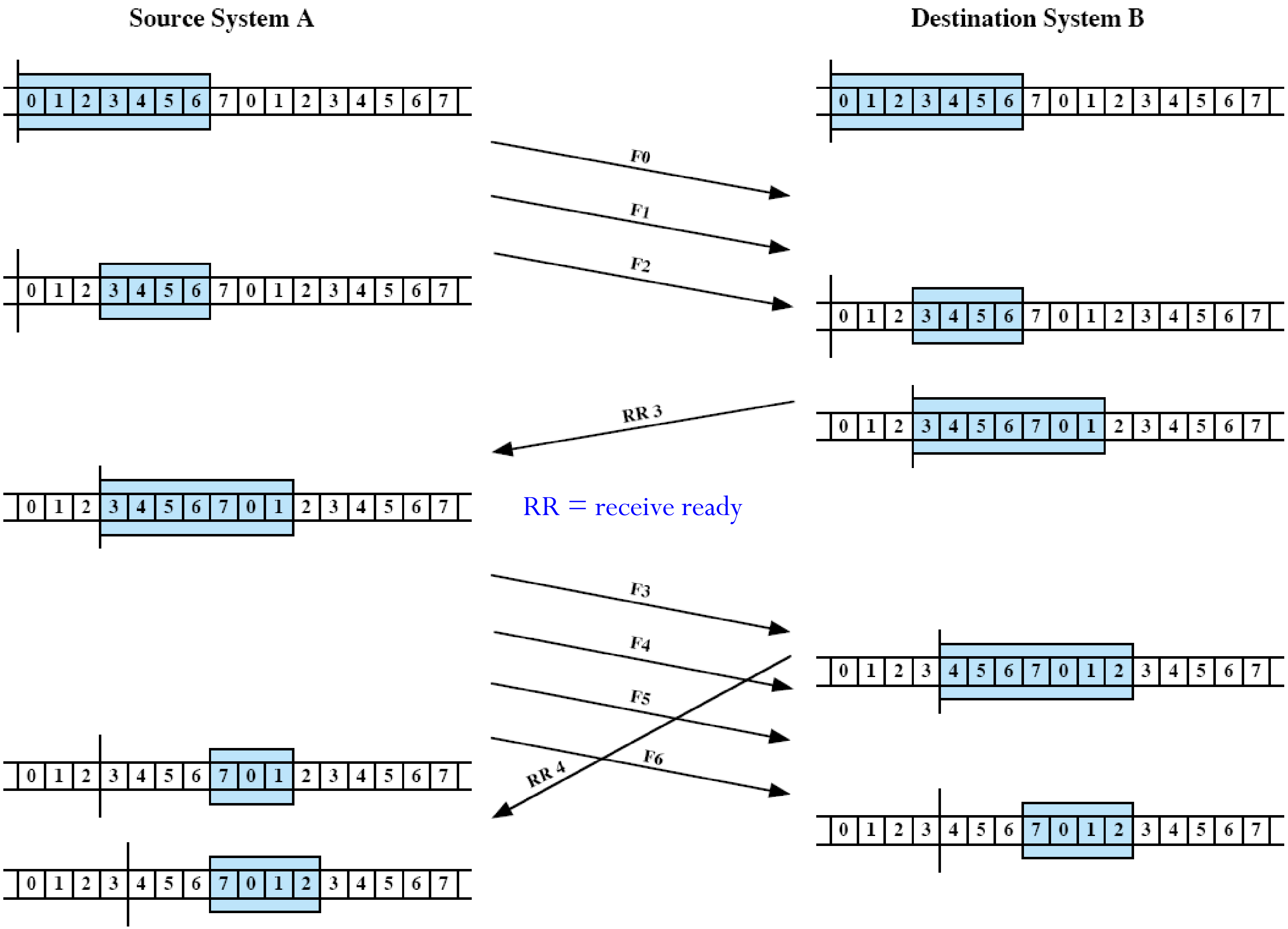


(a) Sender's perspective



(b) Receiver's perspective

Example of Sliding Window



Link Utilization for Sliding Window Flow Control

- For error-free sliding window flow control, the throughput on the line depends on both the window size W and the value of a.

❖ For convenience, transmission time = 1

propagation time = a

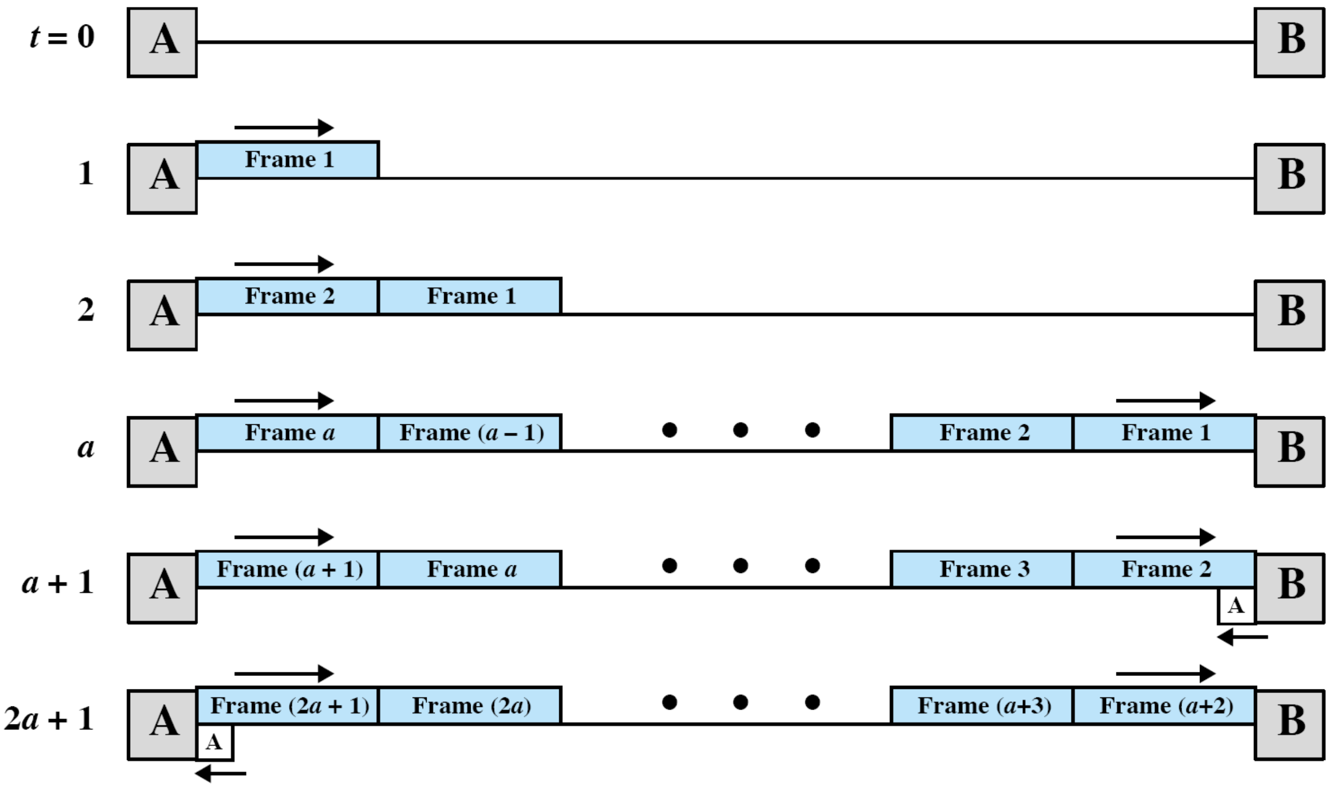
❖ *Consider a full duplex point-to-point line:*

❖ *The sender begins to transmit at time $t = 0$, then the ACK for the first frame reaches it at $t = 2a+1$. (Ignore the transmission time of the ACK frame)*

Link Utilization for Sliding Window Flow Control

Case 1: $W \geq 2a + 1$

The acknowledgement for frame 1 reaches A before A has exhausted its window. Thus A can transmit continuously with no pause and **utilization is 1**.

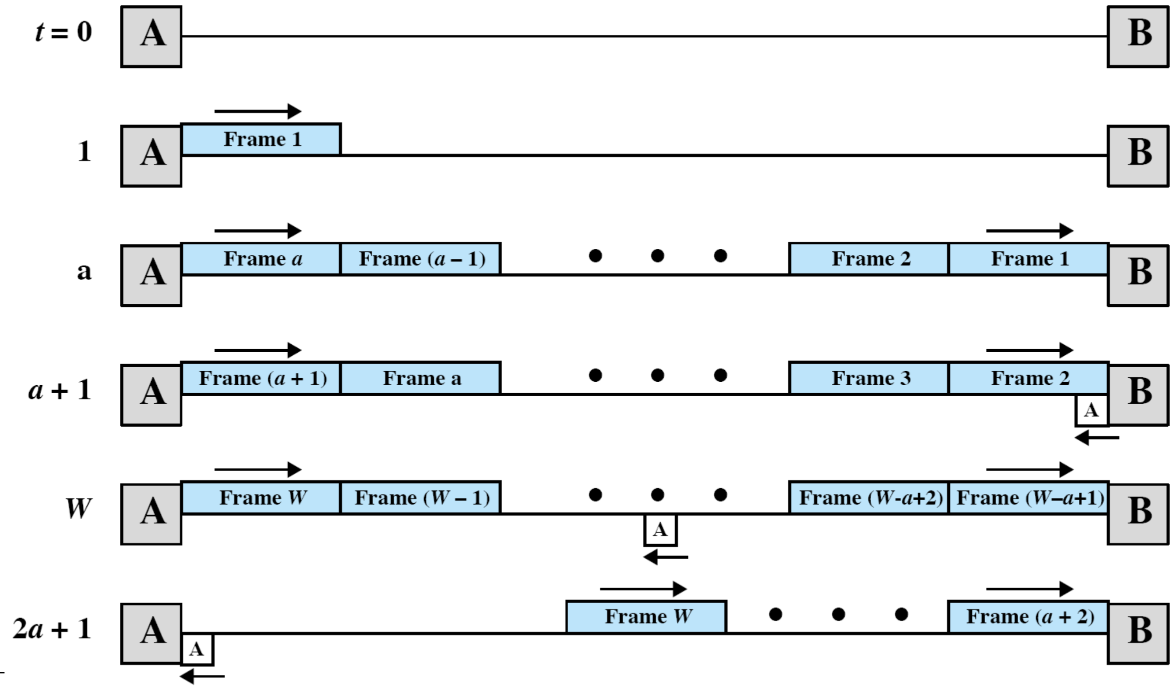


Link Utilization for Sliding Window Flow Control

Case 2: $W < 2a + 1$

- A exhausts its window at $t = W$ and cannot send additional frames until $t = 2a+1$. Hence

