

*Data Link
Control
and
Protocols*

Data link control

Data link control, a function of the data link layer of the OSI model, comprises three functions:

- **line discipline**—controls which device can transmit over a link and when it can transmit
- **Flow control** refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment
- **Error control** in the data link layer is based on automatic repeat request, which is the retransmission of data.
 - **error control**—allows the receiver to inform the sender of any lost data and to request retransmission
- These functions enable the two devices that are communicating to coordinate with each other.

Utopia protocol

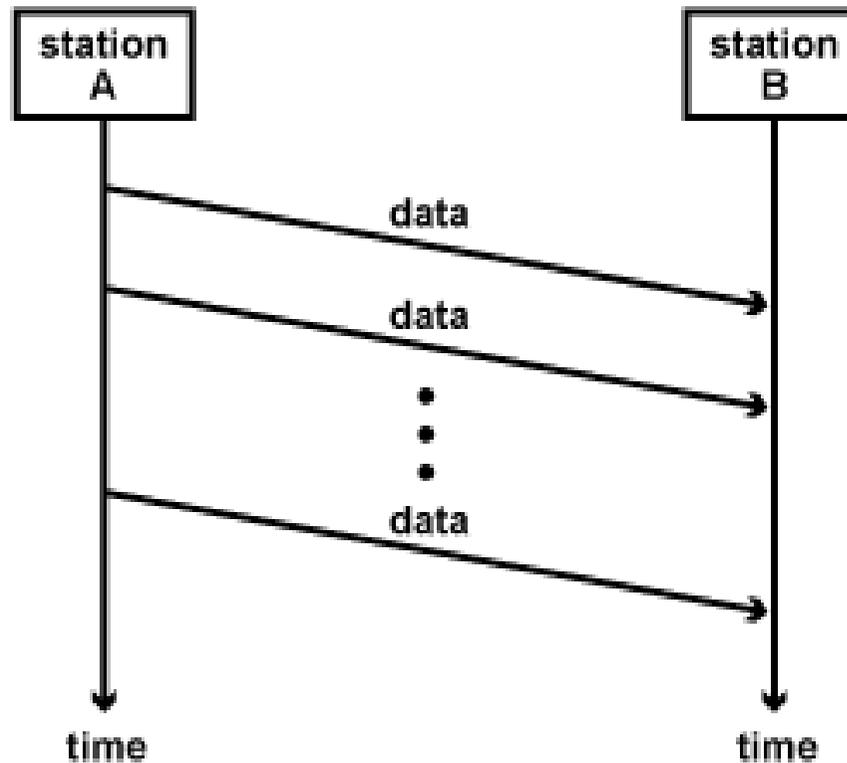
- Utopia protocol is an unrestricted simplex protocol.
- This protocol is thoroughly unrealistic but gives us a starting point. It is the simplest possible protocol.

In it:

- data are transmitted in one direction only, and the receiver is always ready, thus line discipline is not required
- processing time can be ignored, and there is infinite buffer space, thus the receiving station cannot be overwhelmed
- no frames are lost or damaged in the communication link, and thus error control is not required

Example

- The station A's data link layer receives packets from its network layer and just keeps sending data to station B.
- Station B's data link layer continually receives the data and passes it up to its network layer.



Line Discipline

- The "utopia" has three unrealistic simplifications. Line discipline covers one of those simplifications:
 - data are transmitted in one direction only, and the receiver is always ready
 - Before a sender transmits signals to a receiver, it needs to know that the receiver is ready to receive the signals.
 - Line discipline determines who should send at a particular time.

Line discipline

- **Point-to-Point** the line discipline is based on ENQ/ACK
- **Multipoint** A requirement for a multipoint configuration is that each device have a unique address that can be used to identify the destination of a message.
- **Polling** -The polling mode is used for messages or data that are to be sent from a secondary station to the primary station.
 - determine which of several secondary stations is communicating with the primary station
 - determine if that secondary station is available
 - the primary station polls each secondary station in turn to see if it has data to send.
 - If there are no data to be sent, a NAK is returned.
 - If a station has data to send, then it sends that data.

Flow Control

Flow control in the data link layer is achieved through set of procedures that control how much data the sender can send before overwhelming the receiver.

- This control is achieved by the sender's receiving a set of acknowledgments before additional data can be sent.
- The two methods used to control the flow of data are:
 - stop-and-wait
 - sliding window

Stop and Wait

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

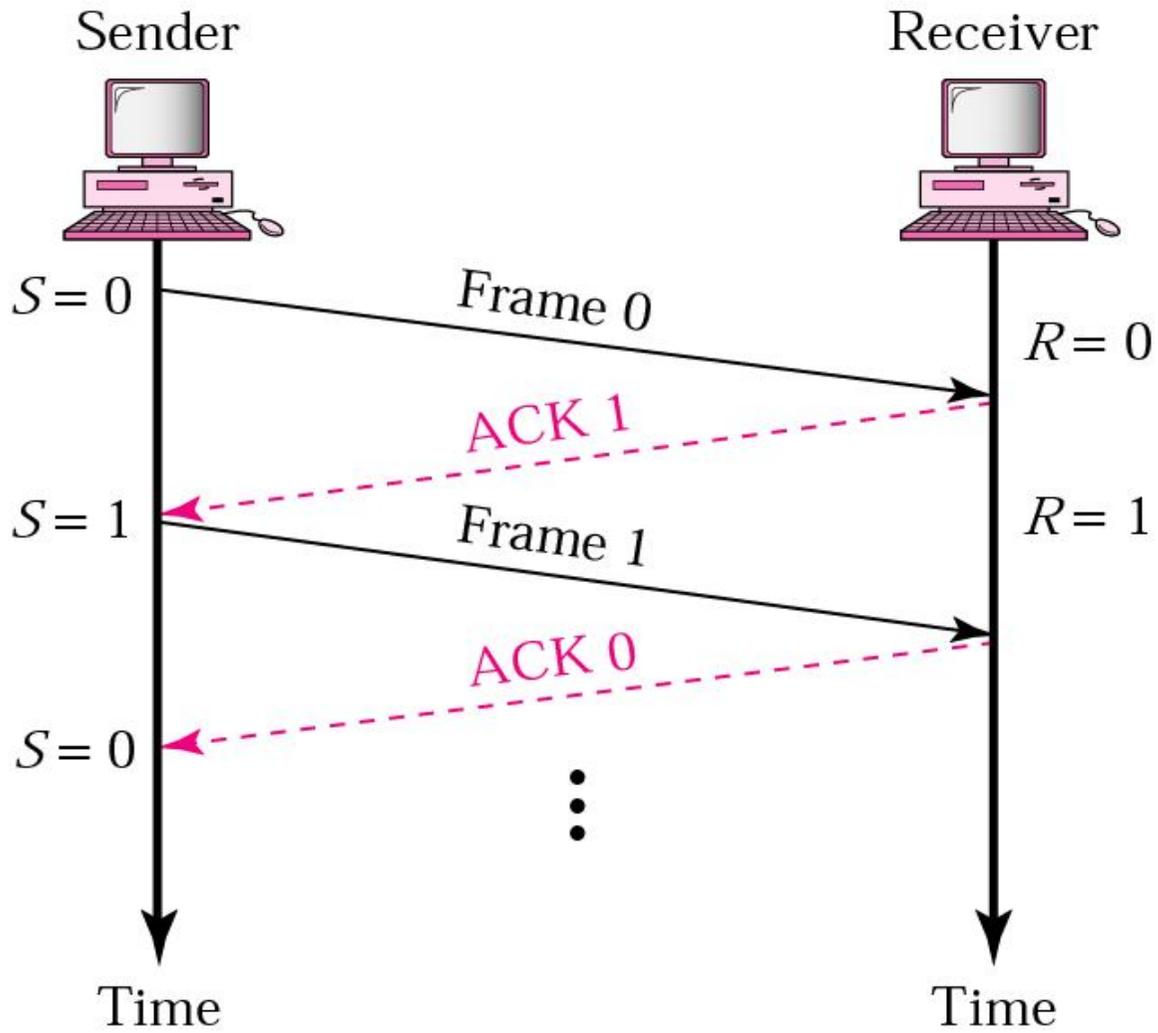
Stop-and-Wait ARQ

- **In stop-and-wait, the sending station waits for an ACK before sending the next frame**
- This process is called Automatic Repeat Request (ARQ)
- The sending device keeps a copy of the last frame transmitted until it receives an acknowledgement
- Both the data frames and acknowledgement (ACK) frames are numbered alternately 0 and 1
- A damaged or lost frame is treated in the same manner by receiver.
 - If the receiver detects an error it simply discards the frame and sends no acknowledgement.
 - If the receiver receives a frame but is out of order (0 instead of 1 or 1 instead of 0) it knows that the frame is lost. It discards the out-of-order received frame
- **The sender has a control variable, call S; The receiver has a control variable, call R**
- The sender starts a timer when it sends a frame. If the timeout is happened the frame is resends it

