The Data Link Layer

Section 3.1

Tanenbaum

Data Link Layer Design Issues

- Network layer services
- Framing
- Error control
- Flow control

Data Link Layer Design Issues

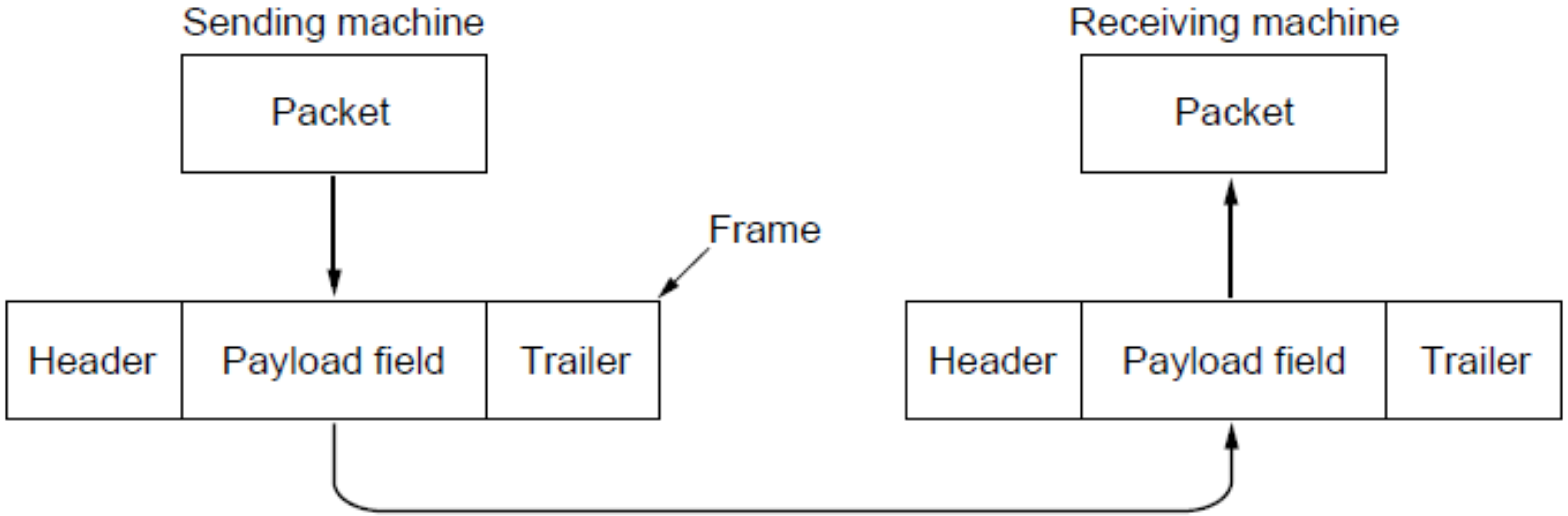
- Physical layer delivers bits of information to and from data link layer. The **functions of Data Link Layer are:**

  1. **Providing a well-defined service interface to the network layer.**
  2. **Dealing with transmission errors.**
  3. **Regulating the flow of data so that slow receivers are not swamped by fast senders.**

- Data Link layer
  - Takes the packets from Physical layer, and
  - Encapsulates them into **frames**

- **Each frame has a**
  - **frame header**
  - **a field for holding the packet, and**
  - **frame trailer.**

Relationship between packets and frames.

