Data Link Protocols
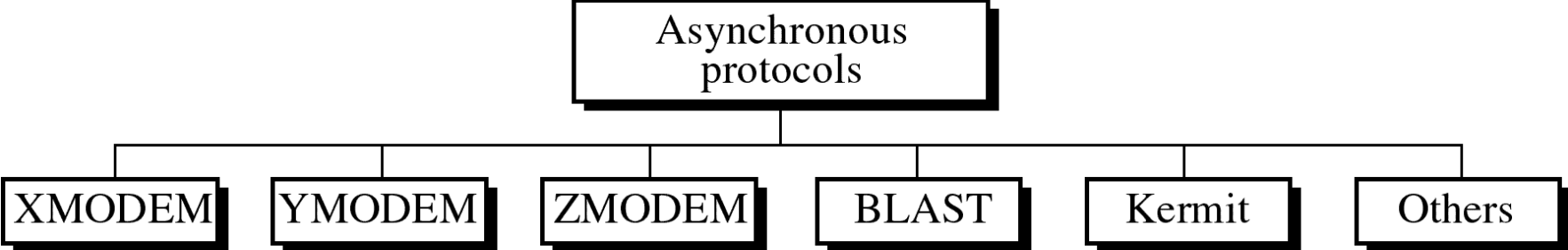
- **Asynchronous Protocols**
- **Synchronous Protocols**
- **Character-Oriented Protocols**
- **Bit-Oriented Protocols**

```
                    ┌─────────────────────┐
                    │     Data link       │
                    │     protocols       │
                    └──────────┬──────────┘
               ┌───────────────┴───────────────┐
    ┌──────────┴──────────┐         ┌──────────┴──────────┐
    │    Asynchronous     │         │     Synchronous     │
    │     protocols       │         │      protocols      │
    └─────────────────────┘         └─────────────────────┘
```

```
                        ┌─────────────────┐
                        │   Asynchronous  │
                        │    protocols    │
                        └─────────────────┘
                                 │
    ┌──────────┬──────────┬──────┴──────┬──────────┬──────────┐
┌────────┐ ┌────────┐ ┌────────┐ ┌────────┐ ┌────────┐ ┌────────┐
│ XMODEM │ │ YMODEM │ │ ZMODEM │ │ BLAST  │ │ Kermit │ │ Others │
└────────┘ └────────┘ └────────┘ └────────┘ └────────┘ └────────┘
```

XMODEM

Each character contains start and stop bits
(dark portion of the box). Characters are
separated from each other by gaps.
The header consists of two bytes: sequence
number and its one's complement.

SOH

CRC

Header

Data: 128 bytes

```
                    ┌─────────────────────┐
                    │     Synchronous     │
                    │      protocols      │
                    └──────────┬──────────┘
              ┌────────────────┴────────────────┐
    ┌─────────────────────┐          ┌─────────────────────┐
    │  Character-oriented  │          │     Bit-oriented     │
    │      protocols       │          │      protocols       │
    └─────────────────────┘          └─────────────────────┘
```

```
                    ┌─────────────────┐
                    │   BSC frames    │
                    └─────────────────┘
                 ┌───────────┴───────────┐
        ┌─────────────────┐      ┌─────────────────┐
        │ Control frames  │      │   Data frames   │
        └─────────────────┘      └─────────────────┘
```