Stop-and-Wait ARQ

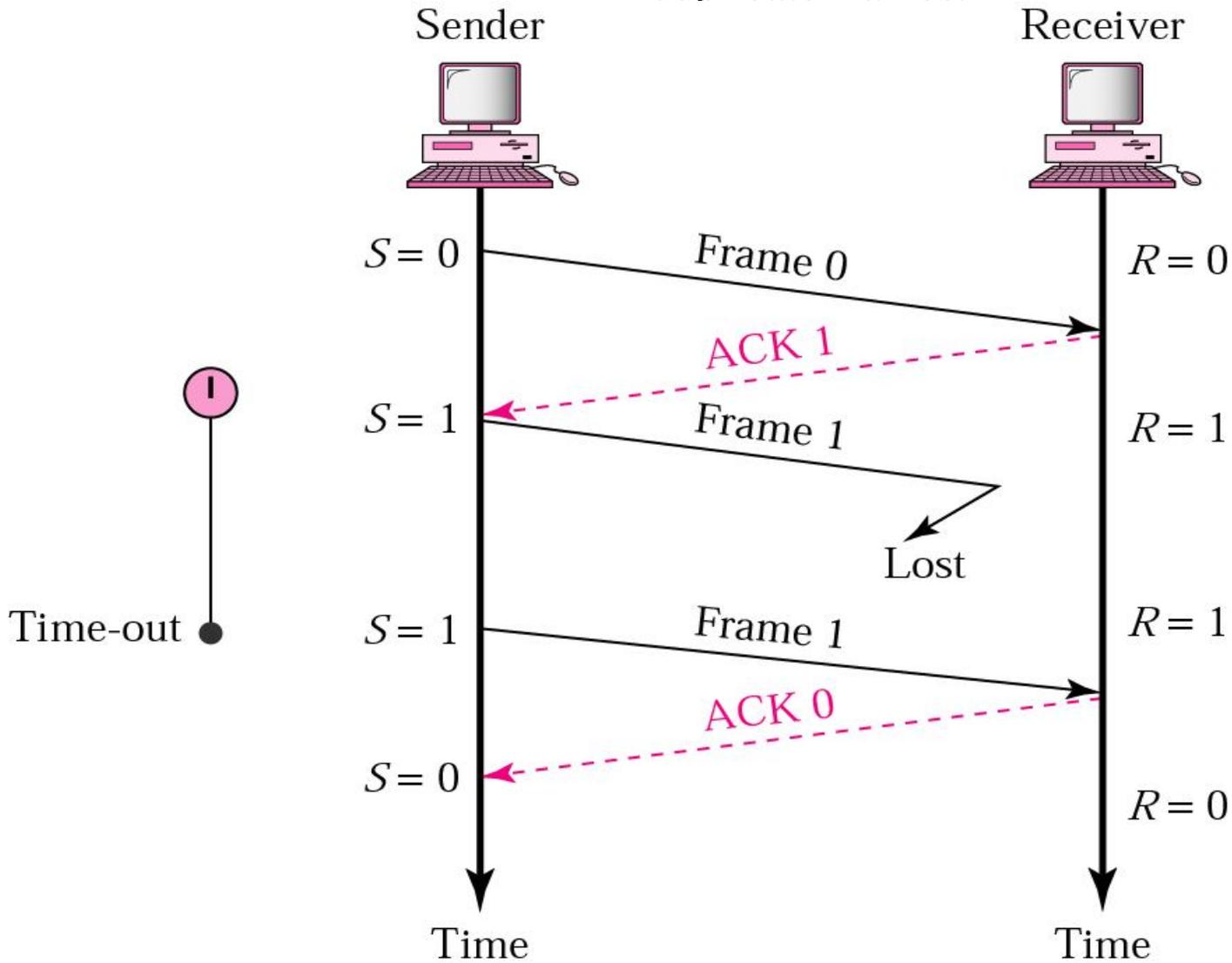
- Source transmits single frame
- Wait for ACK
- If received frame damaged, discard it
 - Transmitter has timeout
 - If no ACK within timeout, retransmit
- If ACK damaged, transmitter will not recognize it
 - Transmitter will retransmit
 - Receiver gets two copies of frame
 - Use ACK0 and ACK1

Normal operation

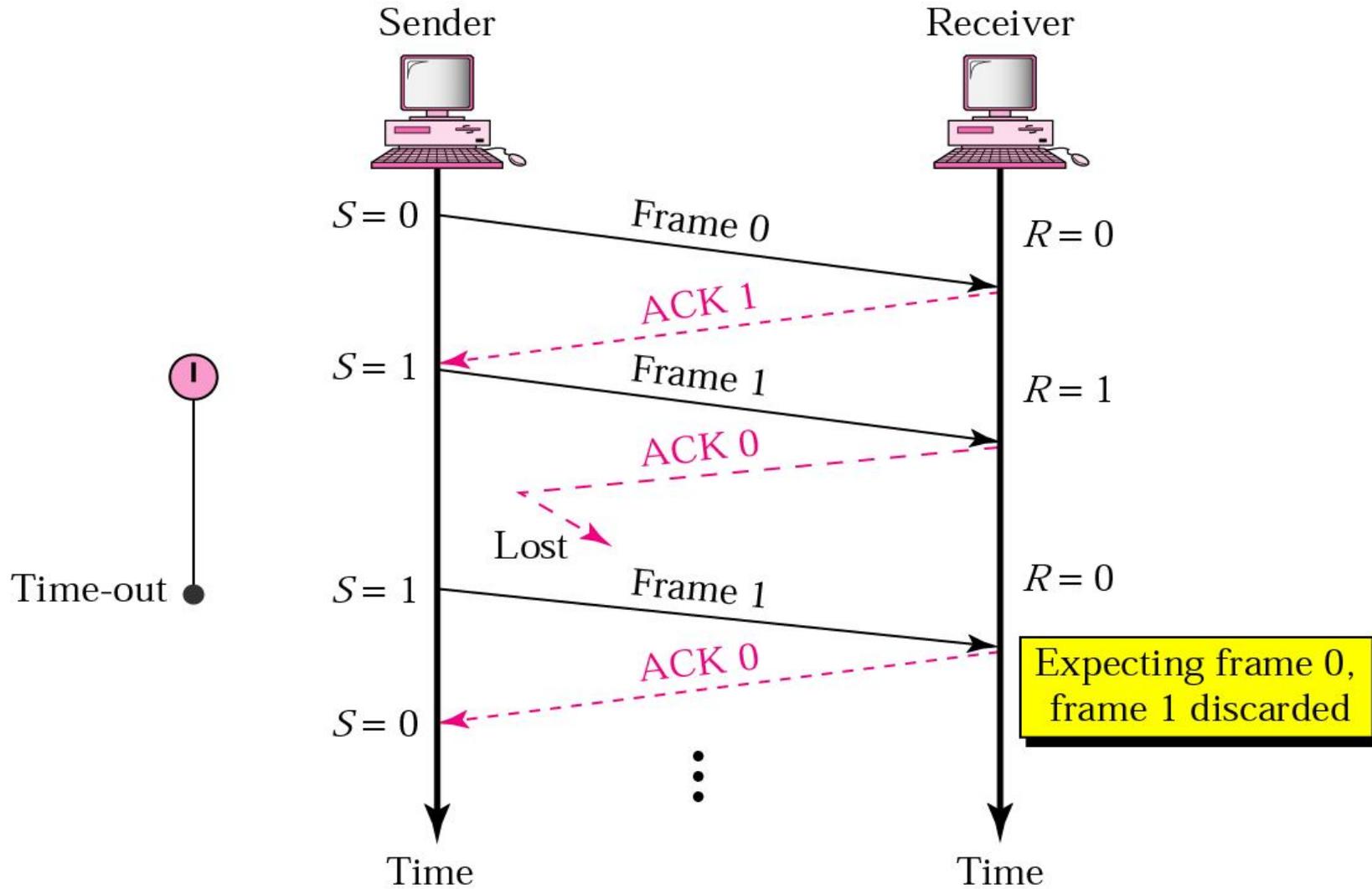


Stop-and-Wait ARQ, lost frame

In Stop-and-Wait ARQ, numbering frames prevents the retaining of duplicate frames.

