Assignment 1
Internet Technologies COMP90007

Name:	Shromann Majumder
Student ID:	967530

April 2, 2023

Problem 1

Let's begin with calculating how many bits the given 0.5-second video clip consists of. We will take the product of the following units and their conversions to conclude in bits.

$$\begin{aligned}
&\text{second} \times \frac{\text{frames}}{\text{second}} \times \frac{\text{pixels}}{\text{frame}} \times \frac{\text{byte}}{\text{pixel}} \times \frac{\text{bits}}{\text{byte}} \\
&= 0.5 \times 30 \times (1280 \times 720) \times 3 \times 8 \\
&= 331776000 \text{ (bits)}
\end{aligned}$$

1.

Since the distance is 10000km, rate of transmission is 1Mbps with signal speed being 200000km/s, the latency is given by

$$\begin{aligned}
\text{Latency} &= \text{Transmission Delay} + \text{Propagation Delay} \\
&= \frac{\text{Message in Bits}}{\text{Rate of Transmission}} + \frac{\text{Length of Channel}}{\text{Speed of Signal}} \\
&= \frac{331776000 \text{ (bits)}}{10^6 \text{ (bits per second)}} + \frac{10000 \text{ (km)}}{200000 \text{ (km per second)}} \\
&= 331.826 \text{ (seconds)}
\end{aligned}$$

2.

Here, we have a satellite in Geostationary orbit that is 36000km away from both sender and receiver, giving us a total distance of $2 \times 36000 = 72000$km. The signal speed is 300000km/s and the rate of transmission is 100Mbps. Therefore the latency is given by

$$\begin{aligned}
\text{Latency} &= \text{Transmission Delay} + \text{Propagation Delay} \\
&= \frac{\text{Message in Bits}}{\text{Rate of Transmission}} + \frac{\text{Length of Channel}}{\text{Speed of Signal}} \\
&= \frac{331776000 \text{ (bits)}}{100 \times 10^6 \text{ (bits per second)}} + \frac{2 \times 36000 \text{ (km)}}{300000 \text{ (km per second)}} \\
&= 3.55776 \text{ (seconds)}
\end{aligned}$$

Problem 2

1.

With a channel bandwidth of 8kHz = 8000Hz and the data rate of 64kbps = 64000bps, we will use **Shannon's Theorem** to calculate the minimum signal-to-noise ratio (as it accounts for noise), then convert the ratio to **dB**.

$$\begin{aligned}
\text{Max Data Rate} &= B\log_2(1 + S/N) \\
\implies S/N &= 2^{(\text{Max Data Rate}/B)} - 1 \\
\implies \text{S/N in dB} &= 10\log_{10}(2^{(\text{Max Data Rate}/B)} - 1) \\
\text{S/N in dB} &= 10\log_{10}(2^{(\text{Max Data Rate}/B)} - 1) \\
&= 10\log_{10}(2^{(64000/8000)} - 1) \\
&= 24.0654 \text{ dB}
\end{aligned}$$

2.

Since we assume the data to be noiseless, we will use **Nyquist's Theorem** to identify the number of levels.

$$\begin{aligned}
\text{Max Data Rate} &= 2B\log_2(V) \\
\implies V &= 2^{\text{Max Data Rate}/2B} \\
&= 2^{64000/2\times 8000} \\
&= 16 \text{ (levels)}
\end{aligned}$$

Thus, we would send the data over using 16 levels using 4 bits (as $2^4 = 16$)

Problem 3

1	8	15	22	29	36	43	50
2	9	16	23	30	37	44	51
3	10	17	24	31	38	45	52
4	11	18	25	32	39	46	53
5	12	19	26	33	40	47	54
6	13	20	27	34	41	48	55
7	14	21	28	35	42	49	56

Figure 1: 32 bit block of data into fragments of Hamming(7, 4) codes
Red and Blue indicate Check and data bits respectively

1.

Hamming(7, 4) codes carry 4 data bits per fragment. Since we're trying to transmit 32-bit blocks of data, we need

$$\frac{\left(\frac{\text{number of bits}}{\text{block}}\right)}{\left(\frac{\text{number of } \textbf{data} \text{ bits}}{\text{fragment}}\right)} = \frac{32}{4} = 8 \text{ fragments}$$

Thus, we need 8 fragments of Hamming(7, 4) codes for each 32-bit block of data.

2.

One fragment is of size (4 data bits + 3 check bits) 7 bits. Therefore to get the total number of bits we transmit for each block

$$\frac{\text{Number of bits}}{\text{fragment}} \times \frac{\text{Number of fragments}}{\text{block}} = 7 \times 8 = 56 \text{ bits}$$

Thus, we have 56 bits for each 32-bit block of message.

3.

