# Unit 4

# The Data Link Layer

# Role of the Data Link Layer

To provide services to the network layer (3) by enhancing the services provided by the physical layer (1).

**Several types of services:**

- **reliable service on a link (no error, no loss, no reordering)**
  - error detection and recovery at the data link layer
  - flow control to protect buffers at receiver
- **unreliable service on a link**
  - error detection done at the data link layer
  - recovery done at higher layers

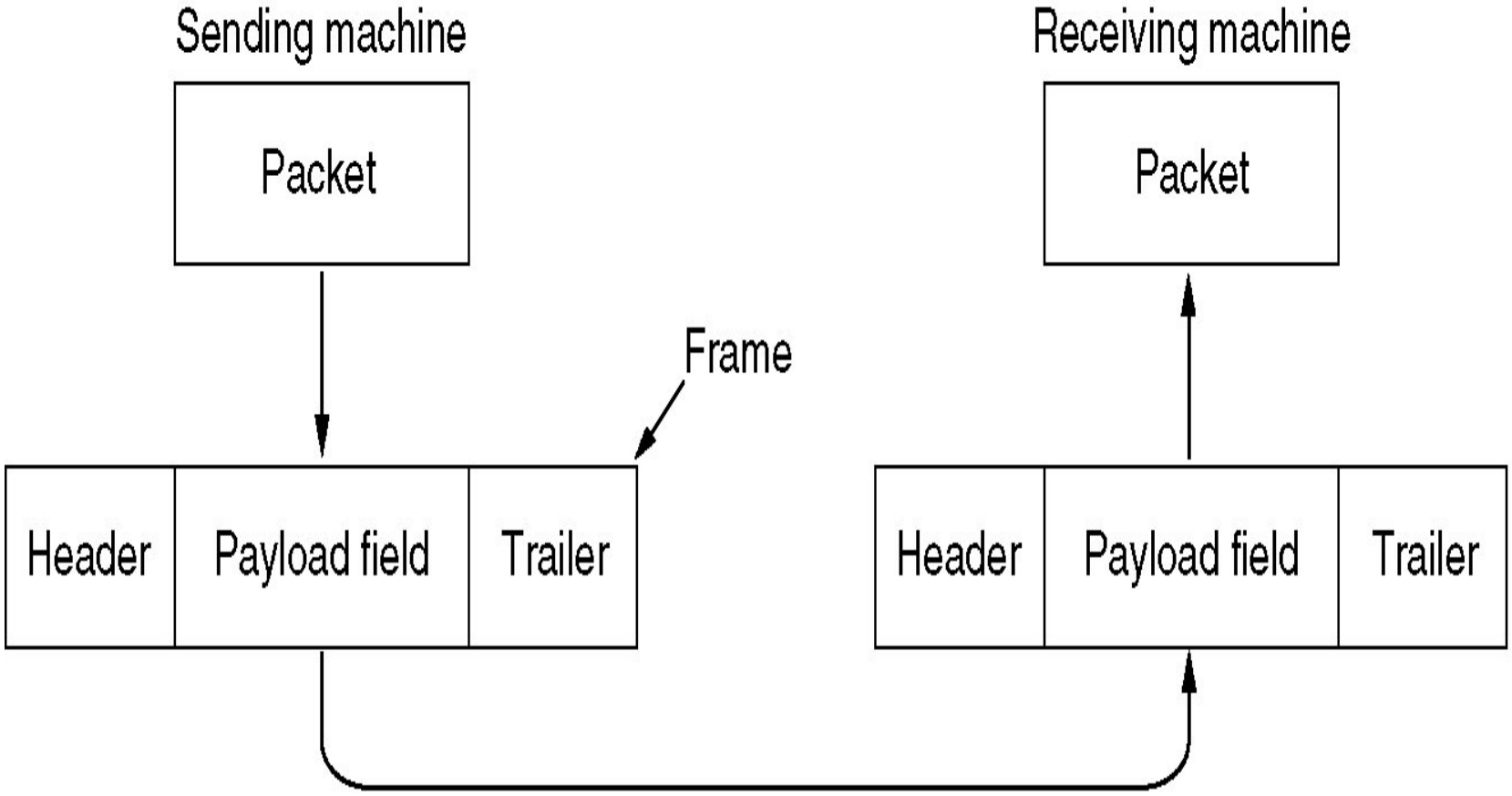
# Data Link Layer Design Issues

- **Services Provided to the Network Layer**
- **Framing**
- **Error Control**
- **Flow Control**

# Functions of the Data Link Layer

- Provide service interface to the network layer
- Dealing with transmission errors
- Regulating data flow
  - Slow receivers not busy by fast senders

# Functions of the Data Link Layer (2)




Relationship between packets and frames.