  Utilization = $W/(2a+1)$

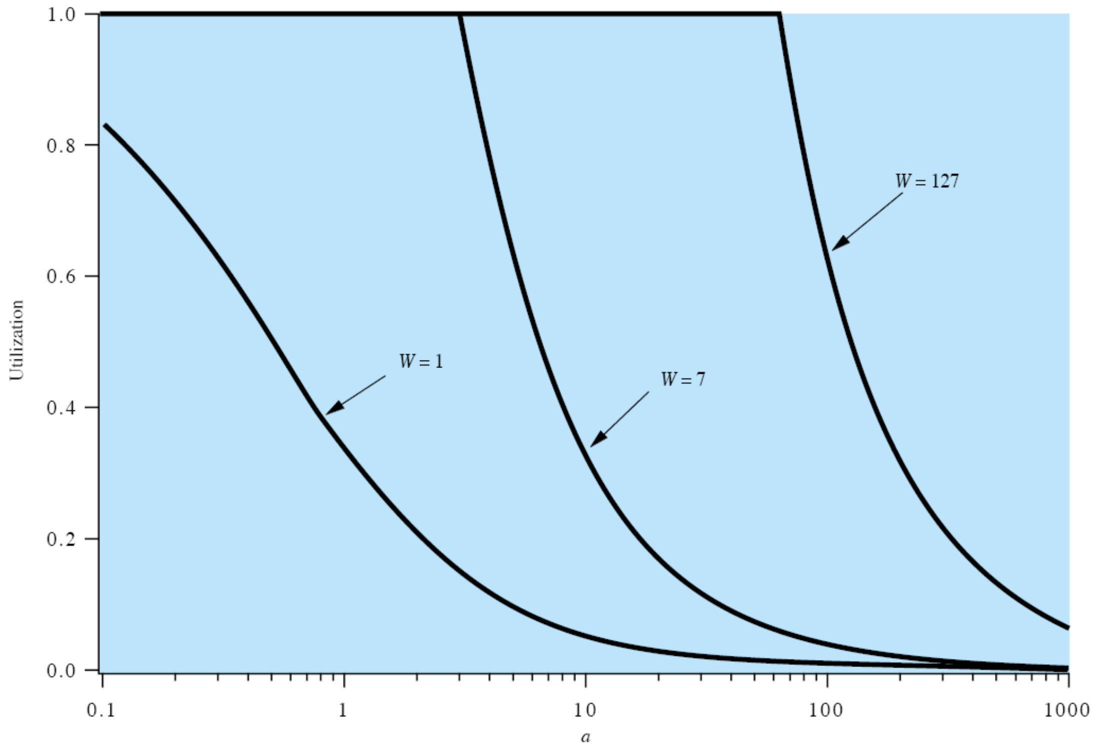


Link Utilization for Sliding Window Flow Control

- Link utilization as a function of *a:*

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

$$a = \frac{\text{Propagation Time}}{\text{Transmission Time}} = \frac{t_{prop}}{t_{frame}}$$

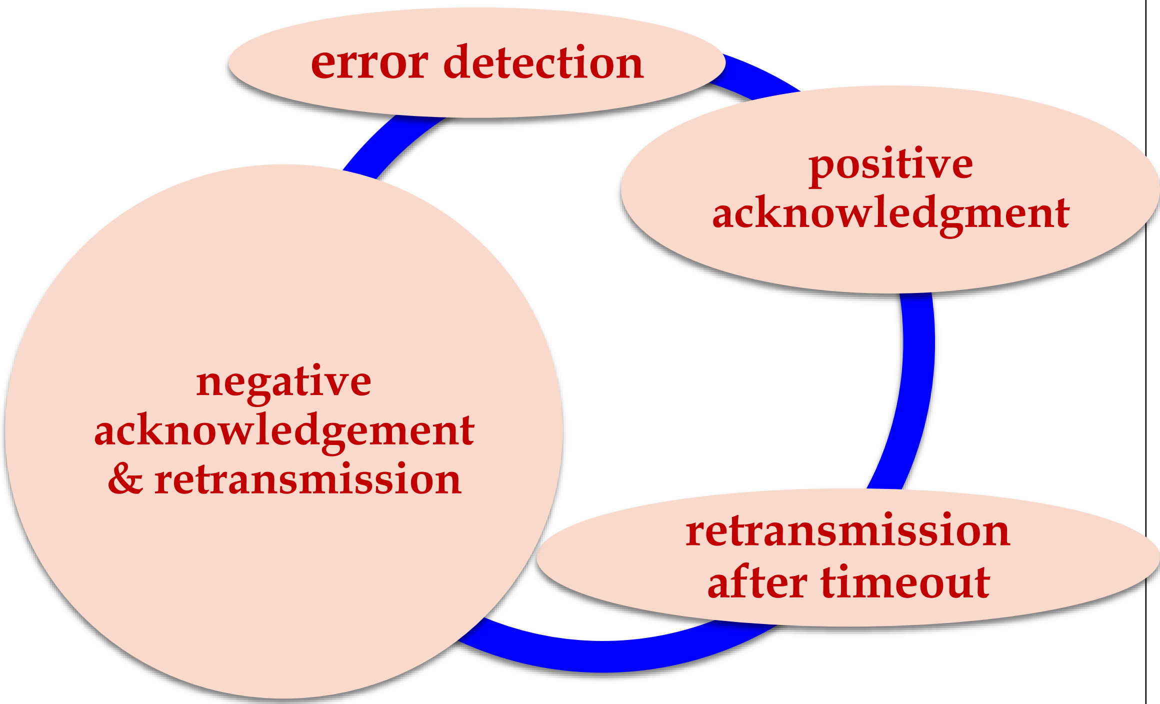


Sliding Window Enhancements

- Receiver can acknowledge frames without permitting further transmission (Receive Not Ready)
- Must send a normal acknowledge to resume
- For full-duplex, use piggybacking
  - If no data to send, use acknowledgement frame
  - If data but no acknowledgement to send, send last acknowledgement number again, or have ACK valid flag (TCP)

Error Control

- Detection and correction of errors
  - Lost frames: a frame fails to arrive at the other side
  - Damaged frames: frame arrives but some of the bits are in error



Automatic Repeat Request (ARQ)

- A collective name for error control mechanisms.
- Effect of ARQ is to turn an unreliable data link into a reliable one
- ARQ – automatic retransmit the frame(s) if ACK doesn’t come back within a fixed period of time
- Different versions of ARQ are:
  - stop-and-wait
  - go-back-N
  - selective-repeat (selective-reject / selective-retransmission)

Stop and Wait ARQ

- The source transmits a single frame, must wait for *ACK*

- ***Two sorts of errors could occur:***

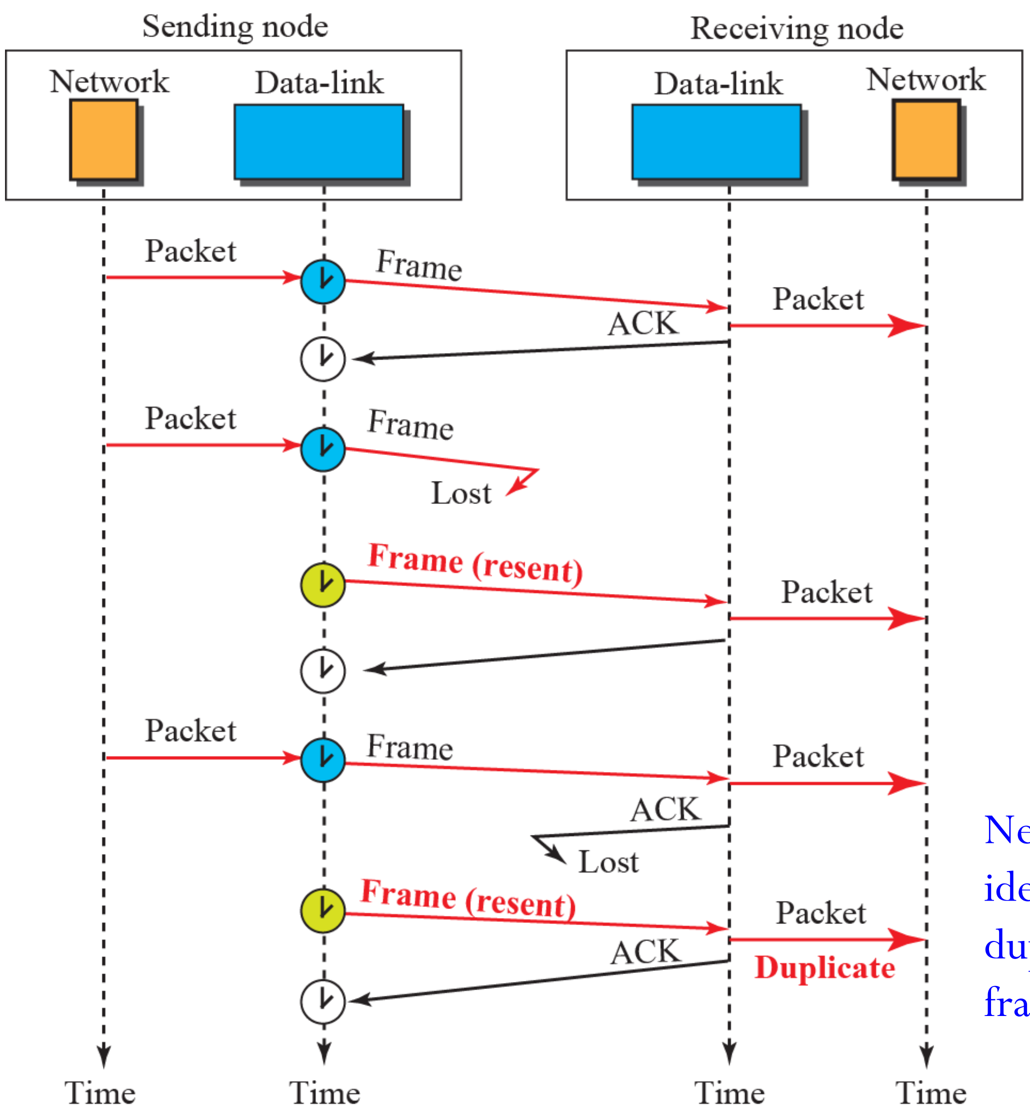
  - If received frame damaged, discard it
  - Transmitter has a timer.
  - If no *ACK* and the timer timeout, retransmit

