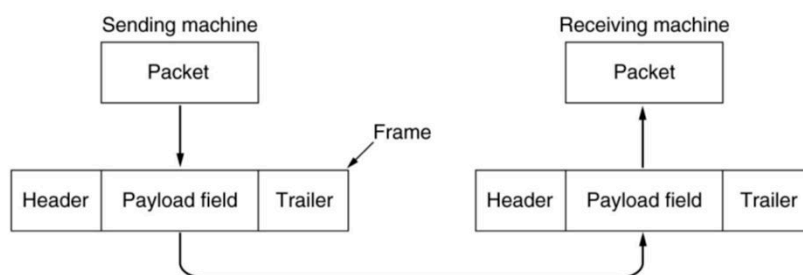


Lecture 5: Data Link Layer Basics

Prof. Mohammed Hawa
Electrical Engineering Department
University of Jordan

EE426: Communication Networks

Layer 2 PDU: Frame



Copyright © Prof. Mohammed Hawa

Electrical Engineering Department, University of Jordan

2

Bit-oriented vs. Byte-oriented

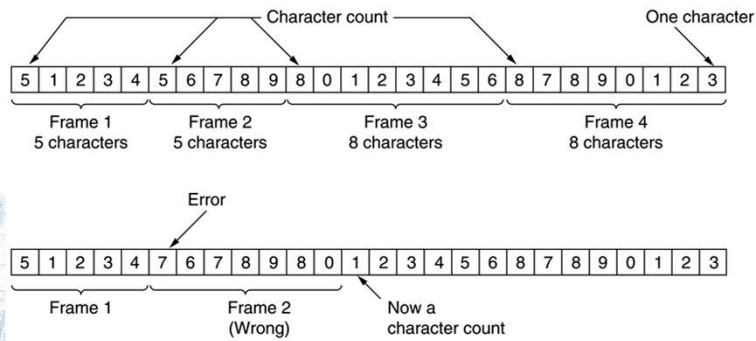
- Layer 2 protocols can be either: byte-oriented or bit-oriented.
- In bit-oriented, frames can contain an arbitrary number of bits.
 - Example: **HDLC** (High-Level Data Link Control) protocol.
- In byte-oriented (character-oriented), frames consist of an integral number of bytes (8 bits).
 - Example: **PPP** (Point-to-Point Protocol).
- In both, the data link layer has the job of dividing a continuous bit stream into identifiable frames.

Framing

- 1. Insert time gaps** of no transmission periods between frames.
 - *Example:* Asynchronous transmission in RS-232 standard.
 - *Example:* Ethernet.
 - *Example:* Wi-Fi.
 - Easiest method.
 - Inefficient because bandwidth is wasted while not transmitting useful data.
 - Unreliable except for short distances (*noise*).

Framing

2. Character Count (used mainly in byte-oriented protocols)

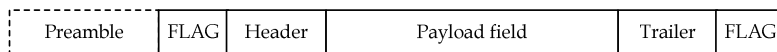


Copyright © Prof. Mohammed Hawa

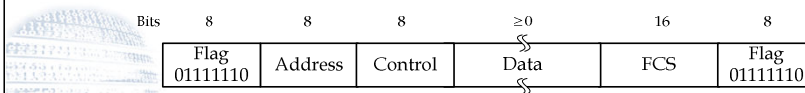
Electrical Engineering Department, University of Jordan 5

Framing

3. Frame Delimiters (used in bit-oriented and byte-oriented).



For example, the HDLC data link protocol (a *bit-oriented* protocol) uses the 8-bit sequence **01111110** as a flag sequence (see figure).



Copyright © Prof. Mohammed Hawa

Electrical Engineering Department, University of Jordan 6

Framing

- PPP (a *byte-oriented* protocol) uses the FLAG = 0x7E = 01111110

Bytes	1	1	1	1 or 2	Variable	2 or 4	1
	Flag	Address	Control	Protocol	Payload	FCS	Flag
	01111110	11111111	00000011				01111110