# Services Provided to the Network Layer

1. Unacknowledged Connectionless Service
2. Acknowledged Connectionless Service
3. Acknowledged Connection-oriented Service

   (Three phases)

   · Connection establishment

   · data transfer

   · disconnection

# 1. Services to Network layer

- **Unacknowledged connectionless service**
  - No connection required and without acknowledgement for data frames.
  - Appropriate for low error rate and real-time traffic.
  - Error recovery is up to higher layer.

# 2. Services to Network layer

- **Acknowledged connectionless service**
  - No connection required but each frame is individually acknowledged.
  - Useful for unreliable channel, such as wireless systems.
  - Transport layer may do ***message recovery*** but is more expensive than ***frame recovery*** at data link layer.

# 3. Services to Network layer

- **Acknowledged connection-oriented service**
  - Guarantee error-free and in sequence delivery of data frames.
  - Consists of three phases
    - **Connection set up** (variables and buffers initialization).
    - **Data frame transmission.**
    - **Connection termination** (free of variable and buffers).

# Framing

Break bit stream from physical layer into frames
for error detection and recovery.

Four framing methods

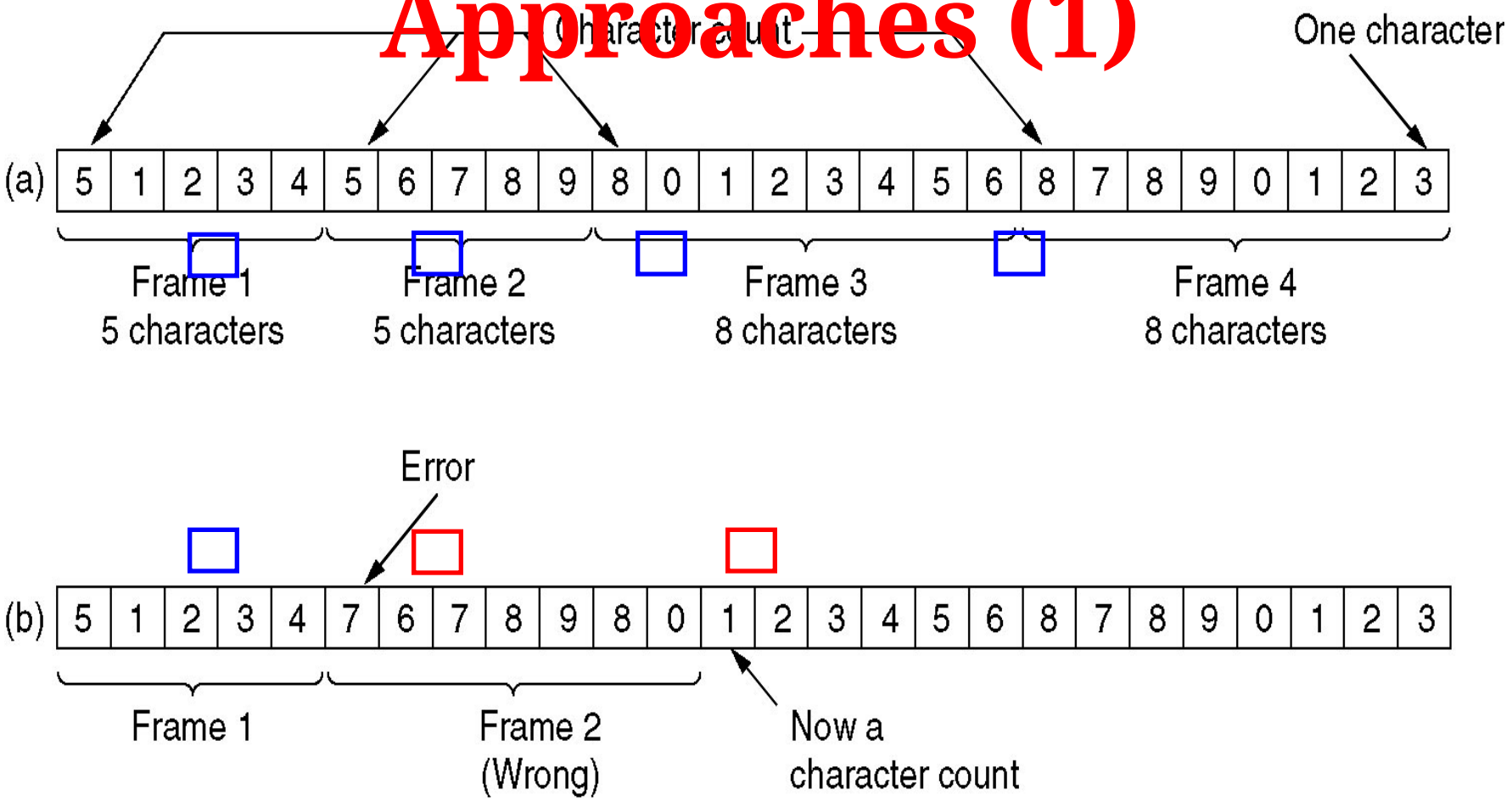
1. Character count.
2. Flag bytes with byte (character) stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

