CSE 350
DATA COMMUNICATIONS

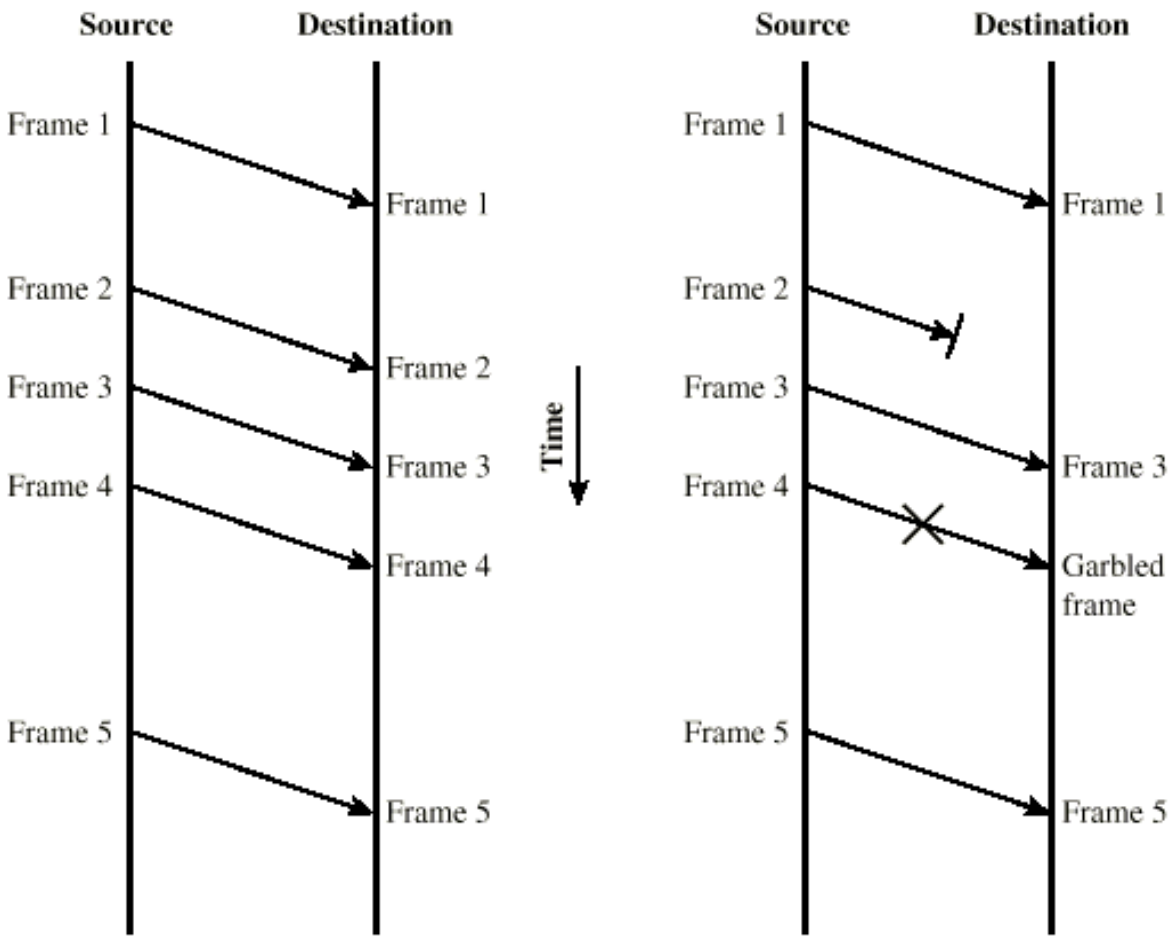
Lecture 6: Data Link Control

Maheen Islam

Flow Control

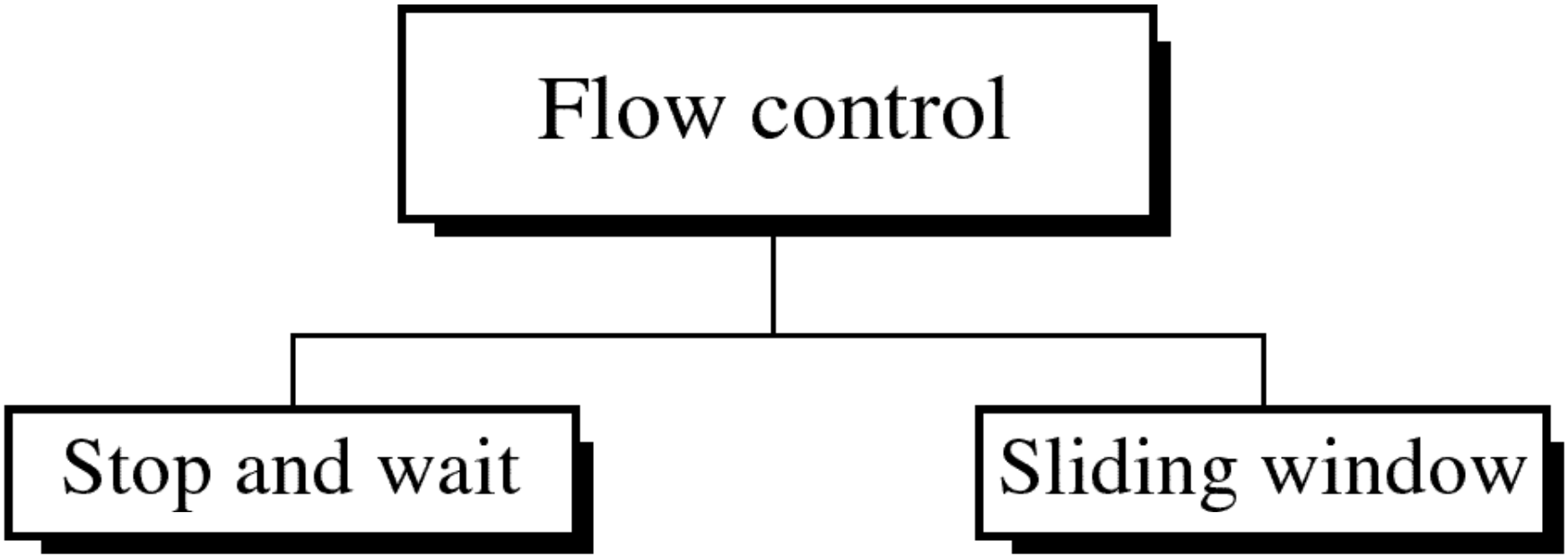
- Ensuring the sending entity does not overwhelm the receiving entity with data, sending and receiving rate same
  - Preventing buffer overflow
- The receiving entity typically allocates a data buffer of some maximum length for a transfer
- In the absence of flow control the receiver buffer may fill up and overflow
- Transmission time
  - Time taken to emit all bits into medium, proportional to the length of the frame
- Propagation time
  - Time for a bit to traverse the link between source and destination