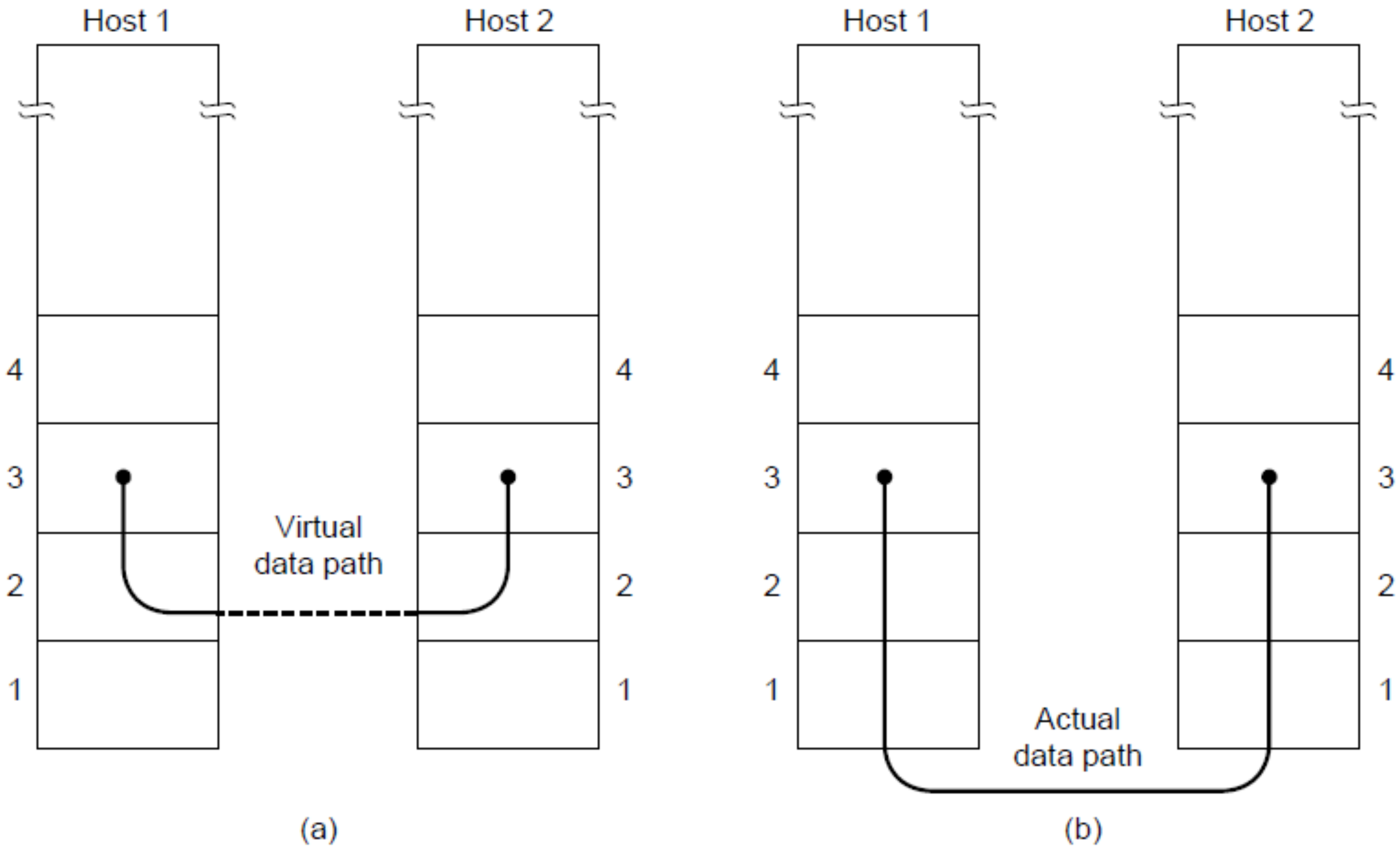


Services Provided to the Network Layer

- Principal Service Function of the data link layer is to transfer the data from the network layer on the source machine to the network layer on the destination machine.

  - Process in the network layer that hands some bits to the data link layer for transmission.
  - Job of data link layer is to transmit the bits to the destination machine so they can be handed over to the network layer there (see figure in the next slide).

Network Layer Services



Possible Services Offered

1. Unacknowledged connectionless service.
2. Acknowledged connectionless service.
3. Acknowledged connection-oriented service.

Unacknowledged Connectionless Service

- It consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them.

- Example: Ethernet, Voice over IP, etc. in all the communication channel were real time operation is more important that quality of transmission.

Acknowledged Connectionless Service

- Each frame send by the Data Link layer is acknowledged and the sender knows if a specific frame has been received or lost.

- Typically the protocol uses a specific time period that if has passed without getting acknowledgment it will re-send the frame.

- This service is useful for commutation when an unreliable channel is being utilized (e.g., 802.11 WiFi).

- Network layer does not know frame size of the packets and other restriction of the data link layer. Hence it becomes necessary for data link layer to have some mechanism to optimize the transmission.

Acknowledged Connection Oriented Service

- Source and Destination establish a connection first.

- Each frame sent is numbered
  - Data link layer guarantees that each frame sent is indeed received.
  - It guarantees that each frame is received only once and that all frames are received in the correct order.