  - If *ACK* damaged, transmitter will not recognize it
  - Transmitter will retransmit after timeout
  - Receiver gets two copies of frame
    - Use *ACK*0 and *ACK*1

Stop and Wait ARQ

Legend

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



Notes:

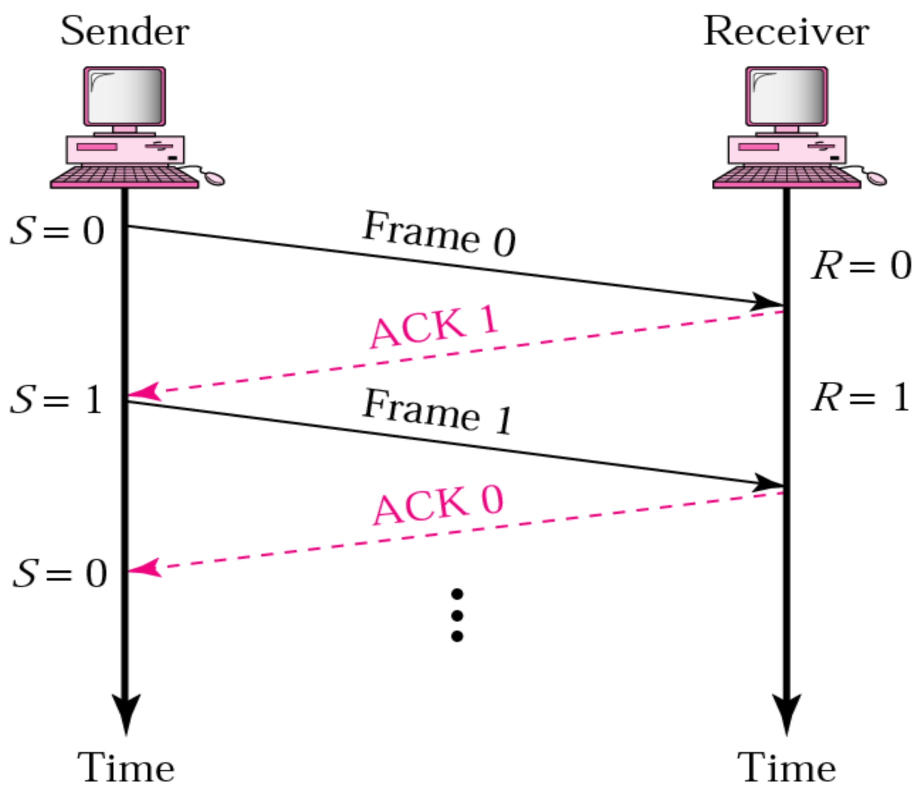
A lost frame means either lost or corrupted.

A lost ACK means either lost or corrupted.

Needs to identify the duplicated frame

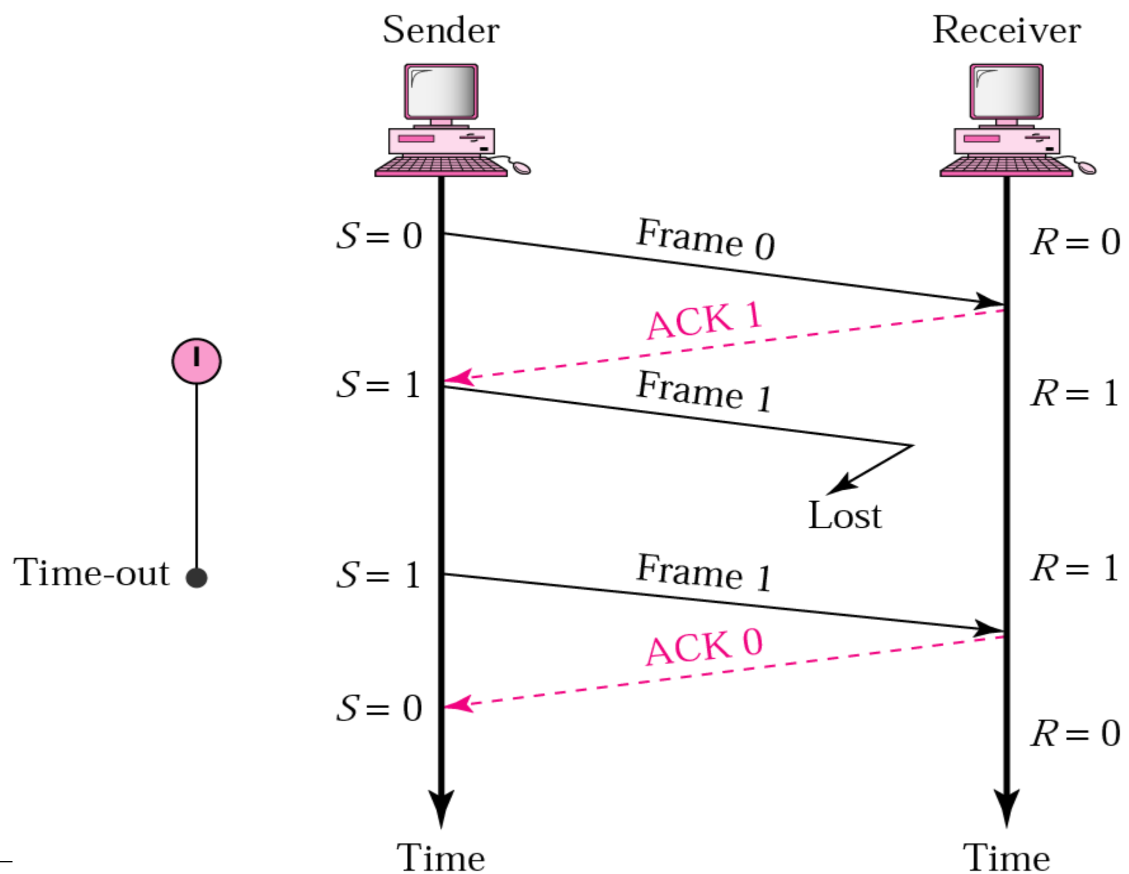
Stop and Wait ARQ

- **Normal Operation:**



Stop and Wait ARQ

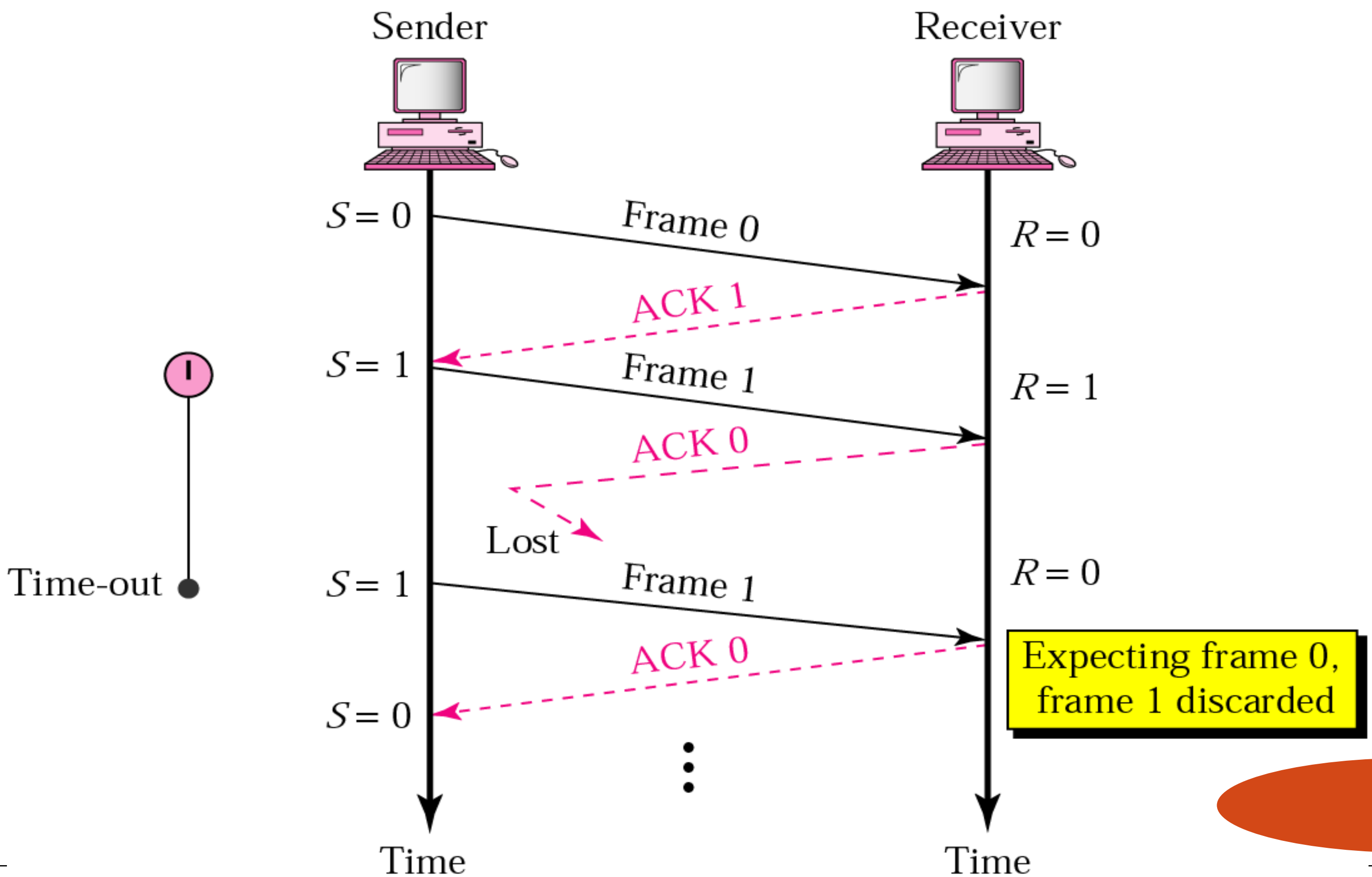
- **Lost frame:**



Note: The sender needs to maintain a copy of a transmitted frame until ACK is received.

