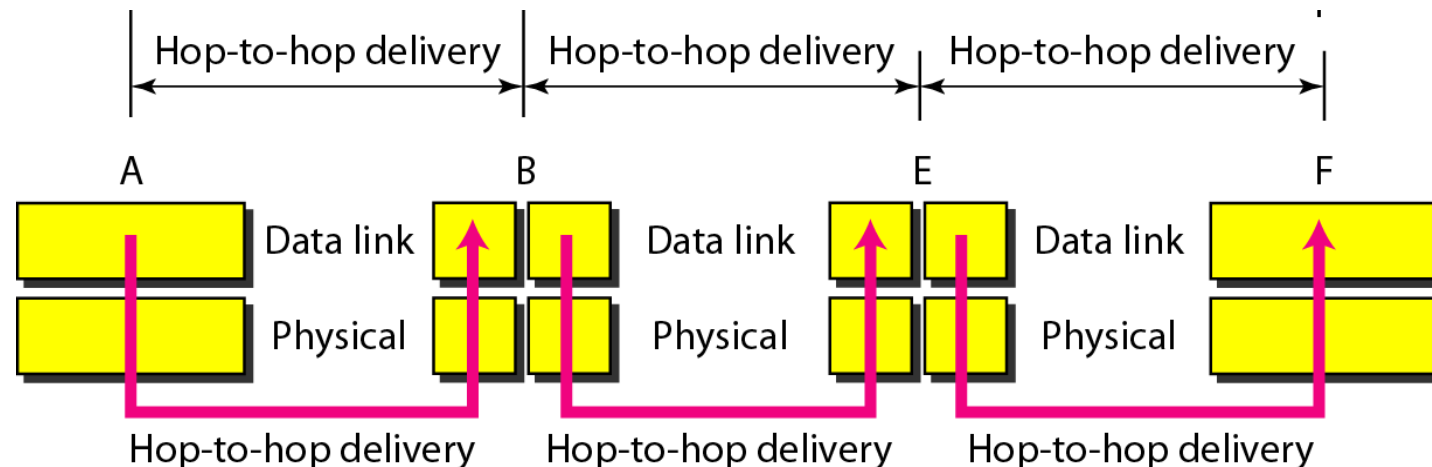


Data link Layer Framing

Data Link Layer

- Data link achieving reliable, efficient communication between two adjacent machines.
- Machines are connected by a communication channel that acts conceptually like wire (coaxial cable, telephone line, or point-to-point wireless channel)

