BLG 337E- Principles of Computer Communications

Assoc. Prof. Dr. Berk CANBERK

20/11/ 2018

-Medium Access Layer - 4

References:

Data and Computer Communications, William Stallings, Pearson-Prentice Hall, 9th Edition, 2010.

-*Computer Networking, A Top-Down Approach Featuring the Internet*, James F.Kurose, Keith W.Ross, Pearson-Addison Wesley, 6th Edition, 2012.

-Google!

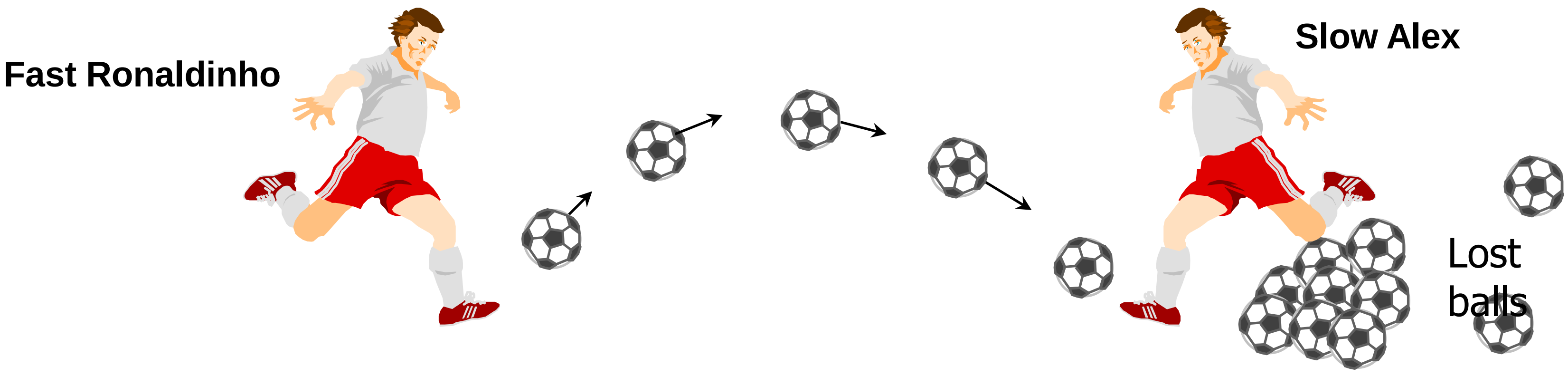
Flow and Error Control

- 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 from the receiver
  - If the channel is **error-free**:
    - Stop-and-Wait flow control
    - Sliding-Window flow control
- Error Control
  - Refers to procedures to **detect and correct** errors
  - Includes the following actions:
    - Error detection
    - Positive Acknowledgement (**ACK**): if the frame arrived with no errors
    - Negative Acknowledgement (**NAK**): if the frame arrived with errors
    - Retransmissions after **timeou**t: Frame is retransmitted after certain amount of time if no acknowledgement was received
  - These actions are called A**utomatic Repeat Request (ARQ)**

Flow and Error Control

- Usually Error and flow control protocols are combined together to provide reliable data transfer service called data link control
    - Stop-and-Wait ARQ
    - Go-Back-N ARQ
    - Selective repeat ARQ
- ARQ provide **reliable data transfer** service over **unreliable networks**
- ARQ ensure that transmitted **data** is delivered accurately to the destination despite errors that occur during transmission and satisfies the following:
    - **Error free**
    - **Without duplicates**
    - **Same order** in which they are transmitted
    - **No loss**

