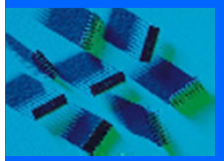


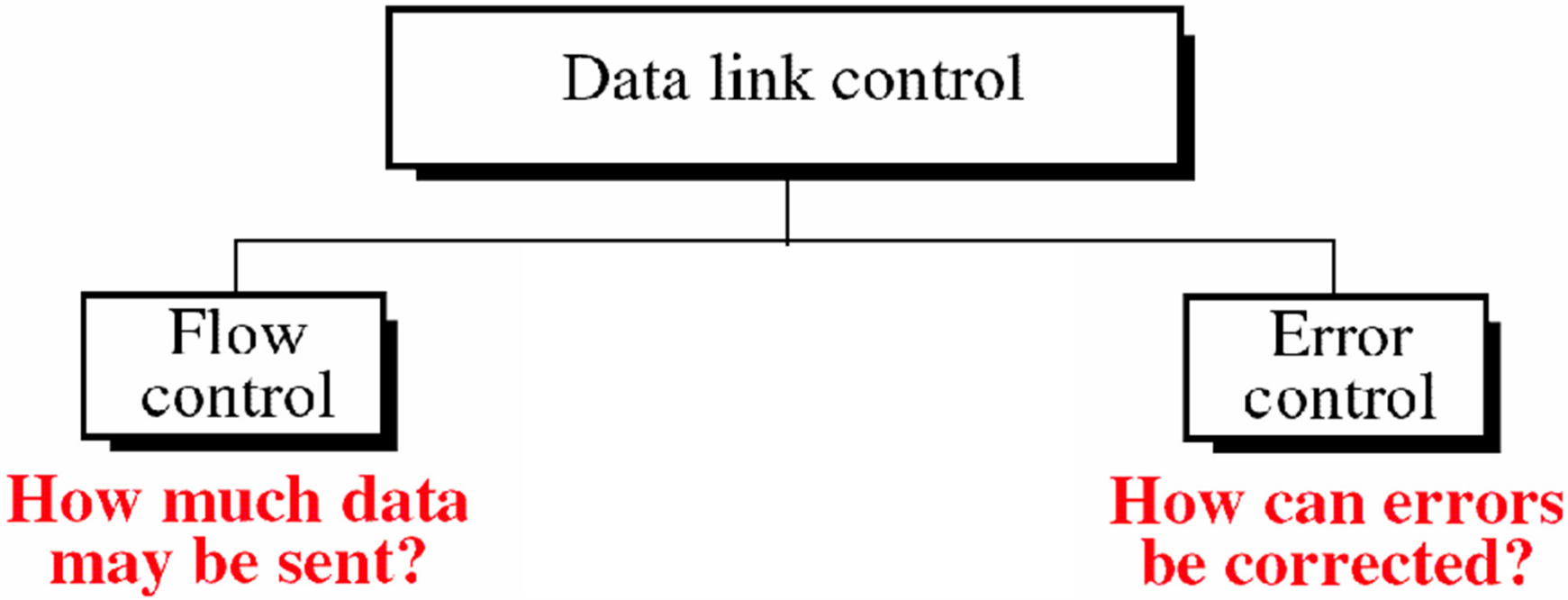
Lecture 4
Data Link Control & Protocols

Textbook: Ch.11

Main Topics

- 11.1 Flow And Error Control
- 11.2 Stop-and-Wait
  - Implicit Retransmission and ARQ
  - Error-control
  - Piggybacking
- 11.3 Framing with HDLC
  - High-level Data Link Control
- 11.1 Bit stuffing

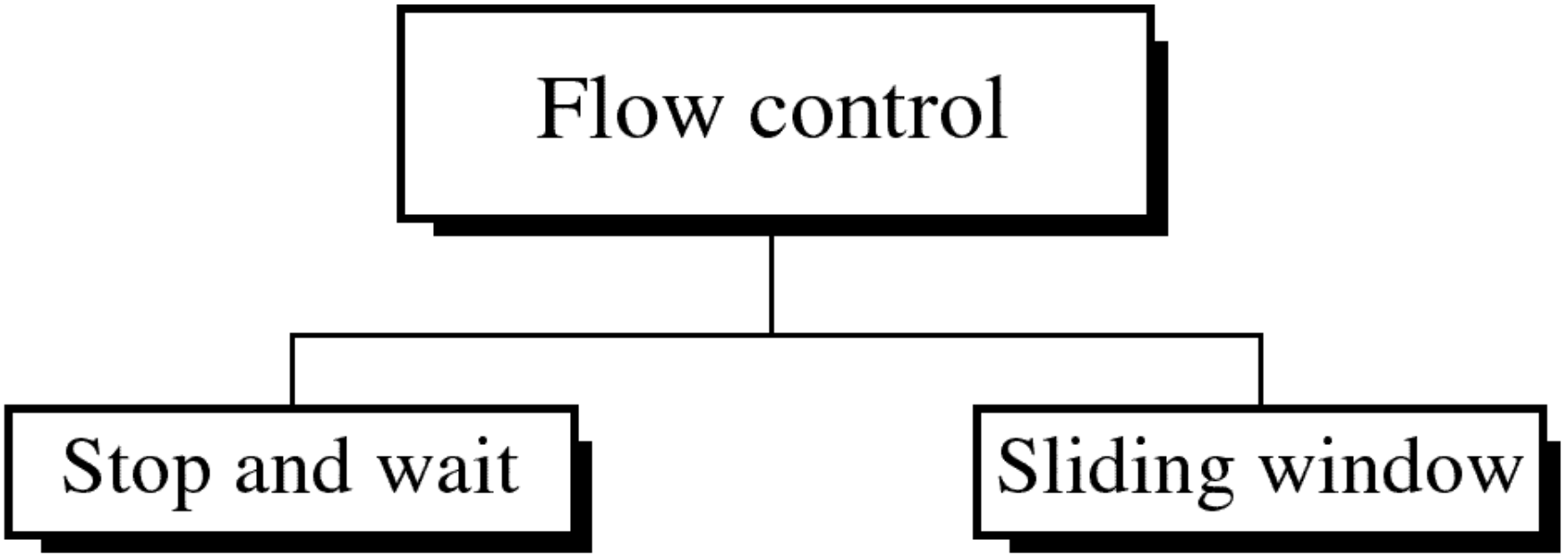
11.1 Data Link Control



Flow Control

- Balance between the sending rate and receiving rate
  - If sender transmits faster than the receiver can handle – data lost
  - If sender transmits too slow, receiver has to wait – less efficient
- Flow control is related to the first issue
  - Prevent data lost
  - Sender waits for acknowledgement (ACK) from receiver