Connection, flow and error control, and disconnection

Transmission of data

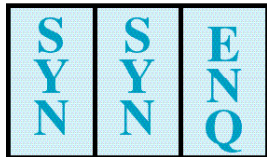# Simple Frame
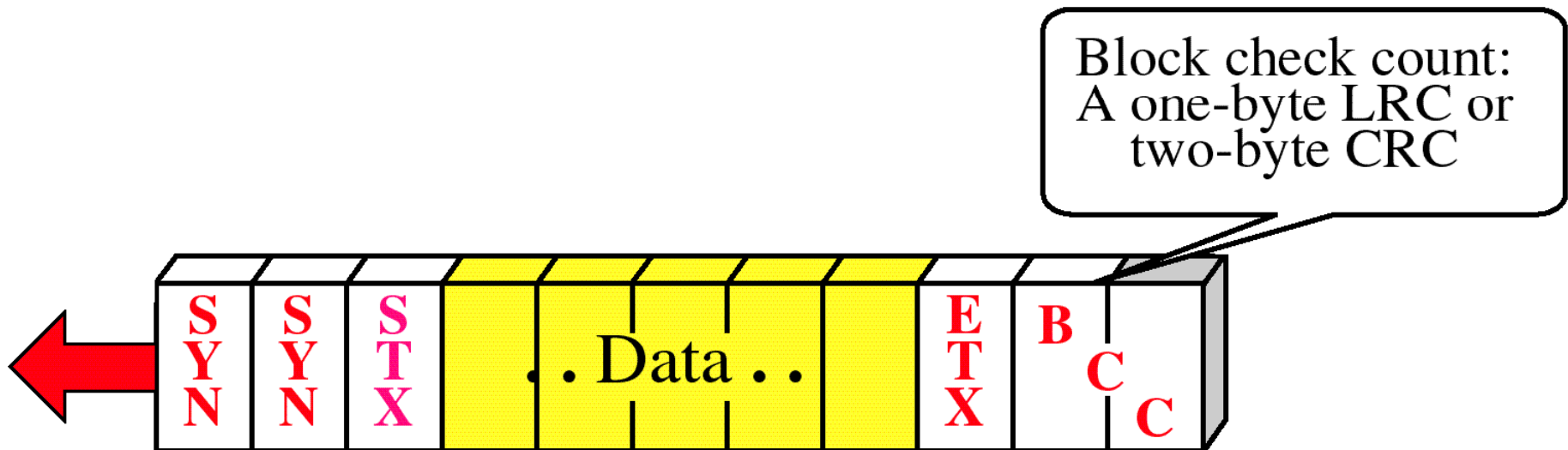
A Frame with Header
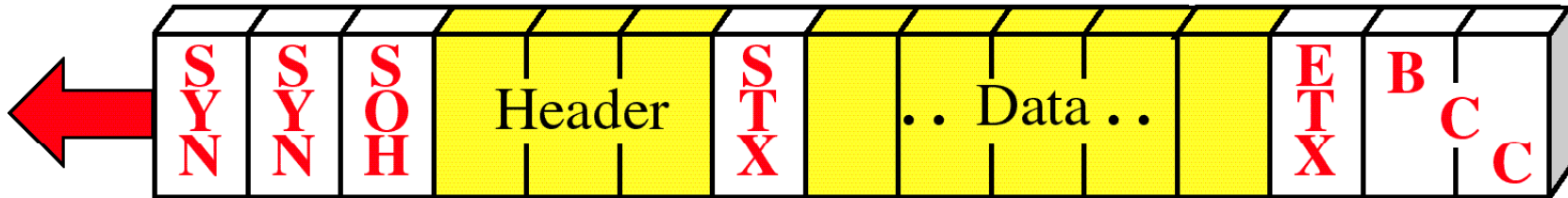
Multiblock Frame
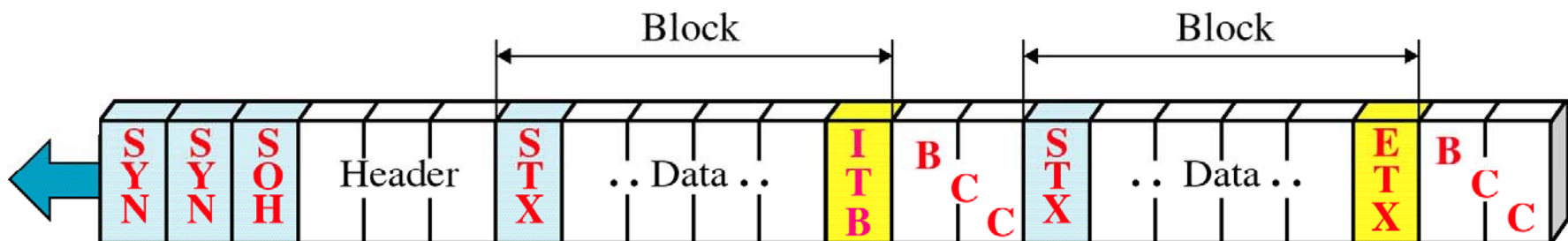
Multiframe Transmission

Header ... Data ... ... Data ...

ACK 1

ETB for all frames except the last one

Header ... Data ... ... Data ...

ACK 0

ETX for the last frame

Header ... Data ... ... Data ...

