# Problem 1

Host A and B are communicating over an already established TCP connection. For simplicity, we assume that this TCP connection does not use delayed ACK and Host B sends an acknowledgment whenever it receives a segment from Host A. Following TCP's convention, ACK is cumulative and ACK number is the next expected byte from the sender.

Suppose at the beginning, Host A has sent data up through byte 233, and Host B has successfully received all the data. Host A then sends two segments to Host B back-to-back. The first and second segments contain 60 and 40 bytes of data respectively. In the first segment, the sequence number is 234, the source port number is 45213, and the destination port number is 8081. Fill in the blanks for questions (a)–(c) directly; work out the diagram in the box for question (d).

(a) In the second segment sent from Host A to B, the sequence number is 334, the source port number is 45213, and the destination port number is 8081.

(b) If the first segment arrives before the second segment on Host B, in the acknowledgment of the first arriving segment, the ACK number is 294, the source port number is 45213, and the destination port number is 8081.

(c) If the second segment arrives before the first segment on Host B, in the acknowledgment of the first arriving segment, the ACK number is 234.

(d) Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after A's timeout intervals for both the first and the second packets.

Draw a timing diagram in the box below, showing these segments and acknowledgments until A receives all the acknowledgments of re-transmitted packets. Assume no additional packet loss. For each segment in your diagram, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the ACK number.

(a) In the second segment sent from Host A to B, the sequence number is 334, the source port number is 45213, and the destination port number is 8081.

(b) If the first segment arrives before the second segment on Host B, in the acknowledgment of the first arriving segment, the ACK number is 294, the source port number is 45213, and the destination port number is 8081.

(c) If the second segment arrives before the first segment on Host B, in the acknowledgment of the first arriving segment, the ACK number is 234.

(d)

# Problem 2

For a TCP connection, suppose at time $T_0$, the EstimatedRTT and DevRTT are 130 ms and 12 ms respectively. The next three measured SampleRTT values after $T_0$ are 140 ms, 120 ms, and 150 ms. Following the algorithm in the lecture slides, compute the following values after **each** of the three SampleRTT measurements is obtained in sequence:

- DevRTT
- EstimatedRTT
- TimeoutInterval

Round your calculation results to two decimals (e.g., 123.45).

Given that at $T_0$, EstimatedRTT = 130ms, DevRTT = 12ms, $\text{SampleRTT}_{1,2,3}$ = 140ms, 120ms, 150ms and assuming $\alpha = \frac{1}{8}$, $\beta = \frac{1}{4}$, we get

$$\text{DevRTT} = (1-\beta)\cdot\text{DevRTT} + \beta\cdot|\text{SampleRTT}_1 - \text{EstimatedRTT}|$$
$$\text{EstimatedRTT} = (1-\alpha)\cdot\text{EstimatedRTT} + \alpha\cdot\text{SampleRTT}$$
$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4\cdot\text{DevRTT}$$

1. *(i)* $\text{DevRTT} = \frac{3}{4}(12) + \frac{1}{4}\cdot|140-130| = 2.5 + 9 = 11.5\text{ms}$
   *(ii)* $\text{EstimatedRTT} = \frac{7}{8}(130) + \frac{1}{8}(140) = 113.75 + 17.5 = 131.25\text{ms}$
   *(iii)* $\text{TimeoutInterval} = 131.25 + 4(11.5) = 177.25\text{ms}$
2. *(i)* $\text{DevRTT} = \frac{3}{4}(11.5) + \frac{1}{4}\cdot|120-131.25| = 8.63 + 2.81 = 11.44\text{ms}$
   *(ii)* $\text{EstimatedRTT} = \frac{7}{8}(131.25) + \frac{1}{8}(120) = 114.84 + 15 = 129.84\text{ms}$
   *(iii)* $\text{TimeoutInterval} = 129.84 + 4(11.44) = 175.6\text{ms}$
3. *(i)* $\text{DevRTT} = \frac{3}{4}(11.44) + \frac{1}{4}\cdot|150-129.84| = 8.58 + 5.04 = 13.62\text{ms}$
   *(ii)* $\text{EstimatedRTT} = \frac{7}{8}(129.84) + \frac{1}{8}(150) = 113.61 + 18.75 = 132.36\text{ms}$
   *(iii)* $\text{TimeoutInterval} = 132.36 + 4(13.62) = 186.84\text{ms}$

# Problem 3

You are designing a reliable, sliding window, byte-stream protocol similar to TCP. It will be used for communication with a geosynchronous satellite network, for which the bandwidth is 1 Gbps ($1G = 10^9$) and the RTT is 200 ms. Assume the maximum segment lifetime is 10 seconds.