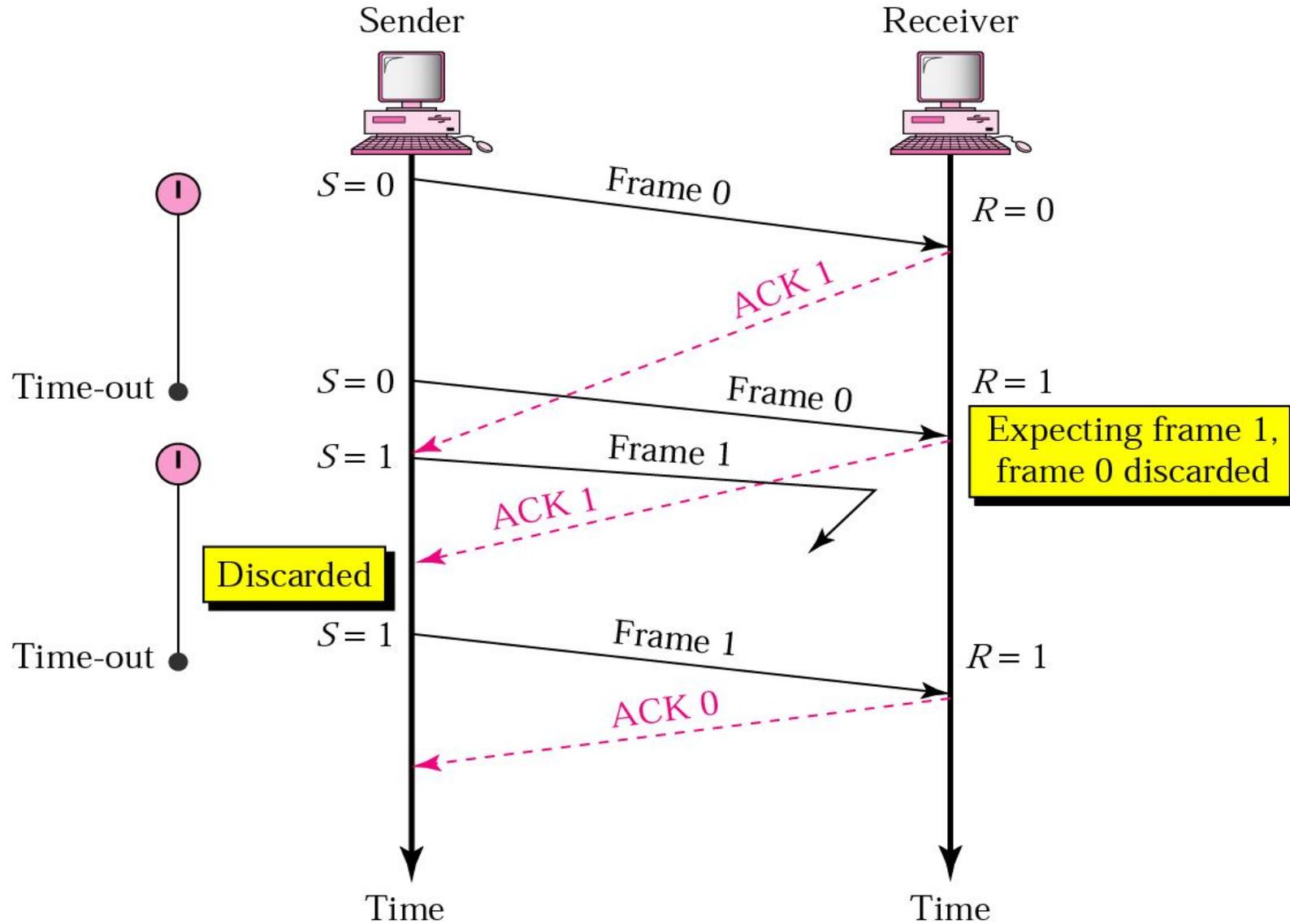


Stop-and-Wait ARQ, lost ACK frame



Stop-and-Wait ARQ, delayed ACK

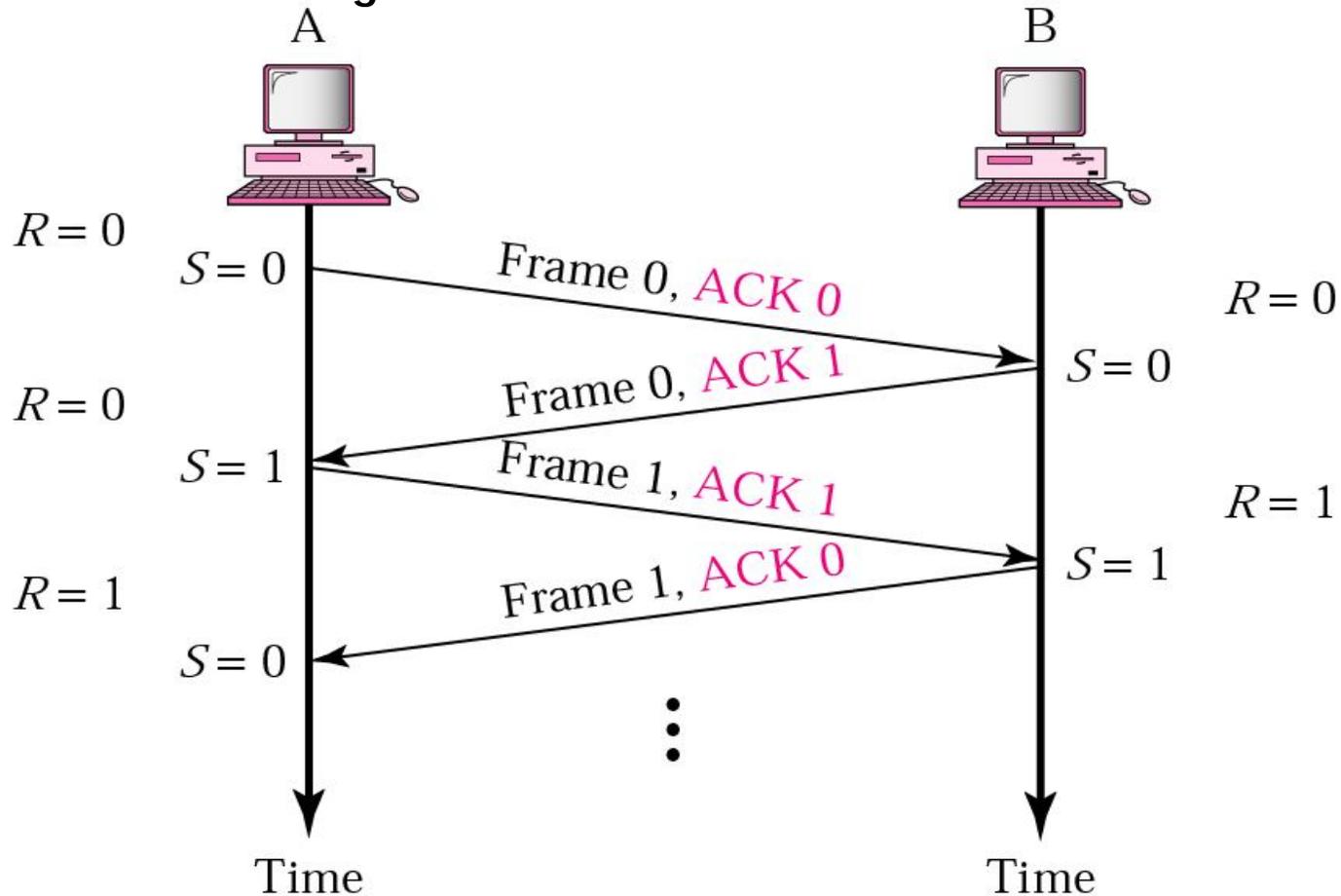
Numbered acknowledgments are needed if an acknowledgment is delayed and the next frame is lost.



Piggybacking

Bidirectional Transmission

- Piggybacking is a method to combine a data frame with an acknowledgement.



Sliding Window

- The sliding window method allows multiple frames to be sent without an acknowledgment, thus eliminating most of the waiting time shown in the stop-and-wait protocol.
- The size of the sliding window is determined by the maximum number of frames, n , that can be sent without an acknowledgment.
- Given n , then:
 - the frames in the sending buffer are numbered modulo n , e.g., 0 to $n - 1$
 - the sliding window is a box with a maximum size of $n - 1$

Example

- If the sending buffer contained 32 frames of data and the maximum number of frames that could be sent without an acknowledgment were 8, the frames would be numbered.
- A sliding window of up to 7 frames at an arbitrary position in the buffer.

0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 **1 2 3 4 5 6 7** 0 1 2 3 4 5 6 7
sliding
window

Sliding Windows

- Both the sender and the receiver have sliding windows.
 - The **sender's** sliding window indicates how many frames can still be sent without receipt of an acknowledgment. The window would start at its maximum value and shrink on the left side every time a frame is sent. It would grow on the right side every time it received an ACK.
 - The receiver's sliding window indicates how many frames can still be received without sending an acknowledgment. The window would start at its maximum value and shrink on the left side every time it received a frame. It would grow on the right side every time an ACK was sent.
- The receiver's ACK usually acknowledges more than one frame. The number of the acknowledgment sent indicates the next expected frame and acceptance of all prior frames.

This example shows how a sliding window works.

■ **Start:**

- Both the sender and the receiver start with a sliding window of frames 0 to 6.



■ **Step 1:**

- The sender sends data frame 0, which is successfully received.
- Thus, both the sender's and receiver's sliding windows shrink from a size of 7 frames (covering frames 0 to 6) to a size of 6 frames (covering frames 1 to 6).

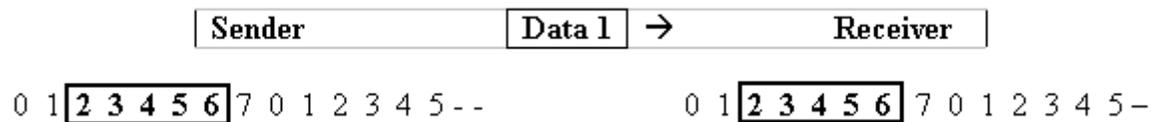


Example cont'

- **Step 2:**

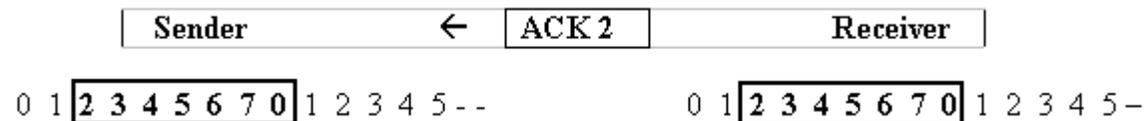
- The sender sends data frame 1, which is successfully received.
- Thus, both the sender's and receiver's sliding windows shrink from a size of 6 frames (covering frames 1 to 6) to a size of 5 frames (covering frames 2 to 6).