Flow control Mechanisms



Send one frame at a time

- *Stop and Wait ARQ*

Send several frames at a time

- *Go-Back-N ARQ*
- *Selective Repeat ARQ*
- *(Details in Lecture 9)*

Error Control

- Error detection by CRC or FCS
- Error correction by retransmission
  - If error is detected, a negative acknowledgment (NAK) is returned and the specified frames are resent.
  - If no error, receiver sends acknowledgment (ACK) to sender, sender sends next frame
  - If no ACK is received after a period of time, sender retransmits

Error Control

- Automatic Repeat Request (ARQ)
  - does not use NAK

- Implicit retransmission in ARQ
  - Receiver discards the error frame and does nothing
  - Sender interprets the absence of an ACK (after a timeout) as an indication that the previous frame was corrupted or lost

Flow control and Error control

- Can be combined
- Acknowledgement is used in both control
  - Sender waits for ACK to transmit the next frame
  - Receiver uses ACK to confirm no error
  - Sender retransmits if no ACK is received

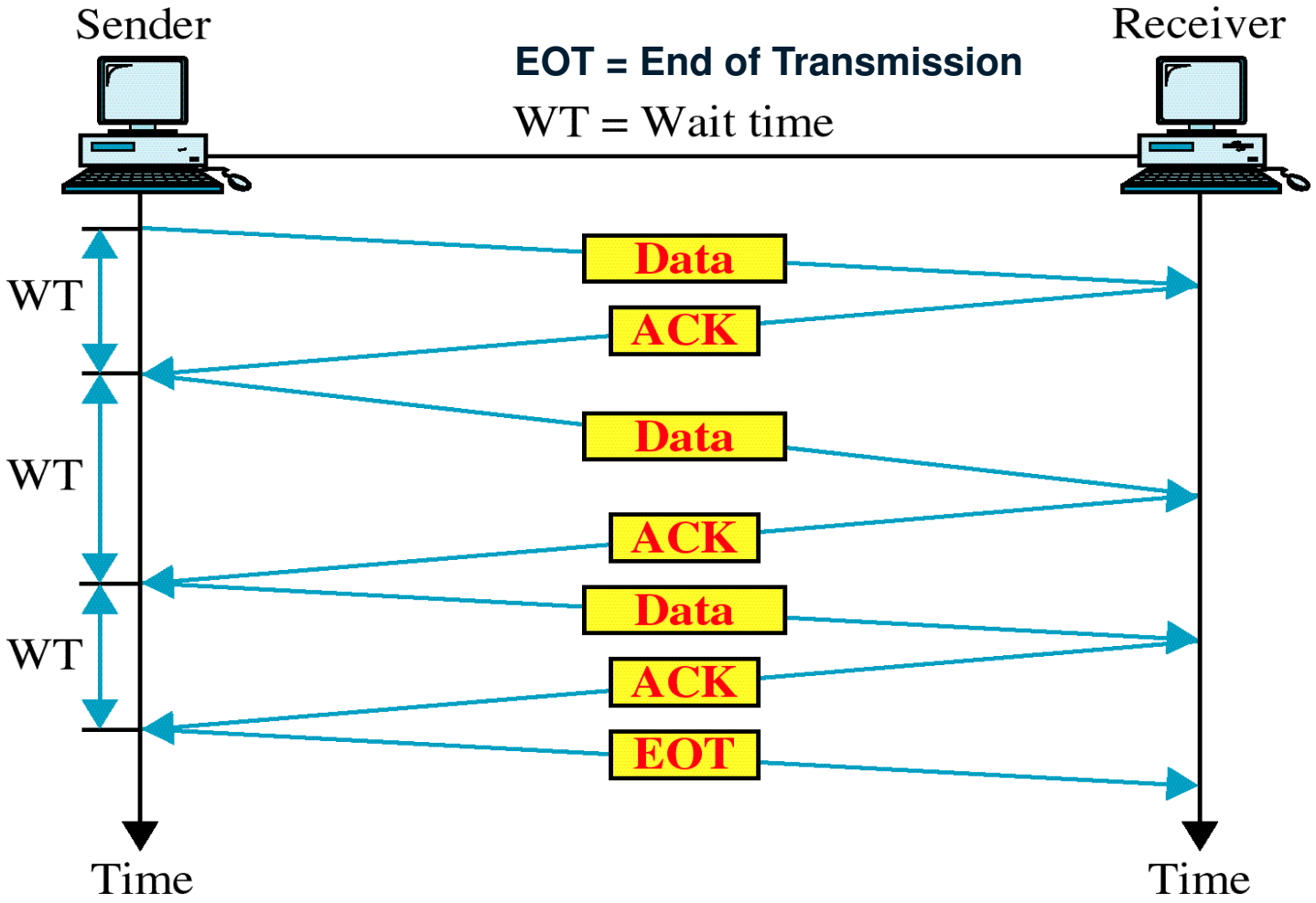
- Stop-and-Wait ARQ
  - The simplest protocol for flow and error control