Model of Frame Transmission



(a) Error-free transmission

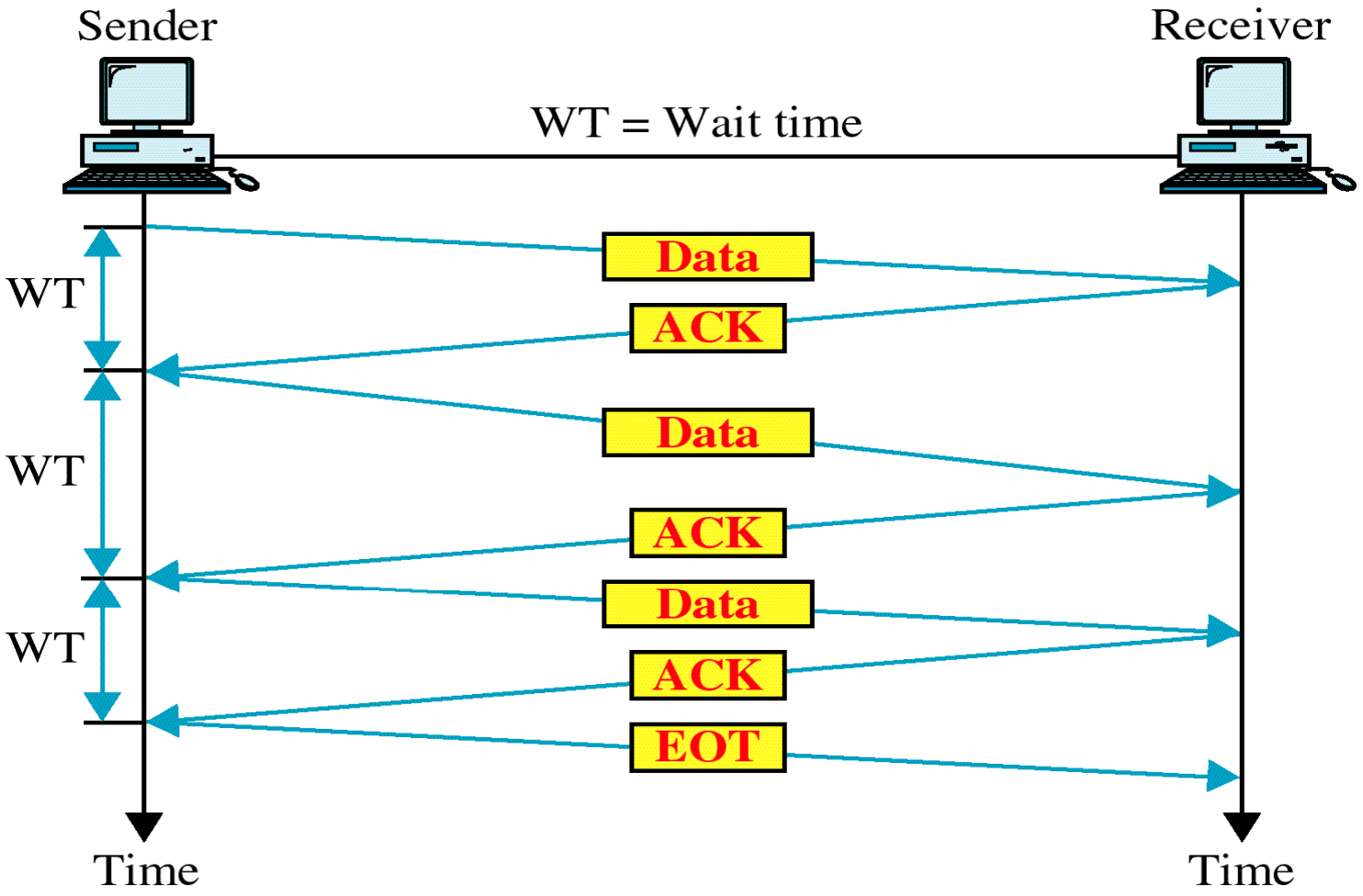
(b) Transmission with losses and errors


Flow control
Stop and wait
Sliding window
Send one frame at a time
Send several frames at a time

Stop and Wait

- Simplest form
- 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 when a message is sent in a few large frames

Stop and Wait



Fragmentation

- Large block of data may be split into small frames
  - Limited buffer size of receiver
  - Longer transmission, greater no. of errors.
  - Errors detected sooner (when whole frame received)
  - On error, retransmission of smaller frames is needed
  - Prevents one station occupying medium for long periods

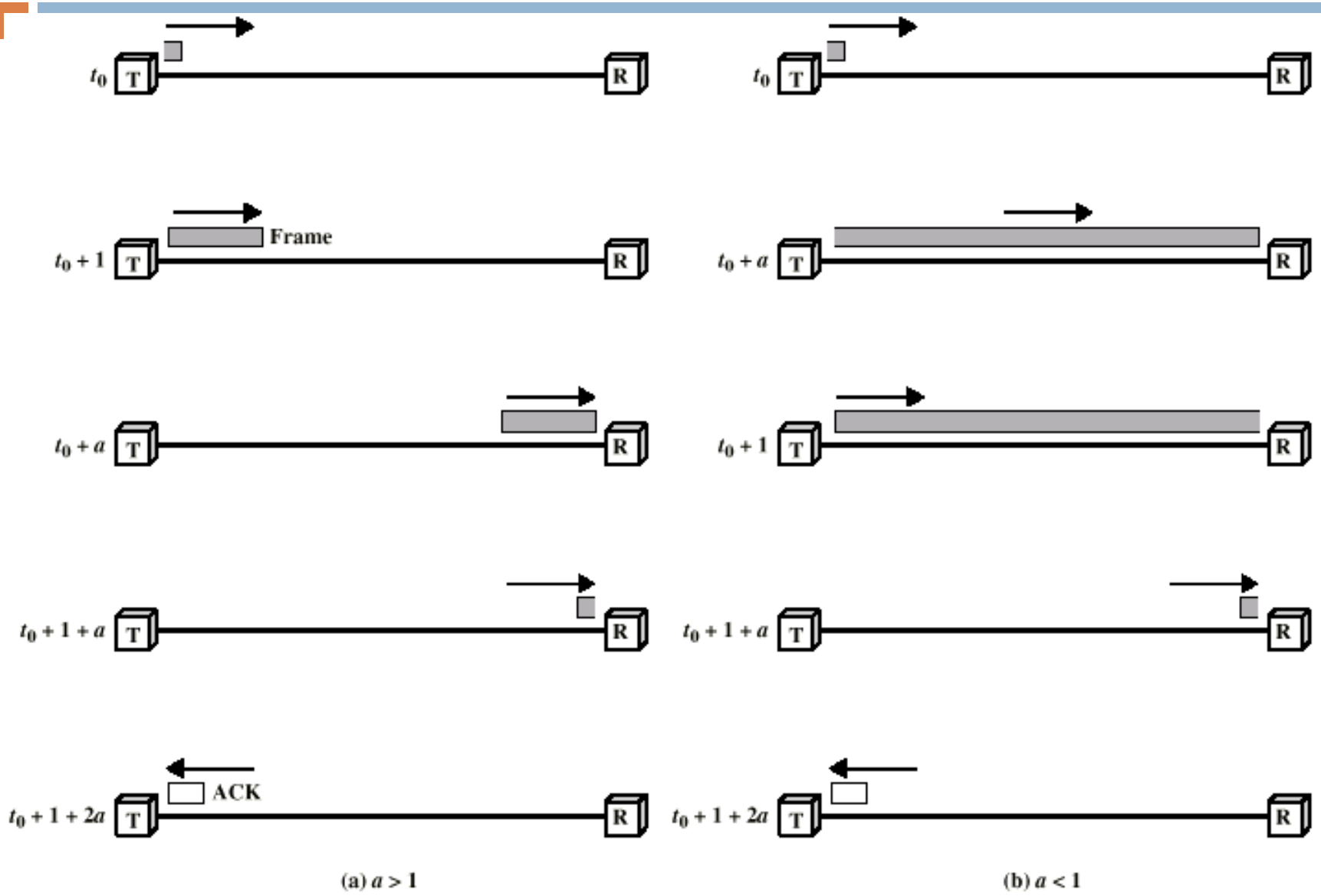
Pros and Cons

- Simplicity –each frame is checked and acknowledged before the next frame is sent
- Inefficiency–
  - Slow—each frame must travel all the way to the receiver and an ack must travel all the way back before the next frame can be sent
  - each frame is alone on the line—each frame sent and received uses the entire time needed to traverse the link
  - use of multiple frames for a single message

Stop and Wait Link Utilization

- Where the bit length of the link is greater than the frame length, serious inefficiencies occur.
- Transmission time is normalized to 1.
- Propagation delay is expressed as the variable a.
- When $a<1$, the line is inefficiently used
- When $a>1$, the line is always underutilized.