-



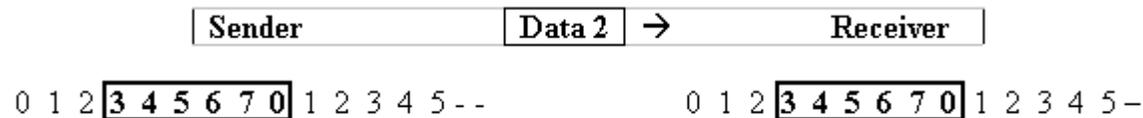
- **Step 3:**

- The receiver acknowledges receipt of frames 0 and 1 and returns an acknowledgment that it is ready to receive data frame 2.
- Thus, both the sender's and receiver's sliding windows expand to 7 frames (frames 2 to 0).

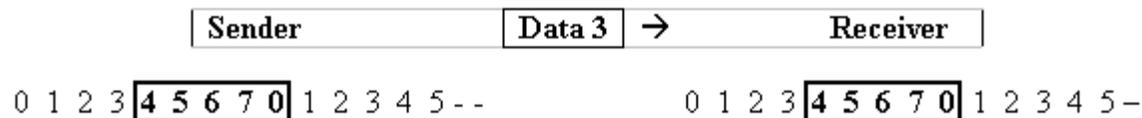


Example cont'

- **Step 4:**
- The sender sends data frame 2, which is successfully received.
- Thus, both the sender's and receiver's sliding windows shrink from a size of 7 frames (covering frames 2 to 0) to a size of 6 frames (covering frames 3 to 0).

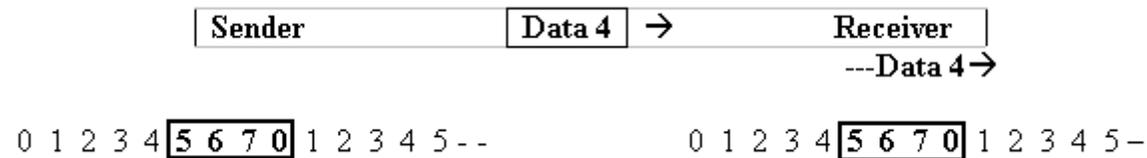


- **Step 5:**
- The sender sends data frame 3, which is successfully received.
- Thus, both the sender's and receiver's sliding windows shrink from a size of 6 frames (covering frames 3 to 0) to a size of 5 frames (covering frames 4 to 0).

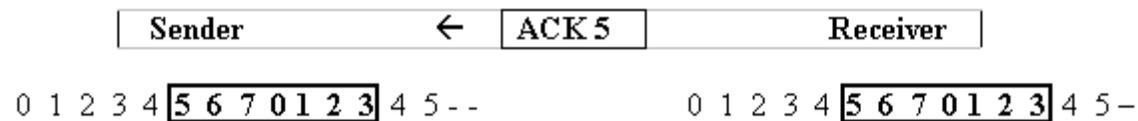


Example cont'

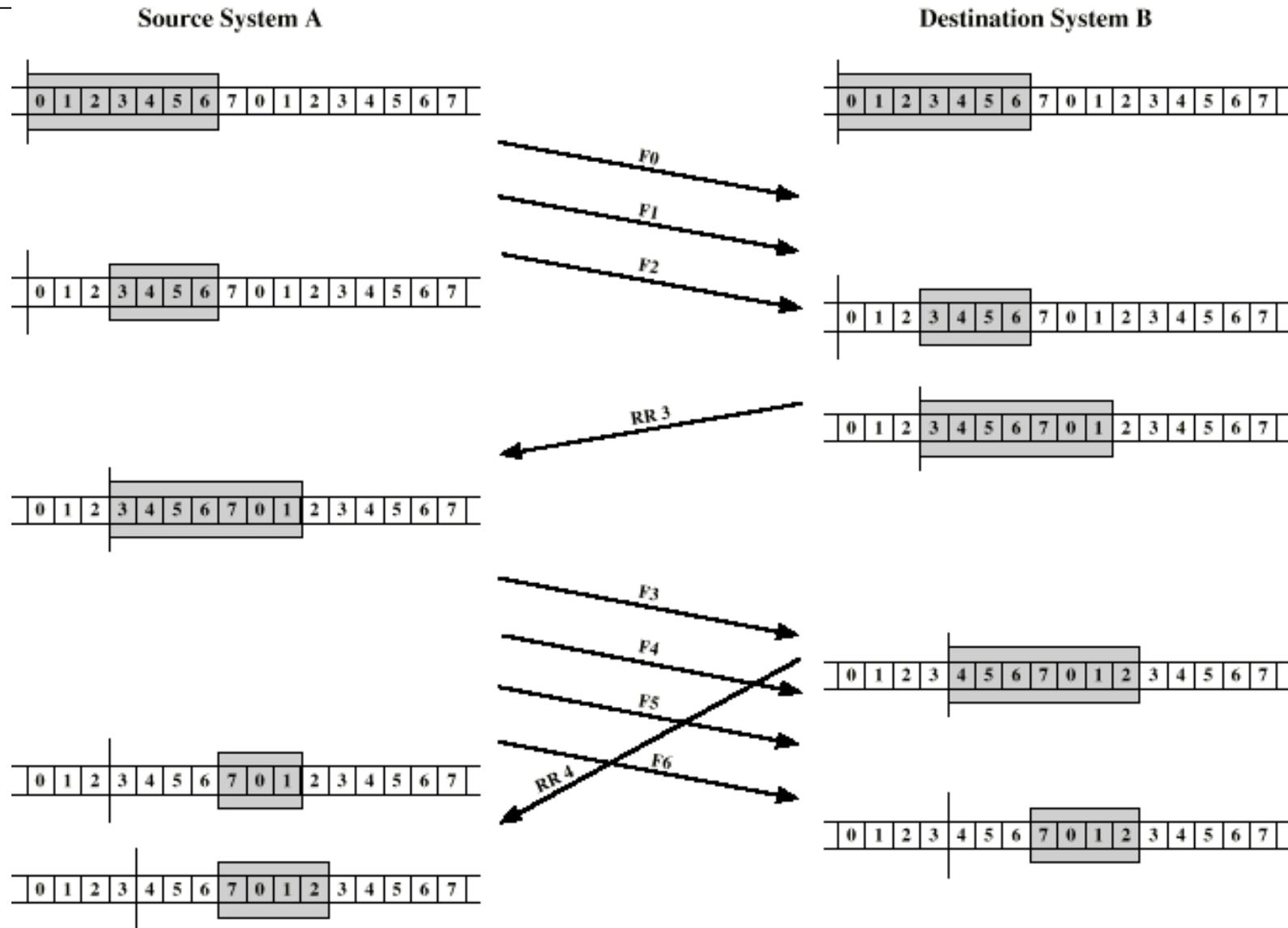
- **Step 6:**
- The sender sends data frame 4, which is successfully received.
- Thus, both the sender's and receiver's sliding windows shrink from a size of 5 frames (covering frames 4 to 0) to a size of 4 frames (covering frames 5 to 0).



- **Step 7:**
- The receiver acknowledges receipt of frames 2, 3, and 4 and returns an acknowledgment that it is ready to receive data frame 5.
- Thus, both the sender's and receiver's sliding windows expand to 7 frames (frames 5 to 3).