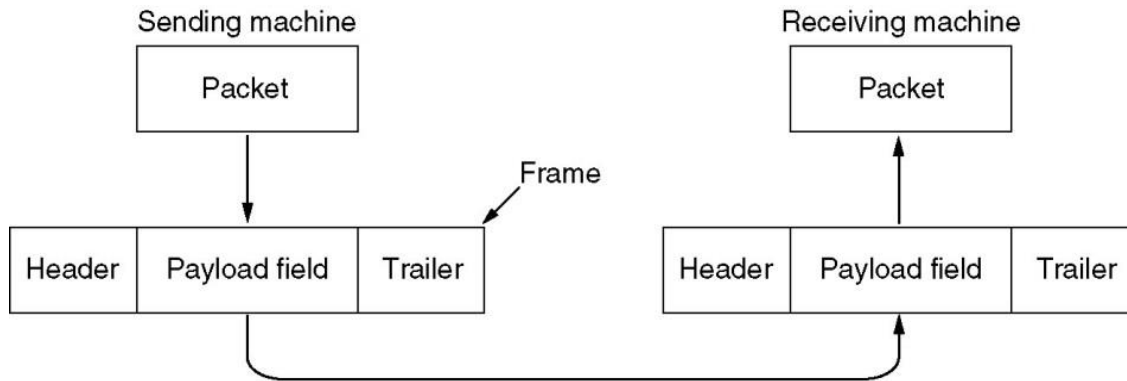


Data Link Layer – design issues

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

Data Link Layer – design issues

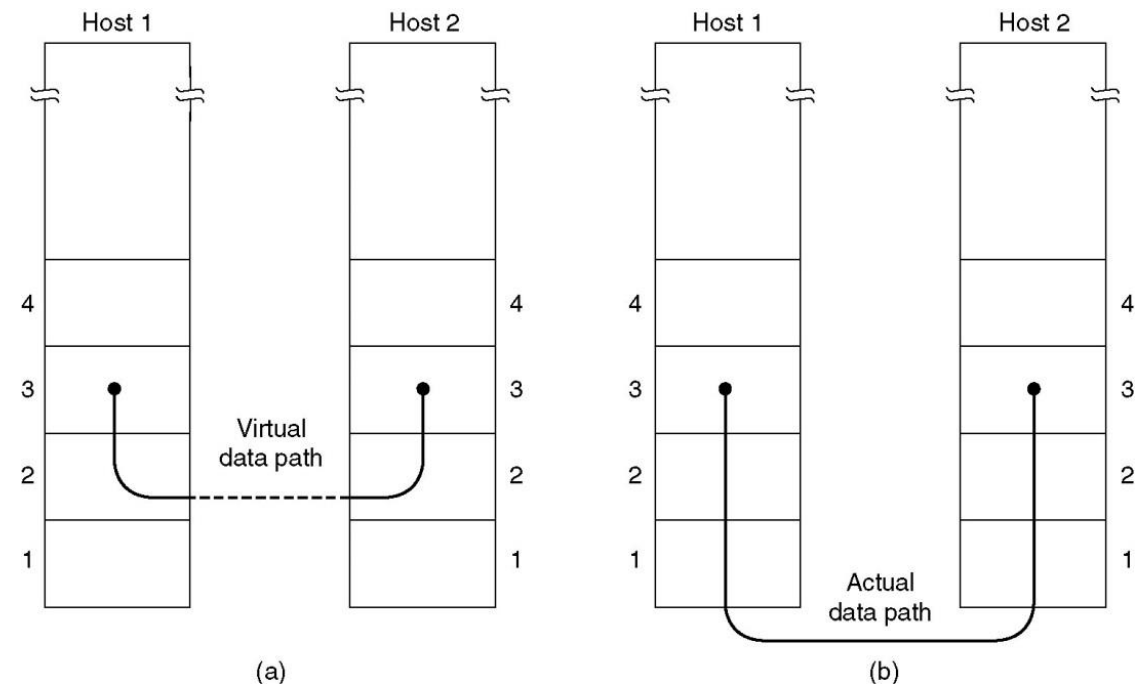
Services Provided to the Network Layer



The actual services offered can vary from system to system.

- Unacknowledged connectionless service.
- Acknowledged connectionless service.
- Acknowledged connection-oriented service.

- (a) Virtual communication
- (b) Actual communication



Framing

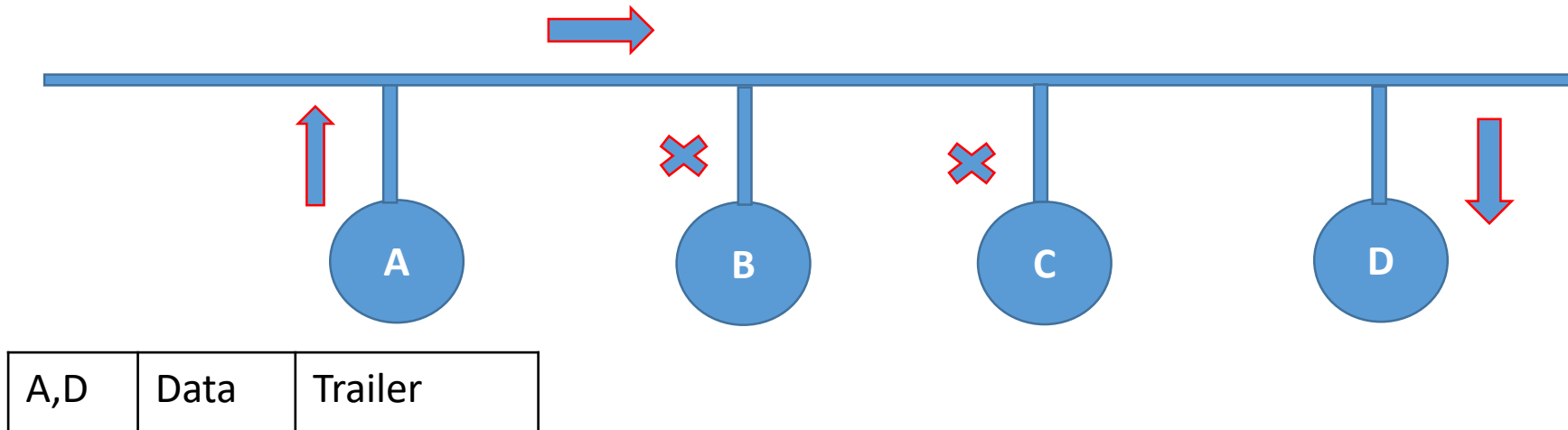
- Framing is a approach to break bit stream into discrete frames and compute the checksum for each frame.
- When a frame arrives at the destination, the checksum is recomputed.
- If the newly-computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it
- Discarding the bad frame and possibly also sending back an error report.

