Aiden Duffy 251008412

Networking 4436 Exam 1

Problem 1

Q1

- The slow start mechanism built into TCP is a feature where once a connection begins, it increases the rate exponentially until the first loss event occurs. Therefore, the rate starts slowing and then rapidly builds.

Q2

- TCP offers the following services:
    - Point to point connections
    - Reliable and in-order byte stream
    - Pipelined to implement congestion control and flow control
    - Duplex data
    - Connection-oriented with handshaking

Q3

- Yes, TCP uses handshaking as it is connection oriented. By contrast, UDP does not use handshaking as it is not connection oriented

Q4

- HTTP is a stateless protocol as it does not retain any data between sessions. Each HTTP request is not associated with any other HTTP request, but simply regarded as their own thing, independent of all others.

Q5

- A reliable data transfer protocol is a crucial topic for networking, where data is transferred without any bit errors or packet loss.

Q6

- HTTP relies upon the lower-level TCP protocol.

Q7

- Pipelining protocols allow for the sending of “in-flight” packets even though prior packets have not yet been acknowledged. The process of pipelining greatly increases the utilization factor by a factor of n, if n is the number of in-flight packets.

Q8

- The AIMD mechanism is used to increase the maximum segment size slowly, but to quickly back off if any loss event occurs. This way, the connection will attempt to use the largest segment size possible, thereby achieving the quickest possible speeds. However, if there is a loss event, the multiplicative decrease allows it to scale back the MSS very quickly, so that another loss event is unlikely to occur. It then begins the additive increase process again.

Q9

- The port number is a 16-bit integer that is used to specify a particular location to send requests and data. Since many different processes could be communicating with a single IP address at a given time, the IP address has many ports to properly divide all its incoming traffic to their correct destinations.

Q10

- The five layers in the TCP/IP internet protocol stack are (from top to bottom): application → transport → network → data link → physical link

Q11

- The DASH over HTTP protocol is a technique for streaming media where a larger media is broken into segments. Each of these segments is available in many different bit rates (qualities) and served over HTTP via a manifest. The client machine will check with the manifest and request the highest bit-rate segment that its network can maintain.

Q12

- The two CDN server placement philosophies are to either enter deep, or to bring home.
    - Enter Deep: deploys a lot of server clusters in different access ISPs across the world
    - Bring Home: deploys a fewer number of large clusters in highly-populated areas near access ISPs, but not within them

Q13

- CDNs have many benefits including:
    - Faster content delivery
    - Ability to place content in servers close to the intended user base
    - No centralized server so the system has no single point of failure
    - More overall servers mean more content that a single CDN can provide (if you don't mind waiting)

Q14

- Different users watch different media. For example, Indian Bollywood movies are not very popular in North America, so a CDN would not need to store these movies in a cluster in North America. Instead, they can store those movies in clusters in India, closer to the users that use those media.

Q15

- Sequence numbers are required for reliable data transfer for each byte of data sent, so that the receiver can build the data in the correct order. There are also needed for the receiver to be able to identify whether any segments have been lost, as they will not have received the segment with a particular missing sequence number.

Q16

- Timers are needed for reliable data transfer so that the receiver does not wait indefinitely for a packet to arrive. Instead, by implementing a timer, if a required packet does not arrive in a specific amount of time, then it simply times out and requests the sender to resend the lost packet, instead of waiting indefinitely.

Problem 2

Q1

b.

Q2

b.

Q3

b.

Q4

a.

Q5

d.

Q6

b.

Q7

a.

Q8

b.

Q9

b.

Problem 3

Q1

4 bytes = 32 bits = 2^{32} sequence numbers

Since each sequence number can refer to a segment of 600 bytes, then the maximum size of L would be (2^{32}) / 600 = 7158278.827 bytes, which rounds up to 7158279 bytes.

Q2

Find total bytes to be transmitted: 2^{32} + header bytes = 2^{32} + header size * number of packets

= 2^{32} + (66 * 7158279) = 4,767,413,710 bytes

Transmission delay = (packet size / bit rate) * number of packets = total file size / bit rate

= 4767413710 bytes * 8 / 200 * 10^6 bps = 190.69 seconds

Therefore, under the given bandwidth, the 4.8GB file will take about 191 seconds, or just over three minutes.