ACK 1
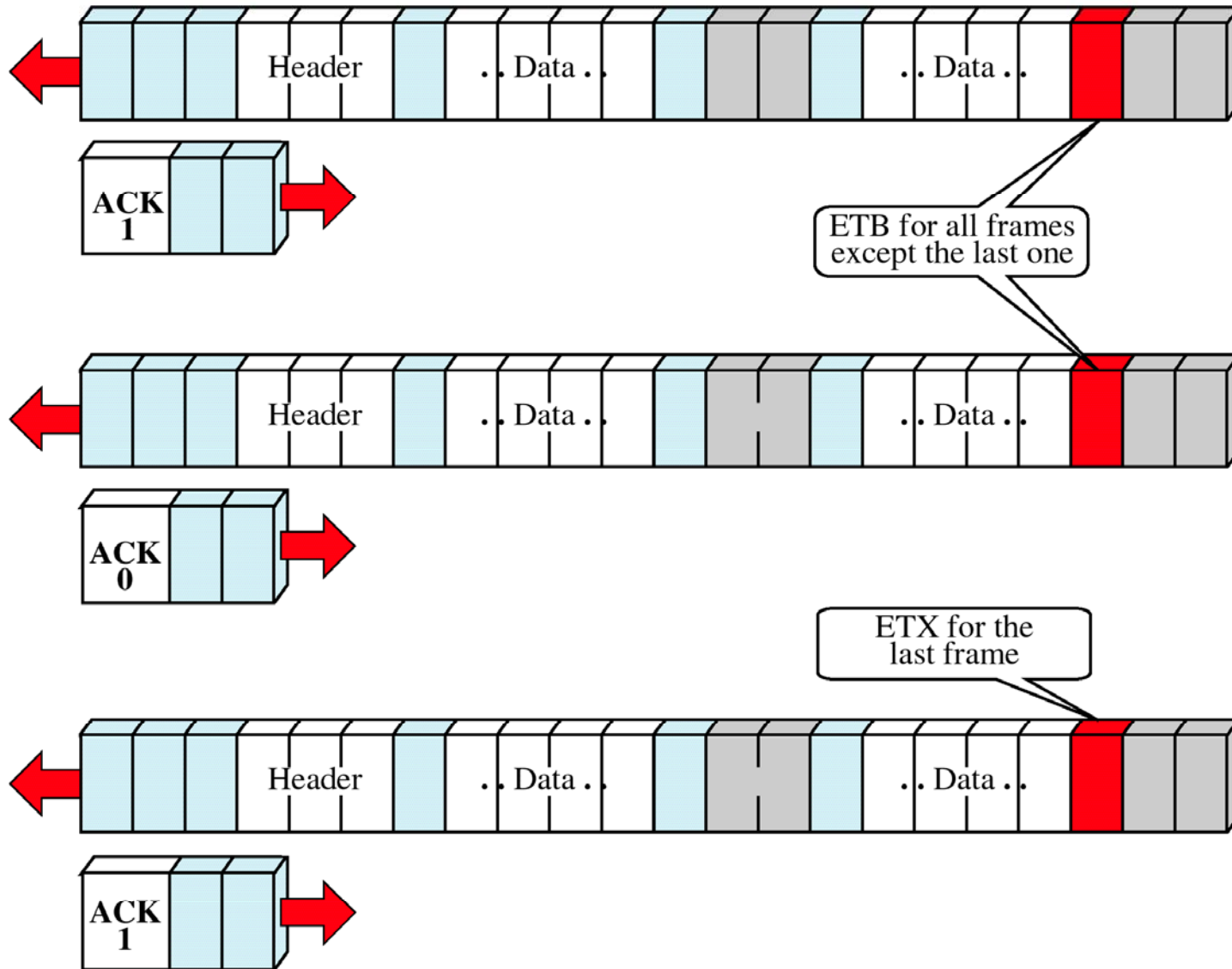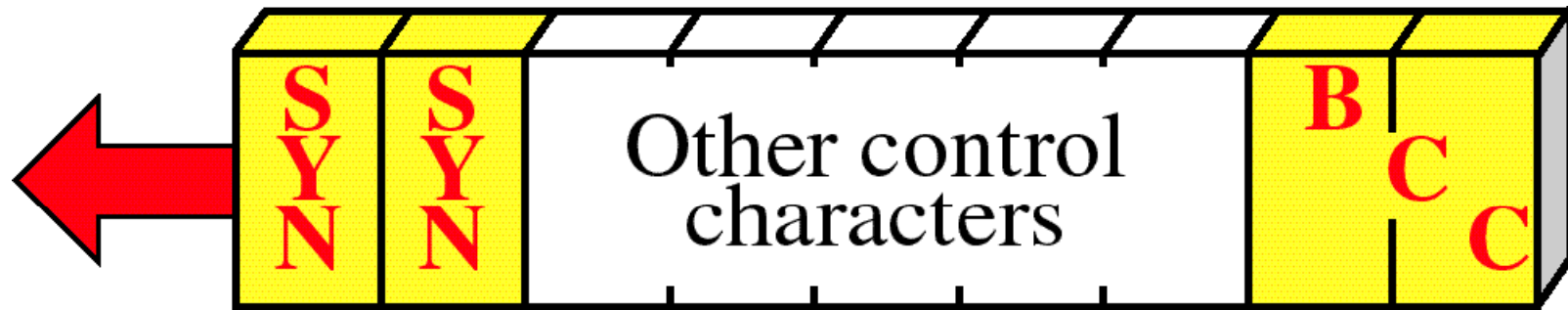
Control Frame