Stop and Wait Link Utilization



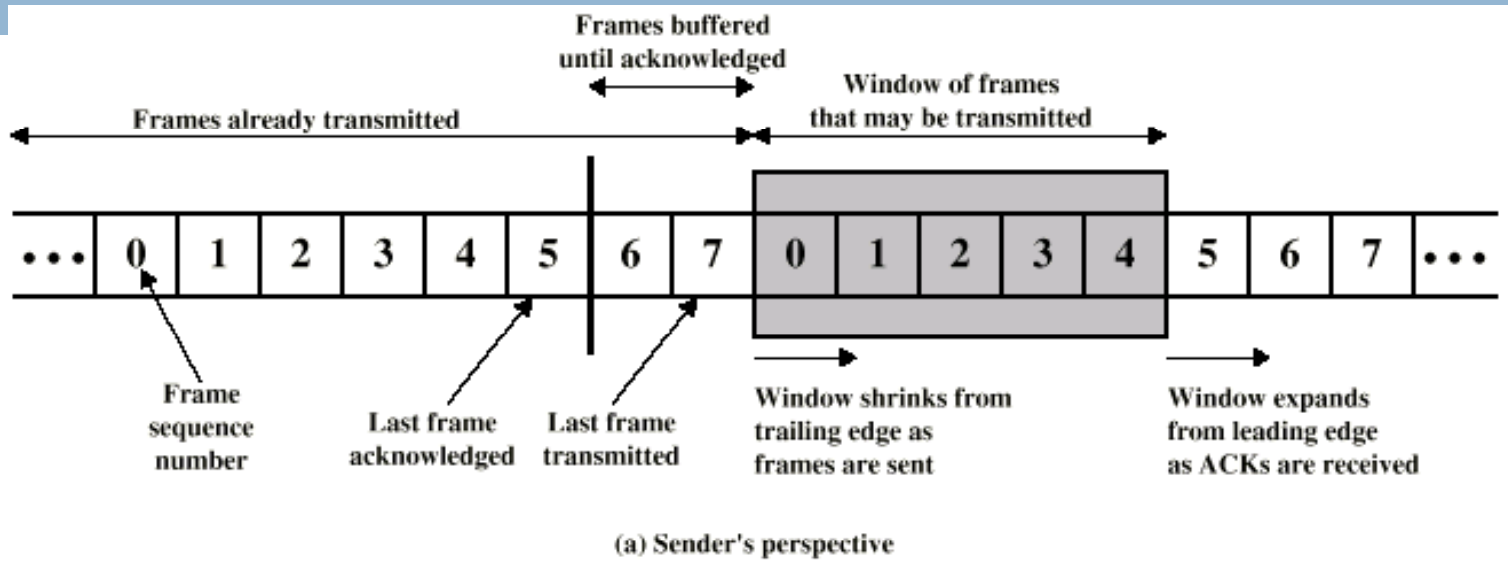
Sliding Window Flow Control

- Allow multiple frames to be in transit
- Receiver has buffer W long
- Transmitter can send up to W frames without ACK
- Each frame is numbered
- ACK includes number of next frame expected
- Sender maintains a list of sequence numbers that it is allowed to send
- Receiver maintains a list of sequence numbers that it is prepared to receive
- Each of these lists can be thought of as a ***window*** of frames

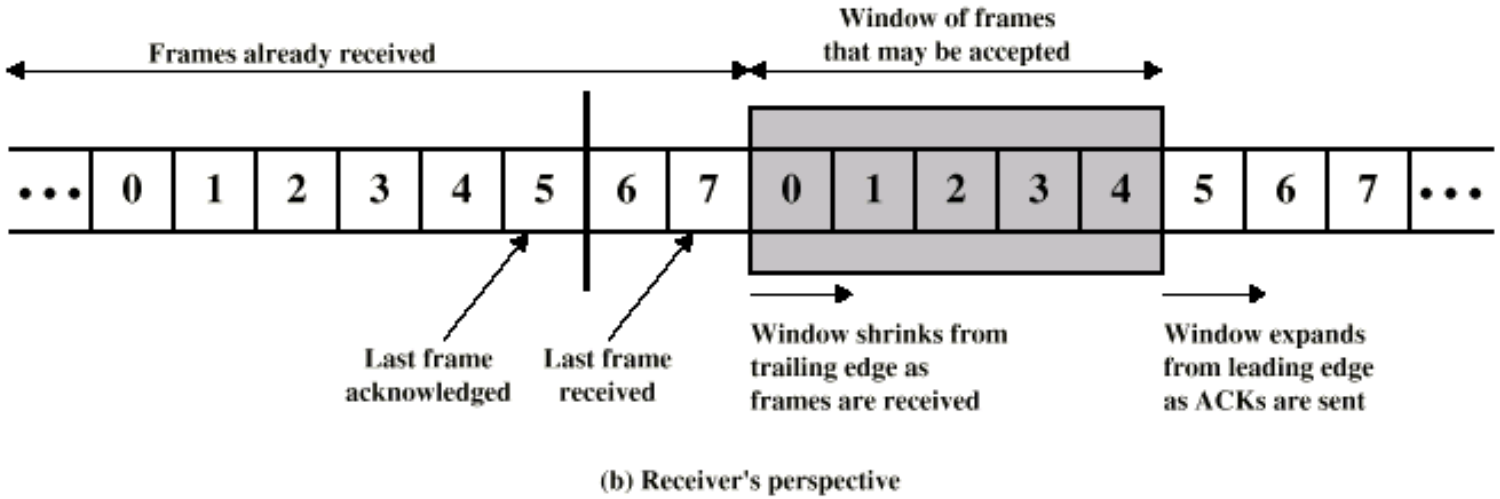
Frame Sequence number and window size

- Sequence number to used occupies a field in the frame, bounded by size of field (k)
  - Frames are numbered modulo 2^k
- For a k-bit field the range of sequence number is 0 through $2^k - 1$
- This is the max window size
- The window size need not be the maximum possible size for a given sequence number length

