Computer Networks HW#5

Submitted By: Neeha Hammad

Problem 5.1

(a)

- SEQ = 1030
- ACK= 3848
- F = ACK
- WIN = 4 000
- Data carried = 2230 - 1030 = 1200

The last sequence number (1030) represents the first byte in the segment. The acknowledgement number (2230) represents the next byte the sender expects to receive. So, subtracting gives us the data carried.

(b) Using an acknowledgement number, the receiver tells us that it has received data uptil this number and we should now send data from here onwards. If a packet is lost, it means there won't be any data transfer. Since the acknowledgement numbers are the same, we can conclude that the receiver received no data between segment 12 and 13.

Along with this, we can also observe that since the receiver reads from the buffer, the window size increases in the 13th segment. This suggests that the server can now accept more bytes (4000).

(c) If the TCP isn't performing properly, it's possible that multiple packets might get lost during the data transfer. If the sender does not know which exact packets were lost, the sender might resend all of them. This will be a waste of time and resources since it's possible that some of the packets resent were already received.

To avoid this problem, selective acknowledgements (SACK) can be used. Using SACK, the receiver can inform the sender about the packets that were received. Now the sender can check which ones got lost and retransmit only those ones.

The SACK block represents the range of the data bytes received. It has 2 numbers (32 bits) in 2 different sections: the Left Edge and the Right Edge. The left edge represents the first sequence number of a block while the right edge represents the sequence number immediately following the last sequence number of the block.

(d) In the 8th segment.

The acknowledgement number is the same between 6th and 7th segment so we know that a packet is lost here. The 8th ACK number is equal to the SEQ number of 6th so we know that data is being received till this point. We can conclude that the lost packet is re-sent by the 9th segment. We know this because the SEQ numbers of the 9th and 6th segments are the same.

Left edge: 3430
Right edge: 4630

(e) The client's TCP state machine will go to the TIME WAIT state since the client is initiating the tear down in the 14th segment by sending ACK, FIN flags. Once these flags are sent, the client enters the FIN_WAIT 1 state.

After receiving this, the server also sends ACK and FIN back to the client (15th segment). After doing so, the server enters the CLOSE_WAIT state. When the client receives this, it enters the FIN_WAIT 2 state. Finally, the client sends one last ACK in the 16th segment and enters the TIME_WAIT state.

When the server receives the last ACK, it enters the CLOSED state. Similarly, after a time out, the client also transitions into the CLOSED state.

Problem 5.2:

(a)
Average Data Rate = Maximum ACK / Time Interval
= 1300000 / 12
= 108333.333333 bytes/s

(b) The minimum receive window size is when the difference between the receive window size line (i.e the green line) and the ACK line is the least. This happens when t = 0. So,

Minimum: 300,000 - 0 = 300,000 bytes.

The maximum receive window size is when the difference between the receive window size line (i.e the green line) and the ACK line is the maximum. This can happen anywhere from t = 4 to t = 12 except where there are dents in the straight parallel lines. Let's take the values at t = 6.
Maximum: 900000-600000 = 300,000 bytes.

(c) At t = 12, 6 packets were lost and not fully transmitted.

If we look at the graph, before t = 12, some red and brown lines are placed on top of each other. These represent stacked up TCP segments, which are not yet combined and transmitted till the end. After t = 12, we see 6 small blue I-beams, which show that TCP segments are being received and acknowledged. As we can see, after this, the ACK values increase significantly.

(d) Let's look at the second graph for this evaluation.

By t = 0.5, TCP segments are already being sent since a connection was already established in the beginning. Uptil t = 1.1, TCP segments are sent very frequently. After that, the transfer rate slows down a bit and eventually becomes constant after t = 1.5.

When we stop seeing the blue I-beams at around t = 1.5, we know that some TCP segments are lost. From t = 1.5 onwards, we also see a lot of red & brown lines stacked on top of each other. These lines depict TCP segments stacking up as they cannot be combined or fully transmitted.

For several next seconds, the ACK line stays at a constant value. Later at around t = 2.85, we see that the stair like structure starts to reappear, thus representing the increasing ACK value. The blue I-beams on the surface of the green line depict the retransmitted TCP segments. After t = 3.3, we don't see any red/brown lines stacking up so we can assume that no more TCP segments are getting lost.

References:

Lecture Notes TCP State Machine Diagram
https://serverfault.com/questions/503345/three-duplicate-acknowledgements7

http://www.networksorcery.com/enp/Protocol/tcp/option005.htm

http://packetbomb.com/understanding-the-tcptrace-time-sequence-graph-in-wireshark/