Control Frames

Connection establishment

| S Y N | S Y N | E N Q | | **Bid** Point-to-point connection request. |

| S Y N | S Y N | E N Q | Polling address | **Poll** Primary polls secondary. |

| S Y N | S Y N | E N Q | Selecting address | **Select** Primary selects secondary. |

| S Y N | S Y N | ACK0 | **Positive response to select or bid** Ready to receive data. |

| S Y N | S Y N | N A K | **Negative response to select or bid** Not ready to receive data. |

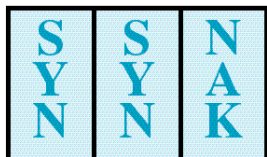| S Y N | S Y N | E O T | **Negative response to poll** Not ready to send data. |
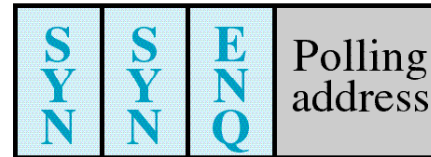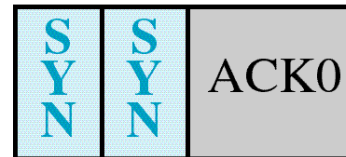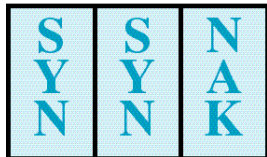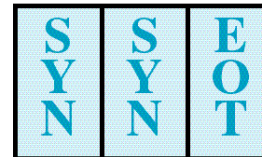
Control Frames

Flow and error control

| SYN SYN ACK0 | **Positive ACK of even frames** Frame number 0 received. | SYN SYN ACK1 | **Positive ACK of odd frames** Frame number 1 received. |

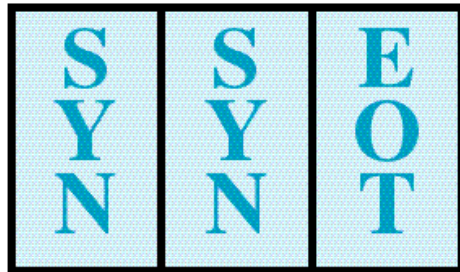| SYN SYN NAK | **Negative ACK of frames** Error in the frame received. | SYN SYN WACK | **Wait & ACK** ACK of previous frame, not ready to receive more. |

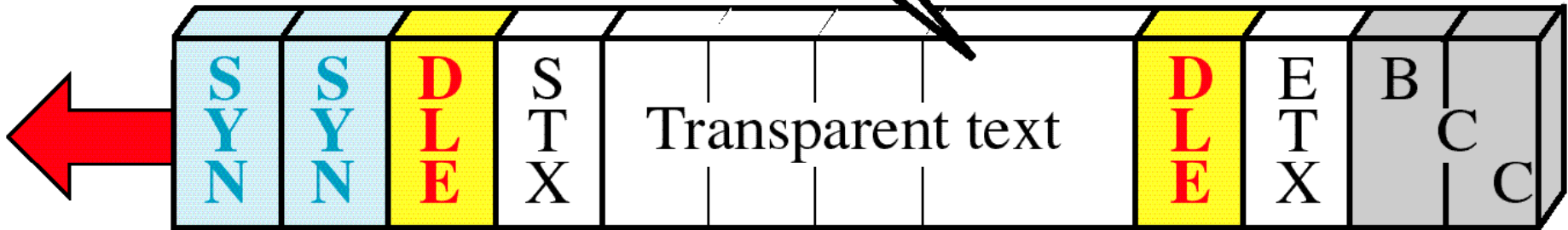| SYN SYN RV1 | **Reverse interrupt** Request for interruption, urgent data to send. | SYN SYN TTD | **Temporary delay** Temporarily delayed but does not relinquish the line. |
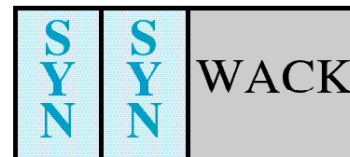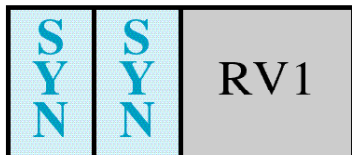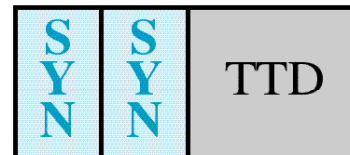
Control Frames

Connection termination

S Y N	S Y N	E O T

End of transmission
Station finished sending data.

Byte Stuffing