11.2 Stop-and-Wait ARQ

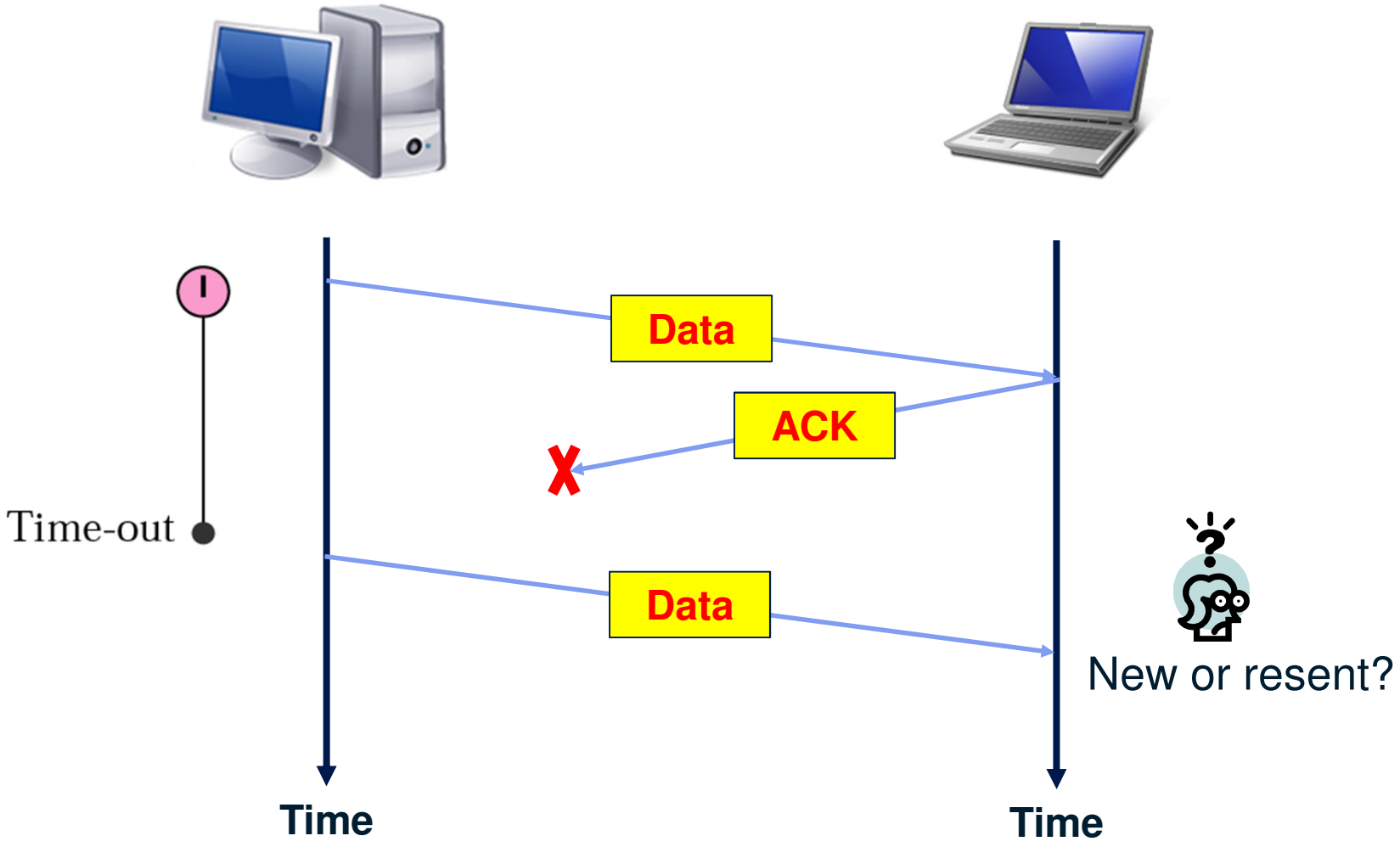
- The sender sends one frame and waits for an ACK before sending the next frame
- If no ACK is received after a period of time (timeout), the sender retransmits
  - Implicit retransmission in ARQ