Flow Control

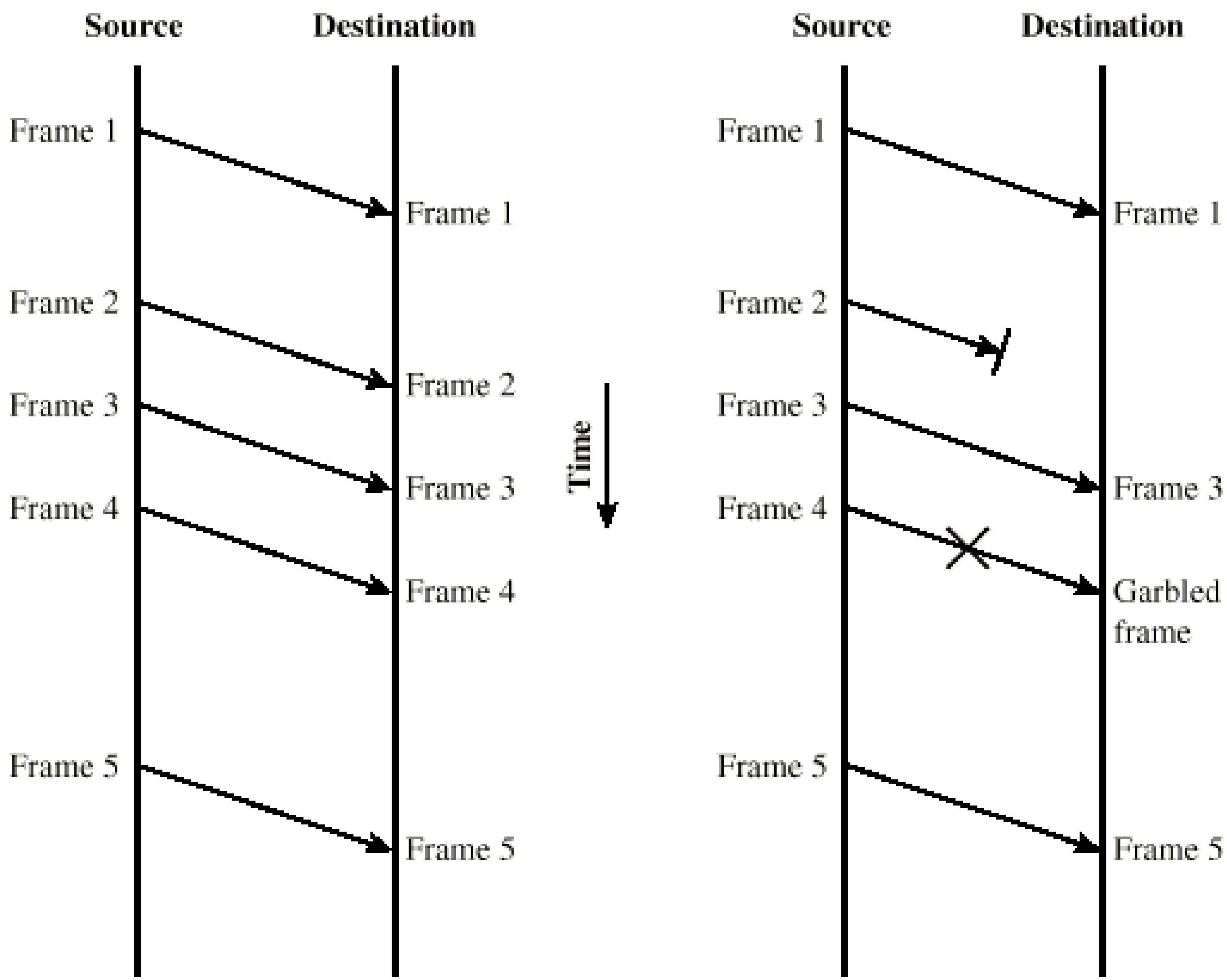


- What to do with a sender that wants to transmit frames faster than the receiver can accept them ???
- Even if transmission is error free, the receiver may be unable to handle the frames as they arrive and lose
  - Might be possible for the sender to simply insert a delay to slow down sufficiently to keep from swamping the receiver
- Two approaches for flow control
  - **Feedback-based flow control:** the receiver sends back information to the sender giving it permission to send more data or at least telling the sender how the receiver is doing.
  - **Rate-based flow control:** the protocol has a built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver.

Flow Control

- Ensuring the sending entity does not overwhelm the receiving entity
  - Preventing buffer overflow
- Transmission time (t_{frame})
  - Time taken to emit all bits into medium
- Propagation time (t_{prop})
  - Time for a bit to traverse the link

Model of Frame Transmission



(a) Error-free transmission

(b) Transmission with losses and errors

Some Flow Control Algorithms

- Simplex protocols
  - Flow control for the ideal network
  - Stop and Wait for noiseless channels
  - Stop and Wait for noisy channels
- Full duplex protocols
  - Sliding window with Go Back N
  - Sliding window with Selective Repeat

Simplex Flow Control

- Data only flows in one direction
- Acknowledgement (ACK) stream may flow in the other direction

Flow control in the ideal network:

Assumptions:

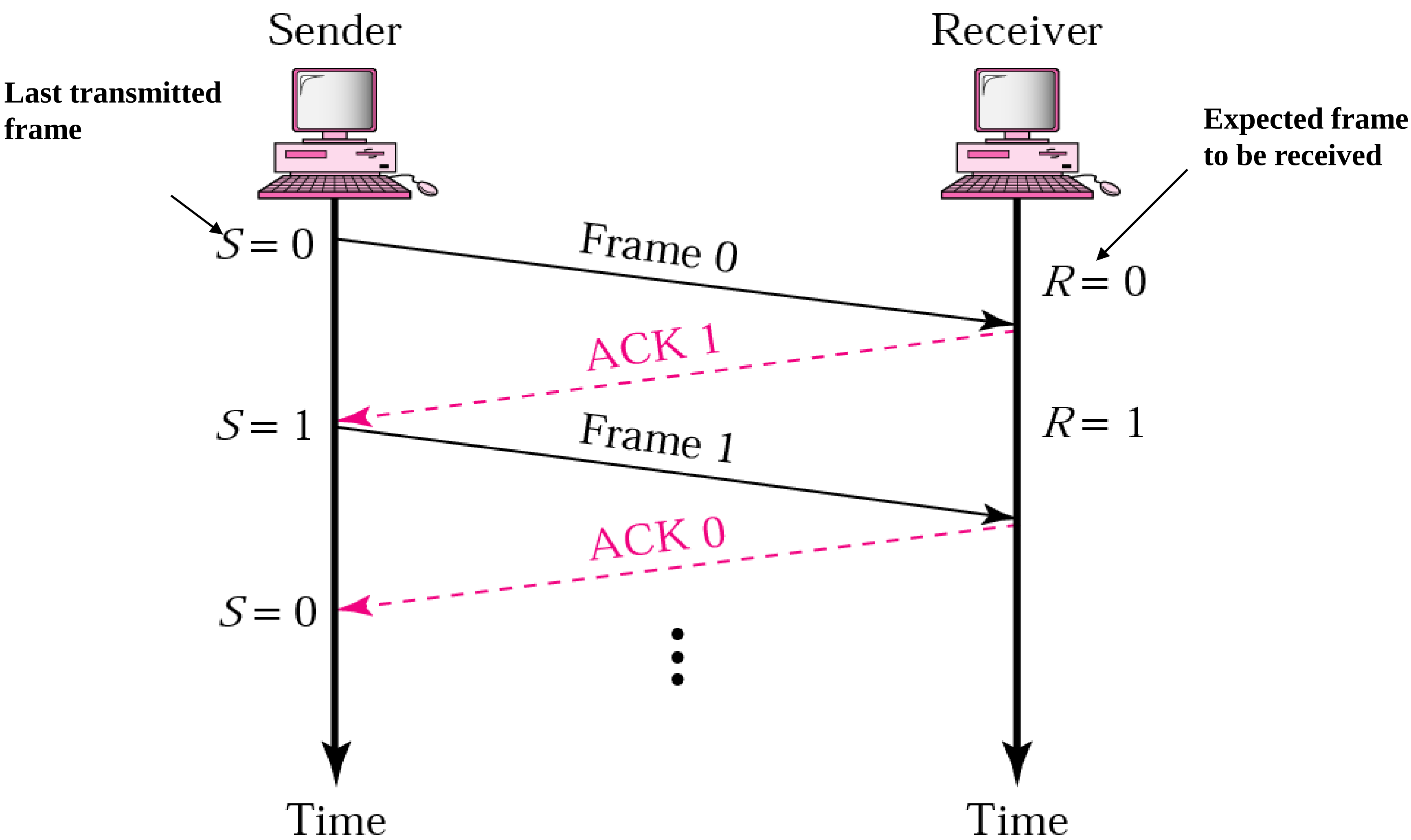
Error free transmission line,

Infinite buffer at the receiver

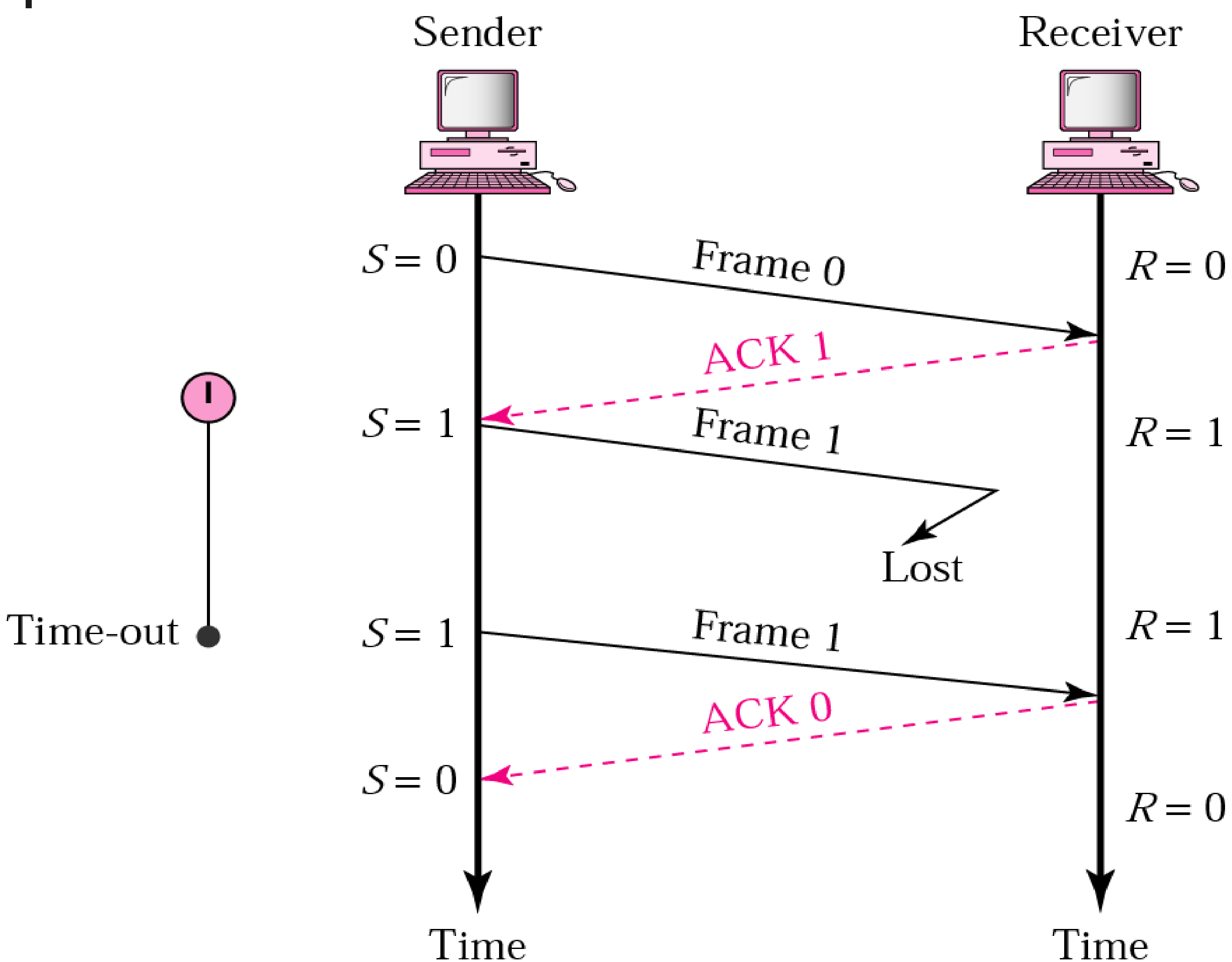
No acknowledgment of frames necessary

Since the transmission line is error-free and the receiver can buffer as many frames as it likes, no packet will ever be lost

Normal operation



Stop-and-Wait ARQ, lost frame



Stop and Wait with Noiseless Channels

Assumptions:

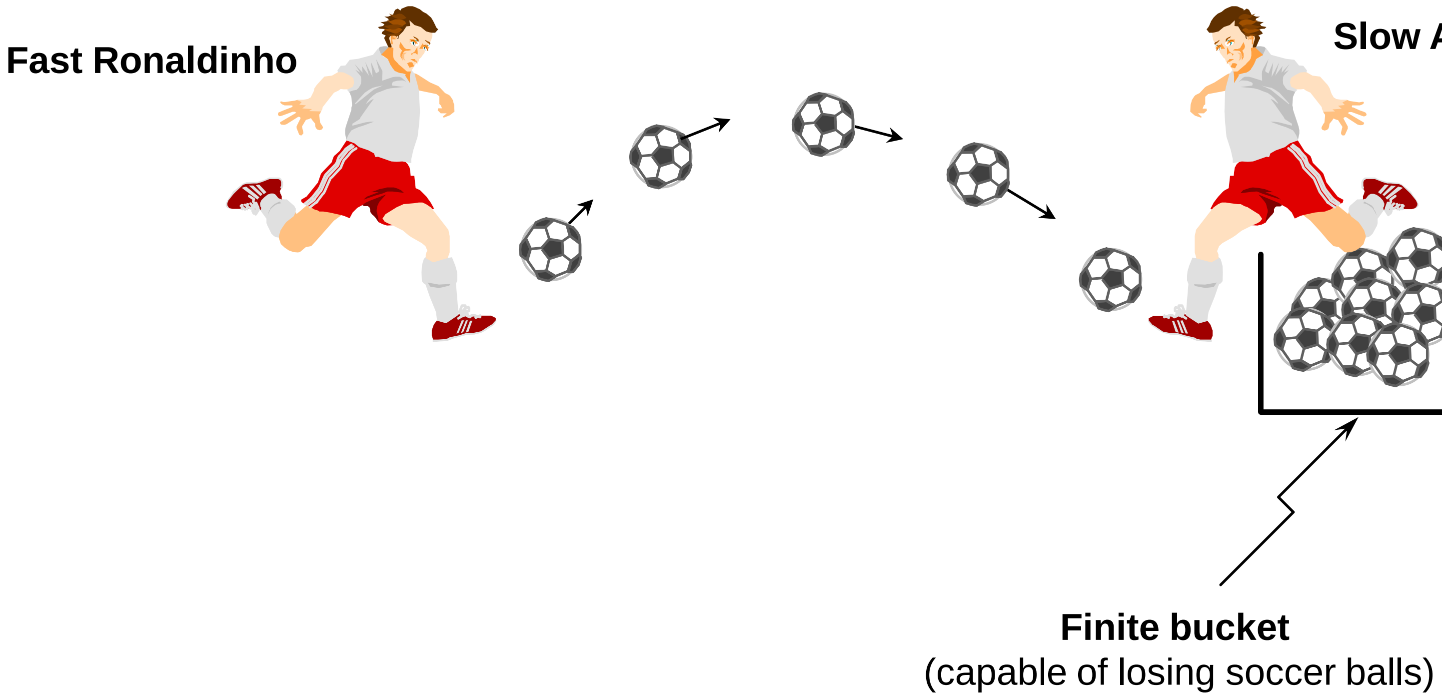
Error free transmission line,

Finite buffer at the receiver

Problem of Buffer overflow at the receiver

- Buffer overflow may happen at the receiver when the sender sends frames at a rate faster than the receiver

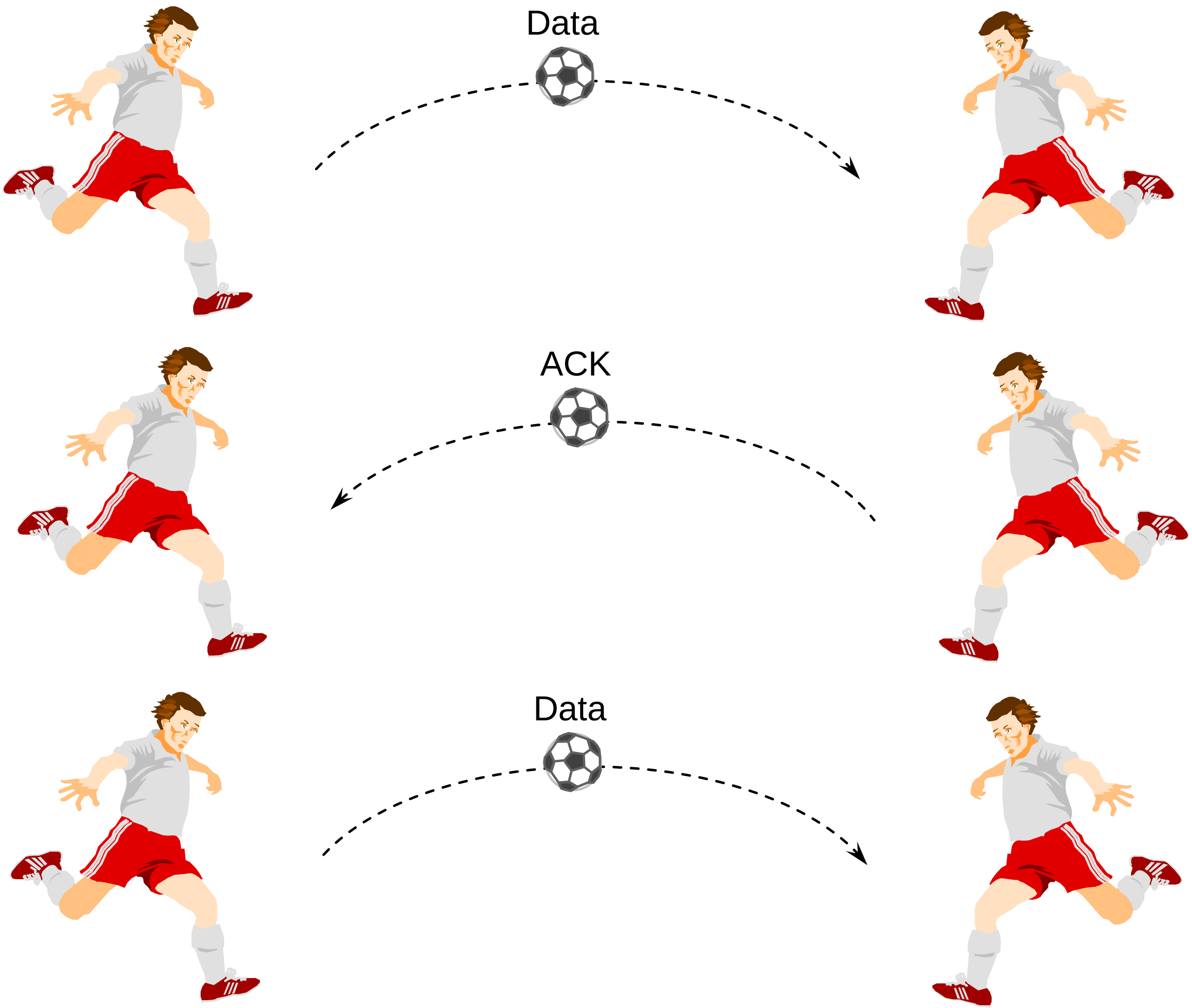
Stop and Wait with Noiseless Channels



Stop and Wait

- Source transmits frame
- Destination receives frame and replies with 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 with Noiseless Channels



