CS 118 Homework 4

Wenxuan Liu 805152602

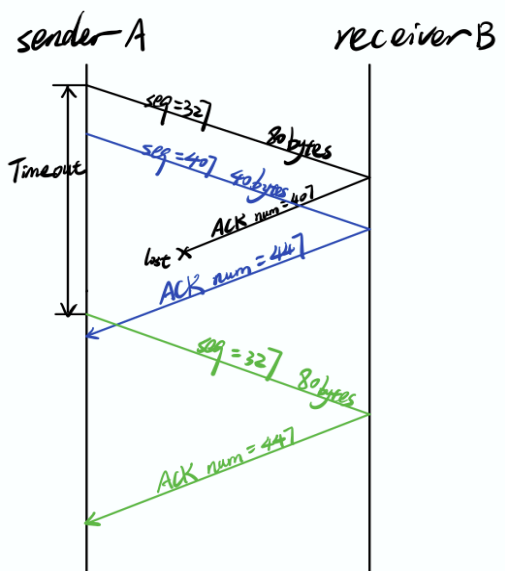
Problem 1

(a) In the second segment sent from Host A to B, the sequence number is 407, source port number is 40200, and destination port number is 80.

(b) If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, the ACK number is 407, the source port number is 80, and the destination port number is 40200.

(c) If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, the ACK number is 327.

(d)



Problem 2

(a)

For Go-back-N, the required buffer size at the receiver is **1**. Because Go-back-N doesn't need to put unordered packets(with unexpected sequence number) into buffer, but only put packet in order one at a time, so the buffer size is 1.

For Selective Repeat, the required buffer size is RWS=SWS=**23**. Because with the same window size as the sender, the receiver can put all unordered packets into buffer and monitor the sequence number from the sender and check whether the packet is new or duplicate.

(b)

(i) The minimum required SeqNumSize for sliding window scheme is **9** with SWS=6 and RWS=3. Because the seq number size should be at least SWS+RWS=6+3=9, so that the expected packets can rank from 1 to SWS for sender's sending packets and SWS+1 to SWS+RWS for buffer packets without confusion.

(ii) The minimum required SeqNumSize for Go-back-N scheme is **7** with SWS=6. Because the seq number size should be at least SWS+RWS=6+1=7, due to the same reason as above.

Problem 3

The formulas for EstimatedRTT, DevRTT and TimeoutInterval are:

$$EstimatedRTT = \alpha * \text{SampleRTT} + (1 - \alpha) * \text{ExtimatedRTT}$$
$$DevRTT = \beta * |SampleRTT - EstimatedRTT| + (1 - \beta) * DevRTT$$
$$TimeoutInterval = EstimatedRTT + 4 * DevRTT$$

So, after SampleRTT of 106ms, 120ms, and 150ms, the EstimatedRTT, DevRTT, and TimeoutInterval are as followed:

(1) After SampleRTT 106ms:
EstimatedRTT = 0.125*106+0.875*100 = 100.75ms
DevRTT = 0.25*(106-100.75) + 0.75*5 = 5.06ms
TimeoutInteral = 100.75+4*5.06 = 121ms

(2) After SampleRTT 120ms:
EstimatedRTT = 0.125*120+0.875*100.75 = 103.15ms
DevRTT = 0.25*(120-103.15) + 0.75*5.06 = 8ms
TimeoutInteral = 103.16+4*8 = 135.15ms

(3) After SampleRTT 150ms:
EstimatedRTT = 0.125*150+0.875*103.15 = 109ms
DevRTT = 0.25*(150-109) + 0.75*8 = 16.25ms
TimeoutInteral = 109+4*16.25 = 174ms

Problem 4

(a) Assume that the sequence number for the first segment being sent from sender to receiver is 1, and each of the following segment's sequence number all increment by 1. The ACK number for segment m is m.

For Go-Back-N:

The sender needs to first send all the 10 segments to the receiver, and then resend the 5 segments from 6 to 10 given that segment 6 is lost. So the there are 10+5=**15** segments being sent by Host A. The sequence number of segments sent by Host A are: **1,2,3,4,5,6,7,8,9,10,6,7,8,9,10.**

Host B first sends ACK for the first 5 segments that it successfully received, and then it sends ACK for the fifth one for 4 more times given that 6 was lost, and there are 4 more unordered packets 7,8,9,10 received by B. And then it sends the later received 5 segments. So, there are **5+4+5=14** ACKs sent by Host B. The ACK number that B sends are: **1,2,3,4,5,5,5,5,5,6,7,8,9,10.**

For Selective Repeat:

The sender needs to first send all the 10 segments to the receiver, and then resend the segment 6 given that it was lost. So, there are 10+1 = **11** segments sent by Host A, and the sequence number of them are: **1,2,3,4,5,6,7,8,9,10,6.**

The receiver first sends 9 ACKs for all the segments except segment 6. Then, when segment 6 was resent to the receiver, the receiver sends ACK for packet 6. So there are 9+1=**10** ACKs sent by Host B. The ACK number of them are: **1,2,3,4,5,7,8,9,10,6.**

For TCP:

The sender needs to first send all the 10 segments to the receiver, and then resend the segment 6 given that it was lost. So, there are 10+1 = **11** segments sent by Host A, and the sequence number of them are: **1,2,3,4,5,6,7,8,9,10,6.**

The receiver sends 5 ACKs for the first 5 received segments, 4 ACKs '6' for the remaining 4 segments after lost segment 6, and 1 ACK for the ending with seq number 11. So, there are

5+4+1=10 ACKs sent by Host B. The ACK number of them are: **2,3,4,5,6,6,6,6,6,11.**

(b)

TCP protocol can successfully deliver all 10 data segments in shortest time interval. Because comparatively TCP transmits the least number of segments from sender to receiver, and TCP fast-transmit doesn't need to wait for the long-time timeout, but resends the lost packet immediately after receiving the corresponding ACK.

Problem 5

(a)

HTTP doesn't use timer. Because it indirectly relies on its TCP connection with the server to use timeout when a host sends a request and didn't receive a response for a long time.

(b)

DNS doesn't use timer for DNS query, because it mainly replies on UDP for transport layer, which doesn't implement timer by default. However, the DNS caching on the local DNS resolver has a timer which records the TTL(Time to live) for each DNS record. Once the TTL expires, the DNS record will be removed in the caching.

(c)

TCP uses timer to trigger retransmission. At the moment of a segment being sent from sender to receiver, the timer is set on, and it would then timeout if after the period of time an ACK from receiver for that packet is not received by sender. Then, the sender will consider the packet being lost and retransmit it, so that TCP becomes reliable transport layer protocol without packet loss.

(d)

UDP doesn't use timer. Because UDP is unreliable transport layer protocol, and UDP doesn't guarantee that there is no packet loss. In order to get the greatest efficiency, UDP doesn't use ACK and doesn't implement retransmission mechanism. So it doesn't have corresponding timer.