Example Sliding Window



Go-Back-N ARQ

Sequence Number

Sender and Receiver Sliding Window

Control Variables and Timers

Acknowledgment

Resending Frames

Operation

Go Back N (1)

- Based on sliding window
- If no error, ACK as usual with next frame expected
- Use window to control number of outstanding frames
- If error, reply with rejection
 - Discard that frame and all future frames until error frame received correctly
 - Transmitter must go back and retransmit that frame and all subsequent frames

Go Back N - Damaged Frame

- Receiver detects error in frame i
- Receiver sends rejection- i
- Transmitter gets rejection- i
- Transmitter retransmits frame i and all subsequent

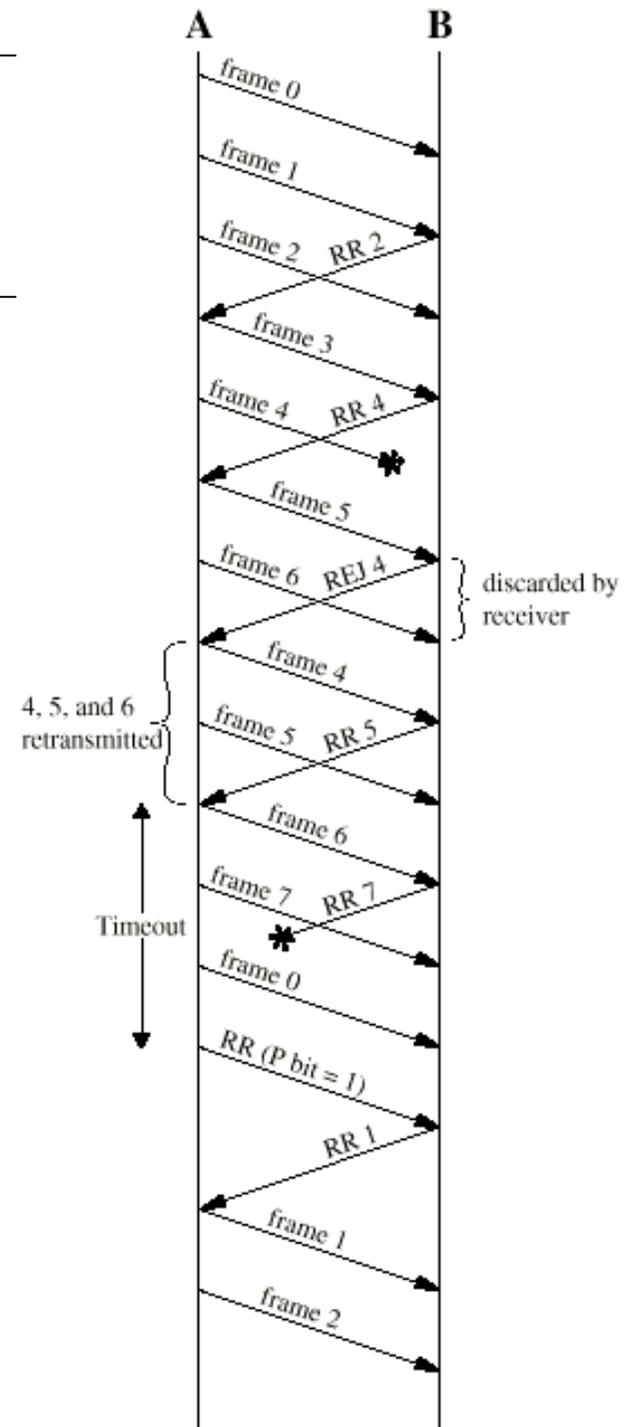
Go Back N - Lost Frame (1)

- Frame i lost
- Transmitter sends $i+1$
- Receiver gets frame $i+1$ out of sequence
- Receiver send reject i
- Transmitter goes back to frame i and retransmits

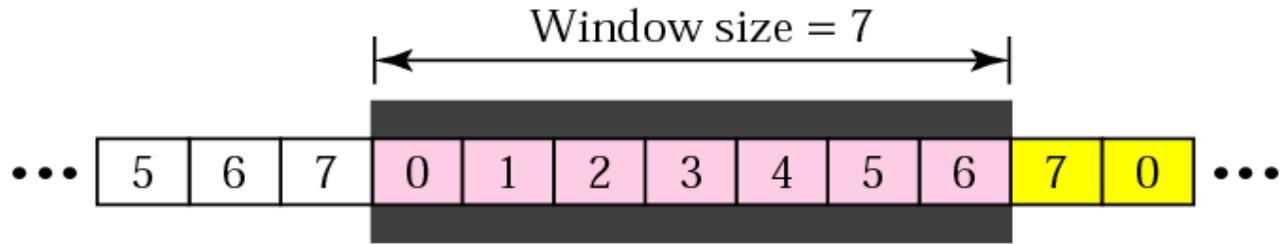
Go Back N - Lost Frame (2)

- Frame i lost and no additional frame sent
- Receiver gets nothing and returns neither acknowledgement nor rejection
- Transmitter times out and sends acknowledgement frame with P bit set to 1
- Receiver interprets this as command which it acknowledges with the number of the next frame it expects (frame i)
- Transmitter then retransmits frame i

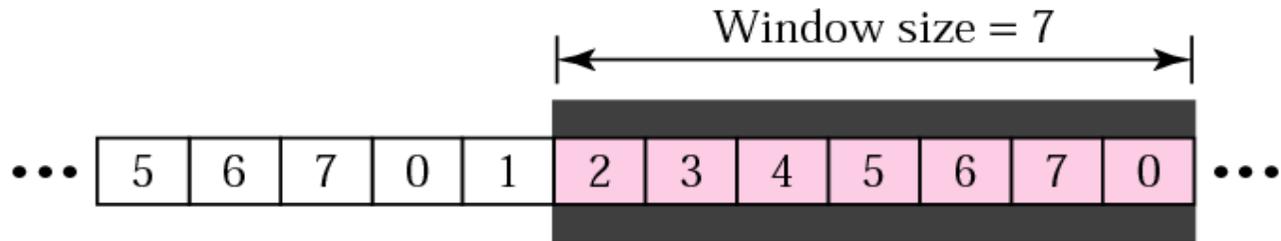
Go Back N - Diagram



Sender sliding window



a. Before sliding

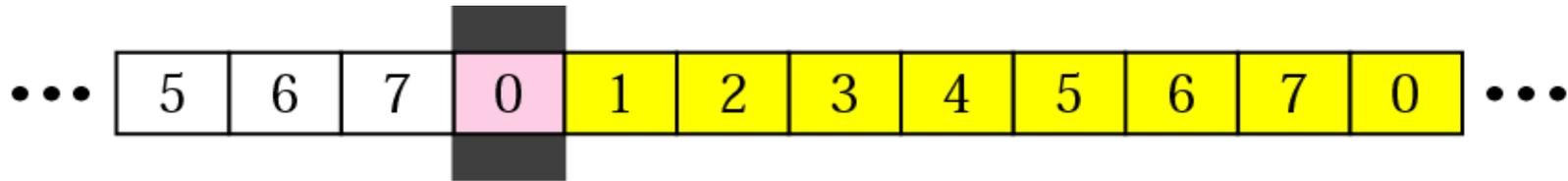


b. After sliding two frames

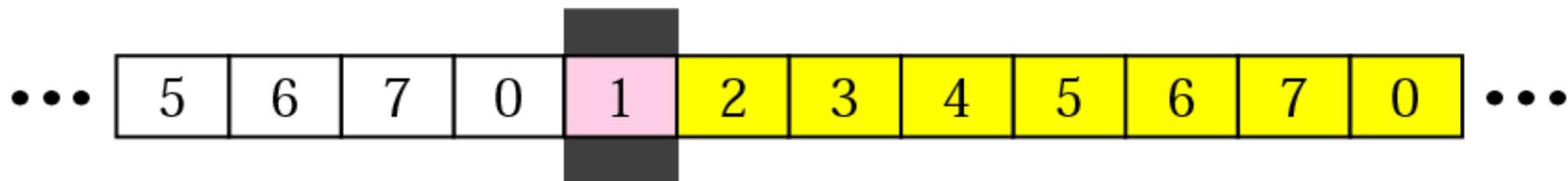
Frames from a sending station are numbered sequentially modulo n

The size of window is $2^m - 1$

Receiver sliding window



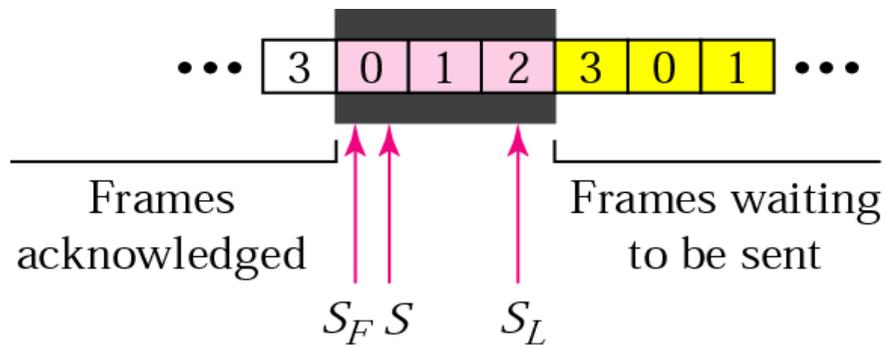
a. Before sliding



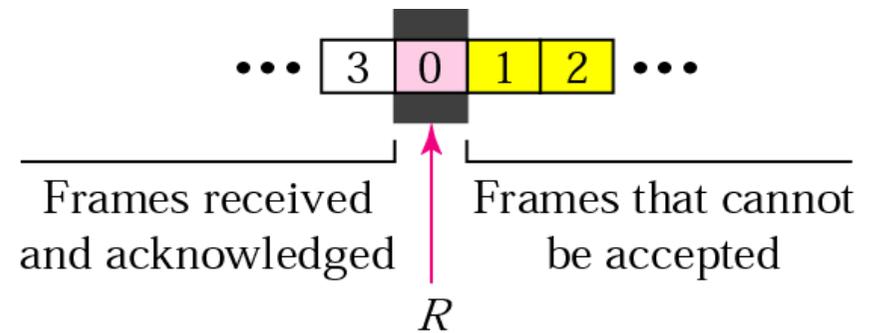
b. After sliding

- *The size of the windows at the receiver site in this protocol is always 1.*
- *The receiver is always looking for a specific frame to arrive in a specific order.*
- *Any frame arriving out of order is discarded and needs to be resent*