We build the fragments by filling in the bits vertically downwards and continue on to the next fragment vertically downwards until all 56 bits are in. Referring to figure 1, the columns represent fragments, and the data is filled in the sequence 1, 2, 3, ..., 55, 56. We can then send data by row; in the order 1, 8, 15, ..., 42, 49, 56. By interleaving this block transmission mode, we can make our system robust to sequences of errors. This way we can correct up to 8 errors; 1 per fragment/column.

Problem 4

Some benefits of a layered model are

- **Security**: A multi-layered architecture creates a more robust system, making it less vulnerable to network attacks.
- **Flexibility**: The ability to change protocols in a certain layer without affecting the operation of other layers.
- **Information hiding**: Network engineers don't need to know detailed information on the physical layer to work on improvements to another layer

Problem 5

1.

I have chosen `http://handbook.wikidot.com/en:http` as my HTTP-based website. The IP address of the source is `192.168.1.113` (my laptop) and the destination is `107.20.139.170` (the HTTP-based website).

We can validate the source IP address by running the `ipconfig`/`ifconfig` on Windows Command Prompt or MacOS Terminal respectively and observing and matching the IP address shown there and the Source IP address shown on WireShark.

We can validate the destination address (`107.20.139.170`) by typing its address on the URL form of any web browser and matching the retrieved web page with the web page retrieved by the URL `http://handbook.wikidot.com/en:http`.

2.

CONNECT primitive is regarding *establising a connection with a waiting peer*. We see this within the flow graph in figure 2 where my laptop (`192.168.1.113`) and the destination (`107.20.139.170`) conduct a 3-way handshake to establish a connection. This happens when my laptop sends a **SYN** to the destination as in the first line. Then on the second line, the destination sends a [**SYN, ACK**], which contains its own **SYN** and an **ACK** to acknowledge my laptops **SYN**. Finally, on the third line that is highlighted in blue, my laptop sends an **ACK** to acknowledge the destination's **SYN** from the second line and a connection is fully established.

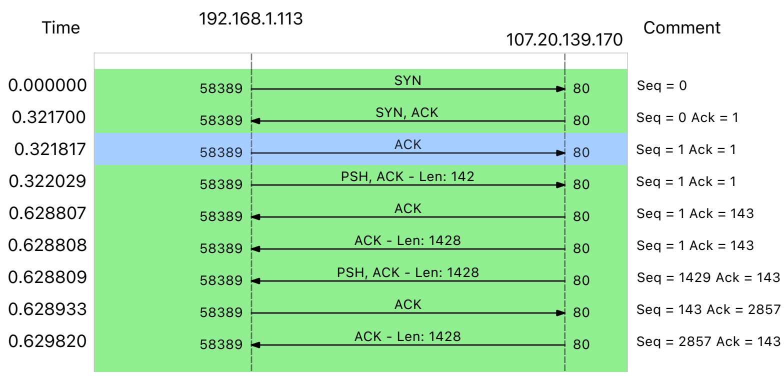

Figure 2: CONNECT

continued on next page...

DISCONNECT primitive is to *terminate a connection.* We see this within the flow graph as shown in figure 3. The disconnection process commences when my laptop sends a FIN to the destination on the fourth last line, which signals to the destination that my laptop wants to terminate the connection formed earlier. The destination then sends a **[FIN, ACK]**, which acknowledges the FIN sent by my laptop, and sends its own FIN. Then my laptop acknowledges the destination's **FIN** from the line above and sends a **FIN** to complete the disconnection. Finally, on the last and highlighted line, the destination acknowledges my laptop's last **FIN** from the line above and terminates the connection.

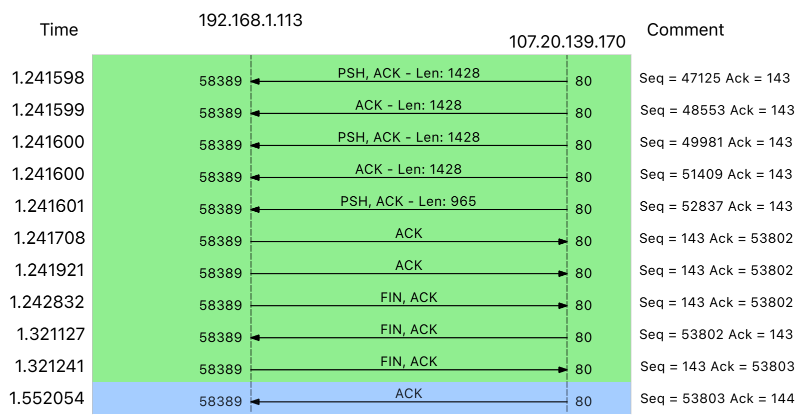

Figure 3: DISCONNECT