Fixed-Size Framing

Variable-Size Framing

Framing

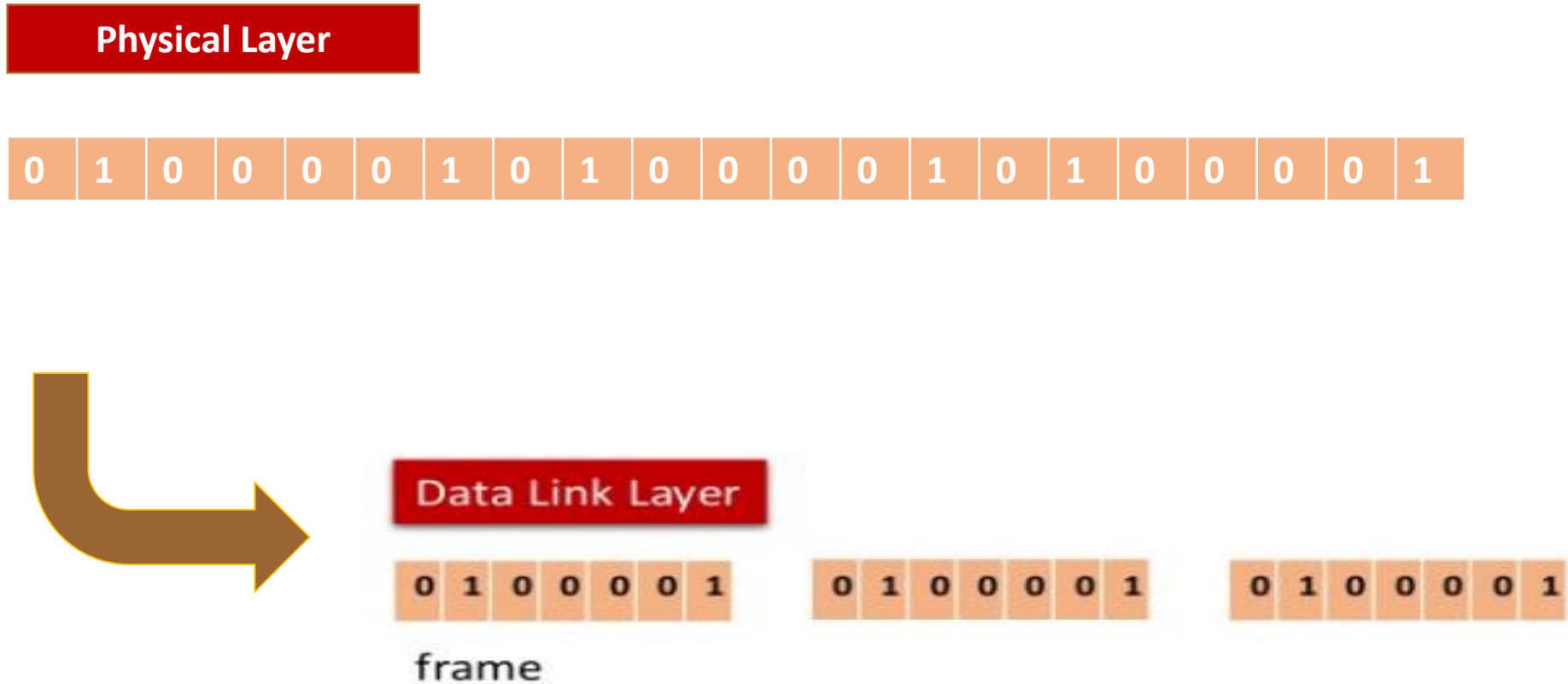
Header	Payload field	Trailer
--------	---------------	---------