- Advantage: Simple
- Disadvantage: Inefficient

Normal Situation



If ACK is lost…

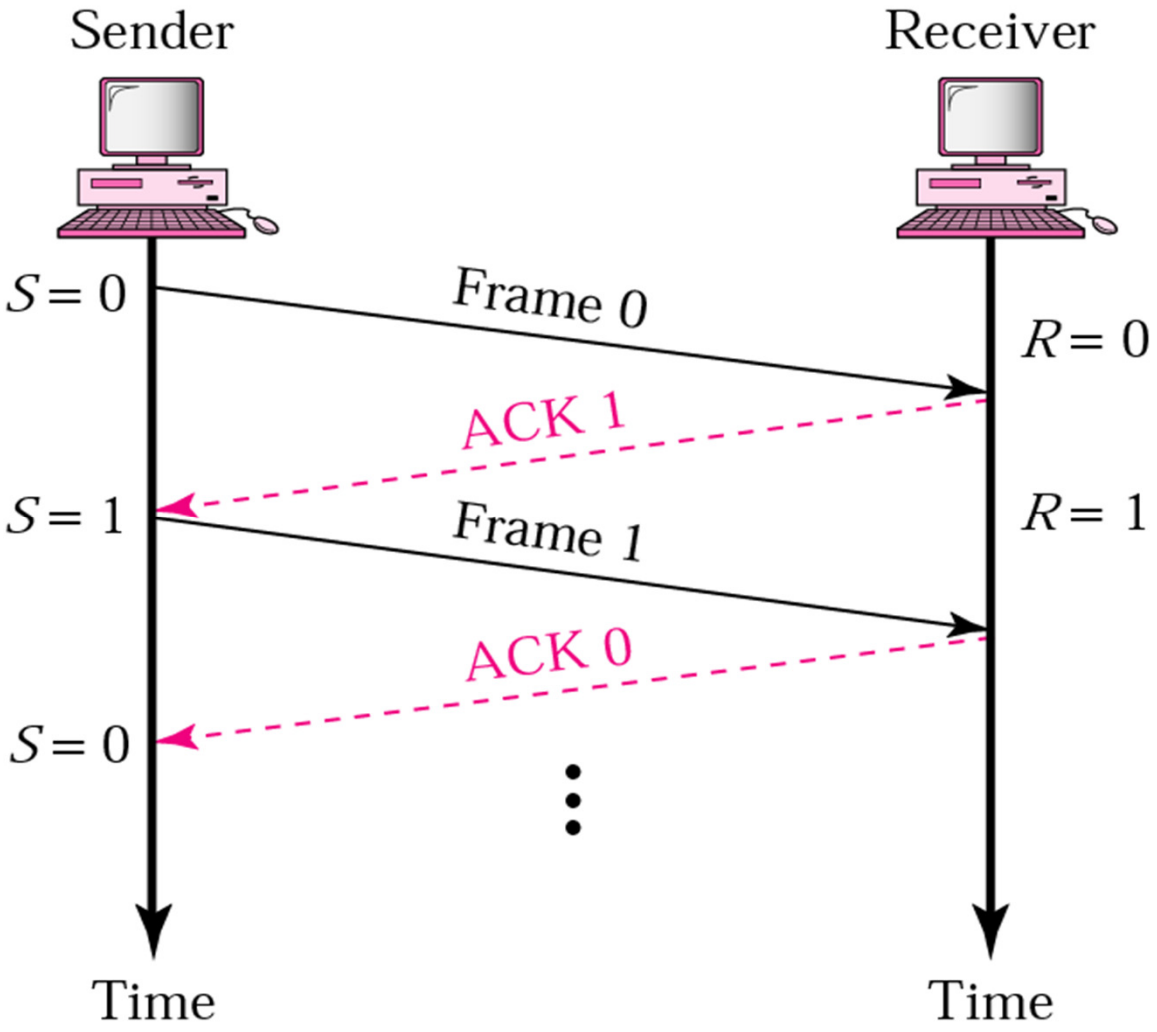


Sequence Number

- Use 1 bit sequence number to distinguish the frame is newly transmitted or resent of previous frame

- ACK confirms the correct receive of frame
- ACK also contains the sequence number of the expected frame
  - Sender knows what frame the receiver is expecting

Normal Operation



Normal Operation

- Sender can have only ***one frame ready*** to send at a time
- When sender initiates a transmission of a frame, it starts a ***timer***
- If the frame is received without error, the receiver sends ***ACK***
- If sender receives ACK, it sends ***another frame***

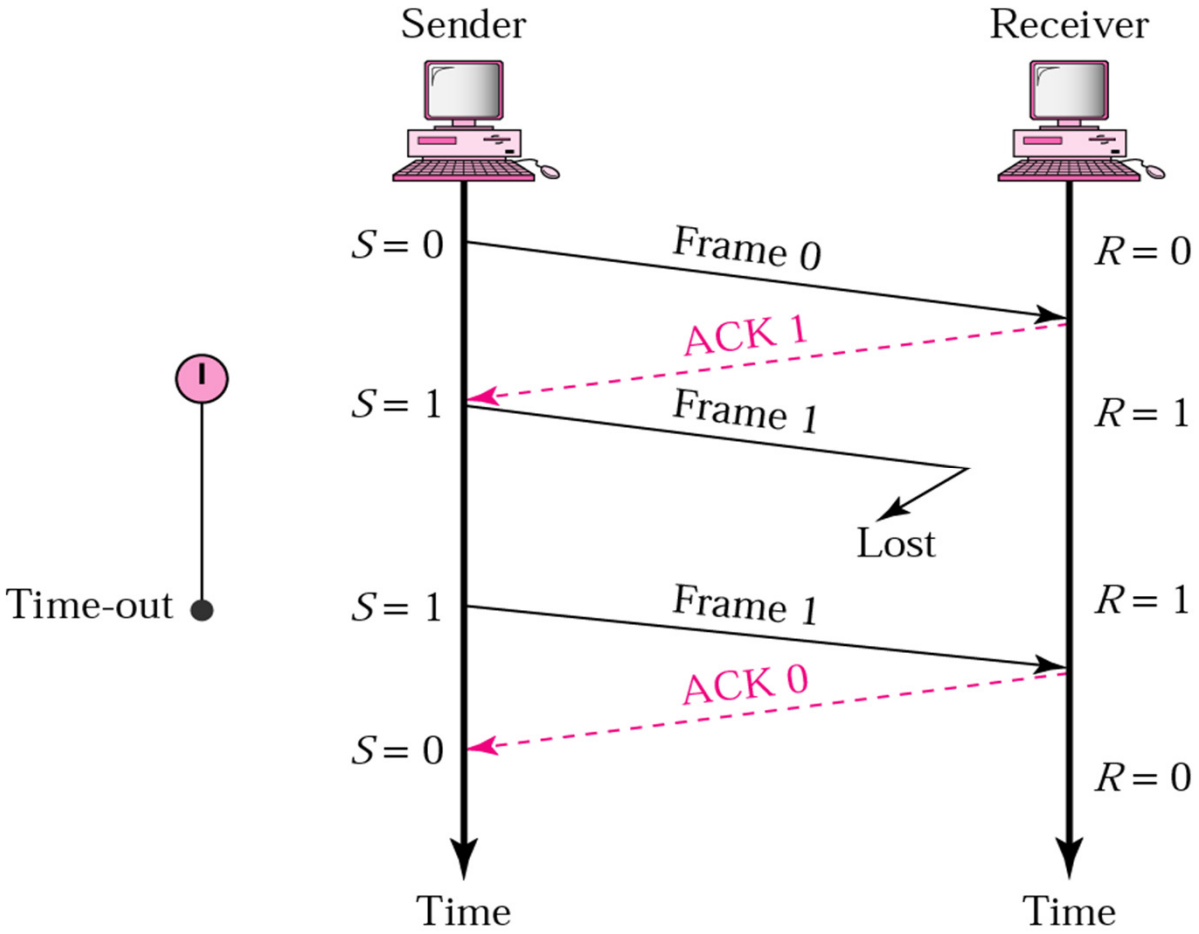
Implicit Retransmission

- Receiver discards the frame if it contains ***errors***
  - No ACK is sent
- If sender does not receive an ACK within a predefined ***time-out interval***, it retransmits the frame in the buffer
- Receiver checks the frame identifier (sequence number)
  - Accept if it is the expecting frame
  - Discard if the frame has been correctly received previously