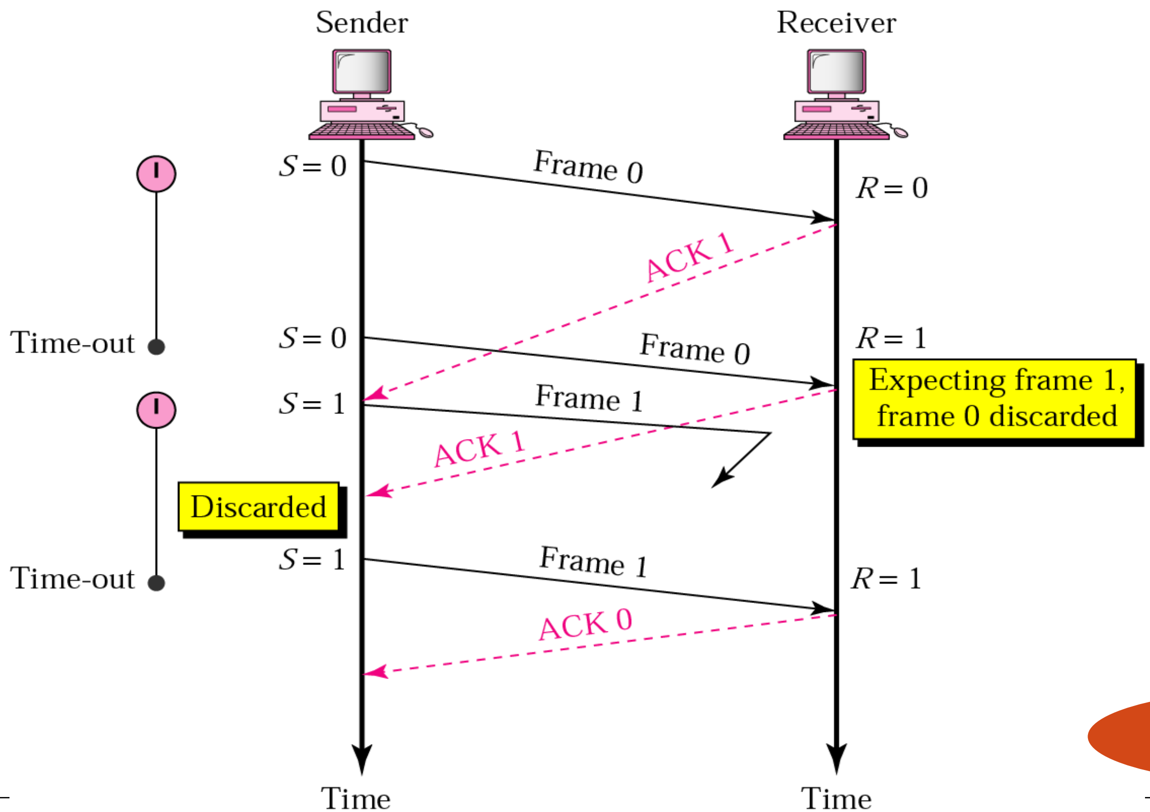
Stop and Wait ARQ

- **Lost ACK:**



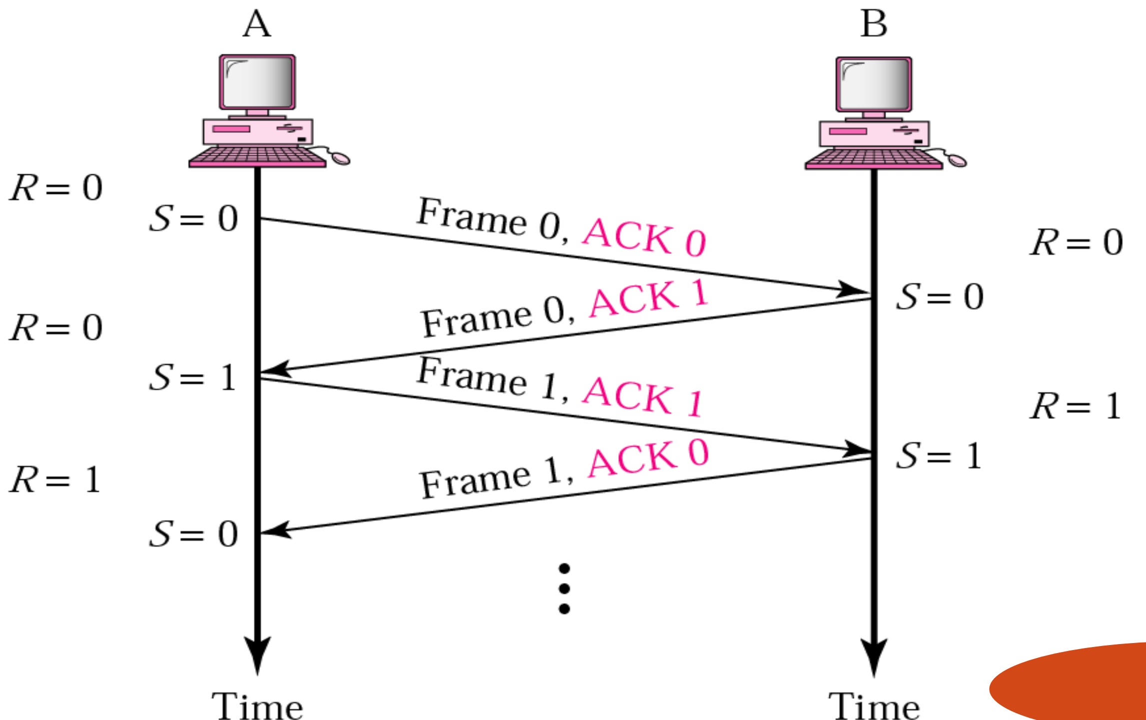
Stop and Wait ARQ

- **Delayed ACK:**
  - Numbered acknowledgments are needed if an acknowledgment is delayed and the next frame is lost.

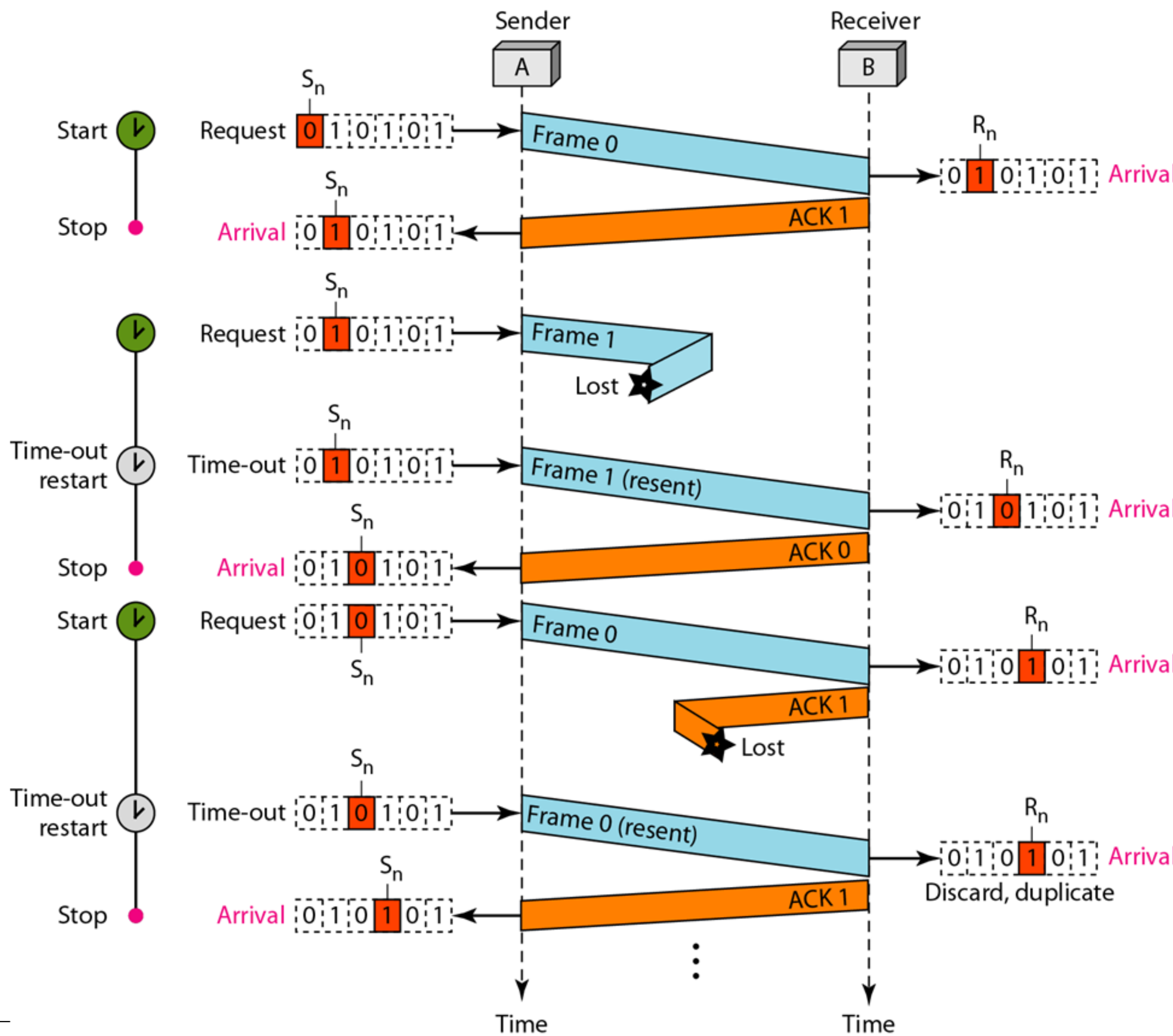


Stop and Wait ARQ

- **Piggybacking**
  - The data in one direction is piggybacked with the acknowledgment in the other direction.