Control variables



a. Sender window



b. Receiver window

S – Holds the sequence number of the recently sent frame

SF – Holds the sequence number of the first in window

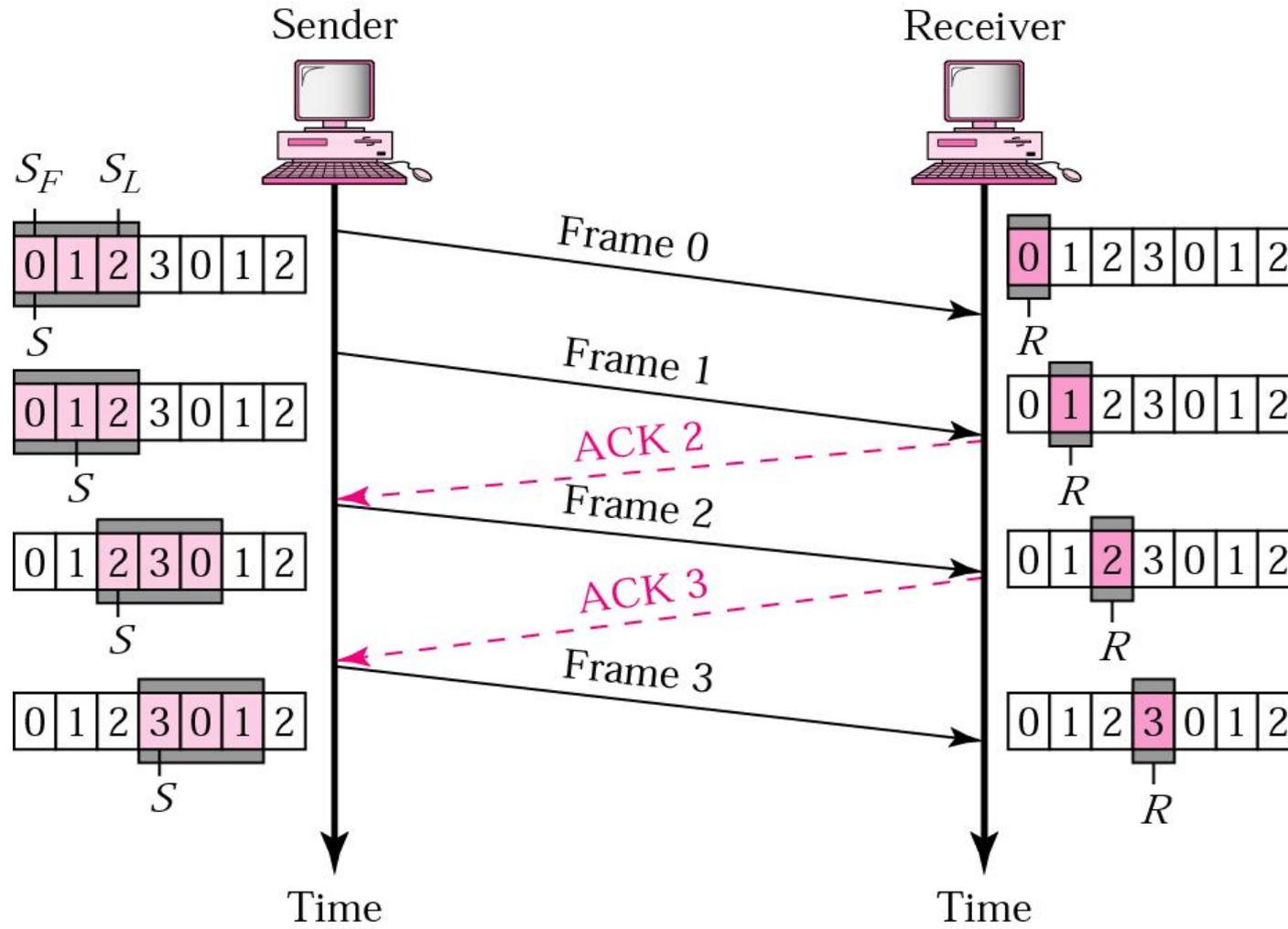
SL – Holds the sequence number of the last in window

The size of window is **$W = SL - SF + 1$**

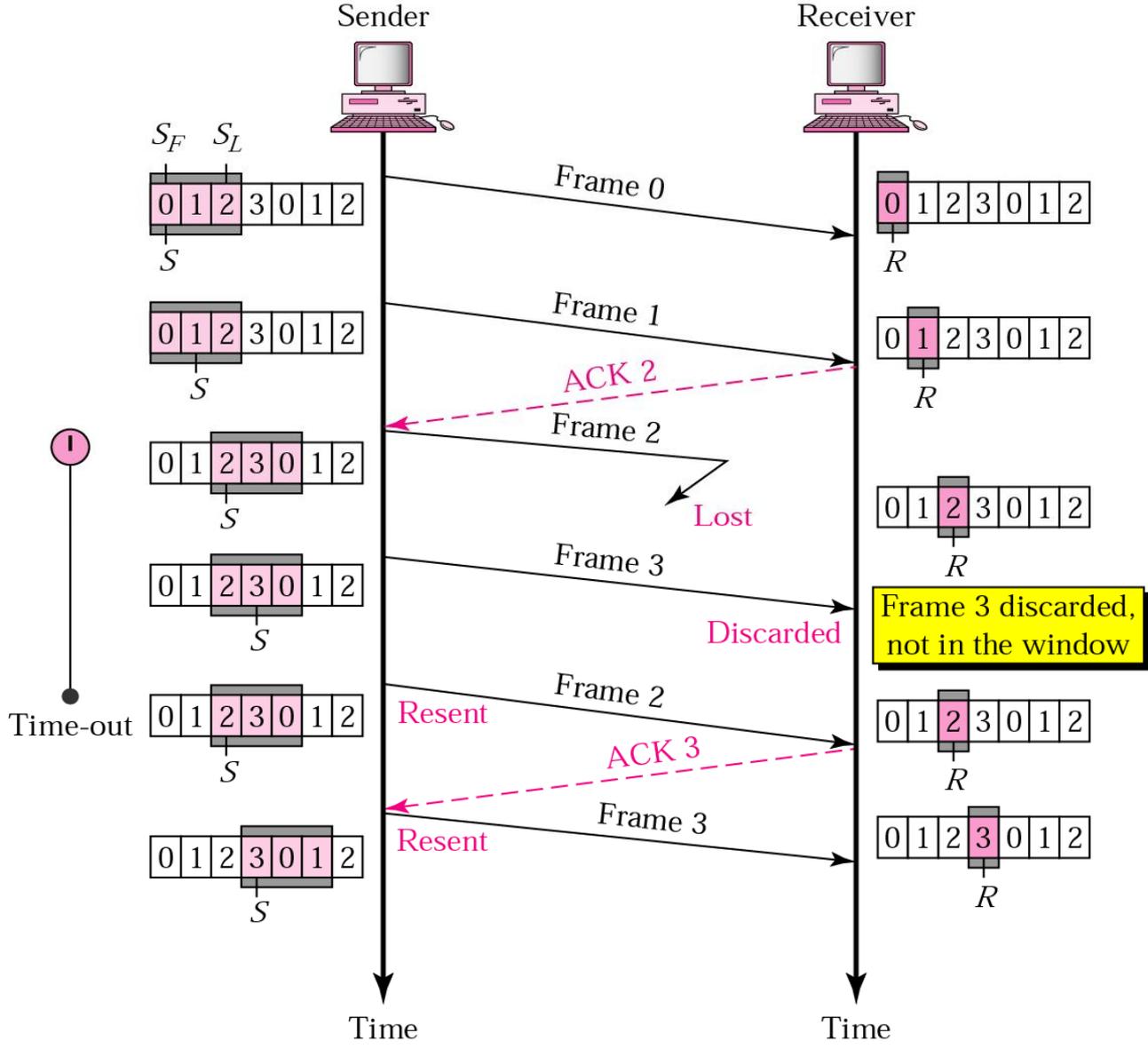
R – Holds the sequence number of the frame is expected to receive.