Sliding Window Diagram

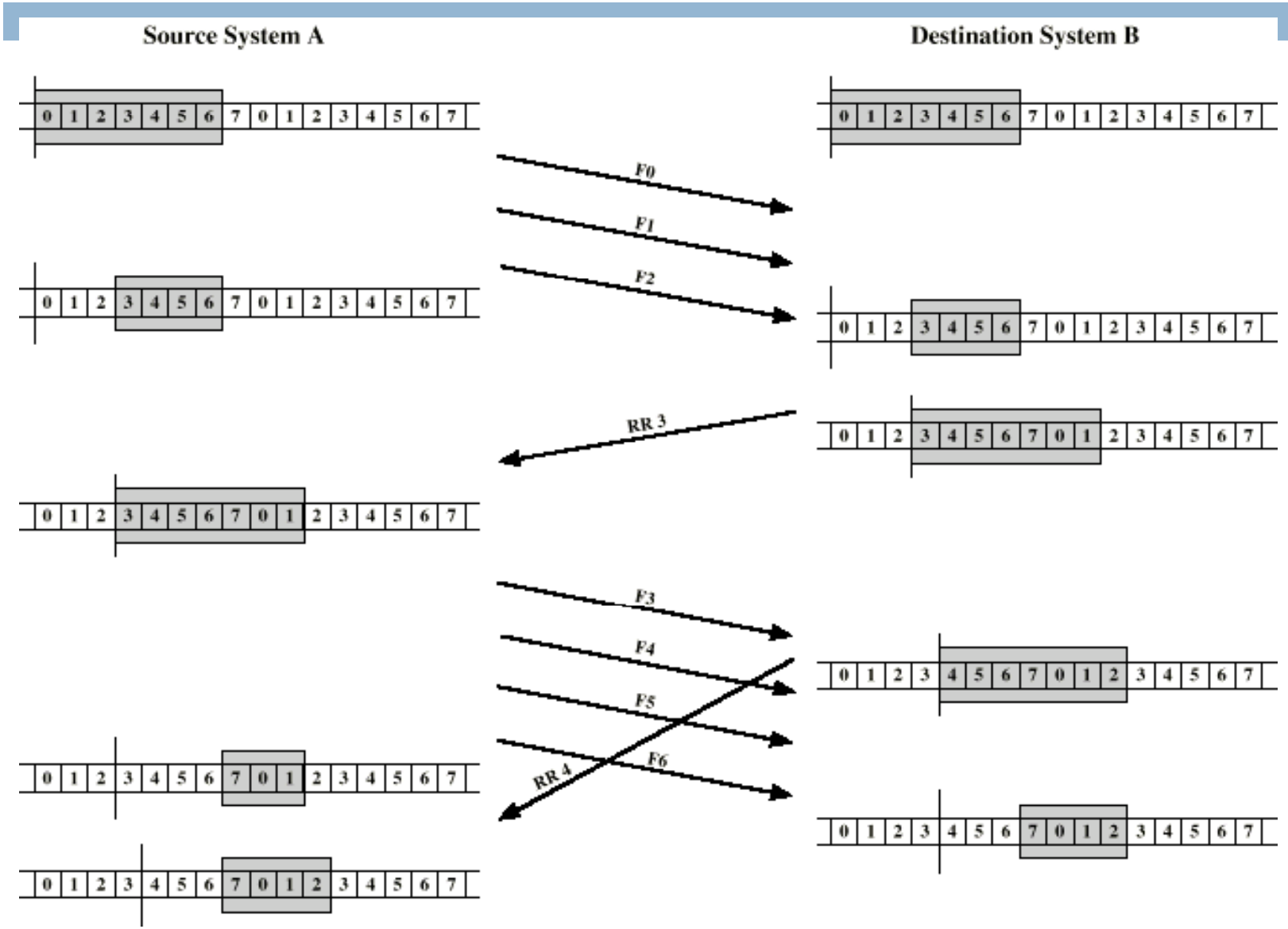


(a) Sender's perspective



(b) Receiver's perspective

Example Sliding Window



Sliding Window Enhancements

- Receiver can acknowledge frames without permitting further transmission (Receive Not Ready--RNR)
- Must send a normal acknowledge to resume
- If duplex, use piggybacking
  - If no data to send, use acknowledgement frame,RR,RNR
  - If data and ack to send, sends both together in one frame
  - If data but no acknowledgement to send, send last acknowledgement number again

Advantage

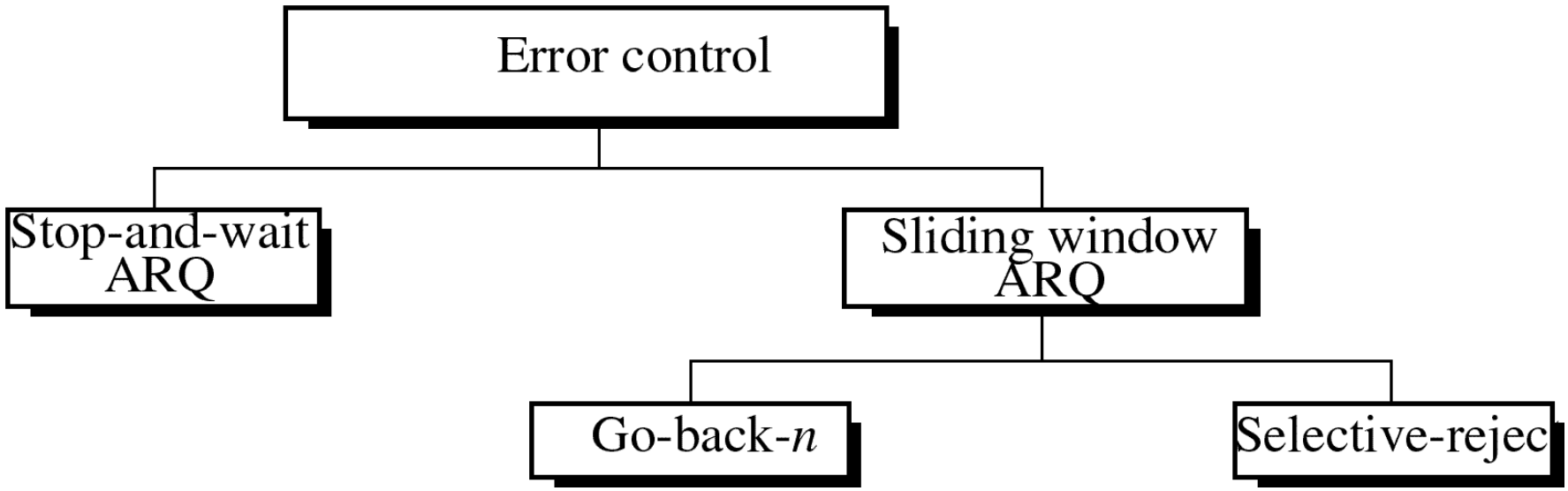
- Sliding window is more efficient than stop-and-wait flow control.
  - The transmission link is treated as a pipeline that may be filled with frames in transit

Error Control

- Detection and correction of errors
- Possibility of two types of errors: -
  - Lost frames -- frame fails to arrive at the other side
  - Damaged frames – a frame does arrive, but some of the bits are in error
- Common techniques for error control
  - Error detection
  - Positive acknowledgment – successfully received error free frames
  - Retransmission after timeout
  - Negative acknowledgment (NAK) and retransmission

•Collectively those mechanisms are referred to as **Automatic repeat request (ARQ)** – turn an unreliable data link into a reliable one.

•3 types of ARQ have been standardized:



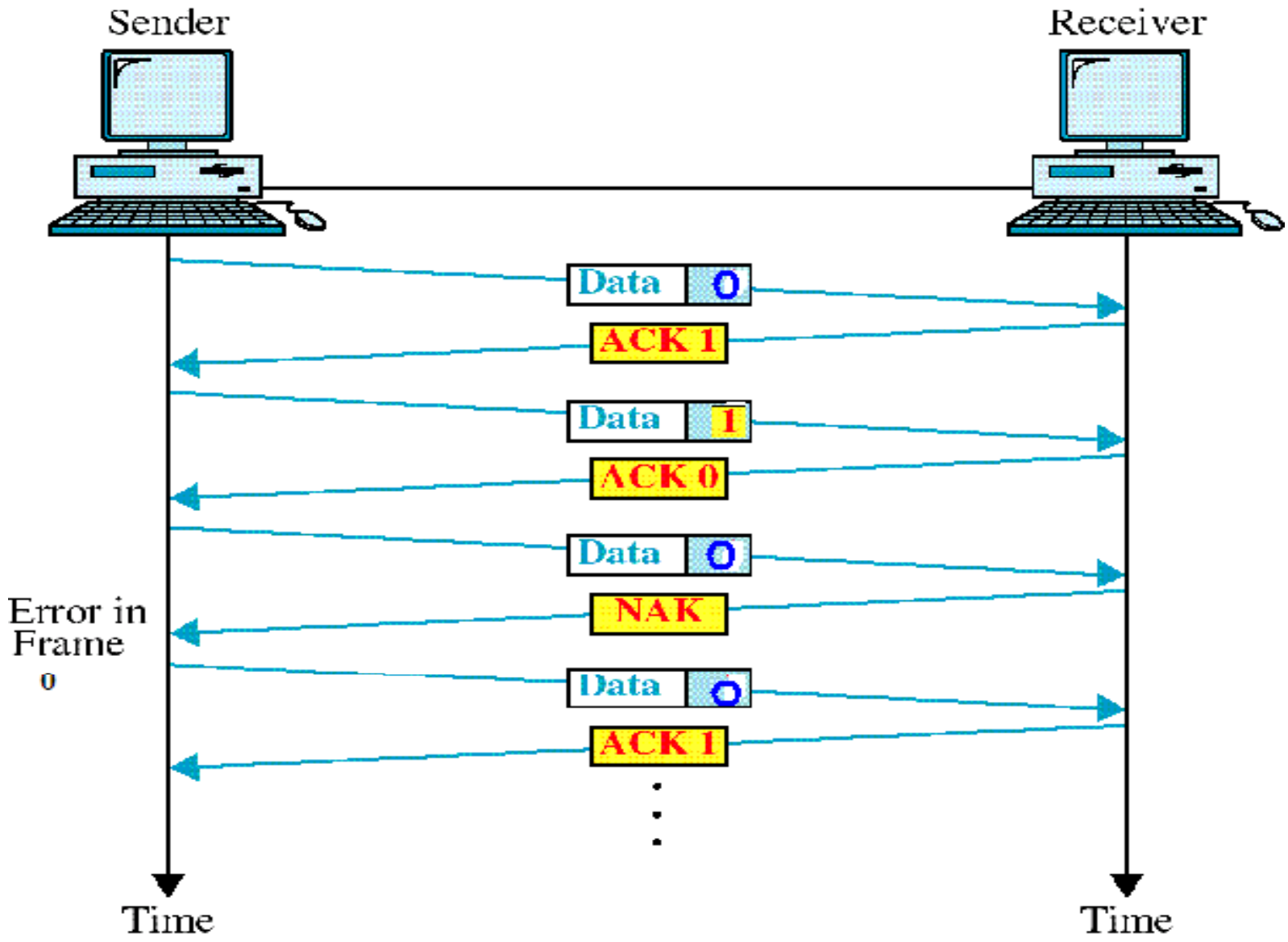
Stop and Wait ARQ

- Form of stop-and-wait flow control
- Extended to include retransmission of data in case of lost or damaged frames
- Four features are added to basic flow control
  - Sender keeps a copy of the last frame transmitted until it receives an ack
  - Both data and **ACK** frames are numbered alternately 0 and 1
  - If an error is discovered in a data frame, a **NAK** frame is returned. NAK frames are not numbered, tell the sender to retransmit the last frame sent after the last ack.
  - A timer is set for sending device. If an expected ack is not received within an allotted time period, the sender assumes that the last data frame was lost in transit and sends it again.

Damaged frame

- If there is an error in a frame, the receiver returns a NAK frame
- The sender retransmits the last frame

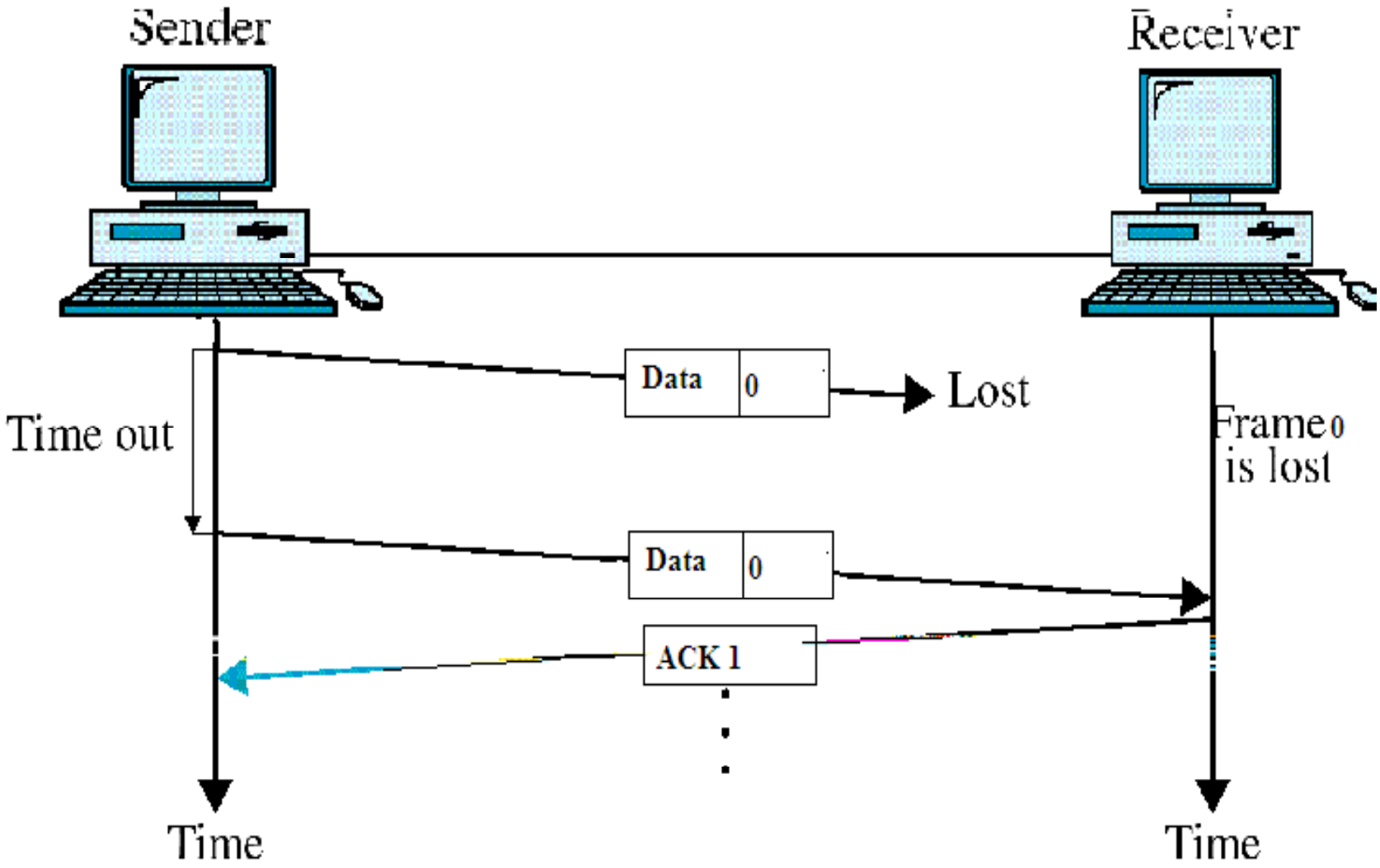
Damaged Frame



Lost data frame

- The sender sets a timer that starts every time a data frame is transmitted
- If the frame never reaches to the receiver, the receiver can never ack
- The sending device waits for an ACK or NAK frame until its timer goes off
- The sender retransmits the last data frame and restarts its timer and waits for an ack.