Example: Stop and Wait ARQ



Performance of Stop-and-Wait ARQ

$$\text{Utilization} = \frac{\text{Time for transmitter to emit a single frame}}{\text{Total time that line is engaged in the transmission of a single frame}}$$

- For Error Free

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

- With Error

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

where N_r is the expected number of transmissions of a frame

Stop and Wait ARQ

- Let ***P*** be the probability that a single frame is in error.
- Assume that ACKs and NAKs are never in error, the probability that it will take exactly ***k*** attempts to transmit a frame successfully is $\boldsymbol{P^{k-1}(1-P)}$

$$N_r = \mathrm{E}(\text{transmissions}) = \sum_{i=1}^{\infty} (i \times P_r[i \text{ transmissions}])$$

$$= \sum_{i=1}^{\infty} (iP^{i-1}(1-P)) = \frac{1}{1-P}$$

- The Utilization for Stop and Wait ARQ is

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

Go-Back-N ARQ

- Based on sliding window flow control
  - Sender sends up to W frames before worrying about acknowledgements
  - It keeps a copy of these frames
- If no error, ACK will be sent as usual with next frame expected
- If error, reply with rejection
  - destination will discard that frame and all future frames until frame in error is received correctly
  - transmitter must go back and retransmit that frame and all subsequent frames
- Frames are numbered sequentially
  - Use *m* bits to identify the sequence number of each frame
  - If m is 3, the sequence numbers are 0 to 7

Go Back N - Handling

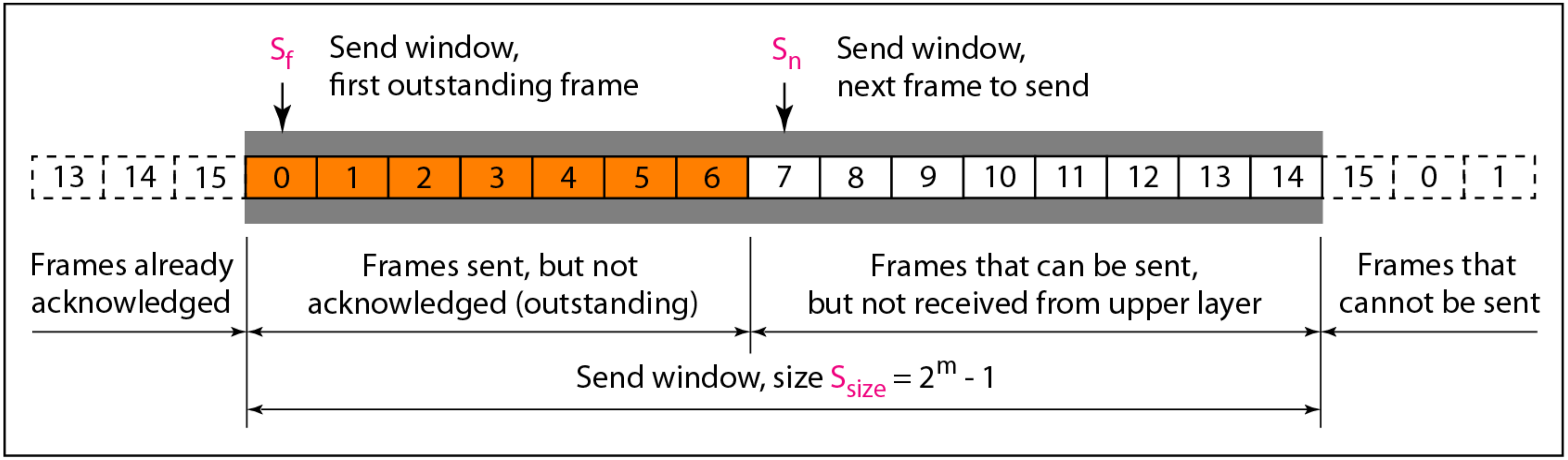
- **Damaged frame**
  - error in frame *i* so receiver discards frame *i*
  - Receiver will either
    - do nothing and wait until the transmitter times out
    - send a **Reject(i)** signal to the transmitter
  - Transmitter retransmits frames from *i*

Go Back N - Handling

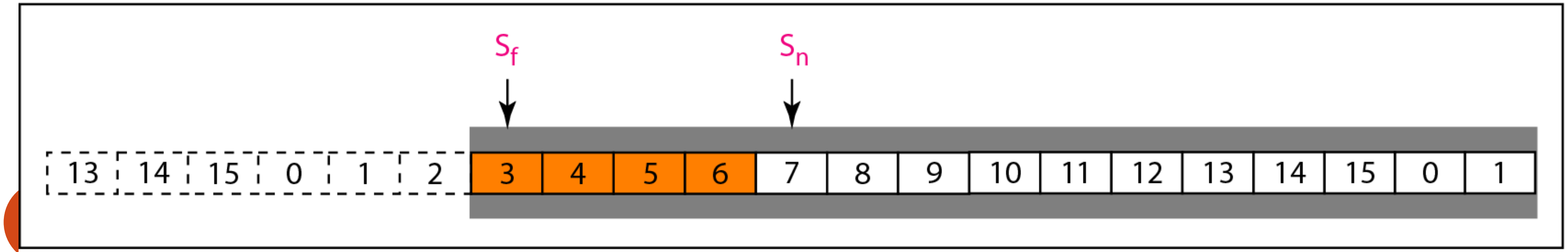
- **Lost frame**
  - frame i is lost and either
  - transmitter sends $i+1$ and receiver gets frame $i+1$ out of sequence and discards frame $i+1$. Receiver will either
    - do nothing and wait until the transmitter times out
    - send a **Reject(i)** signal to the transmitter
  - transmitter then retransmits frames from i

