

Lab 6

Q 1.

Answer is A: 1233.

Since the you're about to send 1234 but haven't yet, you have to assume 1233 is the highest sequence number that was been processed. I would have received this back as an ACK # from the receiver.

Q2.

Answer is 1: Yes.

It is possible that the receiver has received the byte number of 1335 if the window is large enough for many packets to be sent but the Seq # of 1234 had timed out, it may have received 1335 and be about to retransmit 1234.

Q3.

Answer is 2: Will increase timeout

With the information given the smooth value will change, this will change the Timeout Interval because of:

TimeoutInterval = SmoothedRTT + 4*DevRTT

1 is untrue because if the smooth changes, the timeout will change

3 is untrue because the smooth doesn't depend on the Deviation

4 is untrue because even the smooth change, it will change so it doesn't depend on Deviation

Q4.

Answer is: 132ms

Smooth: $101 = (0.875) * 100 + 0.125 * 108$, Alpha = 0.125

DevRTT_i: $7.75 = (0.75) * 8 + 0.25 * |108 - 101|$ Beta = 0.25

TimeoutInterval: $132 = 101 + 4 * 7.75$

Q5.

Answer is C: Flow Control

Number of bytes willing to accept

Q6.

Answer is B: 1.5RTT

Because of the 3 way handshake used for connection management, to establish a connection state, it requires an initial call to the receiver to ready it, this can be considered a half trip, then the receiver send back and the sender sends again with the packet making 3 trips.

Q7.

Answer is b: Triple duplicate ACKS

This is a mechanism to ensure that the whole window isn't retransmitted but only the lowest Seq # is retransmitted as it assumes everything before this is ACK'd and that everything after might not be reliable.

Q8.

Answer is True

If you underestimate the RTT then you might timeout during a standard sending just because it wasn't finished and lose information unnecessarily.

Q9.

Answer is SMTP, HTTP, FTP

Q10.

Answer is c: 0.8 seconds

$100,000/125,000 = 4/5 = 0.8 \text{ seconds}$

Q11.

Answer is C: 4

Client: 150,000 per second

Can only upload 125,000 per second so 25,000 goes into the buffer.

This builds by 25,000 bytes per second till it reaches it's maximum of 100,000 where it then will drop any so the answer should be 4 seconds before the first drop

Q12.

Answer is B: 40 pkts/s

Capacity of 100 pkts/s

RTT is 0.1 seconds

Fixed window of 4 means 4 allowed at a time so:

$0.1 \times 10 = 1$ second so $10 \times 4 = 40$ pkts