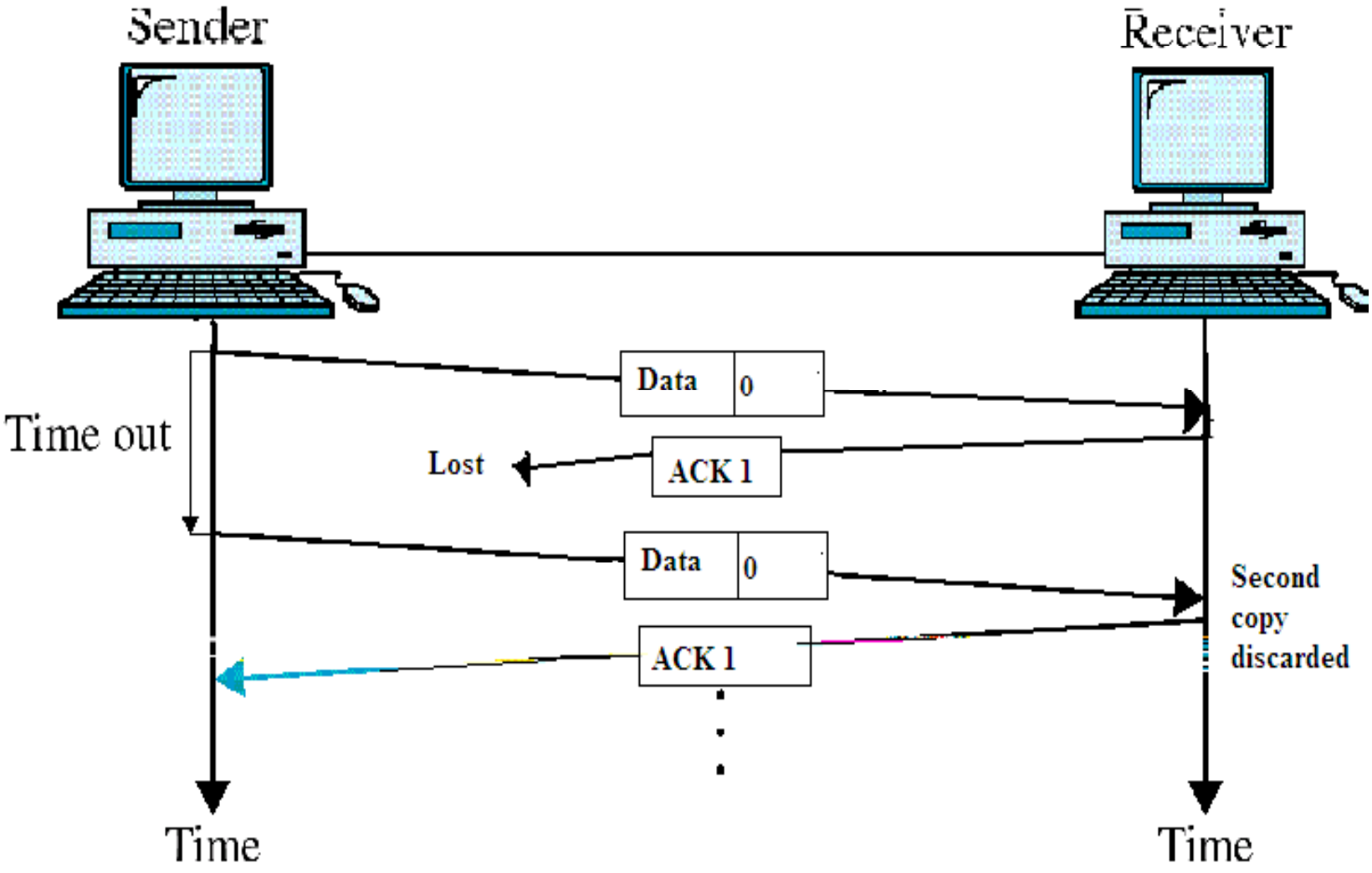
Lost Frame


Sender
Receiver
Data
0
Lost
Time out
Frame 0
is lost
Data
0
ACK 1
Time
Time

Lost acknowledgment

- The ACK or NAK frame is lost
- The sending device waits until its timer goes off
- Then retransmits the data frame
- The receiver checks the number of the new data frame
- If the lost frame was a NAK, the receiver accepts the new copy and returns ack
- If the lost frame was an ACK, the receiver recognizes the new copy as a duplicate, ack its receipt, discards it.
- Wait for the next frame

Lost ACK frame



Stop and Wait - Pros and Cons

- Simple
- Inefficient

Sliding window ARQ

- To extend sliding window flow control :-
  - The sending device keeps copies of all transmitted frames until they have been acknowledged.
  - The receiver has the option of returning a NAK frame.
    - Both ACK and NAK frames must be numbered for identification.
    - ACK frames carry the number of the next frame expected
    - NAK frames carry the number of damaged frame
    - For both ACK AND NAK The message to the sender is the number of the frame that the receiver expects next
    - Data frames that are received without errors do not have to be ack individually.
    - Every damaged frame must be acknowledged

- The sender sets a timer to handle lost acks
  - The whole window may be sent before an ack
  - If all frames are waiting ack, the sender starts a timer and waits before sending any more
  - If the allotted time has run out with no ack, the sender assumes the frames were not received
  - And retransmits one or all of the frames depending of the protocol.

