College of Computer Sciences and Engineering

Information and Computer Science Department

ICS 343: Fundamentals of Computer Networks (3-3-4)

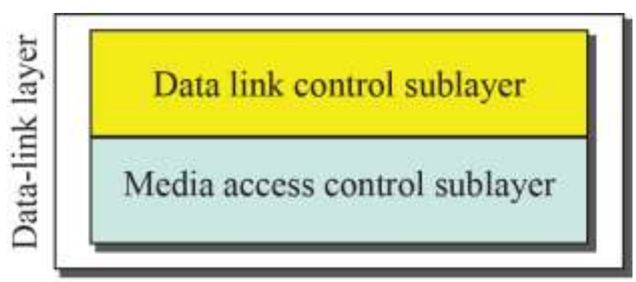
Quiz#4 Key

Name: **ID:**

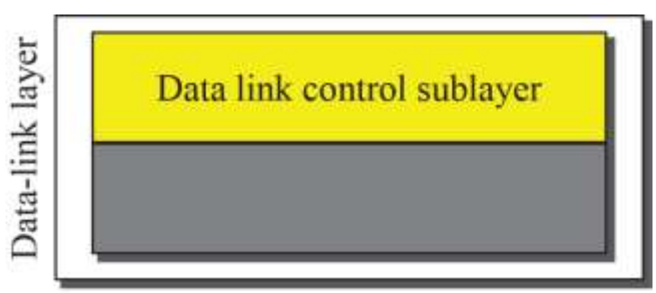
Part I (Chapter 9)

1.1. [10 points] According to the textbook, what are the two main sub-layers of the Data-Link Layer of the TCP/IP model? How do they differ?

- Data Link Control (DLC) sublayer deals with all issues common to both poit-to-point and broadcast links.
- Media Access Control (MAC) sublayer deals only with issues specific to broadcast links.

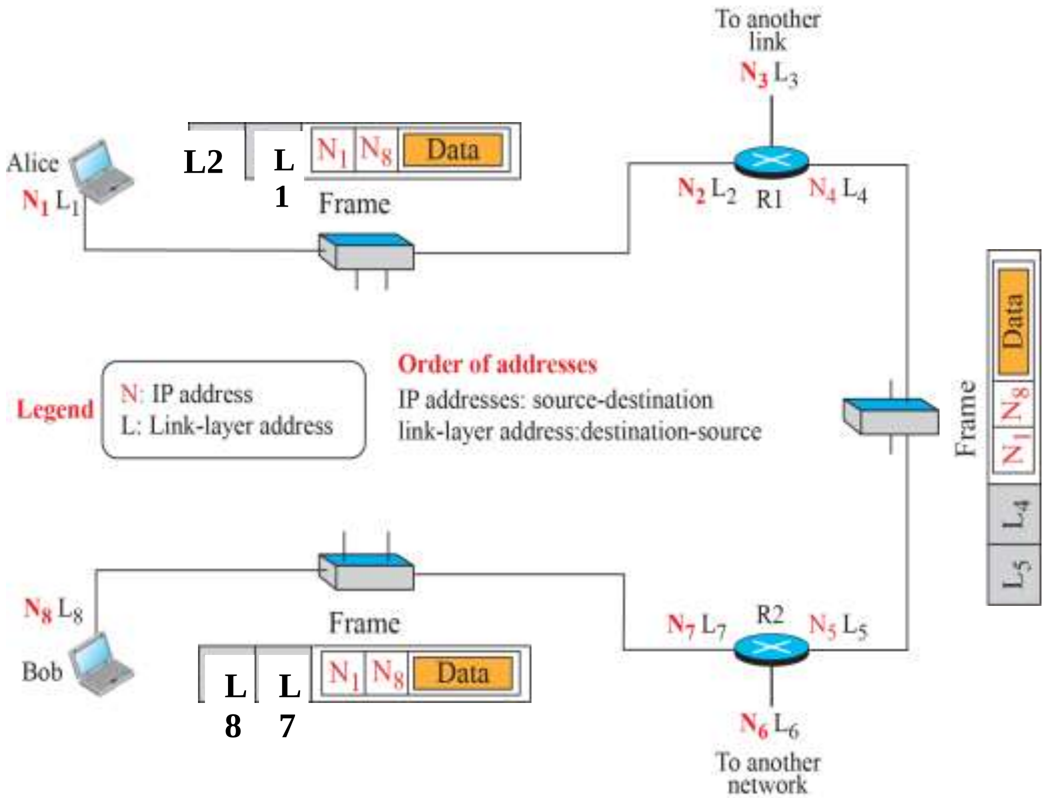


a. Data-link layer of a broadcast link



b. Data-link layer of a point-to-point link

1.2. [10 points] Based on your understanding of the relation between IP Addresses (N) and Link-Layer Addresses (L), Complete the missing 8 fields in the frames in the figure below.



Part II (Chapter 10)

2.1. [5 points] What is the minimum Hamming distance of a code that is (should be) able to detect any 3-bit errors?