Framing

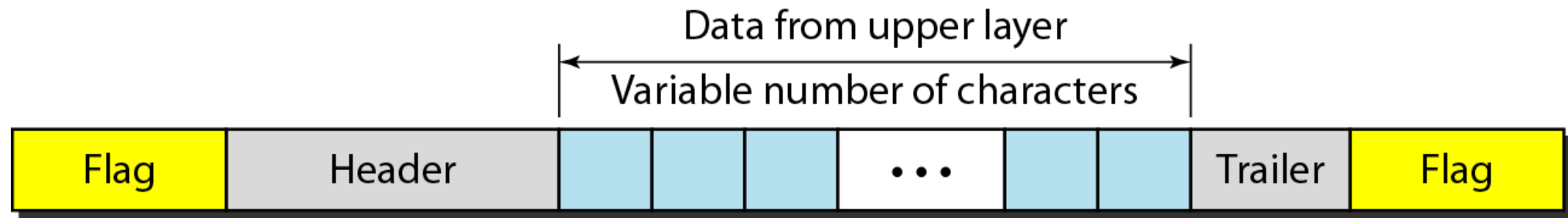
- Fixed –Size Framing
- Variable – Size framing

Fixed Size Framing



Problem: if data size is small then frame then we require to do padding

Variable Size Framing



Framing Methods

- Character Count
- Byte stuffing
- Bit stuffing
- Physical layer coding violations

Character/Byte Count

- Uses a Field in a header to specify the number of bytes into frame

A byte stream:



Framing:



Byte count

Frame 1



Frame 2



Frame 3



Character/Byte Count

- Problem with byte count

Without error:

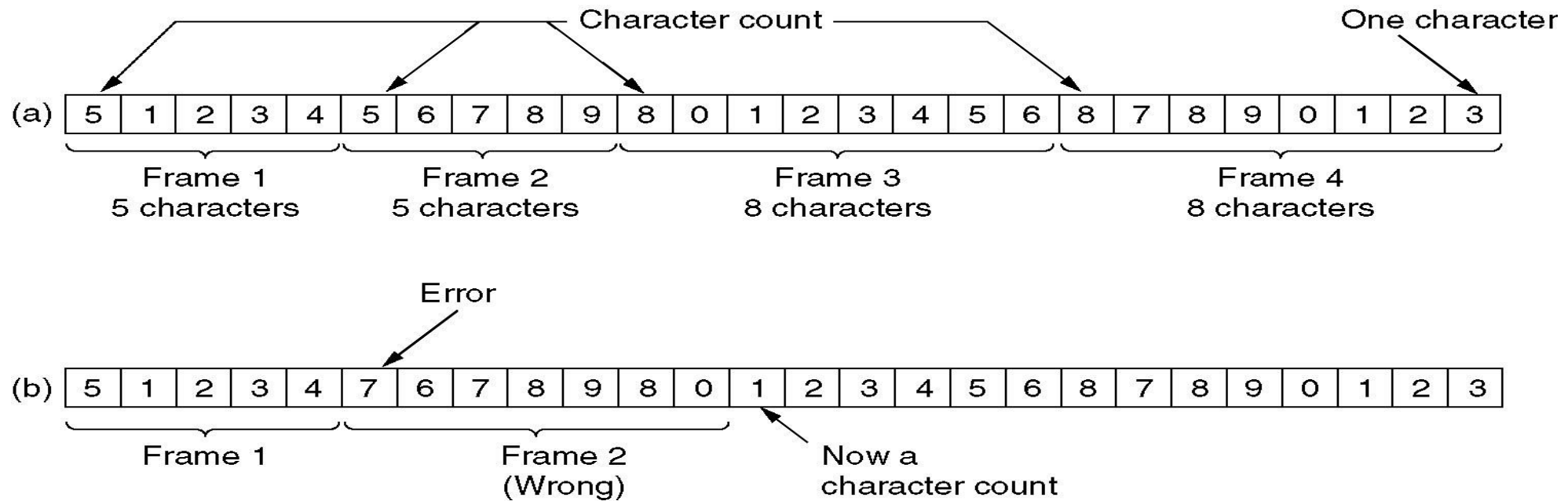


With error:



Character/Byte Count

- Problem with byte count



The trouble with this method is that the count can be garbled by a transmission error.

Parts of Frame

- **Frame Header** – It contains the source and the destination addresses of the frame.
- **Payload field** – It contains the message to be delivered.
- **Trailer** – It contains the error detection and error correction bits.
- **Flag** – It marks the beginning and end of the frame.



Byte Stuffing (character oriented protocol)

- Insert FLAG byte before each frame starting and ending



Frame 1



Frame 2



Frame 3



Byte Stuffing (character oriented protocol)

- Problem : FLAG byte is occur in data

1: FLAG byte occurs in the data



Byte Stuffing

the technique of inserting a special byte (ESC) just before each accidental flag byte in the data.

Original bytes



After Stuffing



Byte Stuffing

Byte Stuffing (character oriented protocol)

2: ESC occurs in the data

Original bytes



After Stuffing



3: ESC and FLAG occurs in the data

Original bytes



After Stuffing



Byte Stuffing (character oriented protocol)

4: Two ESC occurs in the data

Original bytes



After Stuffing

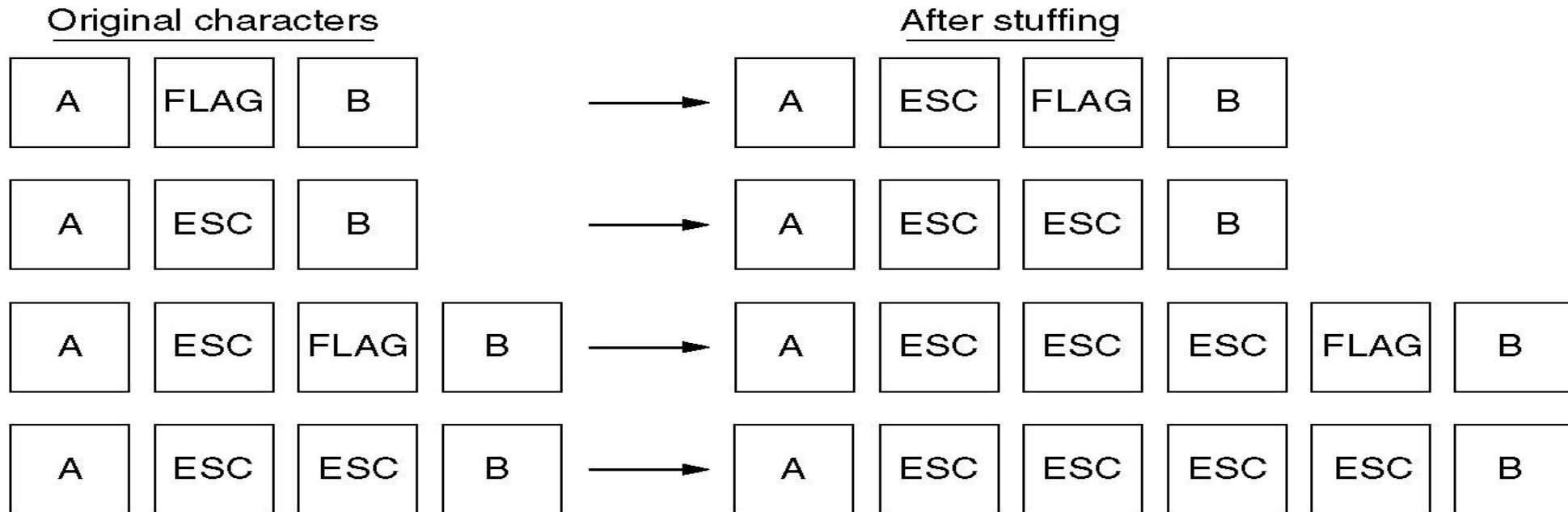


Byte Stuffing (character oriented protocol)

- Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

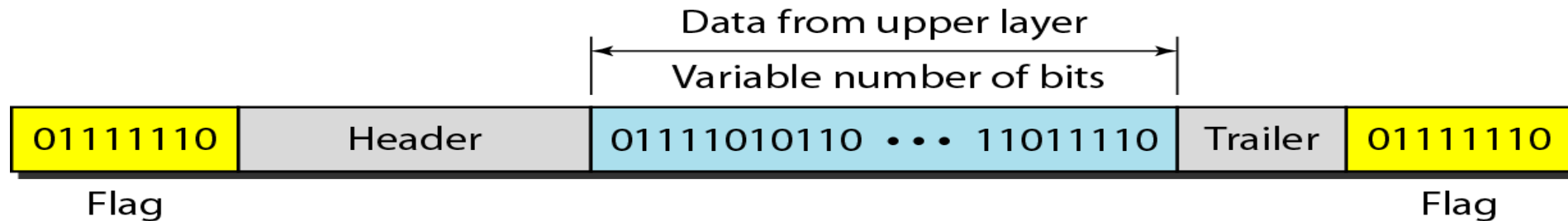


(a)

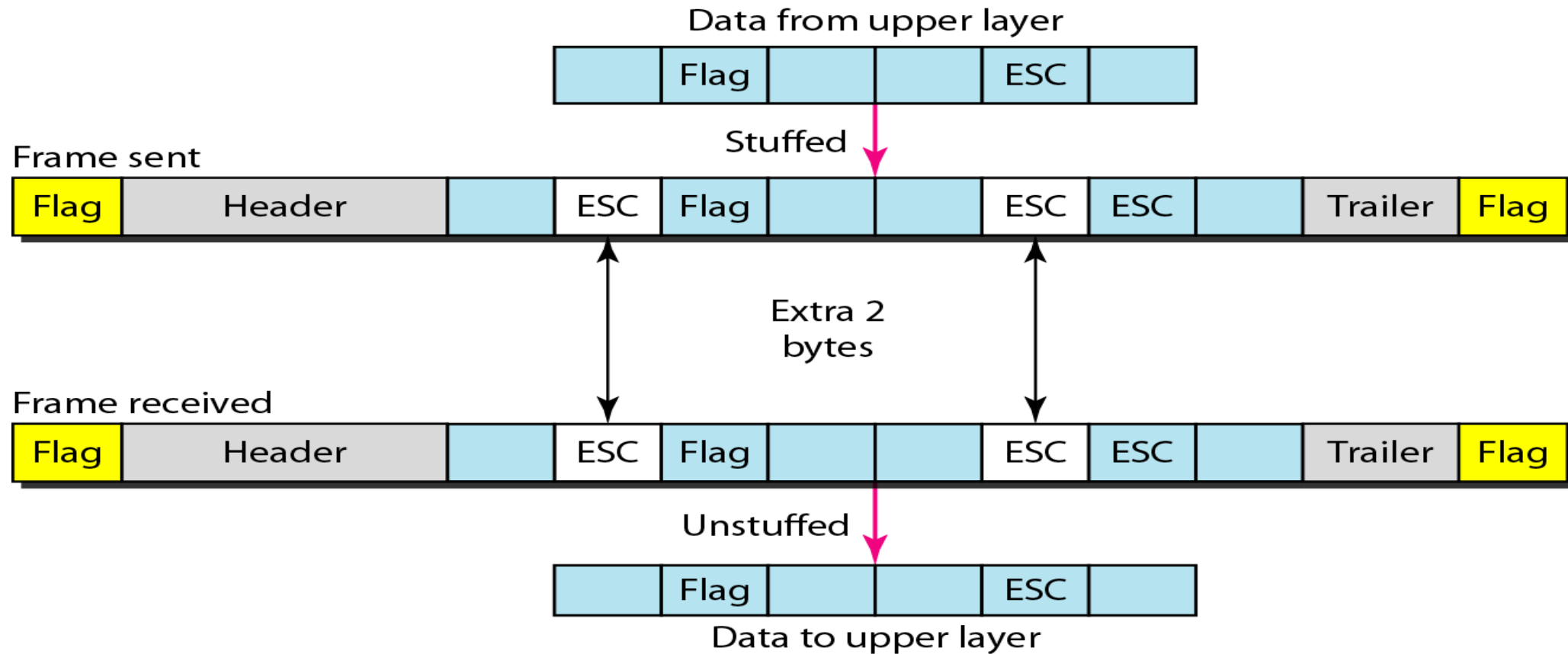


Byte Stuffing

- Uses a special 8- bit pattern flag 01111110 as