Computer Networks and Distributed Systems

Solutions to Questionnaire no. 5

© 2016-17, José María Foces Morán

Context

- Based on this lecture presentation: http://paloalto.unileon.es/cn/ch2-2017.pdf
  - Framing: BiSync, HDLC, etc
  - Error detection: Horizontal parity, bidimensional parity, CRC

Exercises

1. What is the essential difference between a bit-oriented protocol and a byte-oriented one? State the difference in a precise manner.

   Bit-oriented protocols accept payloads of any size, whereas byte-oriented protocols require payloads to be byte-aligned, *i.e.,* their size should be a multiple of 1 byte or 8 bits.

2. HDLC is bit-oriented protocol that uses the following sentinel whose name is *flag:* 01111110. The flag marks the start of a new HDLC frame and is also used to mark the end a the frame, much in the style of the quotation mark used for the start and the end of constant string in the C language. Ethernet is also a bit-oriented protocol since it allows a payload of any size, not necessarily a multiple of 8 (One byte). Let's recall the structure of an Ethernet frame (Consult slide no. 64 in the presentation mentioned above):

   [Preamble] [Dest MAC] [Src MAC] [Ethertype] [PAYLOAD] [CRC]

   The preamble is a 64-bit pattern much like HDLC's flag, but longer; the preamble is the Ethernet's sentinel that marks the start of a new frame.

   a. What field in the Ethernet frame constitutes the sentinel marking its end?
      Ethernet, today, constitutes a varied group of subnetwork-layer technologies, all of them honor the basic datalink frame structure, however, a vast amount of phy (Physical-layer) transceivers are available. For the time being, we will limit ourselves to the original Ethernet which start-of-frame sentinel is the physical-layer preamble: 64-bit pattern 1010…1011. The original Ethernet system specifies no particular end-of-frame sentinel.
   b. If you can't identify the end of frame sentinel in the Ethernet frame, can you speculate how the receiver discovers the end of the received frame?
      It's the absence of carrier signal after transmission of the last bit of the CRC field that implicitly indicates end-of-frame.

3. Solve the following textbook exercises (Ch. 2): 7 (Consult slide no. 49), 8 (Consult slide no. 41)
   Ex. 7: Assume the role of receiver, then, follow the guidelines in slide no 49. Check your solution with the textbook's.

Ex. 8: The transmitter, whenever it finds a sentinel or any protocol-special character embedded in the payload, it escapes it by prepending a DLE character. In the case of this exercise, it will prepend DLE before the first DLE to be transmitted and another before the ETX, therefore, the sequence of characters transmitted before the frame's CRC is this:

[DLE] [DLE] [DLE] [ETX] [ETX]

The last ETX marks the end of the payload field within the frame and implicitly, also marks the end of the frame since the next field, the CRC is of a fixed width (16 bits).

4. Calculate the even, 1's parity of each of the following two 7-bit blocks: 110 0100, 1100 000 (Slide no. 77)

Even 1's parity of 110 0100 = 1 xor 1 xor 0 xor 0 xor 1 xor 0 xor 0 = 1
Even 1's parity of 110 0000 = 1 xor 1 xor 0 xor 0 xor 0 xor 0 xor 0 = 0

5. Assume a datalink protocol that uses bidimensional parity in which each 7-bit block is appended one bit of horizontal parity and, for each 6-byte block a 1-byte vertical parity is computed (The same scheme employed in the textbook example at slide no. 78). The data which bidimensional parity is to be calculated are the following:

1010100 1110000 0001110 0000000 1111111 1000001

Calculate the parity bit of each of the preceding six 7-bit blocks and, finally, calculate the vertical parity byte.

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

6. Solve the CRC calculation of the example in pg. 101 and 102 of the textbook, which is also explained in slide 92 and on. Checkpoint yourselves as you compute each row by checking you obtain the correct result per the table in slide 95.

We solved this CRC calculation in the B1 sessions held on Thursday and Friday, nevertheless, I want to note that once the transmitter has computed the CRC, it sends Message M alongside the CRC to the receiver: [M|CRC]. The receiver applies the CRC calculation algorithm to M, and instead of appending a number of 0's equal to the order of the generator, it appends the received CRC and, if the resulting CRC' is 0, then, it concludes that most probably no error took place which leads it to accept the frame and begin its processing.

7. A layer-2 protocol uses the CRC generator following polynomial: $C(x)=1+x+x^3$. We want to transmit the message $M(x)=x^3+x^2+1$.

Compute the CRC by following the guidelines in slide 92 and on. (Exam solution at this URL: http://paloalto.unileon.es/cn/CN-ExRefSol2013.pdf contains the solution so you can checkpoint the algorithm progress).