V 1.0 6th -May-2022

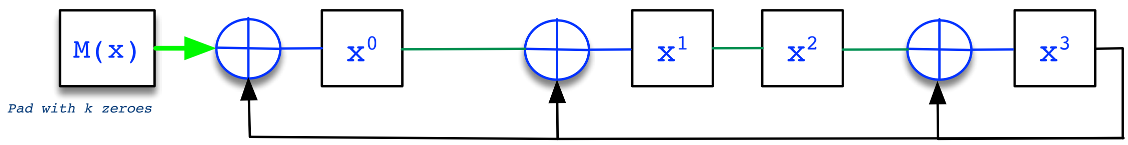
Universidad de León
Ingeniería Informática
Course on Computer Networks

B1 class for solving exercises on the board

Exercises similar to those included in this document can be found in paloalto.unileon.es/cn under heading titled "Weekly Homework while on lockdown in year 2020 (For reference only))"

Exercise 1. Consider the generator polynomial C(x) given by the bit-vector denoted in decreasing order, from left-to-right: (1,1,0,1,1). Check the resulting CRC circuit by feeding the following data bit-vector, again in decreasing order, from left-to-right: (1, 1, 0, 0, 1, 0).

Generator polynomial: $C(x) = 1 \cdot x^0 + 1 \cdot x^1 + 0 \cdot x^2 + 1 \cdot x^3 + 1 \cdot x^4$



M(x)	x^0	x^1	x^2	x^3
1	0	0	0	0
1	1	0	0	0
0	1	1	0	0
0	0	1	1	0
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
0	1	1	0	1
0	1	0	1	1
–	**1**	**0**	**0**	**0**

Exercise 2. Continuing from the preceding exercise, assume that the data bit-vector is sent to the receiver along with the calculated CRC (Data + Redundancy). The receiver must check the CRC for errors. The CRC-checking procedure applied by the receiver consists of:

a. Receive M(x), in our case the data bit vector which size is 6 bits. Store M(x).

   We arrange the bits from M(x) in a column each of which bits will be feed in to the circuit as it evolves from each row to the next one.

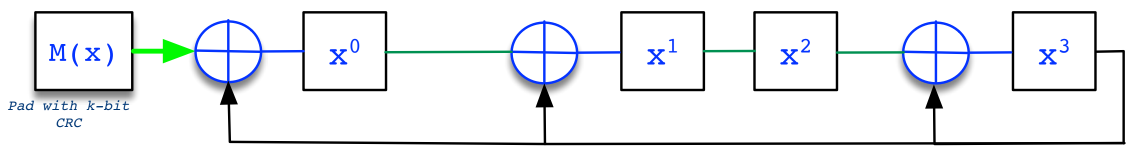
b. Receive C(x), the CRC computed by the sender and store it.

   As we did in the preceding question, C(x) is represented by having a xor gate's output be (Coefficient 1 of term) connected to the respective term's one-bit register or having the term's register directly connected to its preceding one-bit register.

c. Calculate the CRC by applying the procedure explained in the WebConference-lecture of 26th-March, taking into account, though, that instead of padding M(x) (The data polynomial) with as many zeroes as the order of C(x), you will have to pad the M(x) column with the bits from the received CRC. Pay attention not to invert the order of the CRC bits as you use each of them to pad M(x). If no error took place, then the new CRC that you are computing should yield all zeroes, *i.e.*, your CRC should be equal to (0,0,0,0) in this case.[1]

   Search the documents in paloalto.unileon.es/cn for an exercise similar to this.

Generator polynomial: $C(x) = 1 \cdot x^0 + 1 \cdot x^1 + 0 \cdot x^2 + 1 \cdot x^3 + 1 \cdot x^4$



M(x)	x^0	x^1	x^2	x^3
1	0	0	0	0
1	1	0	0	0
0	1	1	0	0
0	0	1	1	0

[1] Since the order of M(x) = 4, the order of the resulting CRC polynomial (The remainder of the integer division) must be = 4 -1 = 3; consequently, the resulting CRC should be equal to (0,0,0,0).

1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
0	1	1	0	1
1	1	0	1	1
–	**0**	**0**	**0**	**0**

In this case, the CRC = (0,0,0,0) which means that no error was *detected*.

Exercise 3. One of the stations that comprise an HDLC point-to-point link wishes to transmit the bit-string 101110110111110101010 to the station on the other end. Explain what the transmitter sends on the line and what the receiver's behavior is.

Let's assume that the transmitter has already sent a start of frame flag (01111110) and that a frame transmission is in progress. The sending upper-layer protocol writes the bit string 101110110111110101010 through HDLC service interface, then, since the written payload contains a sub string comprised of five bits 1, the HDLC transmitter circuit *stuffs* a bit 0 after the last bit 1 in the sub string: 1011101101111100101010. The receiver suppresses the *stuffed* 0 and proceeds with the reception of the ensuing bits.

Exercise 4. As for the Exponential Backoff algorithm which we introduced in the lecture:

a. What is Ethernet's *Channel Capture effect?*

The Channel capture effect represents the fact that when a number of hosts are involved in a collision, the host that has undergone the *least* number of collisions is the one which *most likely* will win the backoff.

b. An Ethernet interface E_1 has undergone 1 collisions in its attempt to transmit a frame A; at the same time, an Ethernet interface E_2 has undergone 3 collisions in its attempts to transmit frame B. After the last collision, both hosts execute *Exponential Backoff*, whereby we ask you to compute the probability that E_1 wins the backoff.

