CS 118 - Homework 4

Kuruvilla Cherian, Jahan
UID: 104436427
Section 1A

October 26, 2016

Problem 1

a

Sequence numbers represent what byte of the overall data we want. Thus it can be calculated using the byte size of the segments sent before. Thus because $segment_1$ required sequence number 257, and took 100 bytes, $segment_2 = 257 + 100 =$ **357**, and the source port number stays the same as **34679** and the destination port number is **80** because it is still from $A \rightarrow B$.

b

The acknowledgment number is the byte number that server is requesting, and so in this case, after receiving the first segment the ACK number from $B \rightarrow A$ would be **257**, while the source and destination port numbers would be switched since we are going in the opposite direction. Source port number is **80** and destination port number is **34679**.

c

The acknowledgment number is dependent of the order of retrieval, in which case the server would require the first segment's byte number which would be **257**.

d

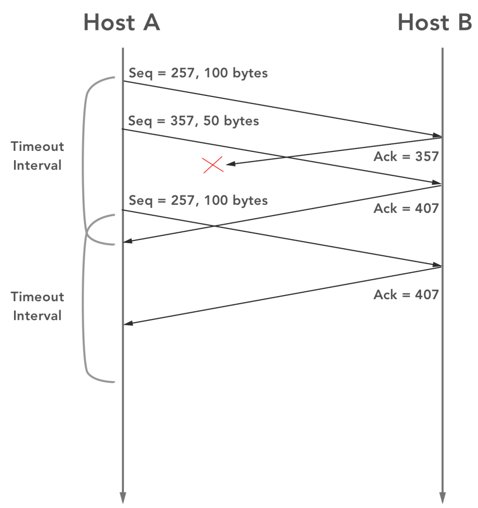

Figure 1: Time diagram representing the segment movement between hosts.

Problem 2

a

$$\mathbf{EstimatedRTT_1} = SampleRTT_1$$
$$\mathbf{EstimatedRTT_2} = (1-a)EstimatedRTT_1 + aSampleRTT_1$$
$$\rightarrow (1-a)SampleRTT_2 + aSampleRTT_1$$
$$\mathbf{EstimatedRTT_3} = (1-a)EstimatedRTT_2 + aSampleRTT_1$$
$$\rightarrow (1-a)^2 SampleRTT_3 + a(1-a)SampleRTT_2 + aSampleRTT_1$$
$$\mathbf{EstimatedRTT_4} = (1-a)EstimatedRTT_3 + aSampleRTT_1$$
$$\rightarrow (1-a)^3 SampleRTT_4 + a(1-a)^2 SampleRTT_3 + a(1-a)SampleRTT_2 + aSampleRTT_1$$

b

$$\mathbf{EstimatedRTT_n} = a\sum_{i=1}^{n-1}(1-a)^{i-1}SampleRTT_i + (1-a)n - 1SampleRTT_n$$
$$\rightarrow \frac{a}{1-a}\sum_{i=1}^{n-1}(1-a)^{i}SampleRTT_i + (1-a)^{n-1}SampleRTT_n$$

Problem 3

a

The MSS does not matter as the maximum size of the file that can be sent is dependent on the initial TCP sequence number and the size of the TCP packet, which is 32 bits. This leads to a total of $2^{32} = 4,294,967,296$ sequence numbers. However, because we start off our sequence numbers at $2^{16} - 1$, we can then only have $2^{32} - 2^{16} - 1 = 4,294,901,761 \approx 4.29Gb$ as the largest size of our file (L).

b

The number of segments we have is $(2^{32} - 2^{16} - 1)/536 \cdot 8 = 1,001,609.55247 \approx 1,001,610$. With 54 bytes per segment being added, we will have a total number of bits as

$$(2^{32} - 2^{16} - 1) + (54 \cdot 8 \cdot 1001610) = 4,727,597,281 bits$$

. This is transferred over a 100Mbps link. Thus we have the total time taken to transfer the file over TCP as follows:

$$4727597281/(100 \cdot 10^6) = 47.3s$$

Problem 4

a

Go-Back-N (GBN): Host A will send a total of 9 segments in total. This is because it will initially send segments 1, 2, 3, 4, 5 but then upon noticing that segment 2 was lost, it will re-transmit segments 2, 3, 4, 5 leading to a total of **9 segments**. Host B will send **8 ACKs**. This will be the first 4 sent for the sequence number 1, and 4 more for sequence numbers 2, 3, 4, 5.

Selective Repeat: Host A will send a total of **6 segments**. This is because it will initially send 1, 2, 3, 4, 5 and later re-send 2 upon noticing that it was lost. Host B will send **5 ACKs**, where the first 4 are with sequence numbers 1, 3, 4, 5 and then the last ACK for the sequence number 2.

TCP: Host A will send a total of **6 segments**. This is similar to Selective Repeat in that it will initially send 1, 2, 3, 4, 5 and then re-send segment 2. Host B will sent **5 ACKs**. This is 4

ACKs for sequence number 2 and 1 ACK for sequence number 6 as the expected number as per TCP's protocol.

b

TCP will be the winner in this case because of its *fast retransmit* which will send the lost segment based on a collection of ACK sequence numbers before the timeout from Host B.

Problem 5

a

There are two possibilities to consider in this situation. If we begin by assuming the receiver has received all packets until k (that is it has received packet $k-1$) and has successfully ACKed (that is the sender receives all these ACKs) all the preceding packets, then we see the sender's window advance to somewhere in the range $[k, k+(3-1)] \to [k, k+2]$.

The second case to consider is if the the ACKs were unsuccessfully transmitted in that the sender received none of them. Thus the sender's window will still contain $k-1$th packet and the 3 packets up to and including $k-1$. Thus the sender's window lies in the range $[k-3, k-1]$. Thus the final range of possible sets of sequence numbers insider the sender's window at time t is

$$[\mathbf{k-3}, \mathbf{k}] \to (k-3, k-2, k-1), (k-2, k-1, k), (k-1, k, k+1), (k, k+1, k+2)$$

.

b

If the receiver is waiting for some packet with sequence number k, then it can be safely assumed that it must have previously received and ACKed all the previous packets with sequence numbers $k-1, k-2$, and $k-3$. Because the sender has sent these packets (which could still be propagating back to the sender) then it means this sender must have received the ACK for $k-4$. Once the receiver would have sent an ACK for $k-4$ it will never send an ACK less than $k-4$ which means the possible values of the ACK field currently propagating back can be in the range of $[\mathbf{k-4}, \mathbf{k-1}]$.