- Examples:
  - Satellite channel communication,
  - Long-distance telephone communication, etc.

Acknowledged Connection Oriented Service

- Three distinct phases:

1. Connection is established by having both side initialize variables and counters needed to keep track of which frames have been received and which ones have not.

2. One or more frames are transmitted.

3. Finally, the connection is released – freeing up the variables, buffers, and other resources used to maintain the connection.

Framing

- To provide service to the network layer the data link layer must use the service provided to it by physical layer.
- Stream of data bits provided to data link layer is not guaranteed to be without errors.
- Errors could be:
  - Number of received bits does not match number of transmitted bits (deletion or insertion)
  - Bit Value
- It is up to data link layer to correct the errors if necessary.

Framing

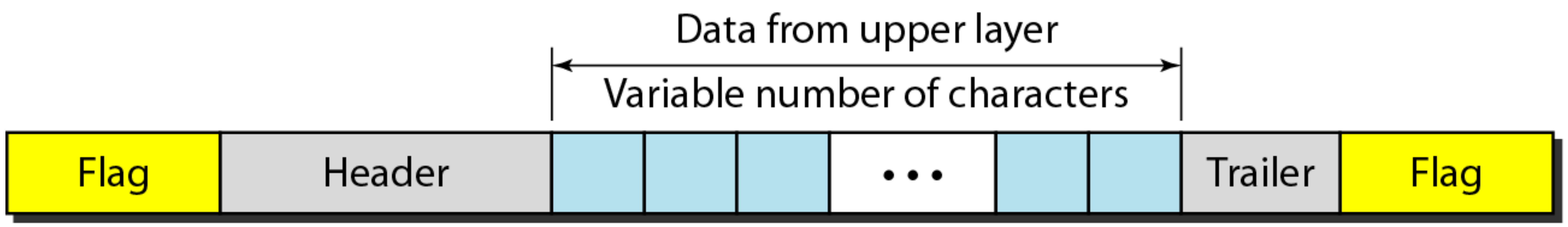
- Transmission of the data link layer starts with breaking up the bit stream
  - into discrete frames
  - Computation of a checksum for each frame, and
  - Include the checksum into the frame before it is transmitted.

- Receiver computes its checksum error for a receiving frame and if it is different from the checksum that is being transmitted will have to deal with the error.

- Framing is more difficult than one could think!

Framing

- Process of wrapping data with certain info before sending out



- A frame typically consists of
  - Flag: indication for start and end of a frame
  - Header: source/destination addresses, as well as other control information
  - Data from the upper layer
  - Trailer: error detection/correction code