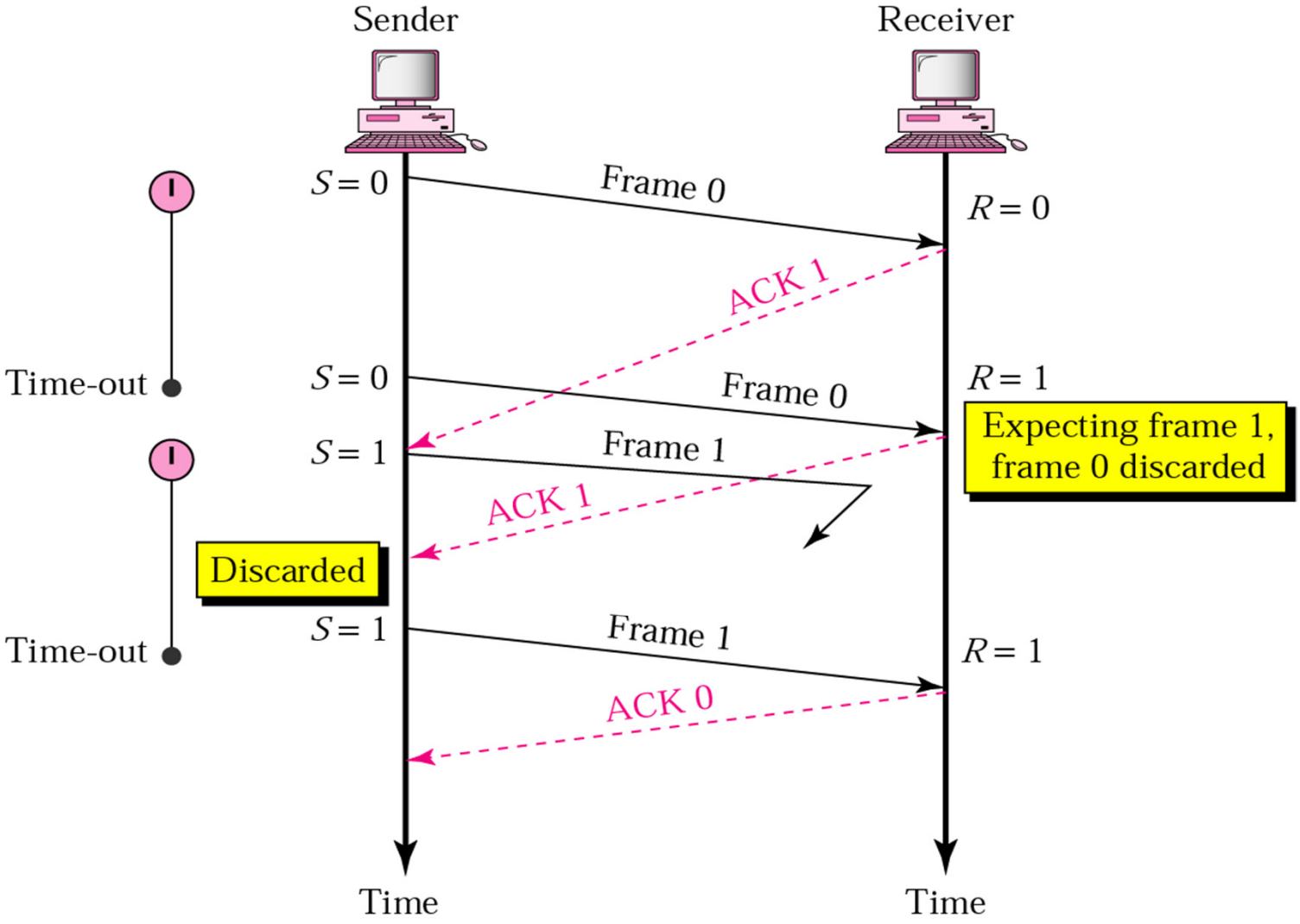
Buffer in Sender

- Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame
- Sender maintains a buffer with size = 1 frame
- Sender may not receive ACK because
  - Receiver detects error in the frame
  - The frame is lost before it reaches the receiver
  - The ACK is lost, delayed or corrupted
- Retransmitting the frame in the buffer when the timer expires