Stop and Wait for Noisy Channels

Assumptions:

Transmission line may cause errors,

Finite buffer at the receiver

ACK frames may now be lost

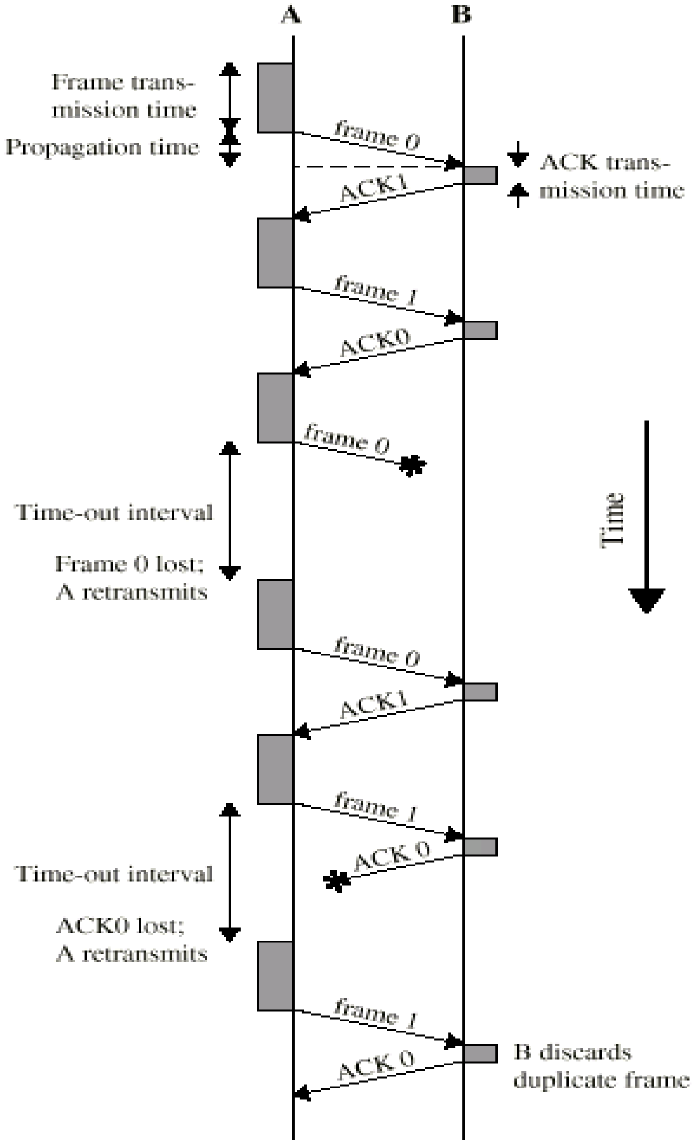
Stop and Wait

- Source transmits single frame
- Keep timeout for every frame transmitted
- Wait for ACK
- If received frame damaged, discard it
  - Transmitter has timeout
  - If no ACK within timeout, retransmit
- If ACK damaged/lost, transmitter will not recognize it
  - Transmitter will retransmit
- If receiver gets two copies of frame (duplicate frames) → discards
  - Use ACK0 and ACK1
- Use 1-bit sequence number (mod 2)

Stop and Wait Operation

- Simple
- Inefficient

Automatic Repeat reQuest (ARQ) protocol!



Automatic Repeat Request (ARQ)

- Stop and Wait
- What if a data or ACK frame is lost when using a sliding window protocol?
  - Sliding Window with Go back N
  - Sliding Window with Selective Repeat (Selective Reject or Retransmission)

Flow Control Effect on Network Performance

- Bandwidth Delay Product (BDP):
  - The product of a data link's capacity (in bits per second) and its round-trip delay time (in seconds).
  - The result, an amount of data measured in bits (or bytes), **is equivalent to the maximum amount of data on the network circuit at any given time, i.e., data that has been transmitted but not yet acknowledged**.

Example

Assume that, in a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and propagation time is 10 ms. What is the bandwidth-delay product? If the system data frames are 1000 bits in length

Solution

The bandwidth-delay product is

$(1 \times 10^6) \times (10 \times 10^{-3}) = 10{,}000$ bits

What is the time needed for an ACK to arrive? (ignore the ACK frame transmission time)?

=Frame Transmission time + 2*Propagation time = $1000/\ 10^6 + 2\text{x}\ 10 \times 10^{-3} = 0.021$ sec

How many frames can be transmitted during that time?

= (time * bandwidth) / (frame size in bits)

$0.021 \times (1 \times 10^6) = 21000$ bits / 1000 = 21 frames

What is the Link Utilization if stop-and-wait ARQ is used?