If the sequence nr of the received frame is the same as the value of R the frame is accepted; if not it is rejected

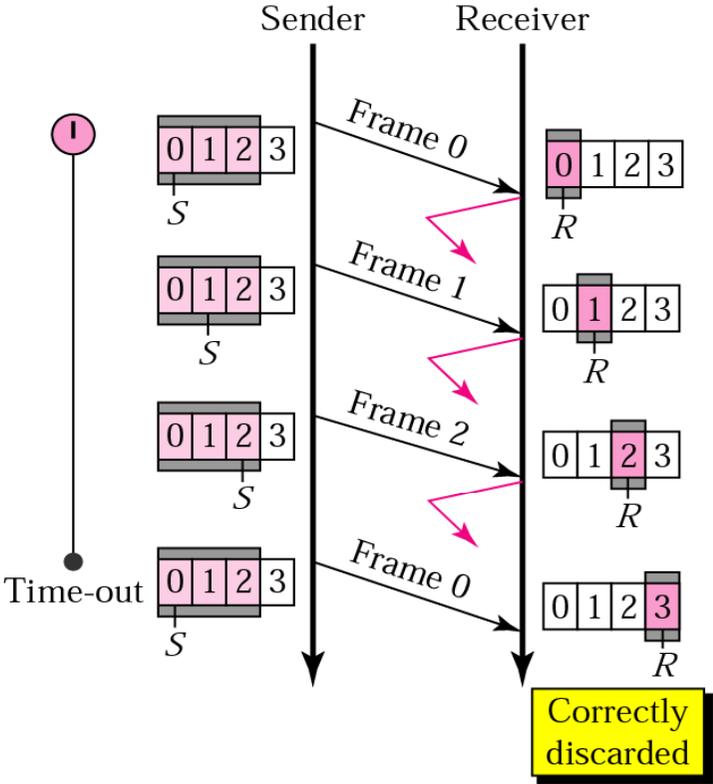
Go-Back-N ARQ, normal operation



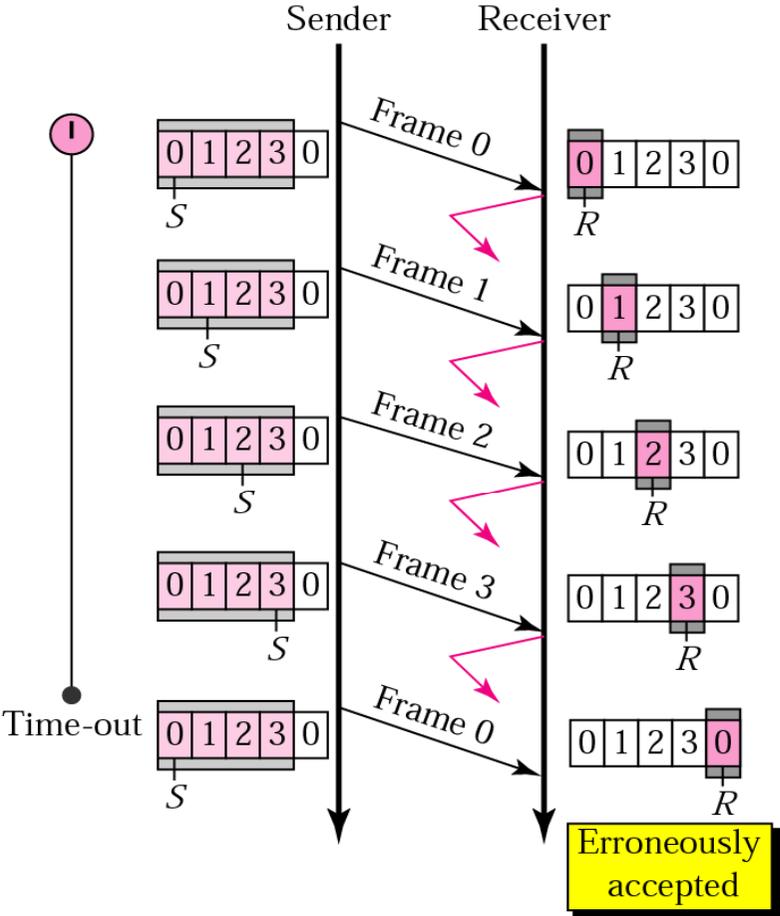
Go-Back-N ARQ, lost frame



Go-Back-N ARQ: sender window size



a. Window size $< 2^m$



b. Window size $= 2^m$

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

Go-Back-N ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and is no need to buffer out-of-order frames.

In a noisy link a frame has a higher probability of damage, wich means the resending of multiple frames.

Selective-Repeat ARQ

Sender and Receiver Windows

Operation

Sender Window Size

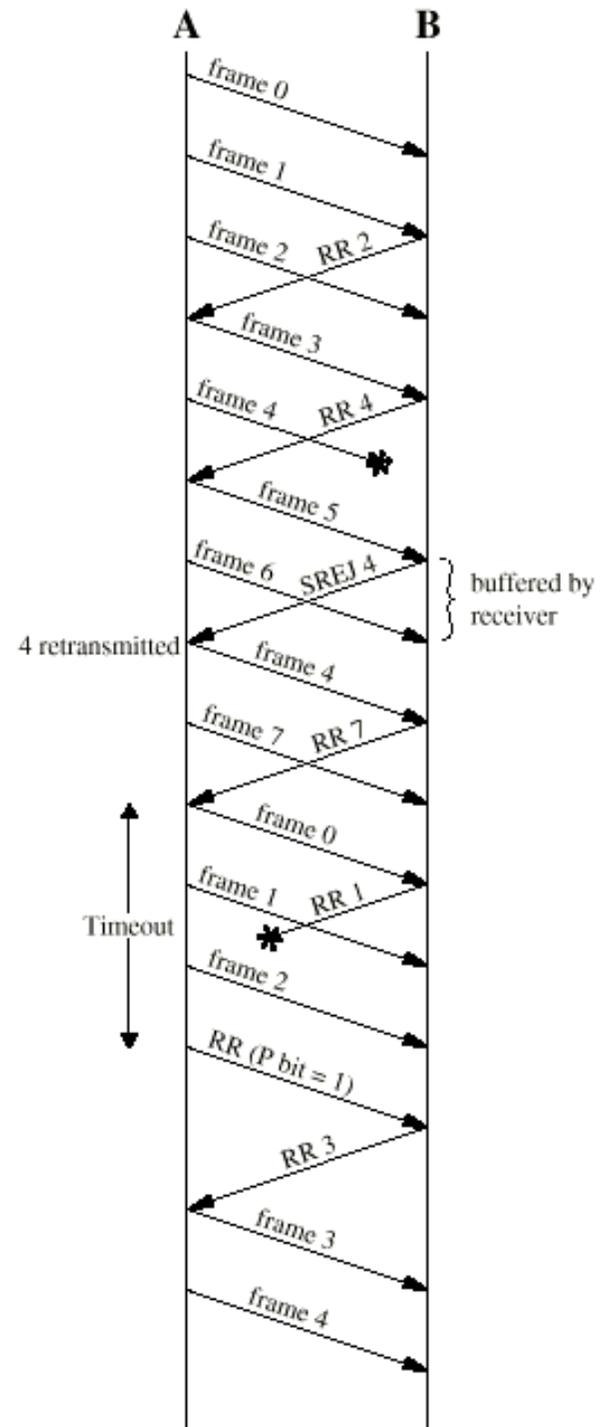
Bidirectional Transmission

Pipelining

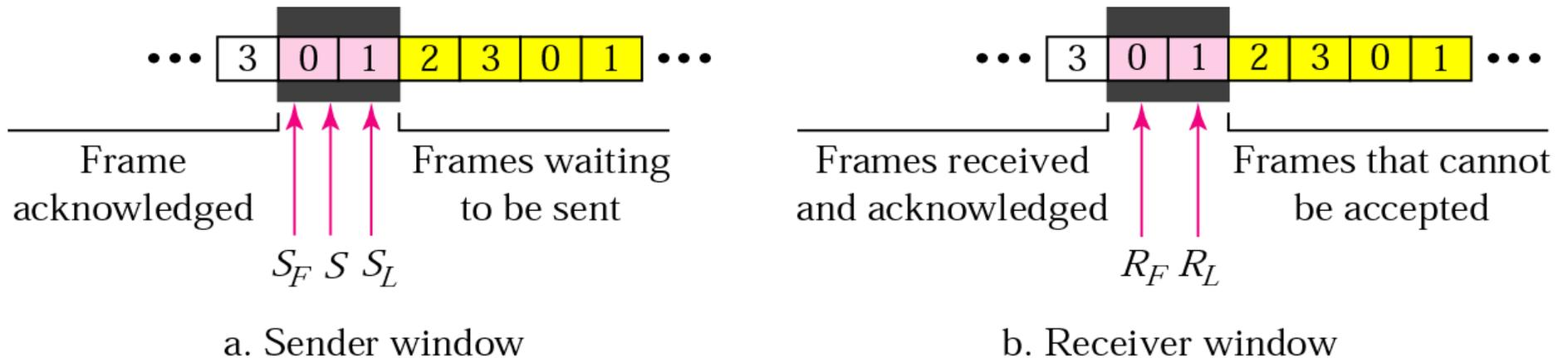
Selective Reject

- Also called selective retransmission
- Only rejected frames are retransmitted
- Subsequent frames are accepted by the receiver and buffered
- Minimizes retransmission
- Receiver must maintain large enough buffer
- More complex logic in transmitter