Sender Sliding Window

- Hold the outstanding frames until they are acknowledged



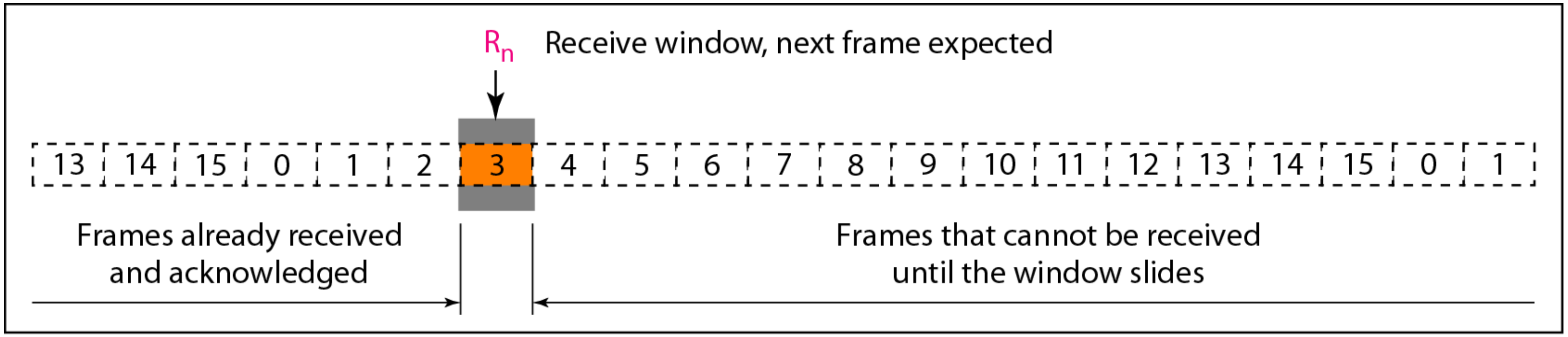
a. Send window before sliding



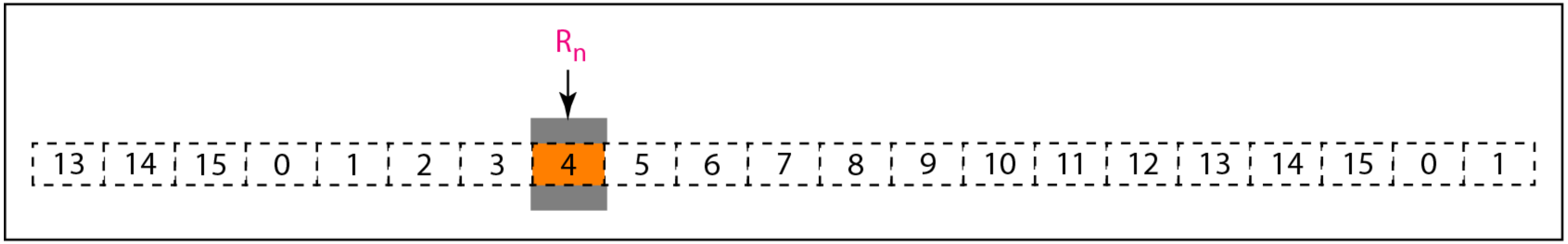
b. Send window after sliding

Receiver Sliding Window

- Look for a specific frame to arrive

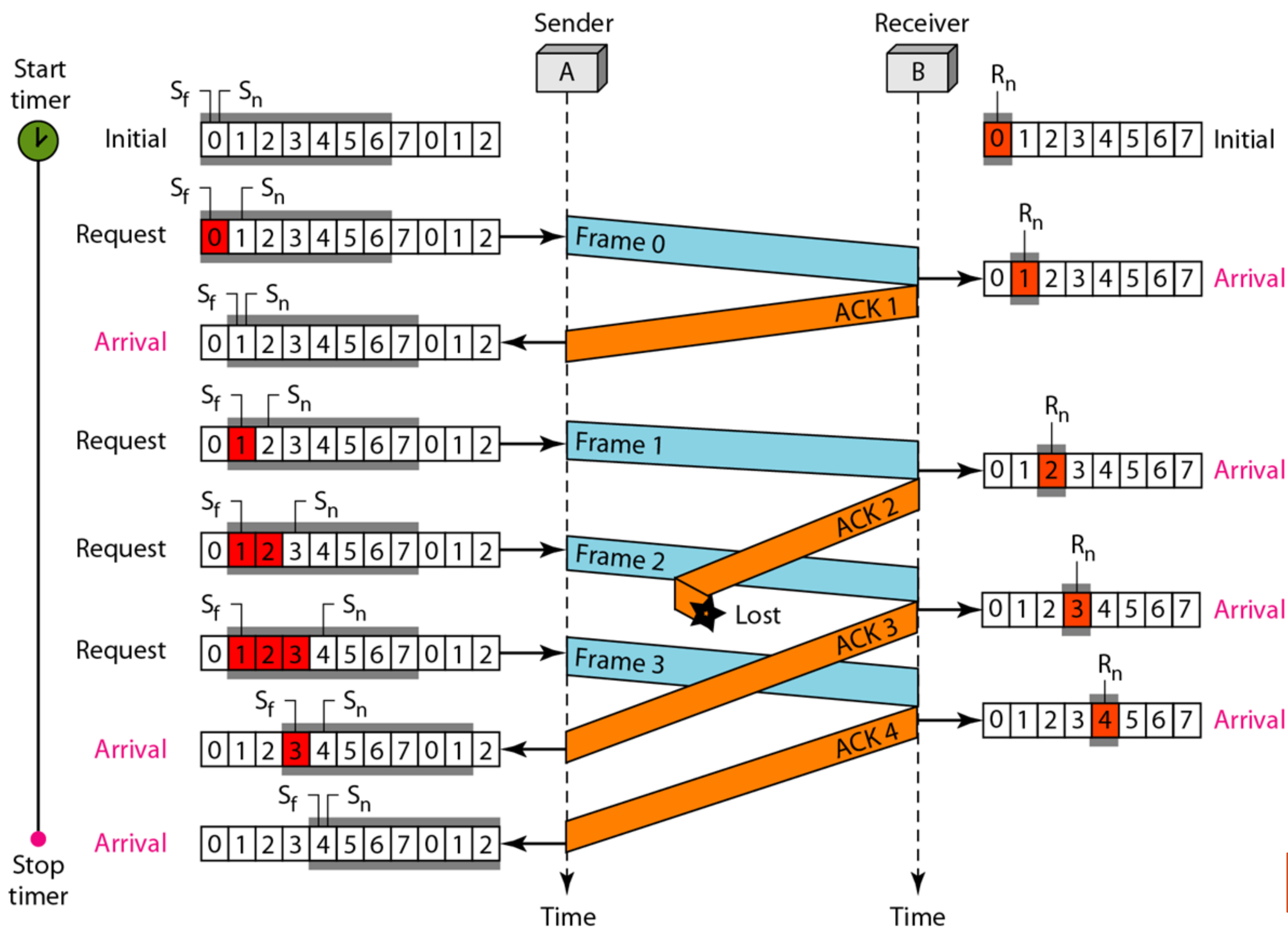


a. Receive window

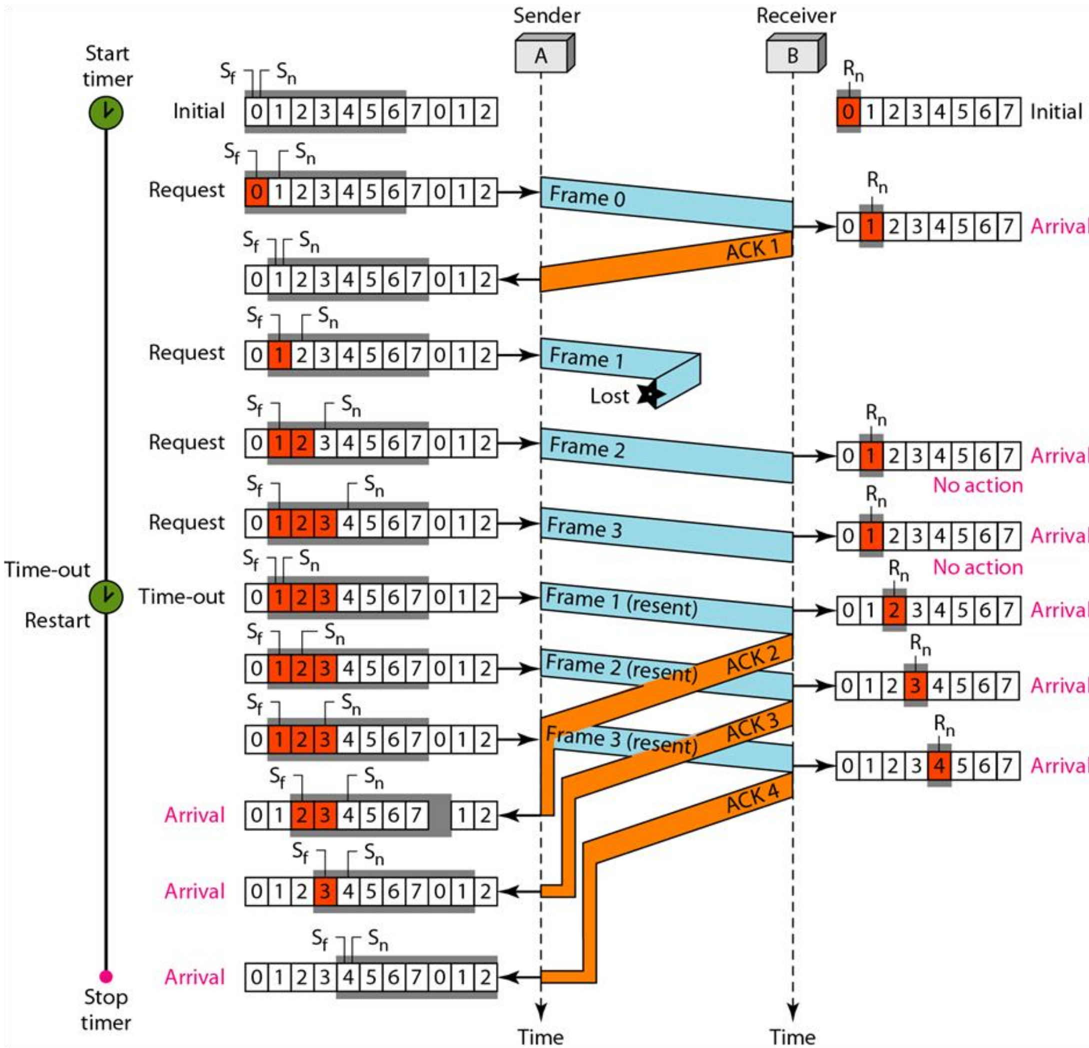


b. Window after sliding

Example 1: Go-Back-N ARQ

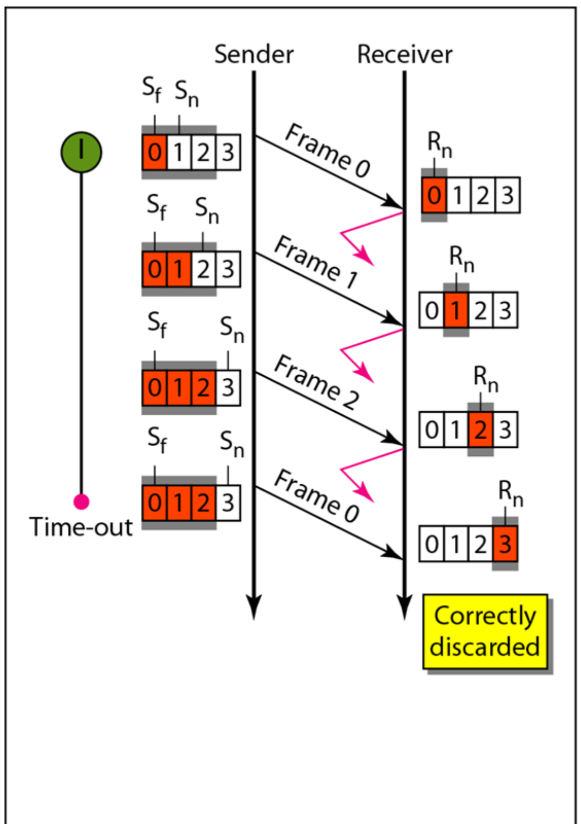


Example 2: Go-Back-N ARQ

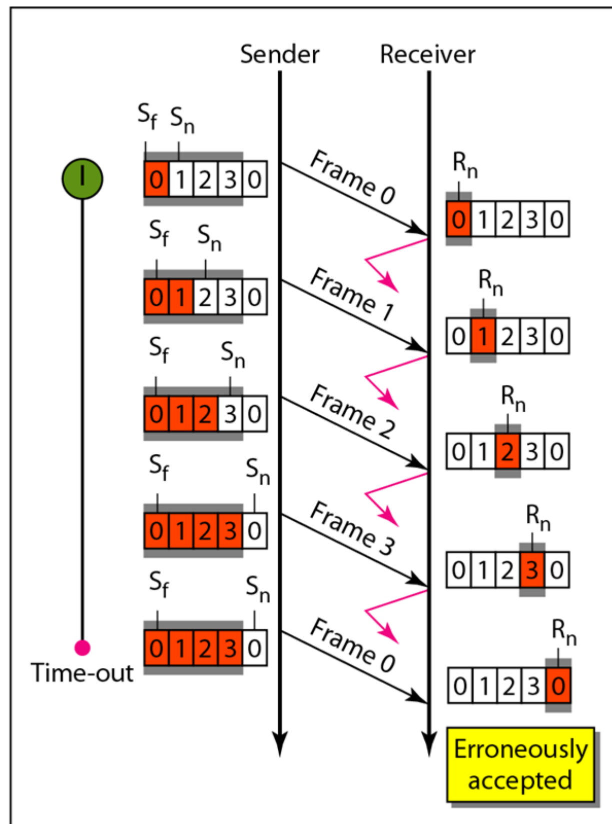


Window Size of Go-Back-N ARQ

- In Go-Back-N ARQ, the size of the send window must be less than 2^m; the size of the receiver window is always 1.



