EE450: Midterm Solutions

1. T, F, F, T, T, T, F, F, T, F, T, T, T, T, F, F, F, T, T, T, F, T, F, F, F

2. 1.3×10^5m, 4.3×10^5bps, 4KHz, 10111110111111, 20Ksec, 40bits/frame, 1Mbps, 25K frames/sec, 500K slots/sec, 294.1Kbps, 9bits, 10bits, 200Kbps, 4.82×10^4 bps, 1.6022 sec, 1.2022 sec, 1.2022 sec, 1.0022sec, 2-Sockets, 7-Sockets

3.

a. FCS bits, 011, Transmitted pattern: 101110**011**. The red bits are the FCS bits.
b. Received sequence = 101110**011** ⊕ 110010100 = 011100111. Receiver will divide this pattern by the generator pattern and observe a zero remainder. The receiver was **NOT** able to detect the error. According to the receiver, the transmitted sequence was indeed 011100111 and hence he will decode the message sequence as 011100. Of course, he is wrong. **Note the receiver does NOT know the error sequence. He only observes the received sequence.**
c. Received sequence = 110010101. When we divide this sequence by the generator pattern, the reminder is 1 which is **NOT** 0 and hence the receiver is able to detect the error (The receiver does not know how many errors or where are they located). There were 5 errors in the received sequence.

4.

Throughput = 5000/11 = 454.4 bps

Link Utilization = 5/11 = 45.4%

Time	Action @ Transmitter	Action @ Receiver	Time
0	F_o is transmitted		0
1	F_1 is transmitted. This frame is Lost		1
2	F_2 is transmitted	F_o is received (No errors), ACK_1 is returned	2
3	ACK_1 is received. Window slides by one unit and F_3 is transmitted		3
4	Sender window is closed	F_2 is received. It is out of order (R/x was expecting F_1), hence dropped	4
5	F_1 is timed out and is retransmitted.	F_3 is received. It is out of order (R/x was expecting F_1), hence dropped	5
6	F_2 is retransmitted		6
7	F_3 is retransmitted	F_1 is received and ACK_2 is returned	7
8	ACK_2 is received. Sender window slides by one unit. F_0 (the new one) is transmitted.	F_2 is received and ACK_3 is returned. This ACK is lost in the channel	8
9	T/x ran out of frames	F_3 is received and ACK_0 is returned	9
10	ACK_0 is received	F_0 is received and ACK_1 is returned	10
11	ACK_1 is received. The End!		11

5.

Step	Action	Delay (sec)
1	Client DNS contact the Local DNS server to obtain address of Web Server	0
2	Local DNS server contact the RNS for the requested IP address (Round Trip)	1
3	Local DNS server contact the TLD for the requested IP address (Round Trip)	1
4	Local DNS server contact the Authoritative for the requested IP address (Round Trip)	0.2
5	Local DNS server caches the IPO address and return to Client	0
6	Client set TCP connection (Hand Shaking) with the HTTP cache and send the request for the HTML file	0
7	The HTTP cache set up a TCP connection (Hand Shaking) with the real HTTP server. Here the HTTP cache is acting as a client on behalf of the real client.	0.2
8	HTTP requests the downloading of the HTML file	0.1
9	Downloading: 1G/1G + 1G/1M + 1G/1G + 0.1	1002.1
10	HTTP Caches the HTML file and send the file to the Client (1G/1G)	1
11	**Total**	**1005.6**
	The end-to-end throughput in this case is ~ 1 Mbps (which is the bottleneck) or more precisely = 1G/1005.6 = 0.9944 Mbps	
12	**For m2.a.com, it will take only 1 sec to download the same HTML file since both the IP address and the HTML file has been cached in Local DNS server and the Local HTTP cache. The End the end throughput in this case is 1 Gbps (since the propagation delay inside the LAN is negligible.**	