Stop-and-Wait ARQ, lost frame



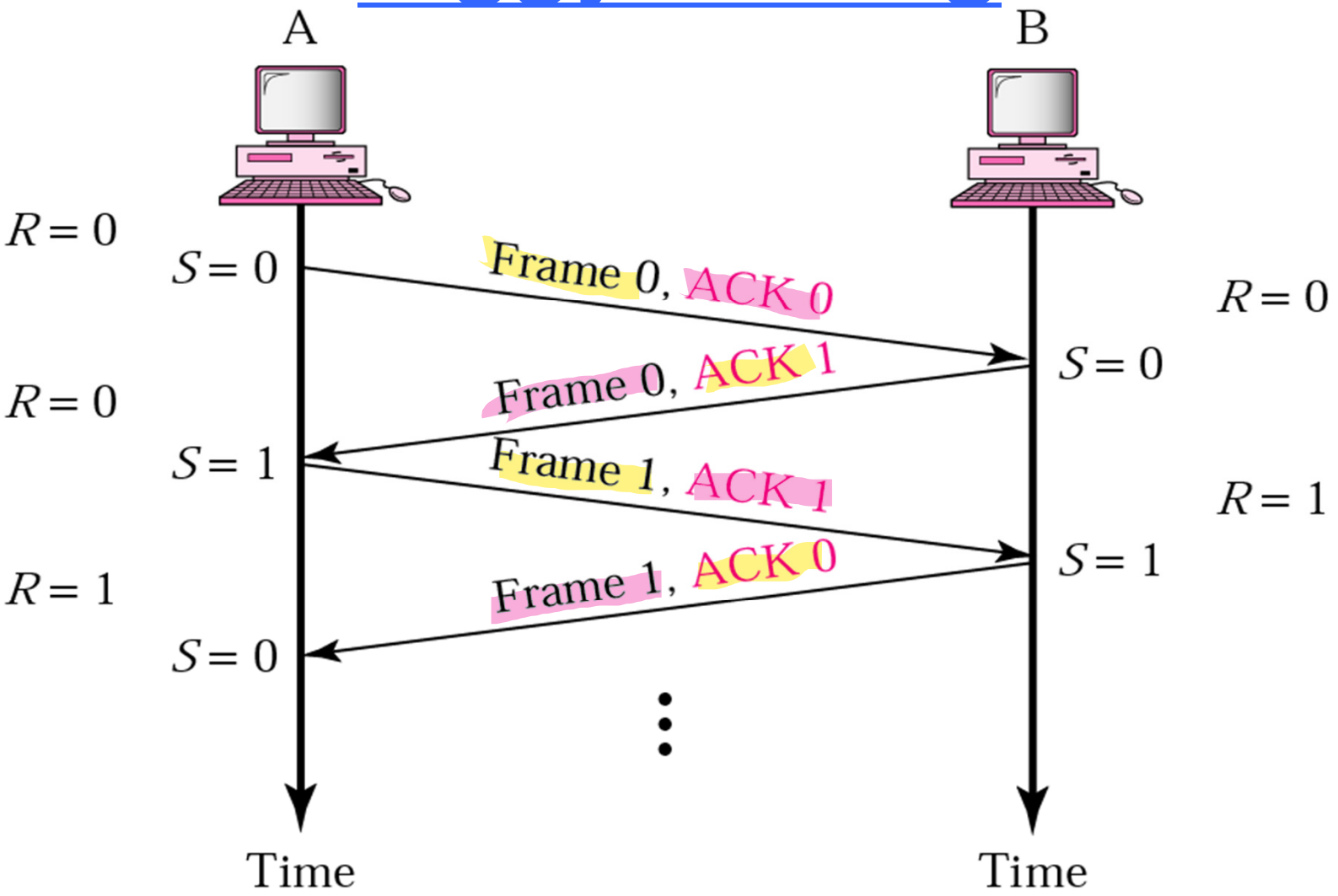
Stop-and-Wait ARQ, delayed ACK



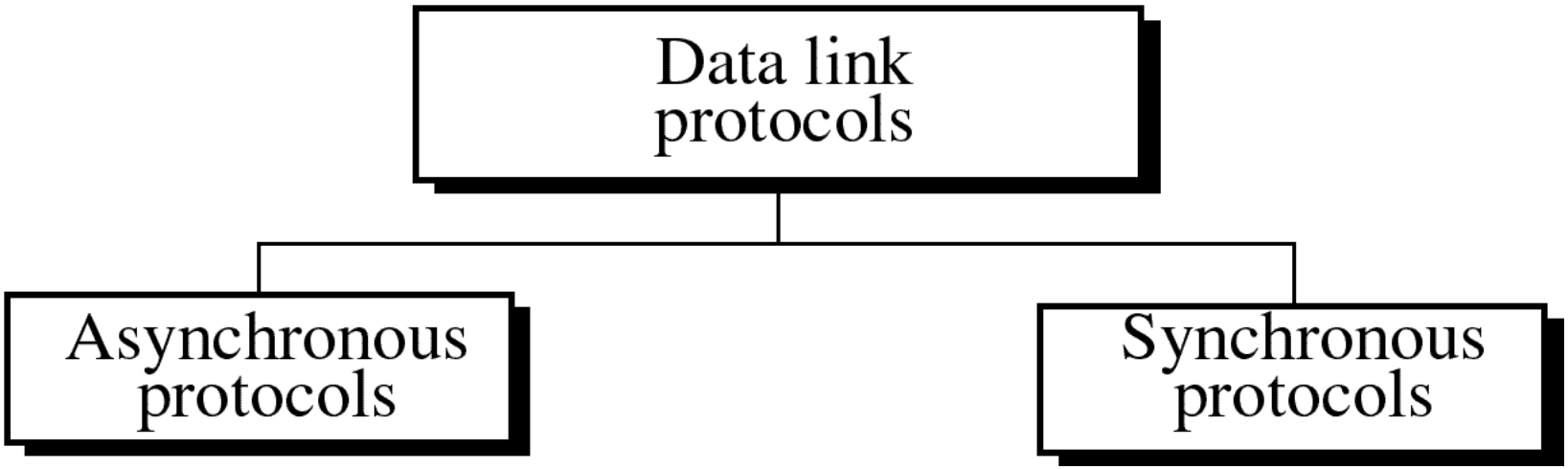
Piggybacking

- For bidirectional transmission
- The technique of temporarily delaying outgoing ACKs so that they can be hooked onto the next outgoing data frame
- A way of improving link utilization
- Normally most links using continuous ARQ are full-duplex and carry data frames in both directions
- Each side contains both a sender & a receiver

Piggybacking

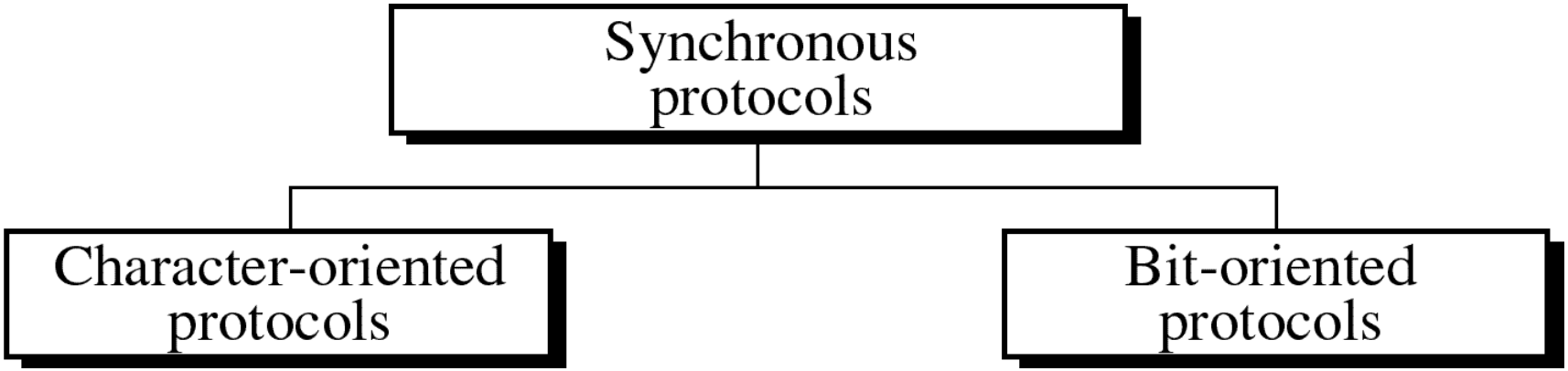


Data Link Protocols



- Asynchronous protocols, used primarily in modems, use start and stop bits and variable length gap between characters
- Due to slow data rate, they are being replaced by higher-speed synchronous protocols

