Name: Taylor Tang
SID: 1323782

Question 1:

- OSI model clearly defines the differences between services, interfaces and protocols, while TCP/IP does not. OSI model is theoretical and conceptual, it distinct these three concepts, which is critical important for software engineering in the designing phase.

- TCP/IP model is not as generalized as OSI model. It has a rigid scope in terms of protocols. In comparison, OSI model is generic and protocol independent, which provides more flexibilities when using OSI model. For example, forming a point-to-point network.

- OSI model clearly separates these data link layer and physical layer, while TCP/IP model combine these two layers and using it as an interface. Physical and data link layers are different on the concept, one is transmission characteristics at physical layer while the other is abstraction of data framing at link layer. A proper model should separate these two layers.

- Replacing protocol in TCP/IP can be difficult as the early protocol foundation was not built strongly nor following a strict guideline. On the other hand, OSI model offers to adapt various types of protocols. For example, OSI can replace the protocols to quickly adapt new technologies if needed.

- OSI model helps to standardize hardware such as router and switch.

Question 2:

4 x images
Each image resolution: 1024 x 256 pixels
Total pixels = 4 x (1024 x 256) = 1,048,576 pixels
Total bytes = 4 x 1,048,576 = 4,194,304 bytes
Total bits =. 8 x 4,194,304 = 33,554,432 bits

1) Over 56kbps modem:
   T-delay = 33,554,432 bits / 56,000 bit/s = 599.186 seconds
   P-delay = 6000 kms / 300,000 km/s = 0.020 second

   Latency = 599.186 + 0.020 = **599.206 seconds**

2) Over 1Mbps broadband link:
   Time = 33,554,432 bits / 1,000,000 bits/per second = 33.554 seconds
   P-delay = 6000 kms / 300,000 km/s = 0.020 second

   Latency = 33.554+ 0.020 = **33.574 seconds**

Question 3:

Nyquist limit:

Max data rate = 2B $\log_2 V$ bits/sec = 2 x 32000 x $\log_2 4$ = <u>128 kbit/s</u>

Shannon limit:

SNR of 20 dB = S/N = 100

Max data rate = B $\log_2(1 + \frac{S}{N})$ bits/sec = 32000 x $\log_2(1 + 100)$

= 32000 * 6.658 bits/sec

= <u>213 kbits/s</u>

In this case, since Nyquist's limit < Shannon's limit in term of max data rate, therefore the max data rate is **<u>128 kbit/s</u>**

Question 4:

n = 2^k – k – 1

Data bits (n)= 4 bits

Check bits (k) required = 3 bits

Put check bits in index P = [2^i] where i = 0, 1, 2 ….

In 7 bits data, we have P1, P2 and P4 as check bits index position.

Index	P1	P2	P3	P4	P5	P6	P7
Data							
Denote	Check bit	Check bit		Check bit			

Next, we fill in data bits **1010** in the rest of index position:

Index	P1	P2	P3	P4	P5	P6	P7
Data			**1**		**0**	**1**	**0**
Denote	Check bit	Check bit	Data bit	Check bit	Data bit	Data bit	Data bit

1st data bit index position is 3, equal to 011 in binary number, so it would be used in parity for P1 & P2, but not P4 (in the backward order of binary number 1 > 1 > 0).

Respectively, we have:
Next data bit at P5, 101 in binary, used in P1 and P4 (1 > 0 (P2 skipped) > 1).
Next data bit at P6, 110 in binary, used in P2 and P4 (0 > 1 > 1).
Next data bit at P7, 111 in binary, used in P1, P2 and P4 (1 > 1 > 1).

P1, P2 and P4 are parity bits, they are used to make the result an even number: Show in formula:
P1 + P3 + P5 + P7 = 0
P1 + 1 + 0 + 0 = 0
P1 = 1

P2 + P3 + P6 + P7 = 0
P2 + 1 +1 +0 = 0
P2 = 0

P4 + P5 + P6 + P7 = 0
P4 + 0 + 1 + 0 = 0
P4 = 1

Fill parity bit to table, now we have:

Index	P1	P2	P3	P4	P5	P6	P7
Data	1	0	**1**	1	**0**	**1**	**0**
Denote	Check bit	Check bit	Data bit	Check bit	Data bit	Data bit	Data bit

So, data sent is **1011010**

Question 5:

Original characters: C B H Z Z K K Z H B C
Flag Byte: Z

First, create a frame with provide characters, including Header and Trailer, assuming header and trailer does not include Z or ESC:

Header C B H Z Z K K Z H B C Trailer

Then, we add flag byte Z to both start and end of the frame:

Z Header C B H Z Z K K Z H B C Trailer Z

To avoid the Z inside the frame being recognized as flag bytes, we add ESC before each of Z in payload field:

Z Header C B H Esc Z Esc Z K K Esc Z H B C Trailer Z

On recipient's end, Z with no Esc in the front will be recognized as header. Any Esc in the payload field will be removed, then we will have the original frame.