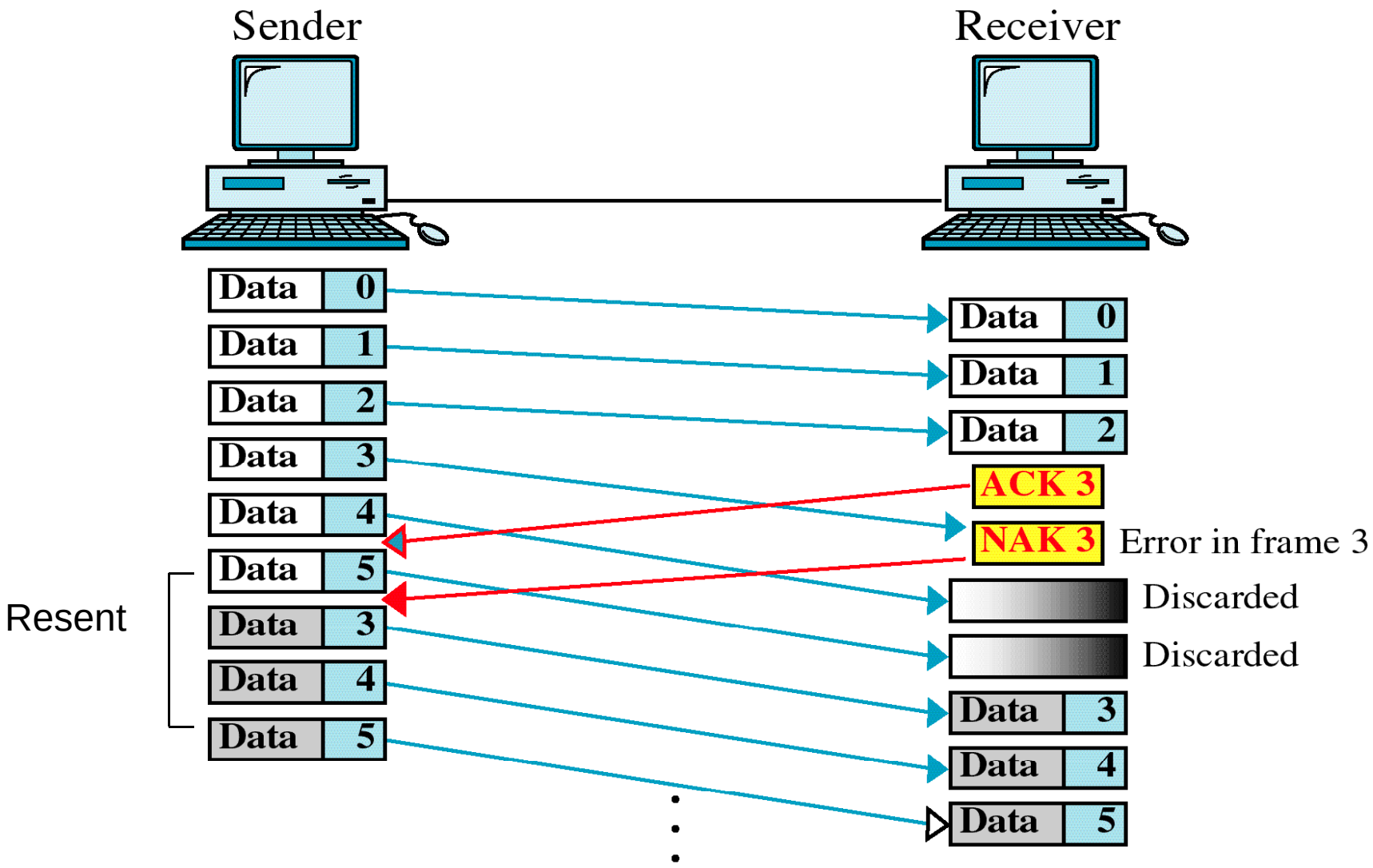
Go Back n ARQ

- If one frame is lost or damaged, all frames sent since the ack are retransmitted

<u>Damaged frame</u>

- NAK means two things:-
  1. A positive ack of all frames received prior to damaged frame
  2. Negative ack of the frame indicated

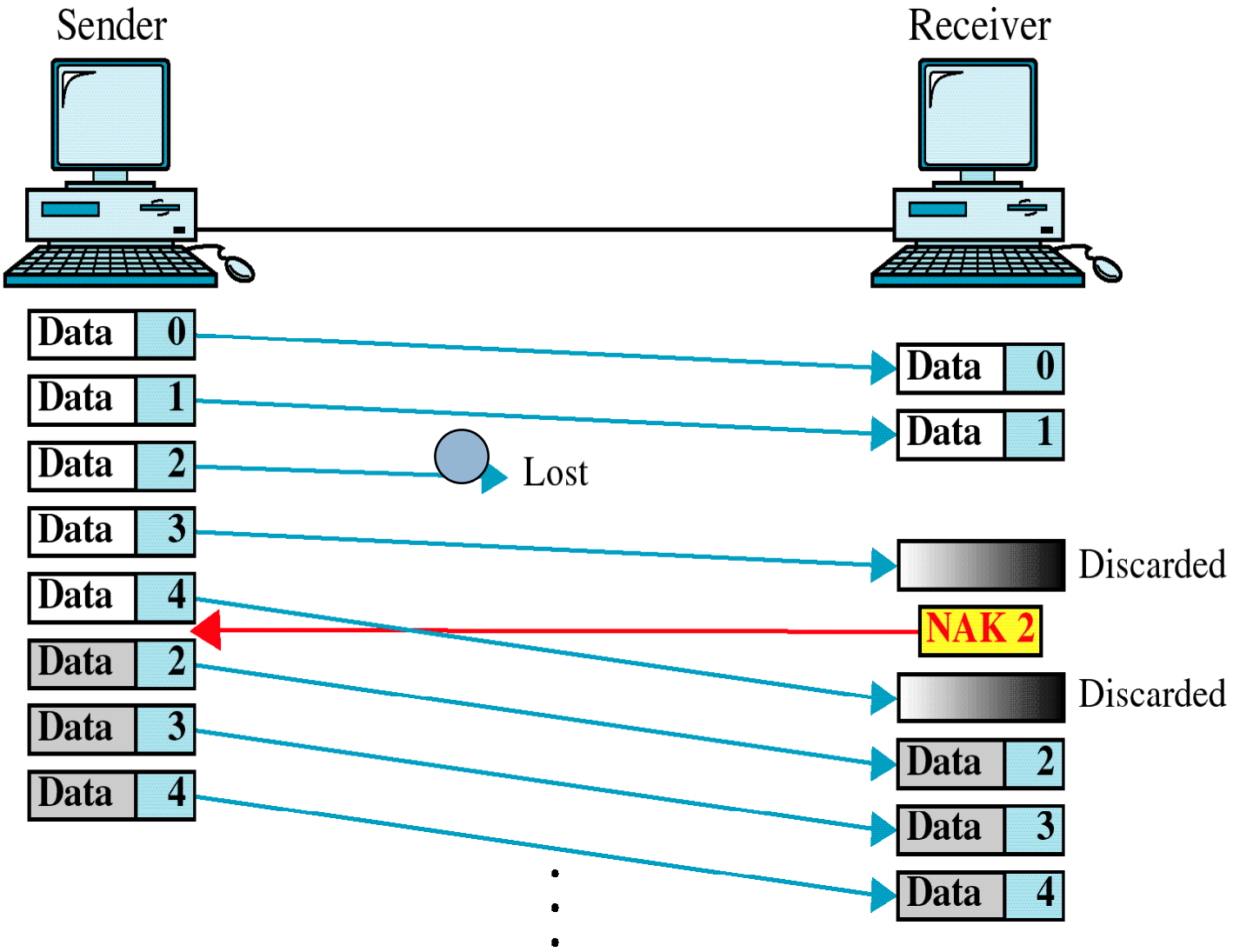
Damaged Frame



Lost Data Frame

- If one or more frames are lost in transit
- The next frame to arrive at the receiver will be out of sequence
- The receiver checks the id number of each frame
- One or more have been skipped
- Returns a NAK for the first missing frame
- The sending device retransmits the frame
- As well as any frames that it had transmitted after the lost one

Lost Frame


Sender
Receiver
Data 0
Data 1
Data 2
Data 3
Data 4
Data 2
Data 3
Data 4
Lost
Data 0
Data 1
Discarded
NAK 2
Discarded
Data 2
Data 3
Data 4