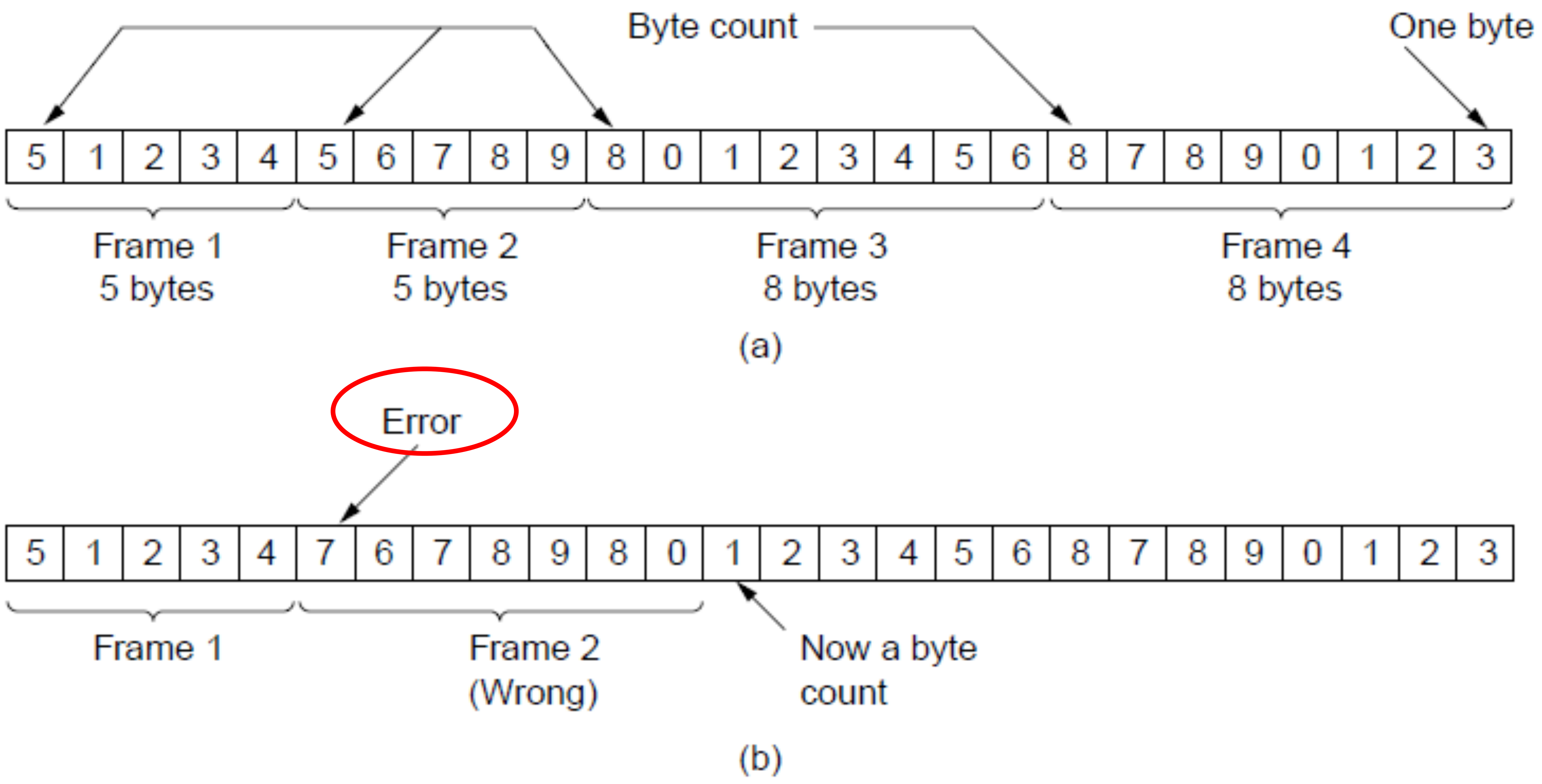
Framing Methods

1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.

Character Count Framing Method

- It uses a field in the header to specify the number of bytes in the frame.
- Once the header information is being received it will be used to determine end of the frame.
- When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is.

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



- Trouble with this algorithm is that when the count is incorrectly received the destination will get out of synch with transmission.
  - For example, if the character count of 5 in the second frame becomes a 7, the destination will get out of synchronization and will not be able to locate the start of the next frame.
- Even if the checksum is incorrect, so the destination knows that the frame is bad, it still has no way of telling where the next frame starts.
- Sending a frame back to the source asking for retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission.
- For this reason, this method is rarely used.