To detect 3-bit errors (s = 3), the minimum Hamming distance of the code should be:

$$d_{min} = s + 1 = 4.$$

2.2. [10 points]

a) What is the Hamming distance between the following two codewords?

d(1101000 , 0101011)

b) What is the maximum number of bit-errors that can be detected by these two codewords?

Answer: a) 3 since 1101000 ***XOR*** 0101011 = 3

b) 2

2.3. [5 points]Given the following Codewords, generated by Simple Parity Check:

10001, 11110, 01111, 11011 and 00110

What are their corresponding Datawords?

It is obvious that the right most bit was added to ensure all codewords contain an even number of 1s. Thus:

Codeword	Dataword
10001	1000
11110	1111
01111	0111
11011	1101
00110	0011

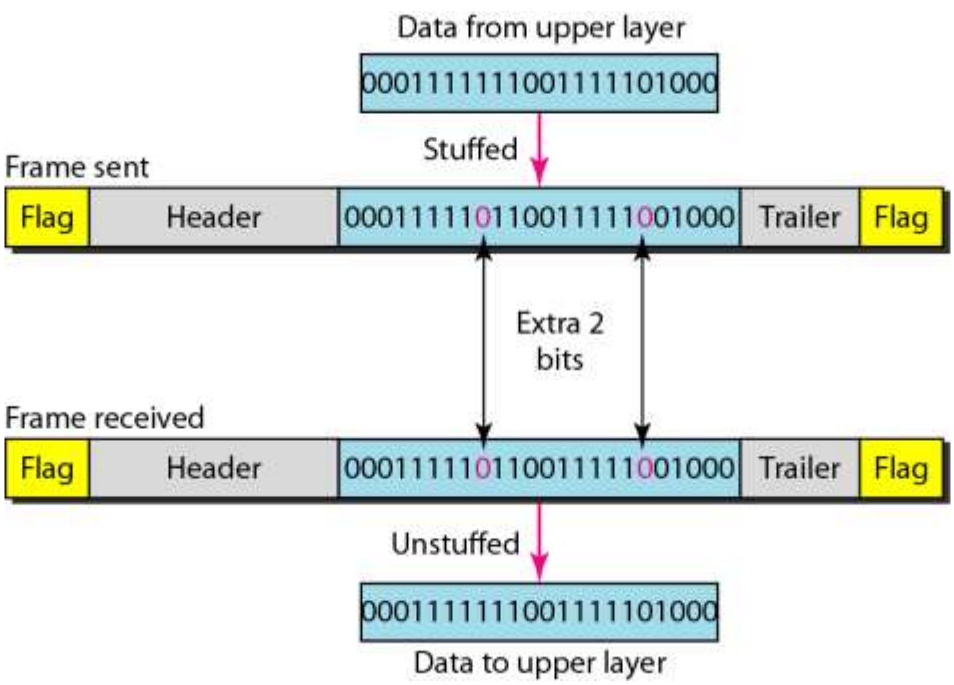
Part III (Chapter 11)

3.1. [15 points] Explain the need of the **ESC** character in framing (at the Data-Link Layer). Why are their **two** cases where **ESC** is being stuffed as an extra byte into a frame?

In Character Stuffing, a special character (*ESC*) is added to the data section of the frame when there is a character with the same pattern as the flag or the special character. Full explanation is on pp 295-296.

3.2. [10 points] What is Bit Stuffing? Why is it used?

Hint:



Answer: 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, which is the delimiter to determine the beginning or the end of a frame.

Part IV (Chapter 12)

4.1. [15 points] Compare between **Pure ALOHA**, **Slotted ALOHA** and **CSMA** in terms of their approach to reduce the chance of collision between frames.

- Pure Aloha uses random Back-off.
- In Slotted ALOHA, the station is forced to transmit only at a beginning of a designated time slot.
- In CSMA, the station checks the state of the medium (listens) before it transmits.

4.2. [5 points] When can the **1-persistent** approach be considered a special case of the **p-persistent** approach?

If $p = 1.0$

4.3. [5 points] Explain how the concept of **Token Passing** is used to organize the access of media among stations connected to the same network.

Token Passing is a controlled access method to avoid collision of packets in a broadcast environment.

Here, the stations in a network are organized in a logical ring. In other words, for each station, there is a predecessor and a successor. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring.

More details are on page 343 of the textbook.

4.4. [5 points] Check to see if the following set of chips can belong to an orthogonal system:
[+1,+1] and [+1,-1]
Explain your answer.

Yes. Since:

- **[+1,+1] . [+1,-1]= +1-1 = 0**
- **[+1,+1] . [+1,+1]= +1+1 = 2 = *n* = number of elements in each chip**

4.5. [5 points] Check to see if the following set of chips can belong to an orthogonal system:

[+1,+1,+1,+1] , [+1,-1,-1,+1] , [-1,+1,+1,-1] , [+1,-1,-1,+1]

Explain your answer.

No. Since two chips are identical ([+1,-1,-1,+1]), which violates the condition: if we multiply each code by another, we get 0.