E_1 1 collision; $2^1 = 2$ ---> {0,1}
E_2 3 collisions; $2^3 = 8$ ---> {0,1,2,3,4,5,6,7}
All cases card(E_1 x E_2) = 2 x 8 = 16, this is the number of possible cases

- E_2 only wins the backoff when it obtains a 0 and E_1 obtains 1
prob(E_2 wins) = 1/16 = 0,062500

- prob(E_1 and E_2 obtain the same value) = prob(ties)
prob(E_1 gets 0 and E_2 gets 0 | E_1 gets 1 and E_2 gets 1) =
(1 + 1)/16 = 0,125

- prob(E_1 wins) = 1 – (prob(E_2 wins) + prob(ties)) = 1 - 0,0625 - 0,125
prob(E_1 wins) = 0,8125

Notice, this is a clear example of the *channel capture effect.*

c. Now, compute the probability that another collision takes place.

A new collision will happen if both senders get the same value (ties):

prob(E_1 and E_2 obtain the same value) = prob(ties)
prob(E_1 gets 0 and E_2 gets 0 | E_1 gets 1 and E_2 gets 1) =
(1 + 1)/16 = 0,125

d. Last, compute the probability that the backoff time generated by E_2 be greater than or equal to 102,4 μs.

Assume generated random number = r;
Backoff time = 51,2μs x r >= 102,4μs; r >= 102,4μs/51,2μs = **2**

$P_{r>=2}$ = p{2,3,4,5,6,7} /p{0,1,2,3,4,5,6,7} = 6/16 = 0,375

Exercise 5. Consider the extended LAN in fig. 1. Solve the following exercises:

a. Develop the evolution of the forwarding tables of all the switches as the following transmissions take place:

1. Ha sends a frame to Hg

   B0 learns Ha. All switches flood this frame since they haven't learned Hg, yet; then all switches learn Ha

2. Ha sends a new frame to Hg

   B0 learns Ha. All switches flood this frame since they haven't learned Hg, yet; then all switches learn Ha

3. H_c sends a frame to H_a

   B2 learns Hc
   Since all switches already learned Ha, the frame travels from Hc to B2, then to B1, then to B0 and finally to Ha, that is, no flooding

4. H_e sends a frame to the broadcast address

   B3 learns He; since the dest MAC is ethernet broadcast, all switches flood the frame, consequently, all switches learn He

5. H_b sends a frame to the broadcast address

   B0 learns Hb; since the dest MAC is ethernet broadcast, all switches flood the frame, consequently, all switches learn Hb

6. H_d sends a frame to H_e

   B2 learns Hd; since all switches learned He in step 4, frame travels through this path: Hd -> B2 -> B1 ->B3 -> He

b. Host H_b sends a frame which SRC MAC is that of H_g. Explain how H_b can do this assuming that it is running a Linux stack and have the forwarding tables updated after the said frame sending by H_b.

· Use a PF_PACKET/SOCK_RAW socket
· Bridge B0 learns Hg at port 2. Switches standing on the path to destination learn Hg

c. Now, H_a sends a frame to H_g. Update the forwarding tables and explain which hosts receive that frame.

· Bridge B0 learns Ha.
· B0 forwards frame onto port 2 according to the learning done on the preceding step

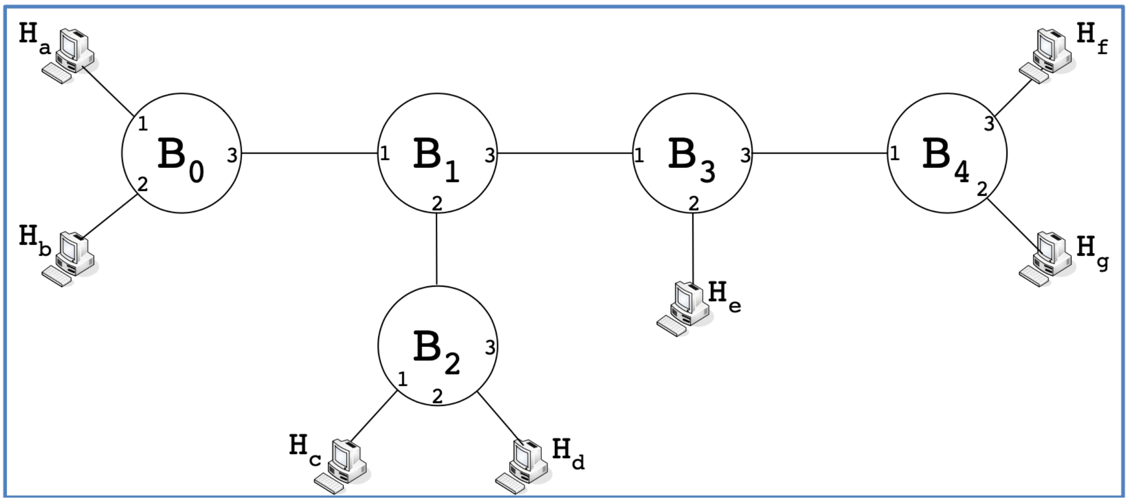


Figure 1. Extended LAN