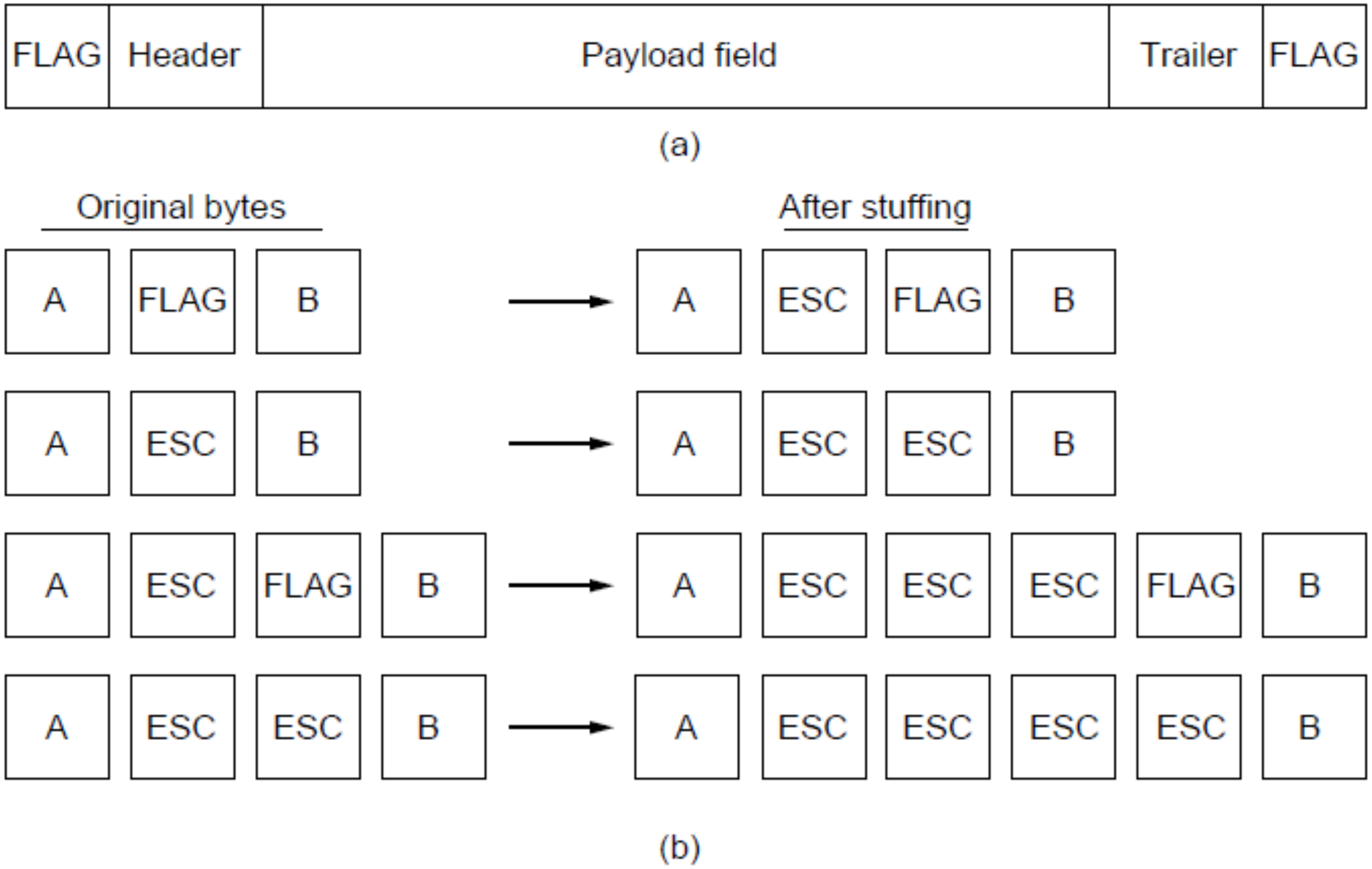
Flag Bytes with Byte Staffing Framing Method

- This methods gets around the boundary detection of the frame by having each appended by the frame start and frame end special bytes.

- If they are the same (beginning and ending byte in the frame) they are called **flag byte**.