the delimiter to define the beginning of the frame

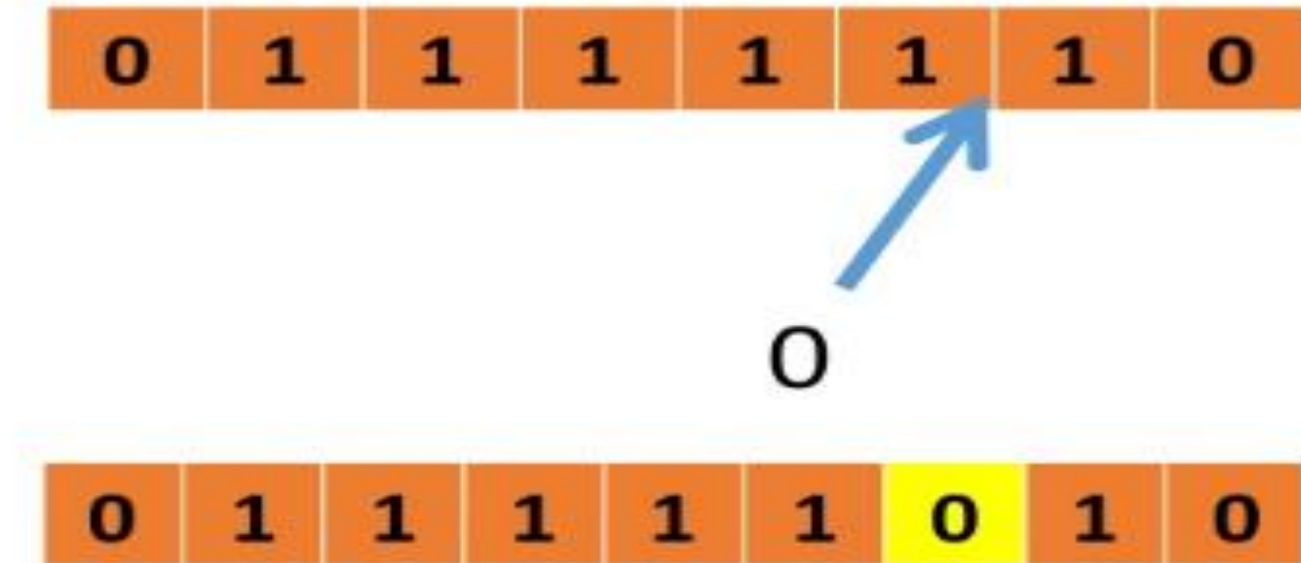


Byte Stuffing



Bit Stuffing (bit oriented protocol)

- Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.