Q3

In short, I do not recommend using MPTCP to transfer the enormous file of L bytes from above, and to instead stick to just regular WiFi (or cellular if that is quicker).

Advantages of MPTCP

- Two data streams ideally means up to double the data transfer rate
- Therefore, down to a minimum of half the data transfer time

Disadvantages and challenges of MPTCP

- How does the protocol decide to partition the file that is being sent?
    - If it splits it evenly, one of the two transfer channels will lag compared to the other unless their speeds are identical
- Segments can become extremely out of order depending on which segments are sent to which destination
- Reassembly of the file from two different destinations is very complex, and may not ever work correctly

Therefore, due to the additional challenges posed by MPTCP, I recommend to just use regular TCP to a single destination. Perhaps in the future when MPTCP's challenges have been removed, then it will be the universally better option.

Problem 4

Q1

Total file size = F

Packet size L = S + 60

Bit rates R_1, R_2 R_3, assume that R_1 is the limiting bandwidth, and so it is the time of each of the packets sent after the first packet sent.

$$T = \frac{L}{R}$$

$$T = \left(\frac{60+S}{R_1} + \frac{60+S}{R_2} + \frac{60+S}{R_3}\right) + \left(\frac{F}{S} - 1\right)\left(\frac{60+S}{R_1}\right)$$

Take the derivatives of each statement to find the minimum value of S

$$0=\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}+\left[\left(-\frac{F}{S^2}\right)\left(\frac{60+S}{R_1}\right)+\left(\frac{1}{R_1}\right)\left(\frac{F}{S}-1\right)\right]$$

$$0=\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}+\left[\left(-\frac{60F+FS}{S^2R_1}\right)+\left(\frac{F}{SR_1}-\frac{1}{R_1}\right)\right]$$

$$0=\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}+\left[\left(-\frac{60F+FS}{S^2R_1}\right)+\left(\frac{F}{SR_1}-\frac{S}{SR_1}\right)\right]$$

$$0=\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}+\left[\left(-\frac{60F+FS}{S^2R_1}\right)+\left(\frac{F-S}{SR_1}\right)\right]$$

$$0=\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}+\left[\left(-\frac{60F+FS}{S^2R_1}\right)+\left(\frac{FS-S^2}{S^2R_1}\right)\right]$$

$$0=\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}+\left[\frac{-60F-S^2}{S^2R_1}\right]$$

$$0=\frac{R_2R_3}{R_1R_2R_3}+\frac{R_1R_3}{R_1R_2R_3}+\frac{R_1R_2}{R_1R_2R_3}+\frac{-60F-S^2}{S^2R_1}$$

$$\frac{R_1R_2+R_2R_3+R_1R_3}{R_1R_2R_3}=\frac{60F+S^2}{S^2R_1}$$

$$S^2R_1\left(\frac{R_1R_2+R_2R_3+R_1R_3}{R_1R_2R_3}\right)=60F+S^2$$

$$S^2\left(\frac{R_1R_2+R_2R_3+R_1R_3}{R_2R_3}\right)-S^2=60F$$

$$S^2\left[\frac{R_1R_2+R_2R_3+R_1R_3}{R_2R_3}-1\right]=60F$$

$$S^2\left[\frac{R_1R_2+R_2R_3+R_1R_3-R_2R_3}{R_2R_3}\right]=60F$$

$$S^2\left[\frac{R_1R_2+R_1R_3}{R_2R_3}\right]=60F$$

$$S^2=\frac{60FR_2R_3}{R_1R_2+R_1R_3}$$

$$S=\sqrt{\frac{60FR_2R_3}{R_1R_2+R_1R_3}}$$

$$S=\sqrt{\frac{60FR_2R_3}{R_1(R_2+R_3)}}$$

Problem 5

R_{AB} = 6 Mbps

D_{prop} = 5 us/km

D_{AB} = 5000 km

L_{data} = 900 bytes

D_{Pdata} = 12 us

D_{BC} = 1000 km

L_{ACK} = 64 bytes

D_{PACK} = 6 us

D_{CD} = 100 km

A-B uses sliding window, 8 frames max

B-C uses sliding window, 4 frames max

C-D uses stop and wait, 1 frame max

Q1

Find Delay between A and B

$Delay_{data}$ = $delay_{proc}$ + $delay_{prop}$ + $delay_{trans}$ + $delay_{queue}$

$$D_{AB} = 12 * 10^{-6} + (5 * 10^{-6})(5000) + \frac{900 * 8}{6 * 10^{6}} + 0$$

$$D_{AB} = 0.026212$$

$$D_{BA} = 6 * 10^{-6} + (5 * 10^{-6})(5000) + \frac{64 * 8}{6 * 10^{6}} + 0$$

$$D_{BA} = 0.02509133333$$

Therefore, the RTT between nodes A and B is 0.0513 s, or 51.3 ms.