Synchronous Protocols



- In character-oriented protocols, the frame is interpreted as a series of characters
  - 8-bit (e.g. ASCII), popular in old days with only text
- In bit-oriented protocols, each bit or groups of bits can have meaning

Bit-oriented Protocols

- Protocols use predefined bit patterns rather than transmission control characters to signal the start and end of a frame. (frame delimiting)
- The receiver searches the received bit stream on a bit by bit basis for the known start and end of frame bit pattern.
- E.g. HDLC

11.3 High-level Data Link Control (HDLC)

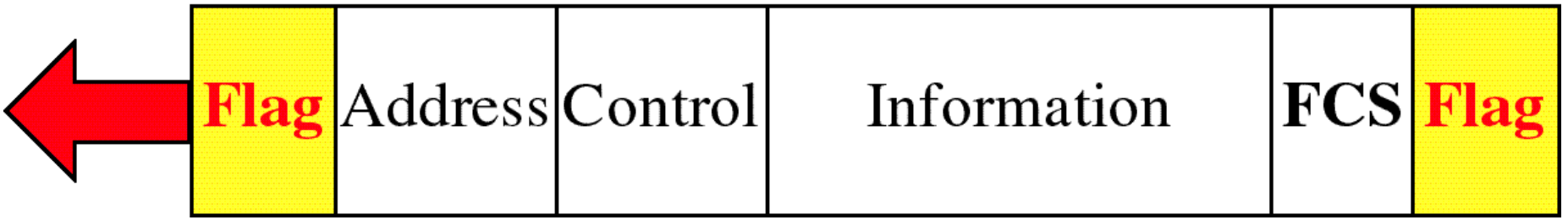
- an ISO international standard used on both point to point and multipoint (multidrop) data links
- supports both half-duplex and full-duplex with error detection
- adopts continuous ARQ with window mechanism
- used extensively in computer networks
- But many large manufacturers still use their own protocols similar to HDLC, e.g.,
  - IBM's SDLC (synchronous data link control)

HDLC Frame Formats

- Both data & control messages are carried in a standard format block

The flag is 8 bits of a fixed pattern.

01111110



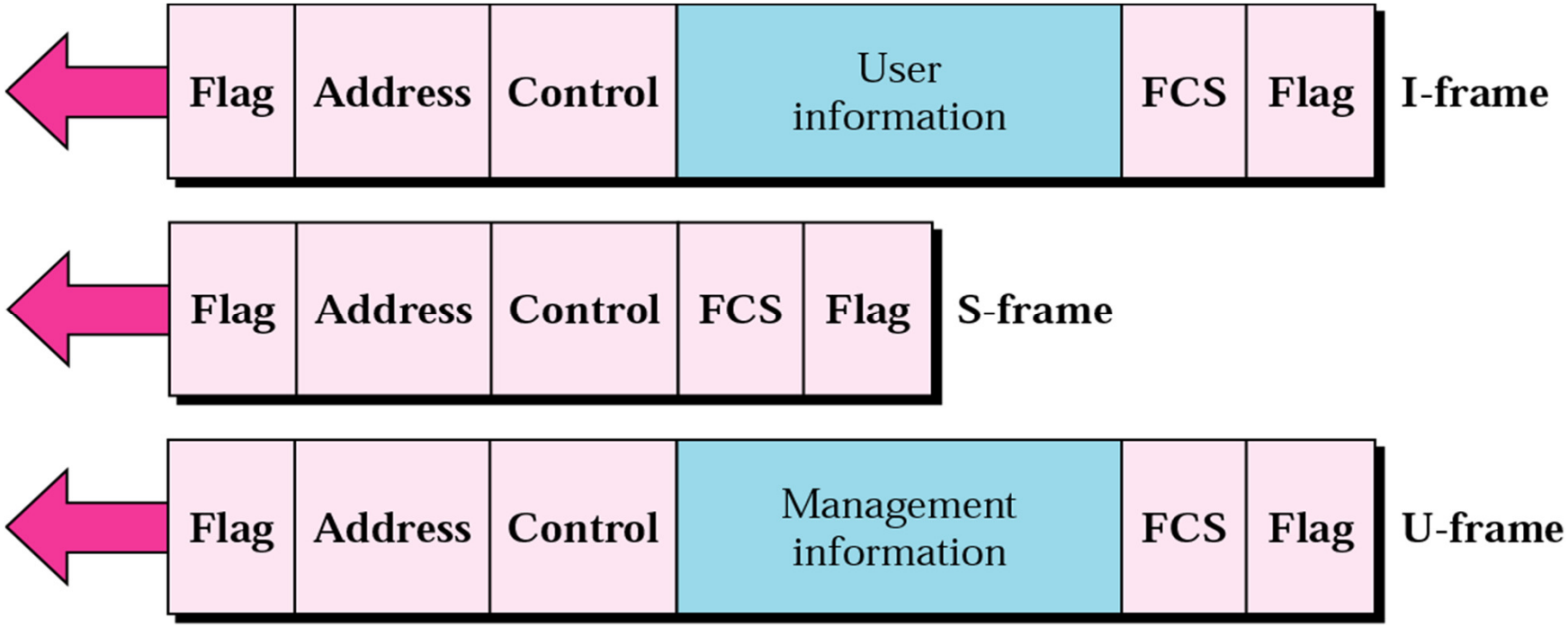
HDLC Frame Formats

- **Flag field**
  - (01111110) indicates start & end of a frame
- **Address Field**
  - Address of the station receiving the frame
- **Control Field**
  - For flow and error control (more details later)
- **Information Field**
  - Contains user's data from upper layer
- **Frame Check Sequence (FCS) Field**
  - For error checking similar to CRC