Link Utilization = (number of actually transmitted frame/ # frames that can be transmitted)*100

(1/21) x 100 = 5 %

Full Duplex Flow Control Protocols

Data frames are transmitted in both directions

Sliding Window

Flow Control Protocols

Sliding Window Flow Control

- Allow multiple frames to be in transit
- Receiver has buffer W long (receiver window)
- Transmitter can send up to W frames without ACK (sender window)
- Each frame is numbered (sequence number)
- ACK includes number of next frame expected
  - Ack for frame n = I am expecting frame n+1 (not "I received fame n")
- Sequence number bounded by size of field
  - e.g. k bits: Frames are numbered modulo 2^k

Sliding Window Protocols
Definitions

- Sequence Number: Each frame is assigned a sequence number that is incremented as each frame is transmitted
- Sender's Window: Keeps sequence numbers of frames that have been sent (or can be sent) but not yet acknowledged
- Sender Window size: The number of frames the sender have already sent and may transmit before receiving ACKs
- Receiver's Window: Keeps sequence numbers of frames that the receiver is allowed to accept
- Receiver Window size: The maximum number of frames the receiver may receive out of order

- The sending and receiving windows do not have to be the same size
- Any frame which falls outside the receiving window is discarded at the receiver

Sliding Window Protocols
Piggybacking Acknowledgements

Since we have full duplex transmission, we can "piggyback" an ACK onto the header of an outgoing data frame to make better use of the channel

When a data frame arrives, instead of immediately sending a separate ACK frame, the receiver waits until it is passed the next data frame to send. The ACK is attached to the outgoing data frame.

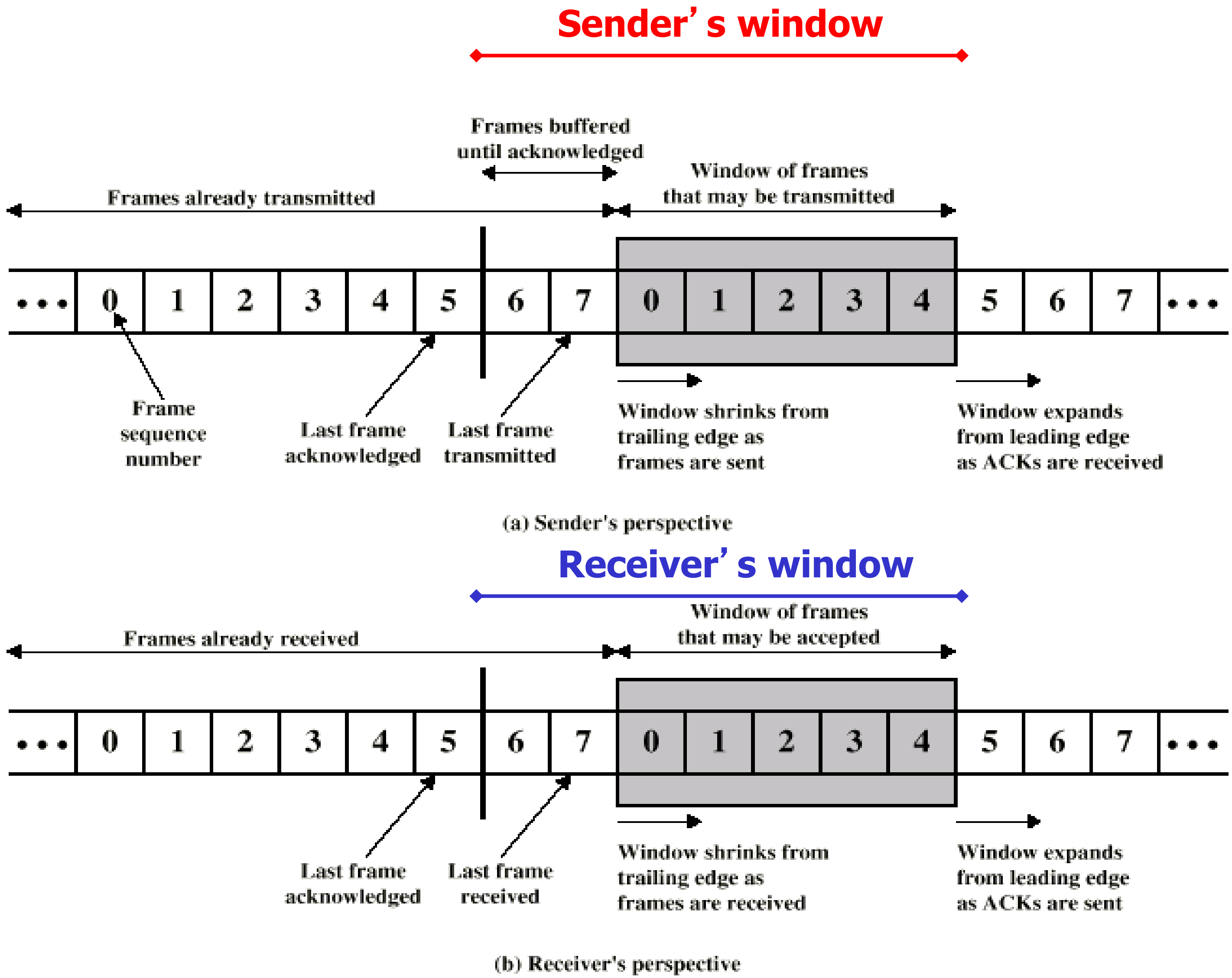
Sliding Window with Window Size W

- With a window size of 1
  - the sender waits for an ACK before sending another frame
  - This protocol behaves identically to stop and wait for a noisy channel!