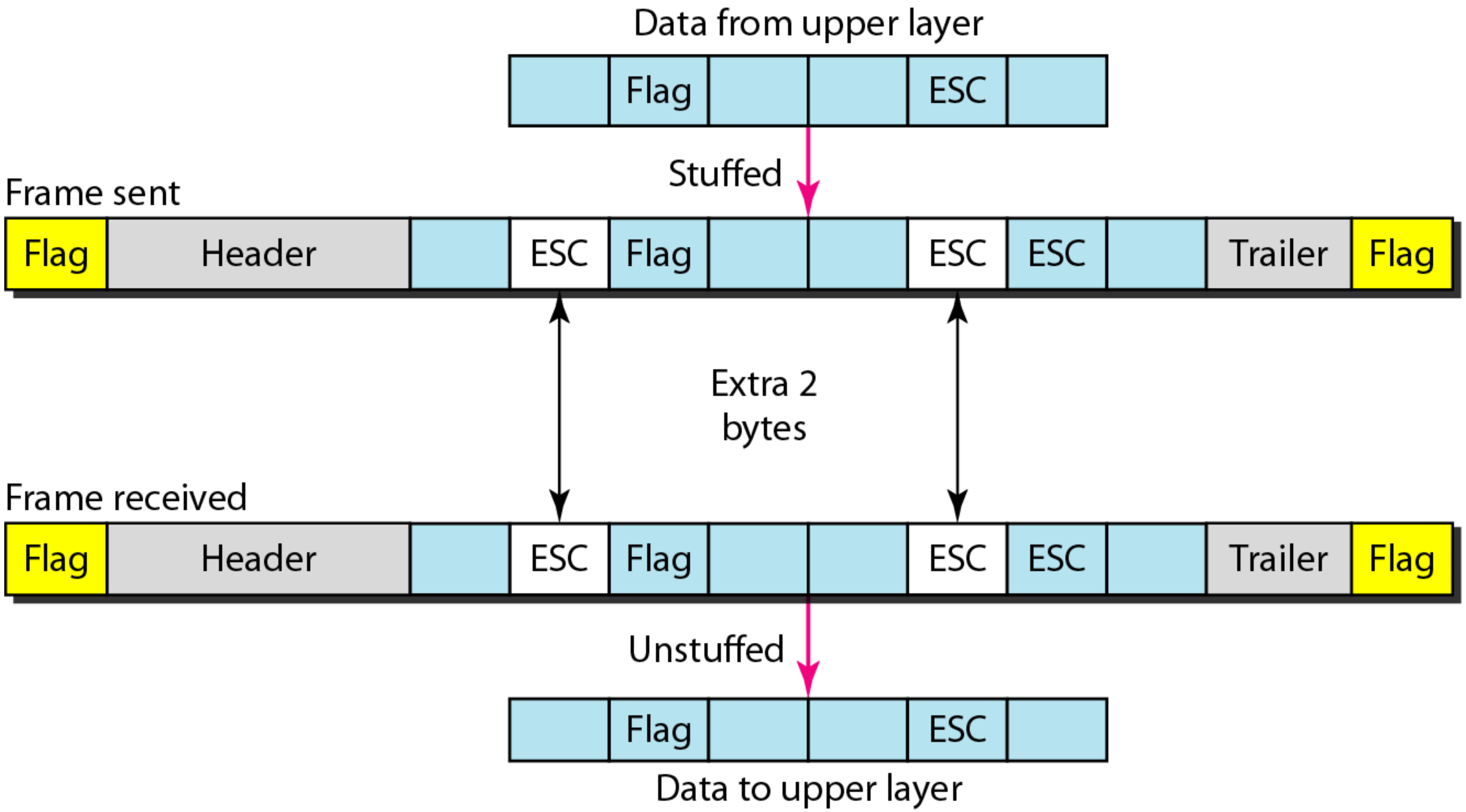
- In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame.

- Two consecutive flag bytes indicate the end of one frame and start of another frame.



(a) A frame delimited by flag bytes.

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


Data from upper layer
Flag
ESC
Stuffed
Frame sent
Flag
Header
ESC
Flag
ESC
ESC
Trailer
Flag
Extra 2
bytes
Frame received
Flag
Header
ESC
Flag
ESC
ESC
Trailer
Flag
Unstuffed
Flag
ESC
Data to upper layer

- A major disadvantage of using this framing method is that it is closely tied to the use of 8-bit characters.

- Not all character codes use 8-bit characters. For example, UNICODE uses 16 bits.