Selective Reject - Diagram



Selective Repeat ARQ, sender and receiver windows

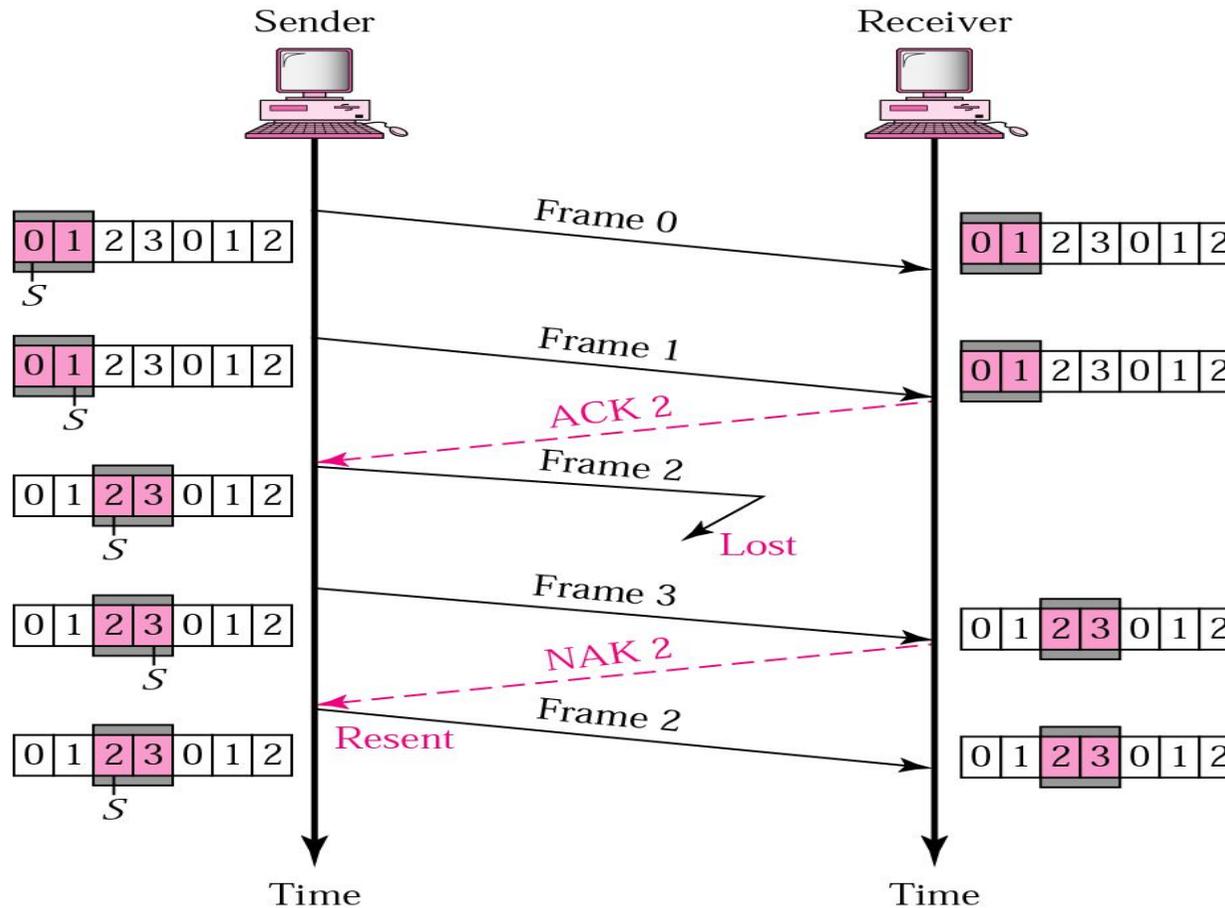


The size of the windows must be at most one-half of 2^m

The selective Repeat ARQ also define a negative acknowledge (NAK)

That reports the sequence number of a damaged frame before the timer expires.

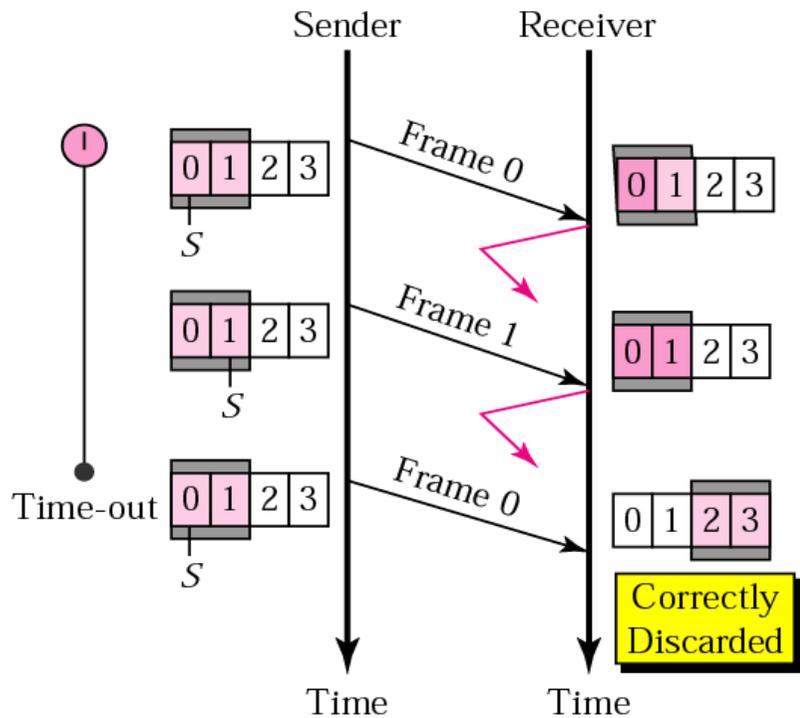
Selective Repeat ARQ, lost frame



- Frame 0 and 1 are accepted when received because they are in the range specified by receiver window.
- When frame 3 is received, it is also accepted for the same reason.
- However, the receiver sends a NAK2 to show that the frame 2 has not received
- When the sender receives the NAK2 it resends only frame 2, which is then accepted because it is in the range of the window

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

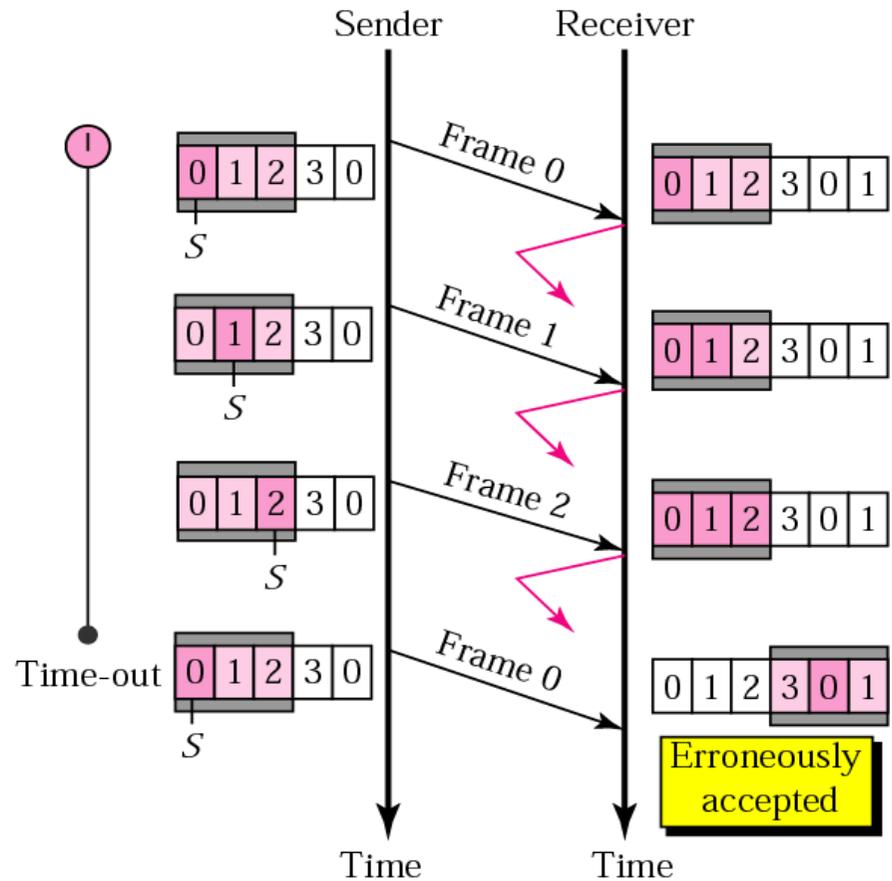
Selective Repeat ARQ, sender window size



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

If the size of the window is 2 and all acknowledgments are lost, the timer for frame 0 expires and frame 0 is resent.

However the window of the receiver is now expecting frame 2, not frame 0 so this duplicate frame is correctly discarded.



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

If the size of the window is 3 and all acknowledgments are lost, the sender sends a duplicate of frame 0.

However, this time, the window of the receiver expects to receive frame 0 (0 is part of the window), so it is accepted not as a duplicate, but as the first frame for the next cycle. This is an error

Example

In a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip.

What is the bandwidth-delay product?

If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

Solution

The bandwidth-delay product is

$$1 \times 10^6 \times 20 \times 10^{-3} = 20,000 \text{ bits}$$

The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and then back again.

However, the system sends only 1000 bits.

We can say that the link utilization is only $1000/20,000$, or 5%. *For this reason, for a link with high bandwidth or long delay, use of Stop-and-Wait ARQ wastes the capacity of the link.*

Example

What is the utilization percentage of the link in the previous example if the link uses Go-Back-N ARQ with a 15-frame sequence?

Solution

The bandwidth-delay product is still 20,000.

The system can send up to 15 frames or 15,000 bits during a round trip.

This means the utilization is $15,000/20,000$, or 75 percent.

Of course, if there are damaged frames, the utilization percentage is much less because frames have to be resent.