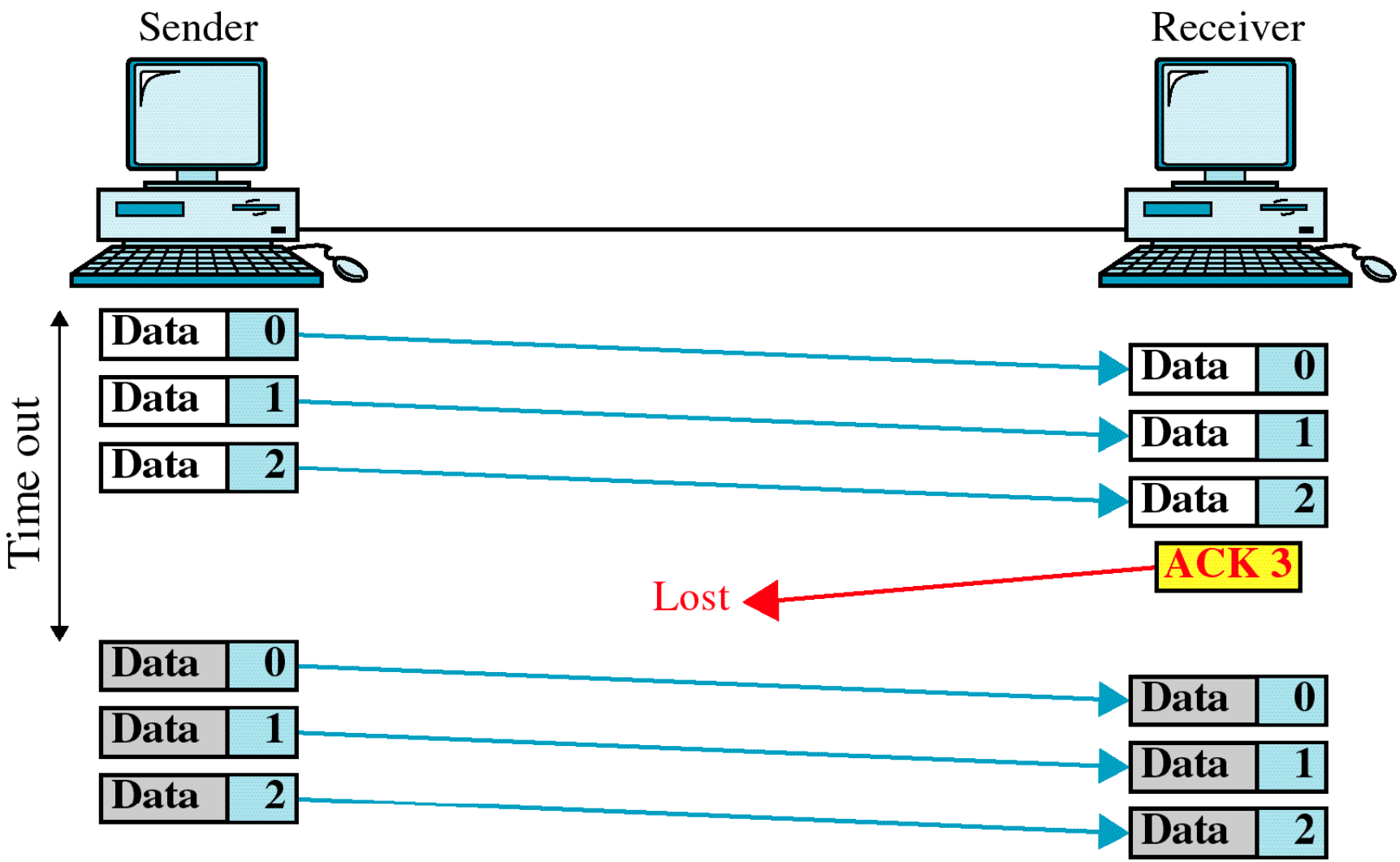
Lost Acknowledgement

- Sender is not expecting to receive an ACK frame for every data frame
- Sender uses a timer
- The sending device can send as many frames as the window size before waiting for an ACK
- Once the time limit has been reached
- Or the sender has no more frames to send
- It must wait
- If the ACK has been lost, the sender could wait for ever

Lost Acknowledgement

- So, the sender sets a timer
- Begins counting when the window capacity is reached
- If an ACK is not received within the time limit
- Sender retransmits every frame transmitted since the last ACK

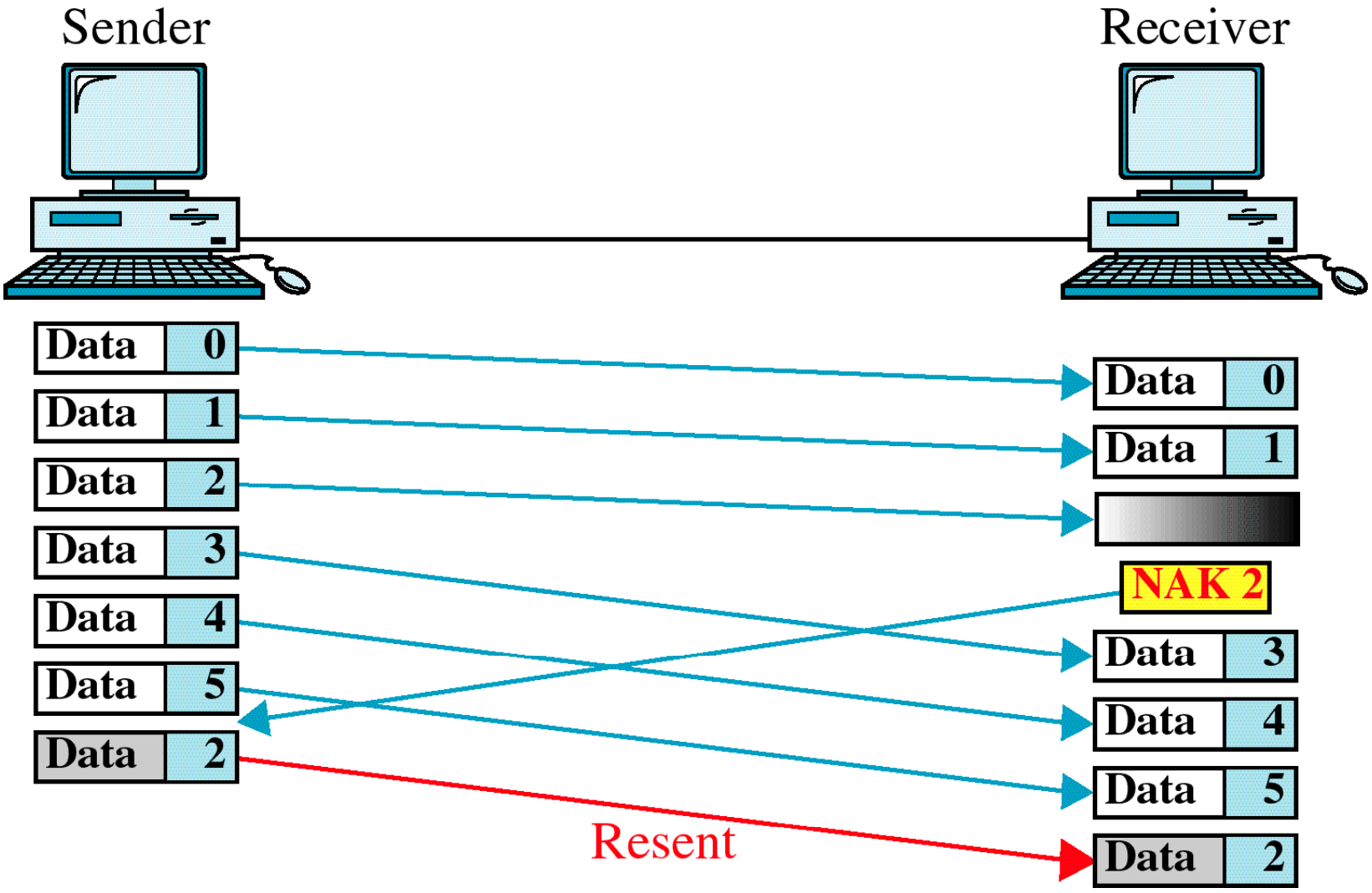
Lost ACK


Sender
Receiver
Time out
Data 0
Data 1
Data 2
Data 0
Data 1
Data 2
ACK 3
Lost
Data 0
Data 1
Data 2
Data 0
Data 1
Data 2

Selective Reject ARQ

- Only the specific damaged or lost frame is retransmitted
- A NAK is returned if a frame is corrupted
- The frame is resent out of sequence
- The receiving device must be able to sort the frames
- Insert the retransmitted frame into its proper place

Selective Reject- Damaged frame



Lost frames

- If a frame is lost the next frame will arrive out of sequence
- Receiver returns a NAK
- The receiver will recognize omission only if other frames follow
- If the lost frame was the last transmitted frame
- The receiver does nothing
- The sender treats the silence like a lost ack

Lost Acknowledgement

- Like Go Back n ARQ

Comparison

- Selective Reject ARQ
  - Expensive—required sorting, searching mechanism and the extra logic needed by the sender to select specific frames for retransmission
  - Not often used
  - Gives better performance
- Go Back n ARQ
  - Simple to implement