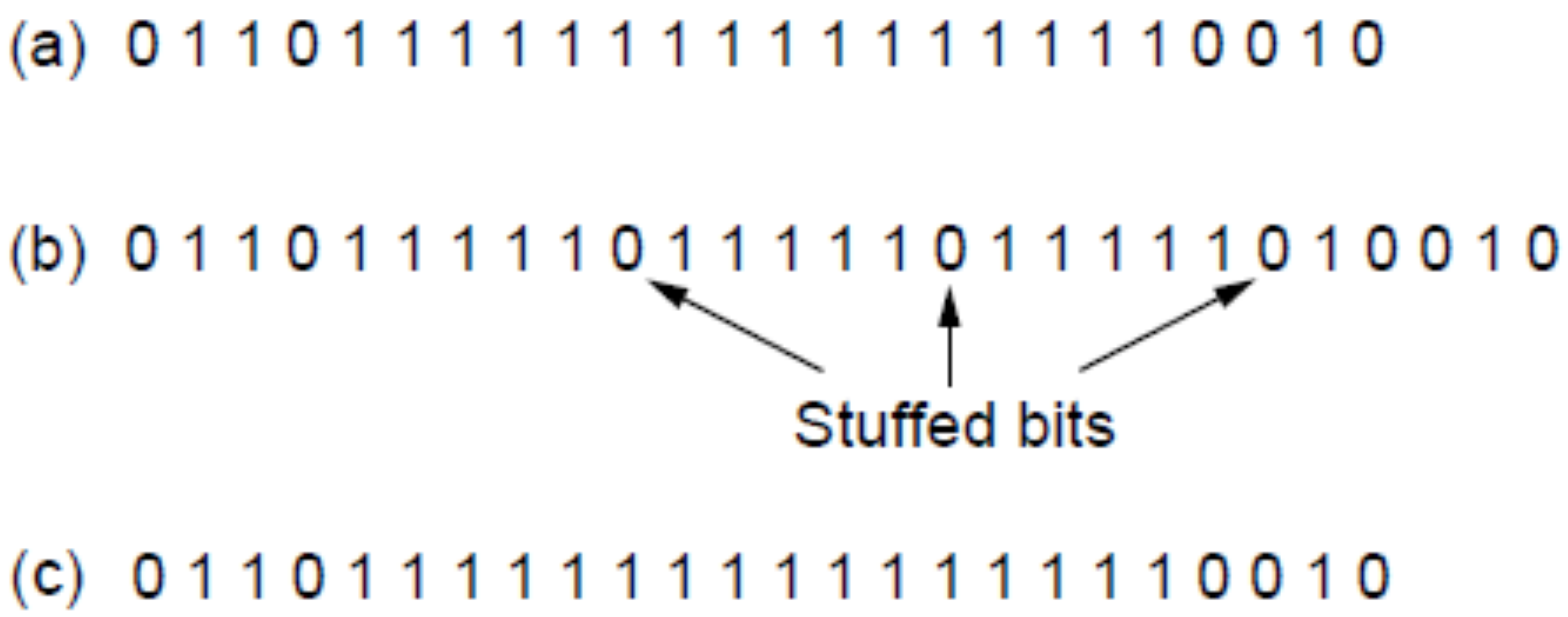
- As network developed, the disadvantages of embedding the character code length in the framing mechanisms became more and more obvious, so a new technique had to be developed to allow arbitrary sized characters.

Flag Bits with Bit Stuffing Framing Method

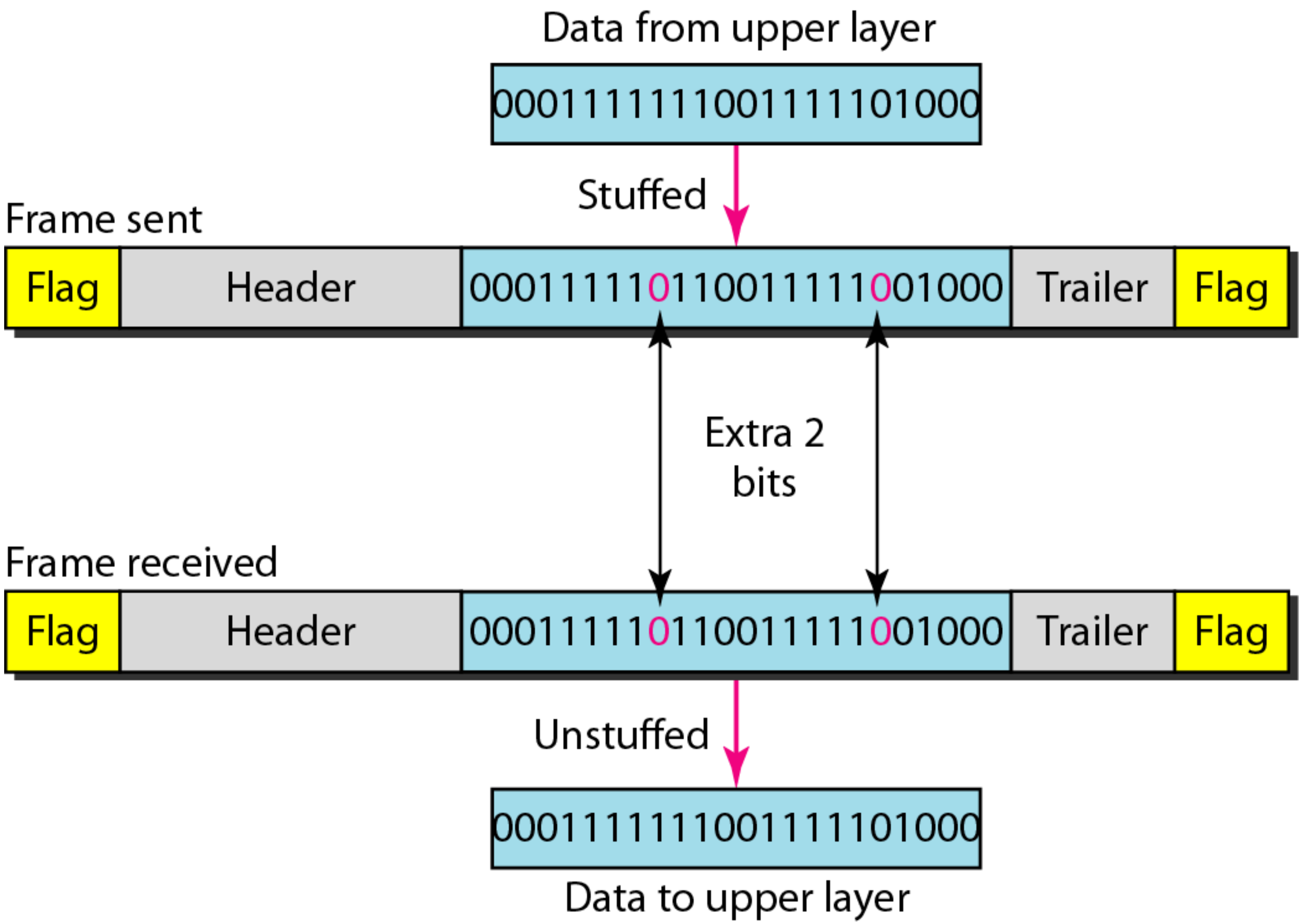
- This methods achieves the same thing as Byte Stuffing method by using Bits (1 Bit) instead of Bytes (8 Bits).