# Framing Approaches (1)

## Character Count

- Indicate the frame boundary by frame length
- Once an error, frame boundary cannot be recognized and thus recovery is impossible.

# Framing Approaches (1)



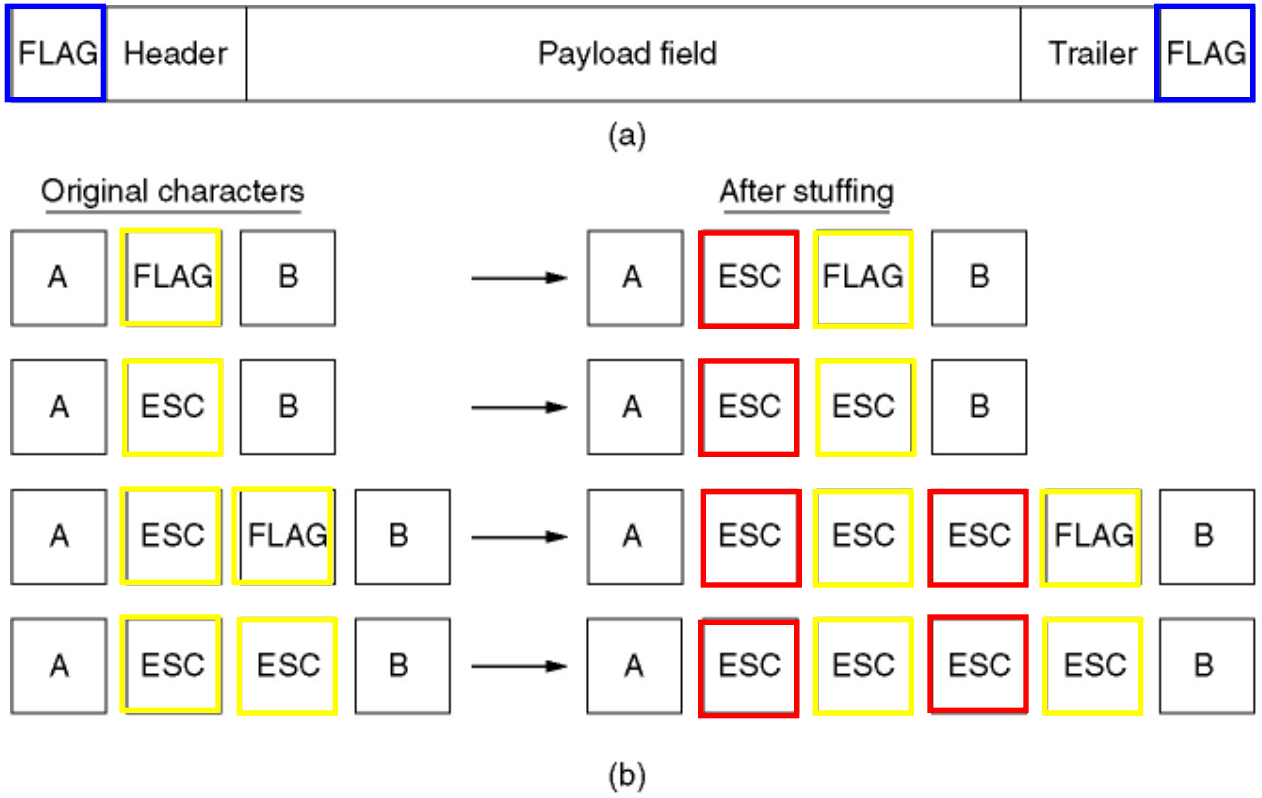

A character stream. (a) Without errors. (b) With one error.

# Framing Approaches (2)

**Flag bytes with byte stuffing (character stuffing)**

- Delimit the frame by flag bytes
- Prevent frame boundary from appearing at the data content by character stuffing.
- Character Stuffing: inserting ESC ahead of accidental flag byte within the data content.

# Framing Approaches (2)




(a) A frame delimited by flag bytes.

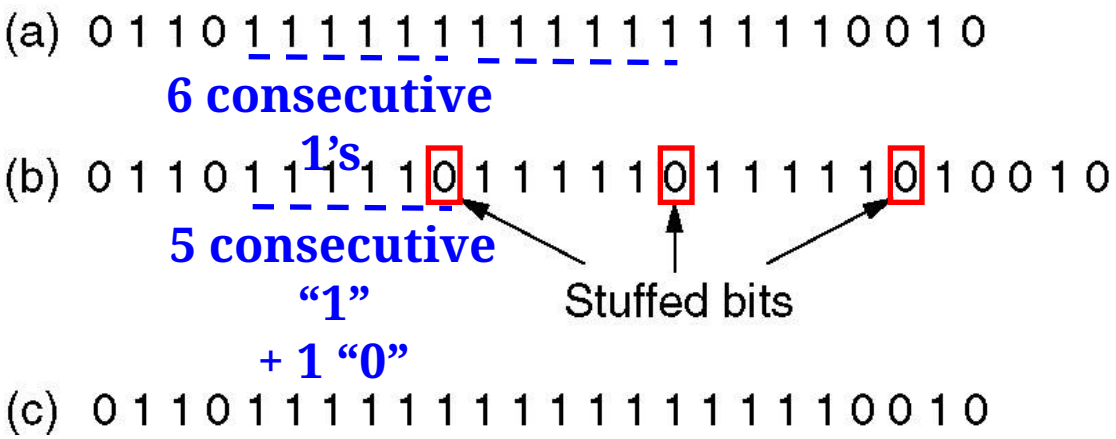
(b) Four examples of byte sequences before and after stuffing.

# Framing Approaches (3)

**Starting and ending flags, with bit stuffing**

- Begin and end with a flag byte "01111110"
- Prevent a flag from appearing in data by bit stuffing.
- Bit stuffing: inserting 0 after five continuous bit "1" data appear.

# Framing Approaches (3)




Bit stuffing

(a) The original data.

(b) The data as they appear on the line.

(c) The data as they are stored in receiver's memory after destuffing.

# Framing Approaches (4)

**Physical Layer Coding Violations.**

- Used when physical layer encoding contains redundancy
- Example: 01 (H), 10 (L) then 00 or 11 can be used to delimit a frame.

Note: One or more combination of the approaches may be used to provide extra protection for framing.

# Error Control

- Need mechanisms such as
  - Acknowledgement or NACK
  - Timer
  - Retransmission
  - Sequence numbering

**Ack ⇒ Timer ⇒ Retransmission ⇒ Sequence numbering**

# Flow Control

- When receiver processes frames slower than the sender, congestion occurs.
- Needs some feedback to prevent sender from sending faster than the receiver can process.