a. Window size $< 2^m$



b. Window size $= 2^m$

Performance of Go-Back-N ARQ

- In this case, each error generates a requirement to retransmit K frames rather than just one frame. Thus

$$N_r = \mathrm{E}[\text{number of transmitted frames to successfully transmit one frame}]$$

$$= \sum_{i=1}^{\infty} f(i) P^{i-1}(1-P)$$

where $f(i)$ is the total number of frames transmitted if the original frame must be transmitted i times:

$$f(i) = (i-1)K + 1$$

Performance of Go-Back-N ARQ

$$N_r = (1-K)\sum_{i=1}^{\infty} P^{i-1}(1-P) + K\sum_{i=1}^{\infty} iP^{i-1}(1-P)$$

$$= 1 - K + \frac{K}{1-P}$$

$$= \frac{1 - P + KP}{1-P}$$

Also, we have

$$K = \begin{cases} 2a+1 & W \geq 2a+1 \\ W & W < 2a+1 \end{cases}$$

Performance of Go-Back-N ARQ

- The utilization for Go-Back-N ARQ is

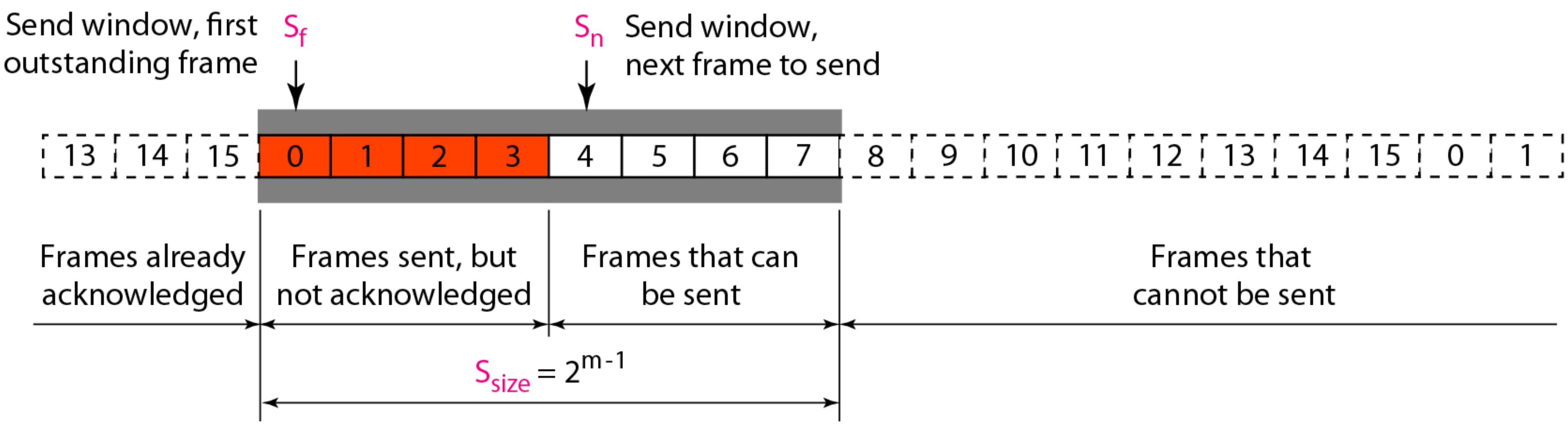
$$U = \begin{cases} \dfrac{1}{N_r} & W \geq 2a+1 \\ \dfrac{W}{N_r(2a+1)} & W < 2a+1 \end{cases}$$

$$= \begin{cases} \dfrac{1-P}{1+2aP} & W \geq 2a+1 \\ \dfrac{W(1-P)}{(2a+1)(1-P+WP)} & W < 2a+1 \end{cases}$$

Selective-Repeat ARQ

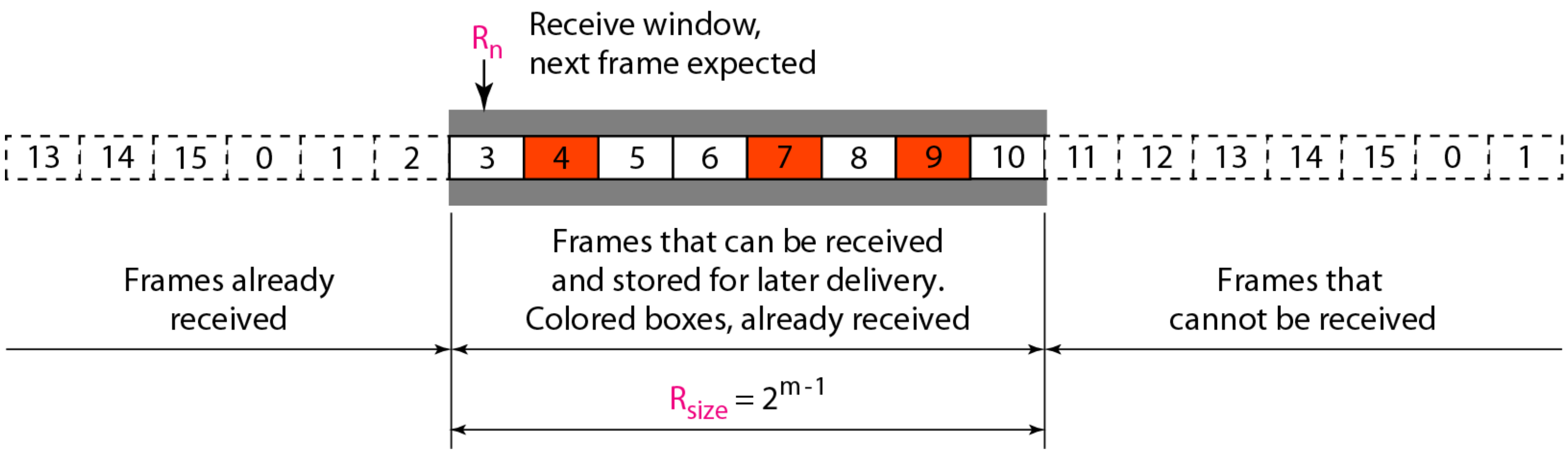
- Also called selective retransmission or selective reject ARQ
- Main ideas:
  - Only resend the corrupted data
  - Allow the receiver to keep track of the received frame
  - Introduce a negative acknowledgment (NAK) that reports the sequence number of a damaged frame
- More efficient: minimizes the amount of retransmission
- Receiver must maintain a large enough buffer
- More complex logic in transmitter to send a frame out of sequence