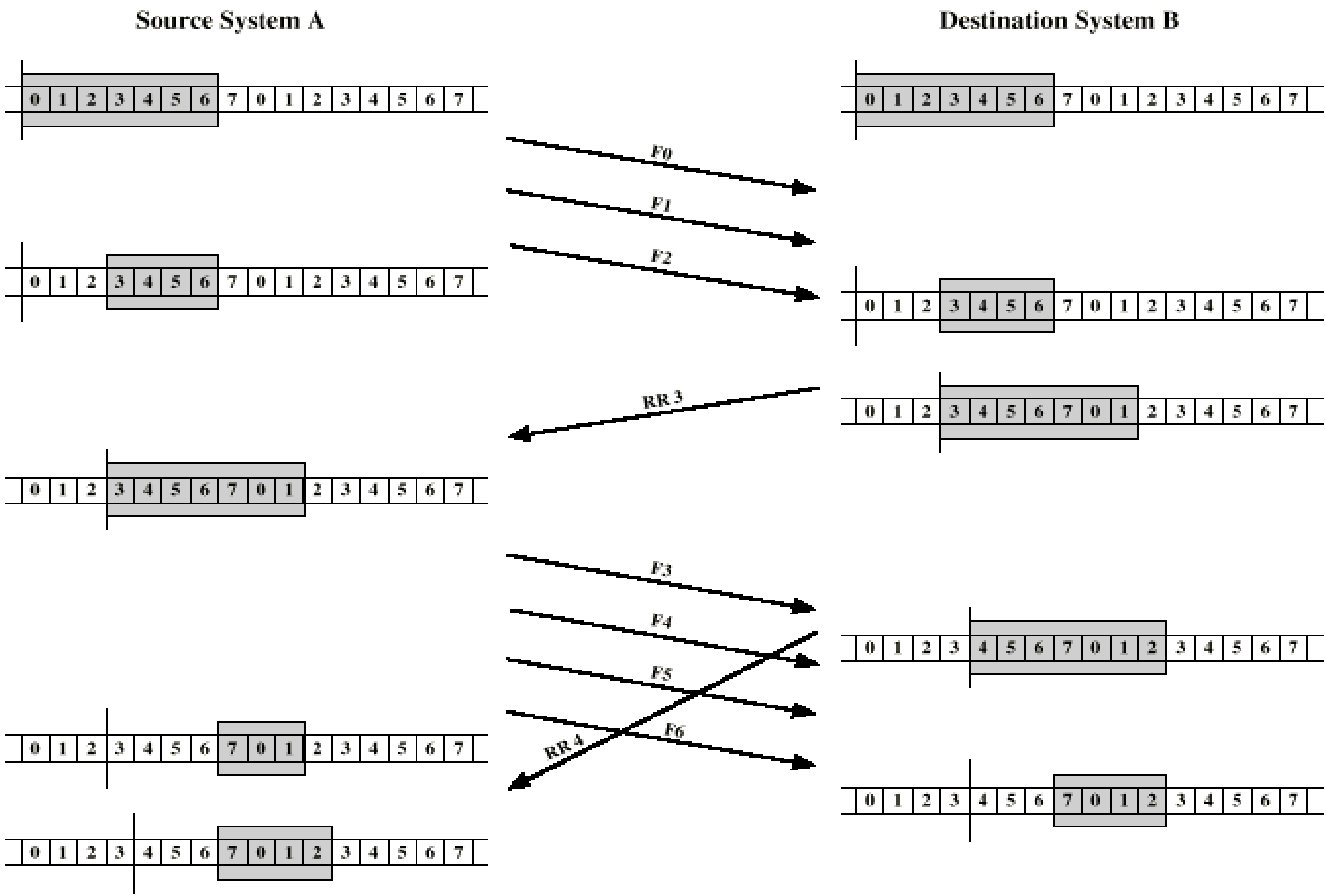
- With a window size of W, the sender can transmit up to W frames before “being blocked”