Copyright © Prof. Mohammed Hawa

Electrical Engineering Department, University of Jordan 7

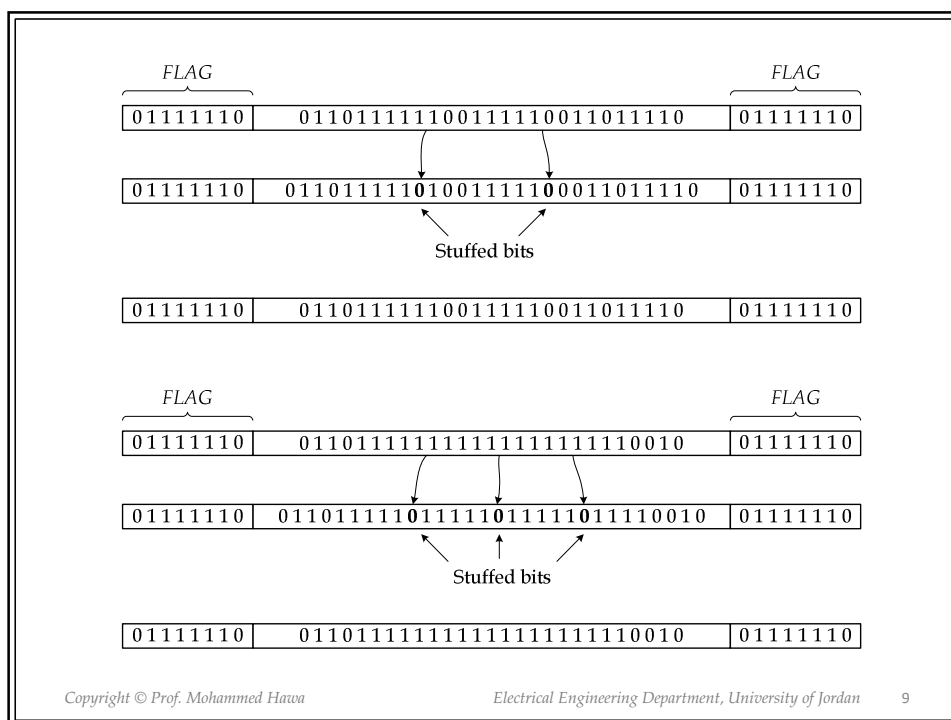
Bit Stuffing

- Prevents flags occurring in the middle of a frame, which would lead the receiver to have a frame slip.
- **TX:** add an extra 0 after a group of five ones that appears in the middle of the frame. This way whatever bit precedes or follows the 11111, there is no chance the flag will appear in the middle of the frame. The start and end flags are not stuffed.
- **RX:** performs *destuffing* by replacing every pattern 111110 by 11111 before the data is handed to the network layer.



Copyright © Prof. Mohammed Hawa

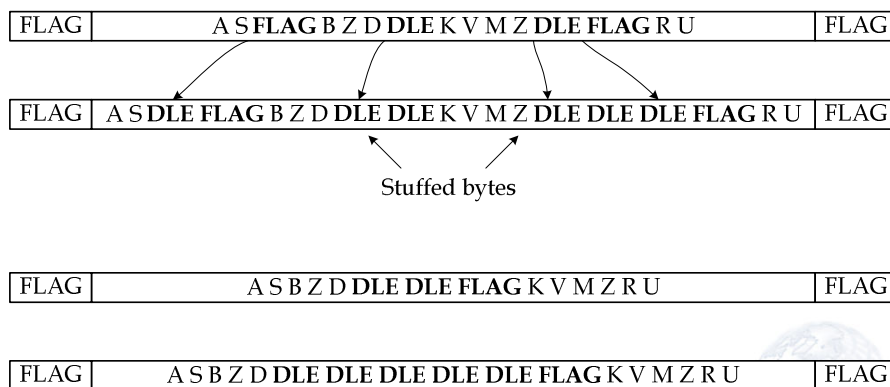
Electrical Engineering Department, University of Jordan 8



Byte Stuffing

- Byte stuffing (character stuffing) consists of using a special *data link escape* character DLE to prevent the occurrence of the FLAG (and DLE) bytes in the middle of the frame.
- The receiving end removes any DEL byte it finds while keeping the subsequent byte.
- For example, PPP uses the FLAG = 0x7E = 01111110, and the DLE = 0x7D = 01111101.

Byte Stuffing



Copyright © Prof. Mohammed Hawa

Electrical Engineering Department, University of Jordan 11

Error Detection and Correction

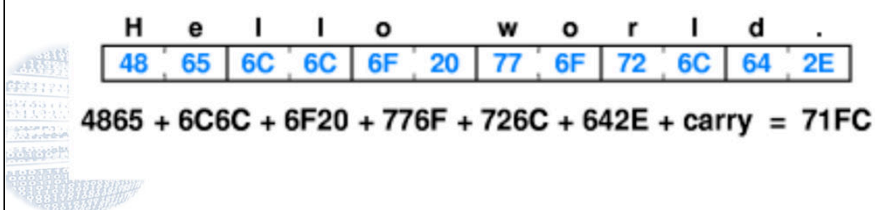
- **Parity:** A parity bit is added to the sent message M . The parity bit is set to 1 or 0 to force the transmitted message to contain **even** or **odd** number of 1's.
- 7 bit data: $M = 1010001$
- 8 bits transmitted (including parity):
 - 10100011 (*even* parity)
 - 10100010 (*odd* parity)

Copyright © Prof. Mohammed Hawa

Electrical Engineering Department, University of Jordan 12

Checksums

- Checksum is the sum (plus carry) of the 16-bit words making up the message M . The receiver checks that the sum of the received message M' matches the checksum sent by the transmitter. If both sums match, no errors have occurred during the transmission of the message.
- Checksums are used in TCP and IP headers.



Copyright © Prof. Mohammed Hawa

Electrical Engineering Department, University of Jordan 13

Cyclic Redundancy Check (CRC)

- Message bits M are divided by a bit sequence called a *generating polynomial* G , and the remainder of the division R (called a CRC) is sent along with the message.
- The receiver divides the received message bits along with the CRC code ($M' : R'$) by the generating polynomial G .
- If the remainder is zero, no errors have occurred. If the remainder is not zero, an error occurred.
- CRC can be implemented in hardware using a shift register and X-OR gates (inexpensive and fast).
- CRC can detect burst errors.
- CRC is used in HDLC, PPP, Ethernet, and many others. It is the most common error detection code nowadays.

Copyright © Prof. Mohammed Hawa

Electrical Engineering Department, University of Jordan 14