Sender Sliding Window

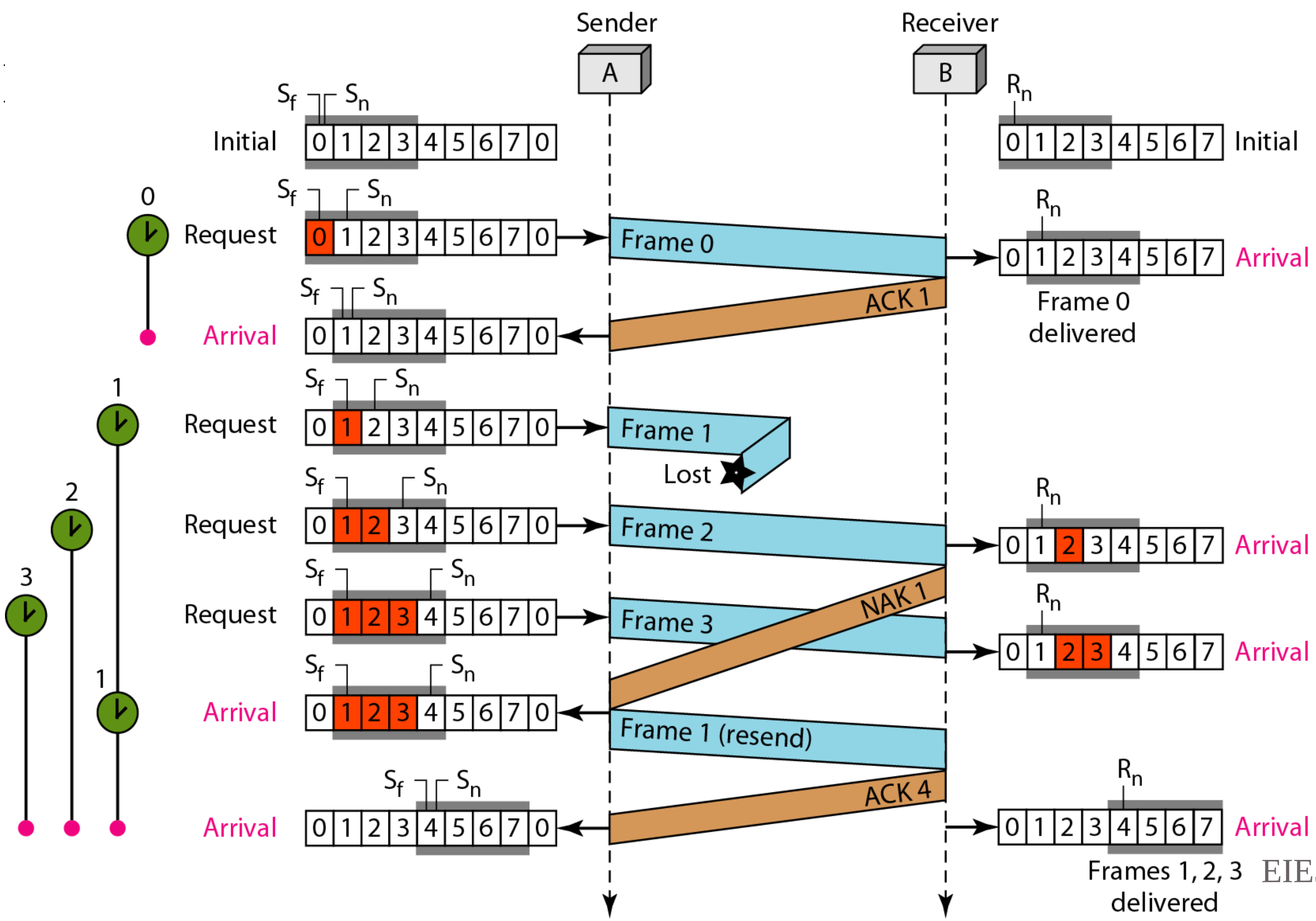


Receiver Sliding Window

- Look for a specific frame to arrive

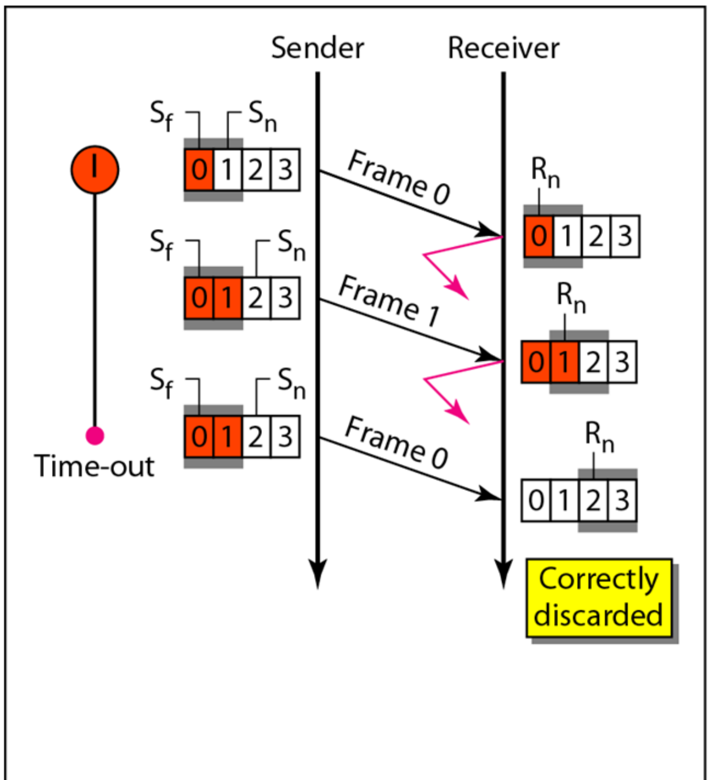


Example 1: Selective-Repeat ARQ

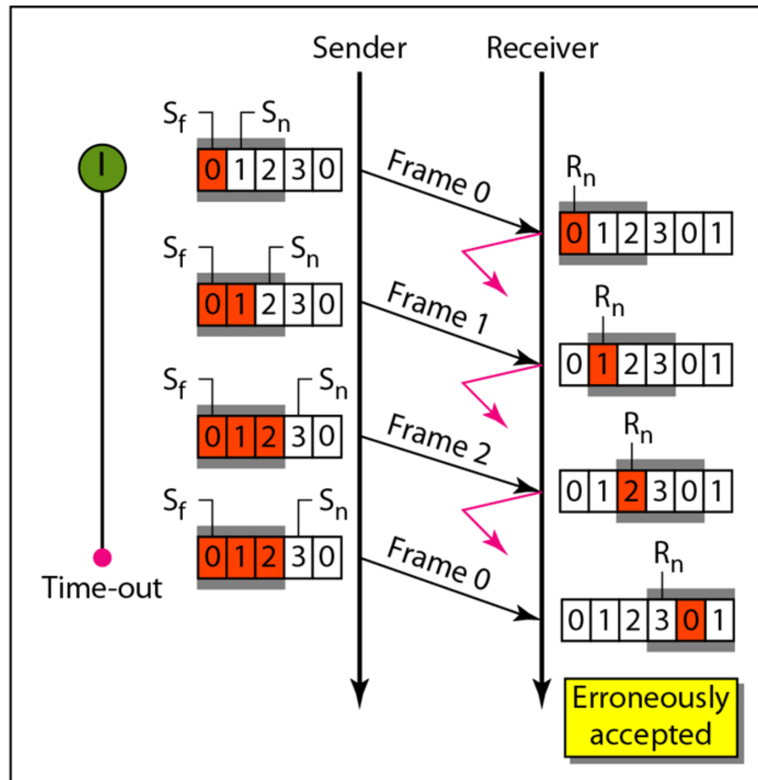


Window Size of Selective Repeat ARQ

- In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.



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



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

Performance of Selective-Reject ARQ

- The utilization for error-free sliding-window protocol is:

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

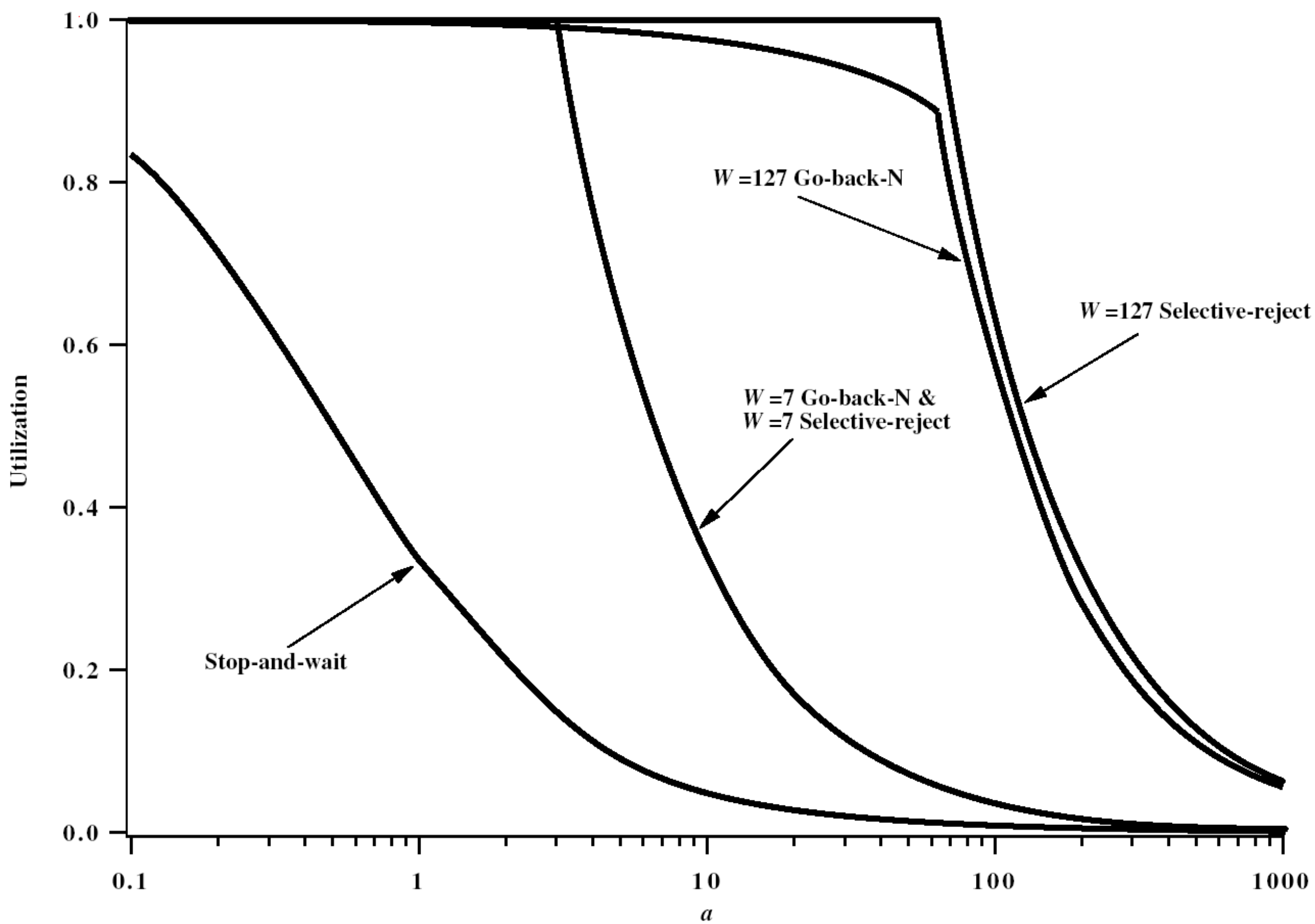
- The utilization for Selective-Repeat ARQ is

$$U = \begin{cases} \dfrac{1}{N_r} & W \geq 2a+1 \\ \dfrac{W}{N_r(2a+1)} & W < 2a+1 \end{cases}$$

$$= \begin{cases} 1-P & W \geq 2a+1 \\ \dfrac{W(1-P)}{(2a+1)} & W < 2a+1 \end{cases}$$

Performance of ARQ

$P = 10^{-3}$



Reading

- B. A. Forouzan, "Data Communications and Networking," 5th Edition, McGraw-Hill 2013 (Chapters 11 and 23)
- William Stallings, "Data and Computer Communications," 10th Edition, Pearson 2015 (Chapter 7)