A then sends 8 packets every 51.3 ms, and B receives 8 packets every 51.3 ms.

Find R_{BC}

Window size for B – C is 4 frames.

The buffer at B is not flooded so long as it can have the same input rate as output rate. Therefore, B must send 8 packets out every 51.3 ms.

Assume that B – C sends 4 packets every 51.3/2 ms = 25.65 ms.

Use this 25.65 ms delay from B to C to find R_{RB}

$$D_{total} = D_{BC} + D_{CB}$$

$$0.02565 = \left(12 * 10^{-6} + (5 * 10^{-6})(1000) + \frac{900 * 8}{R_{BC}}\right) + \left(6 * 10^{-6} + (5 * 10^{-6})(1000) + \frac{64 * 8}{R_{BC}}\right)$$

$$0.02565 = 18 * 10^{-6} + 1 * 10^{-2} + \frac{900(8) + 64(8)}{R_{BC}}$$

$$0.02565 - 1.8 * 10^{-5} - 10^{-2} = \frac{900(8) + 64(8)}{R_{BC}}$$

$$R_{BC} = \frac{900(8) + 64(8)}{0.02565 - 18 * 10^{-6} - 10^{-2}}$$

$$R_{BC} = 493346.9806$$

Therefore, R_{BC} must be at least 493.346 Kbps, or ~0.494 Mbps

<u>Find R_{CD}</u>

Window size for C – D is 1 frame (stop and wait).

The buffer at C is not flooded so long as it can have the same input rate as output rate. Therefore, C must send 4 packets out every 25.65 ms.

Assume that C must send 1 packet out every 25.65/4 ms = 6.4125 ms.

Use this 6.4125 ms delay from C – D to find R_{CD}

$$D_{total} = D_{CD} + D_{DC}$$

$$0.0064125 = \left(12 * 10^{-6} + (5 * 10^{-6})(100) + \frac{900 * 8}{R_{CD}}\right) + \left(6 * 10^{-6} + (5 * 10^{-6})(100) + \frac{64 * 8}{R_{CD}}\right)$$

$$0.0064125 = 18 * 10^{-6} + 10 * 10^{-4} + \frac{900 * 8 + 64 * 8}{R_{CD}}$$

$$0.0064125 - 1.8 * 10^{-5} - 10^{-3} = \frac{900(8) + 64(8)}{R_{CD}}$$

$$R_{CD} = \frac{900(8) + 64(8)}{0.0064125 - 1.8 * 10^{-5} - 10^{-3}}$$

$$R_{CD} = 1429604.227$$

Therefore, R_{CD} must be at least 1429.604 Kbps, or ~ 1.43 Mbps

Q2

Find Buffer Size B_B

Buffer size = RTT * data rate

$$B_B = 5.13 * 10^{-2} * \ 493346.9806$$

$$B_B = 25308.7$$

Therefore, the buffer size at B should be 25.3 Kb

Find Buffer Size B_C

Buffer size = RTT * data rate

$$B_C = 2.565 * 10^{-2} * \ 1429604.227$$

$$B_C = 36669.3$$

Therefore, the buffer size at B should be 37.0 Kb

I recommend that the buffer sizes for nodes B and C be 25.3 Kb and 37.0 Kb respectively. This amount of storage will allow these nodes to hold the amount of data that the node can reliably send at the rate they are receiving the data.

Q3

If there are errors introduced into the links, then the buffer sizes that I recommend will change. Errors and packet loss will both mean that some packets may need to be retransmitted. Therefore, the nodes will have to resend packets, and so those packets must still be in the buffer to send, and not overwritten. I would recommend that nodes B and C therefore should have their buffers doubled, to account for having to resend packets or ACKs or both simultaneously. Therefore, I recommend buffer B holding 50.6 Kb and buffer C holding 74.0 Kb.