- We call using larger window sizes **Pipelining**

Sliding Window Diagram



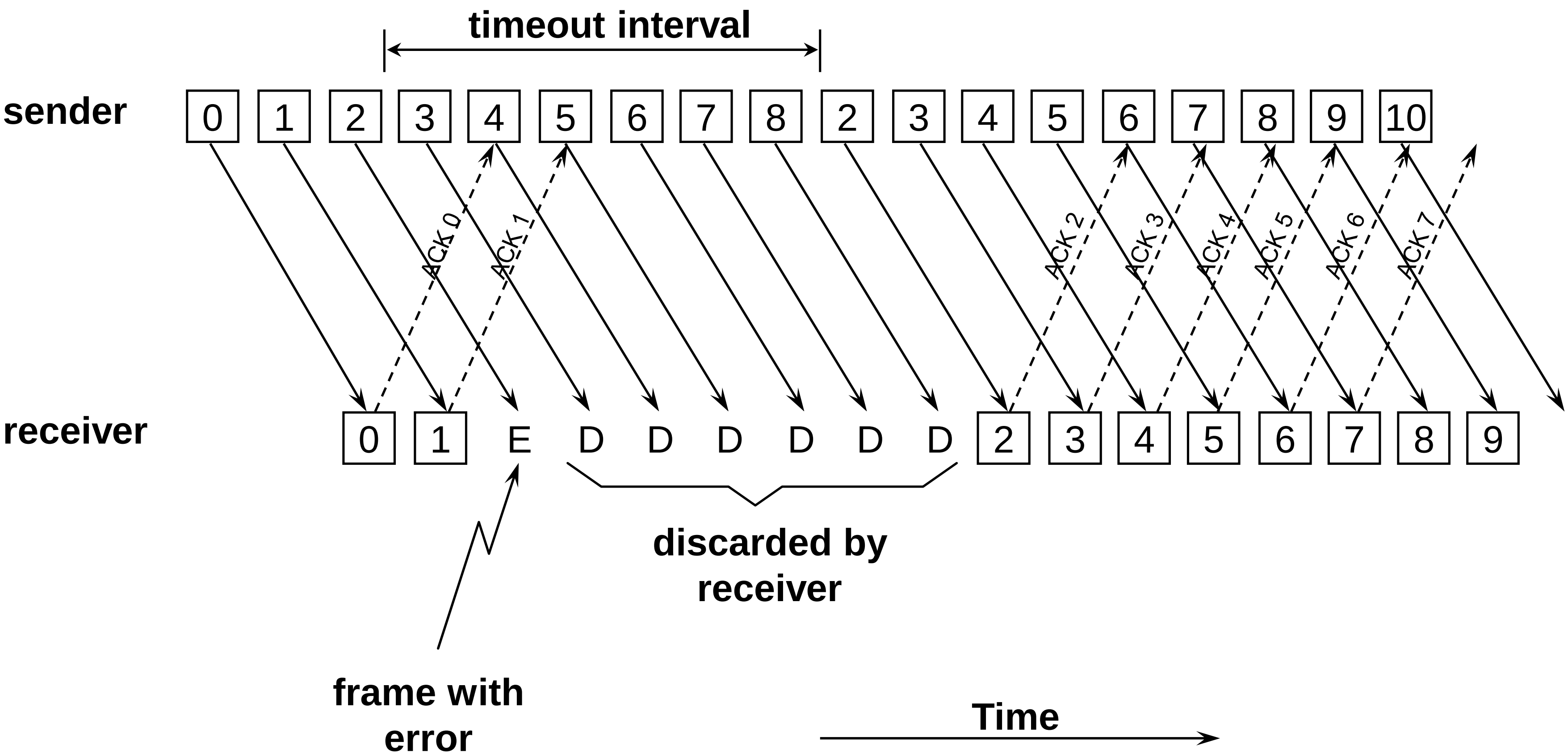
Example Sliding Window



Sliding Window with Go Back N

- 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/lost frame received correctly
  - Transmitter must go back and retransmit that frame and all subsequent frames

Go Back N

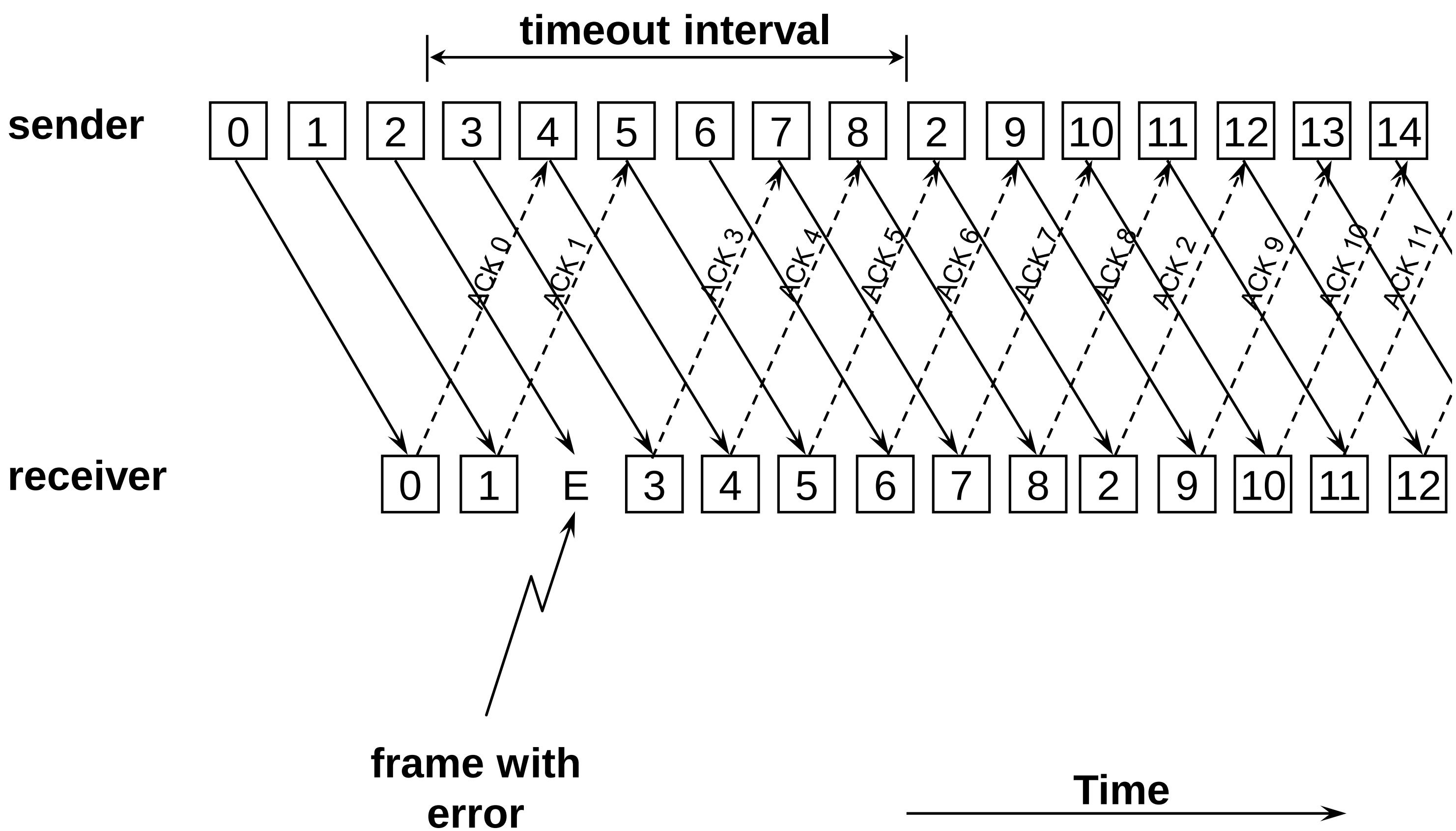


- Assume ACK per frame by receiver (receiver window=1)
- Go Back N can recover from erroneous or missing packets
- Inefficient → if there are a lot of errors, the sender will spend most of its time retransmitting

Sliding Window with Selective Repeat

- The sender retransmits only the frame with errors
- The receiver stores all the correct frames that arrive following the bad one (receiver window > 1)
- Requires a significant amount of buffer space at the receiver
- When the sender notices that something is wrong, it just retransmits the one bad frame, not all its successors
- Might be combined with Negative Acknowledgment (NACK)

Selective Repeat



Selective Repeat with NACK