HDLC Frame Types

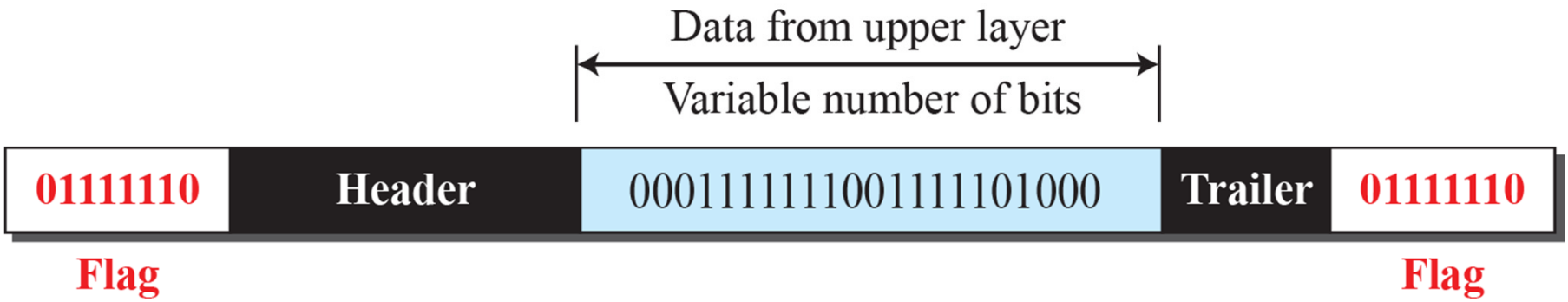
- **Information frames (I-frames)**
  - carry actual data
  - also act as piggyback ACK
- **Supervisory frames (S-frames)**
  - for transporting control information
- **Unnumbered frames (U-frames)**
  - for link set-up and disconnection

HDLC frame types



Data transparency

- Data can be any combination of bits
- Confusion between control information and data is called a **lack of data transparency**
  - E.g. data field contains 01111110



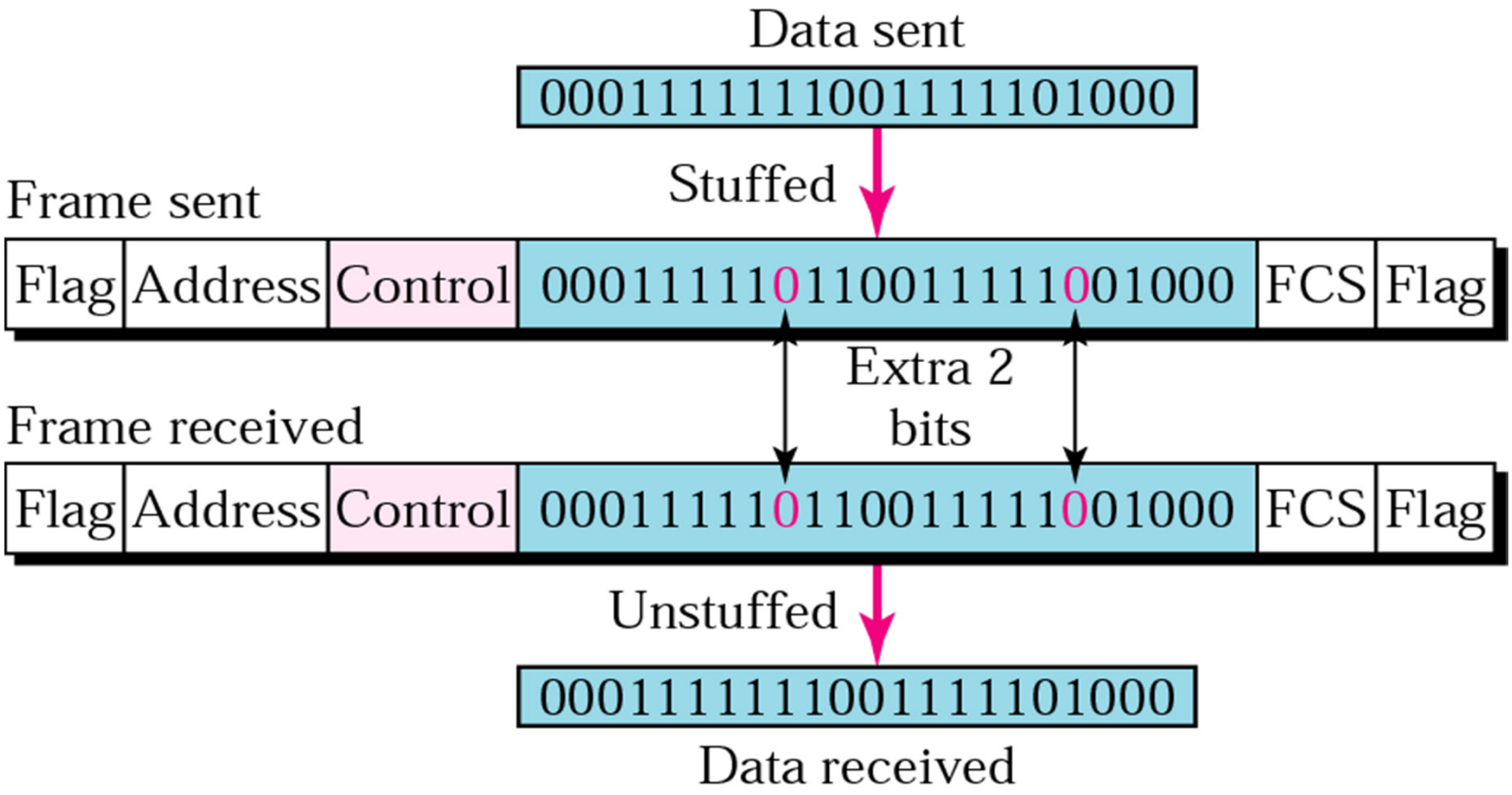
Data transparency

- when data are transparent, which means we should be able to send any combination of bits as data
- Bit Stuffing method is used in HDLC for achieving **data transparency**

Bit Stuffing

- A method used in HDLC for achieving **data transparency**
  - Ensure that the flag pattern is not present in the frame contents
- Sender inserts a “0” bit after transmitting five consecutive “1” bits
- *Exceptions:* when the bit sequence is really a flag
- Receiver removes the “0” bit after receiving five consecutive “1” bits

Bit stuffing and removal



Summary

- Flow Control and Error Control
- Stop-and-Wait ARQ
- High-level Data Link Control (HDLC)
- Bit stuffing

- Revision Quiz
  - http://highered.mheducation.com/sites/0073376221/student_view0/chapter11/quizzes.html