Problem 6

Q1

TCP slow start is operating at the interval of transmission 1 to 7, and again at the interval of transmission 8 to 13. These are the intervals when the window size is growing exponentially from one transmission round to the next.

Q2

TCP congestion avoidance is operating at the interval of transmissions 13 to 21 and 22 to 32. This is the interval when the window size either grows linearly.

Q3

The segment loss detected at round 7 is a timeout, as the MSS is reduced to 1.

Q4

The segment loss detected at round 21 is a triple duplicate ACK, as the MSS is cut in half.

Q5

After transmission round 7, the initial value of the Threshold is 32 segments. This is the size of the window when the TCP connections switches over to AIMD congestion control from slow start, as can be seen by the window size increasing linearly after transmission round 7.

Q6

After the first transmission round, the initial value of the Threshold is 64 segments. This is evidenced because the next Threshold value is 32 segments. When the TCP connection encountered a timeout during the slow start, the Threshold value was halved. Since the value after being halved is 32 segments, the original Threshold must have been 64 segments.

Q7

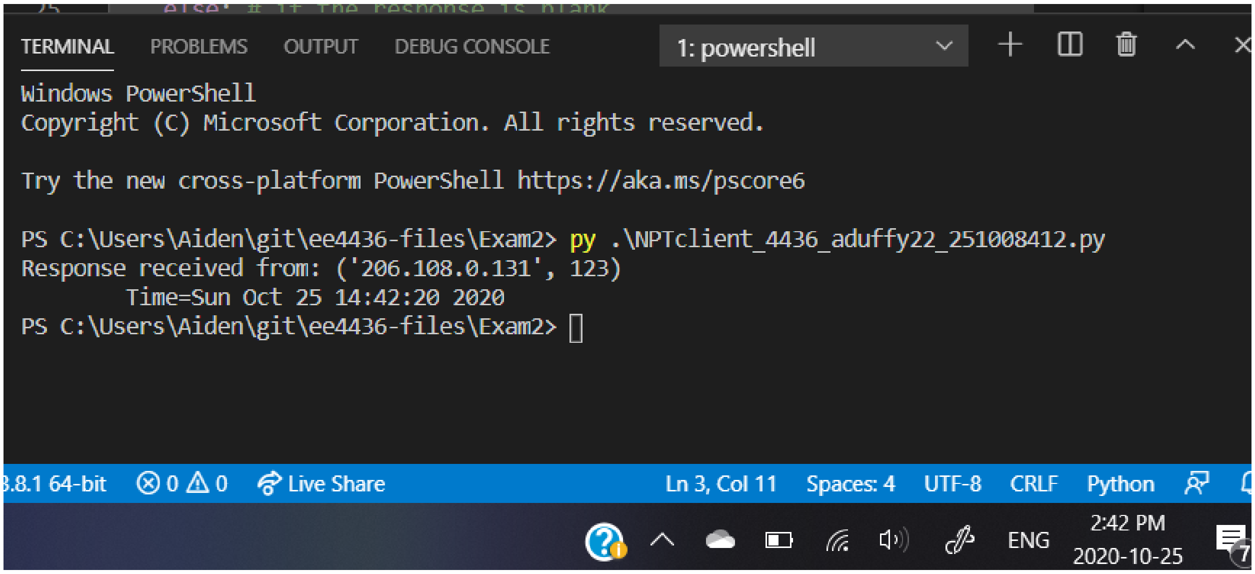
The 55th segment is sent in the 6th transmission round. After 1 round, 1 segment was sent. After 2 rounds, another 2 segments were sent, for a total of 3. After 3 rounds, another 4 segments were sent, for a total of 7. After 4 rounds, another 8 segments were sent, for a total of 15. After 5 rounds, another 16 segments were sent, for a total of 31. After 6 rounds, another 32 segments were sent, for a total of 63 (therefore including segment #55).

Q8

If a packet loss from a triple duplicate ACK is detected after the 32nd transmission round, then the congestion window size will be halved from 30 segments to 15 segments, in accordance with AIMD. Additionally, the Threshold is set to half of the cwnd value right when the triple duplicate ACK occurs, indicating that the Threshold value is also 15 at this point in time.

Aiden Duffy 251008412

Problem 7



The program can be seen running in the snip above. It outputs the time received from the ntp server which perfectly aligns with my system time. See attached code files.