# Error Detection and Correction

- Error-Correcting Codes
- Error-Detecting Codes

# Error Correcting Code

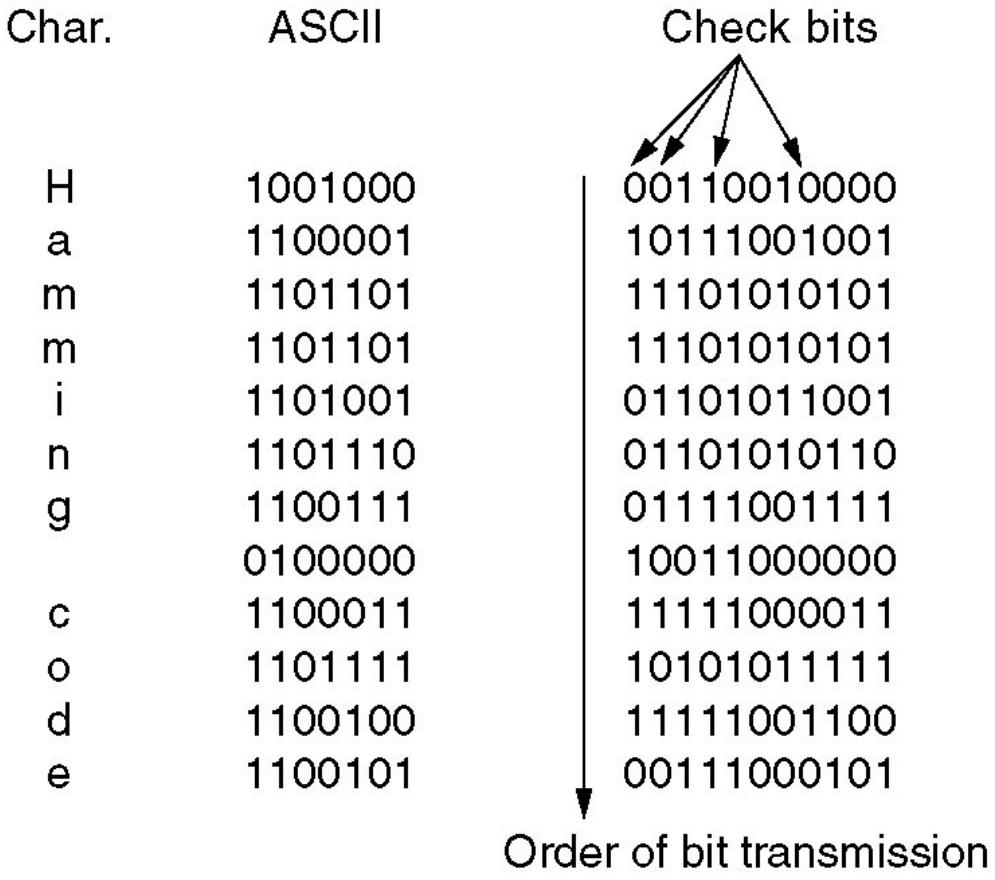
- Codeword = Data + Check-bit
- *n*-bit codeword = *m* data bit + *r* check-bit
- $2^m$ out of $2^n$ are legal
- Hamming distance = The minimum number of bit positions in which two codewords differ.
- H = d+1 --> detect d errors;
- H = 2d+1 --> correct d errors.

# Error-Correcting Codes

- Parity bit:
  - detect single bit error and H = 2.
- Example of H = 5 $\Rightarrow$ d = (5-1)/2 = 2
  - 0000000000
  - 0000011111
  - 1111100000
  - 1111111111
  - 0000000111 ==> two bit errors in 0000011111
  - 0000000011 ==> three bit errors in 0000011111 but be corrected to be 0000000000

Out of Correcting Capability

# Error-Correcting Codes




Use of a Hamming code to correct burst errors.

# Error Detecting Code

- Parity code
  - detect single or odd # of bit errors
  - detect burst error of n-bit by matrix checksum on each column of n-bit wide and h-bit high data and put the checksum at the h+1 row.
- Cyclic redundancy code
  - Polynomial code
  - Using Exclusive OR in addition and subtraction.

# Data Link Protocols

- **ARQ: Automatic Repeat reQuest**
  - Timer
  - Acknowledgement
  - Sequence number
  - Retransmission
- ARQ must handle
  - Garbled frames
  - Lost frames
  - Lost acks
  - Duplicate frames