Bit Stuffing (bit oriented protocol)

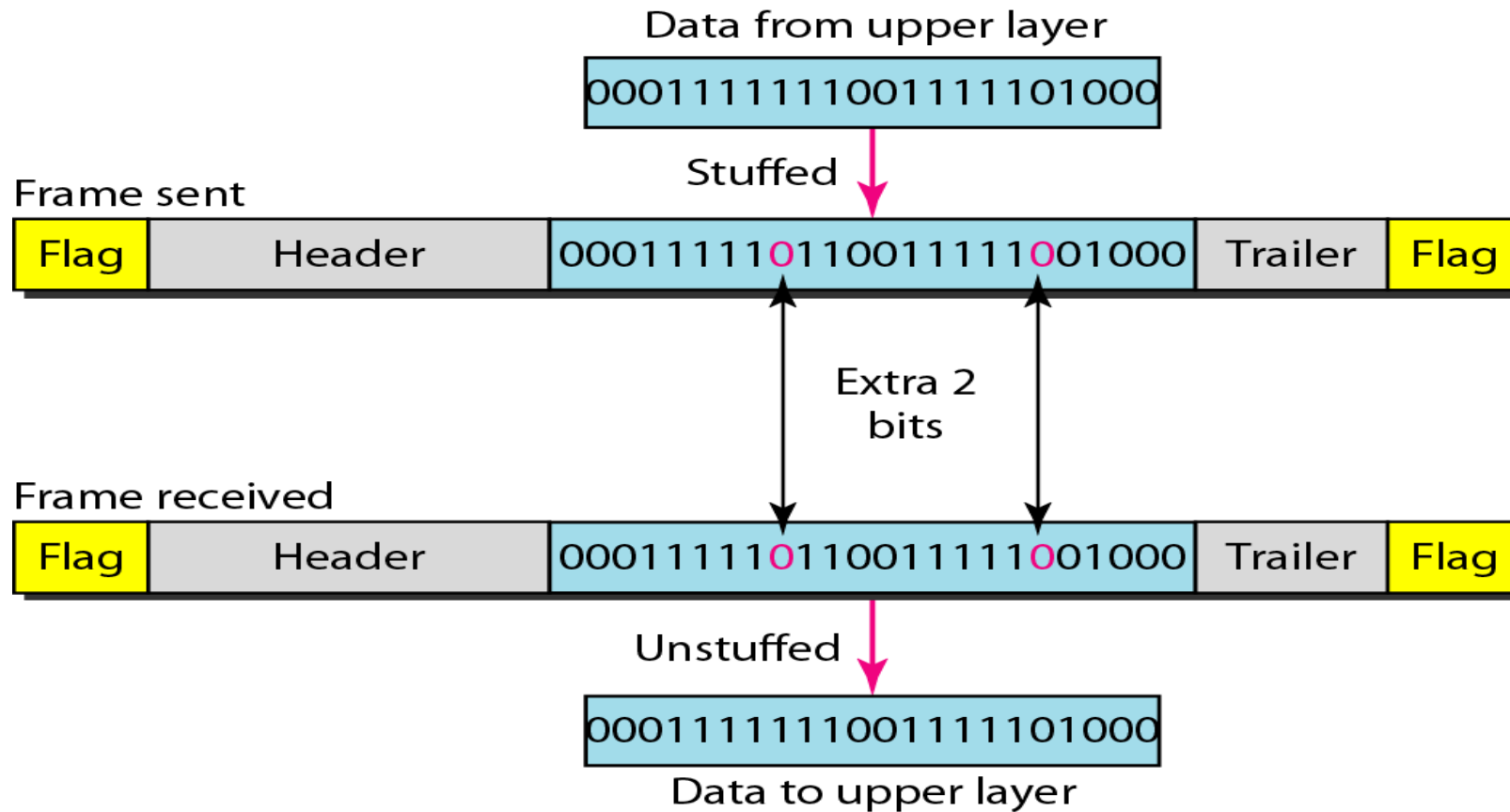
Original data

1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0

Data after stuffing

1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 0

Bit Stuffing and unstuffing



Physical layer coding violations

- It only applicable to network in which the encoding on the physical medium contains some redundancy
- Example
 - LANs encode 1 bit of data by using 2 physical bits.
 - Normally ,a 1 bit is high-low pair and 0 bit is a low-high pair
 - Easy for the receiver to locate the bit boundaries
 - High-High and Low-Low are not used for data but are used for delimiting frames in some protocols