- It was developed for High-level Data Link Control (HDLC) protocol.

- Each frames begins and ends with a special bit patter:
  - 01111110 or 0x7E <- Flag Byte
  - Whenever the sender's data link layer encounters five consecutive 1s in the data it automatically stuffs a 0 bit into the outgoing bit stream.
  - USB uses bit stuffing.



Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.



Data from upper layer
0001111111001111101000
Stuffed
Frame sent
Flag
Header
000111110110011111001000
Trailer
Flag
Extra 2 bits
Frame received
Flag
Header
000111110110011111001000
Trailer
Flag
Unstuffed
0001111111001111101000
Data to upper layer

- With Bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern.

- Thus, if the receiver loses track of where it is, then all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

Error Control

- After solving the marking of the frame with start and end the data link layer has to handle eventual errors in transmission or detection.
  - Ensuring that all frames are delivered to the network layer at the destination and in proper order.

- Suppose that the sender just kept outputting frames without regard to whether they were arriving properly.

- This might be fine for Unacknowledged connectionless service, but would most certainly be not fine for reliable connection-oriented service.

- The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the other end of the line. Typically, the protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgement about the incoming frame.

- If the Sender Receives positive Acknowledgment it will know that the frame has arrived safely.

- On the other hand, a negative acknowledgement means that something has gone wrong, and the frame must be retransmitted.

- An additional complication comes from the possibility that hardware troubles may cause a frame to vanish completely.
  - In this case, the receiver will not react at all, since it has no reason.
  - And, the sender will hang forever in wait of a positive or negative acknowledgement.
  - This possibility is **dealt with by introducing timers** into the data link layer.

- When the sender transmits a frame, it generally also starts a timer. The timer is set to expire after an interval long enough for the frame to reach the destination, be processed there, and have the acknowledgement propagate back to the sender.

- Normally, the frame will be correctly received and the acknowledgement will get back before the timer runs out, in which case the timer will be cancelled.

- However, if either the frame or acknowledgement is lost, the timer will go off, alerting the sender to a potential problem. The solution is just to transmit the frame again.

- To prevent the receiver to accept the same frame multiple times, sequence numbers are assigned to the outgoing frames so that the receiver can distinguish retransmissions from originals.

Flow Control

- Important Design issue for the cases when the sender is running on a fast powerful computer and receiver is running on a slow low-end machine.

- Two approaches:

1. Feedback-based flow control
2. Rate-based flow control

Feedback-based Flow Control

- Receiver sends back information to the sender giving it permission to send more data, or

- Telling sender how receiver is doing.

Rate-based Flow Control

- Built in mechanism that limits the rate at which sender may transmit data, without the need for feedback from the receiver.