(a) What is the minimum number of bits you should use for ReceiveWindow and SequenceNum fields so that full utilization of the channel can be achieved? (Hints: The ReceiveWindow should be able to encode the maximum possible in-flight number of bytes. SequenceNum should be large enough so that the sequence number does not wrap around when the delayed segment is still in the channel.)

(b) If ReceiveWindow is 16 bits, what upper bound would that impose on the effective bandwidth?

10s = 10000ms, bandwidth = 1 $\times 10^6$ bits/ms, RTT = 200ms

(a) $\text{ReceiveWindow} \geq \log_2(200 \times 10^6) = \log_2(2 \times 10^8 \text{ bits}) = \log_2(2.5 \times 10^7 \text{ bytes}) = 25$ bits.
$\text{SequenceNum} \geq \log_2(10 \times 10^9) = \log_2(1 \times 10^{10} \text{ bits}) = \log_2(1.25 \times 10^9 \text{ bytes}) = 31$ bits.

(b) If ReceiveWindow is 16 bits, we have an upper bound of:
$\text{EffectiveBandwidth} = \frac{\text{ReceiveWindow}}{RTT} = \frac{2^{16}}{200} = 327.68$ KB/s

# Problem 4

In this question you are comparing among the following three schemes:

- Go-Back-N: cumulative ACK is used. The ACK number is the sequence number of corresponding segment (Figure 3.22 in the textbook).
- Selective Repeat: individual ACK is used. The ACK number is the sequence number of corresponding segment (Figure 3.26 in the textbook).
- TCP: cumulative ACK is used with no delayed ACK. The ACK number is the next expected sequence number.

Assume that timeout values for all three protocols are sufficiently long, such that 8 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A), respectively.

Suppose Host A sends 8 data segments to Host B, and the 5th segment (sent from A) is lost. No other segments are lost. In the end, all 8 data segments have been correctly received by Host B. The first segment starts from sequence number 1 and each segment contains 1 byte of data. The round trip time is greater than the transmission time for the 8 segments, so that the first ACK arrives after all 8 segments have been transmitted. For TCP, assume that the connection is already established and you don't need to count the connection establishment.

(a) How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.

(b) If the timeout values for all three protocols are very long (e.g., longer than several RTTs), then which protocol successfully delivers all 8 data segments in shortest time interval?

We will assume window size is 8.

(a) *(i)* Host A sends 12 segments in total and Host B sends 12 ACKs in total. Their sequence numbers are: $A = \{1, 2, 3, 4, 5, 6, 7, 8, 5, 6, 7, 8\}$ and $B = \{1, 2, 3, 4, 4, 4, 4, 4, 5, 6, 7, 8\}$

*(ii)* Host A sends 9 segments in total and Host B sends 8 ACKs in total. Their sequence numbers are: $A = \{1, 2, 3, 4, 5, 6, 7, 8, 5\}$ and $B = \{1, 2, 3, 4, 6, 7, 8, 5\}$

*(iii)* Host A sends 9 segments in total and Host B sends 2 ACKs in total. Their sequence numbers are: $A = \{1, 2, 3, 4, 5, 6, 7, 8, 5\}$ and $B = \{5, 9\}$

(b) If the timeout values for all three protocols are very long, then TCP with cumulative and no delayed ACK will successfully deliver all 8 data segments in the shortest time interval.

# Problem 5

Consider the evolution of a TCP connection with the following characteristics. Assume that all the following algorithms are implemented in TCP congestion control: slow start, congestions avoidance, fast retransmit and fast recovery, and retransmission upon timeout. If ssthresh equals to cwnd, use the slow start algorithm in your calculation.

- The TCP sender sends a number of data segments of 1 byte each.
- The TCP receiver follows cumulative ACK convention without delayed ACK, i.e. the receiver acknowledges every successfully received segment immediately.
- The RTT is 100 ms for all transmissions, consists of the network latency of 60 ms in sending a segment (header and payload) from the sender to the receiver and 40 ms in sending an acknowledgment (header only) from the receiver to the sender. Ignore packet-processing delays at the sender and the receiver.
- To simplify the setting, rather than estimating RTO from sampled RTT values, the RTO at the send is set to 500ms and does not change during the connection lifetime.
- Initially ssthresh at the sender is set to 6. Assume cwnd and ssthresh are measured in segments, and the transmission time for each segment is negligible.
- The connection (already established) starts to transmit data at time t = 0, and the initial sequence number starts from 1. TCP segment with sequence number 6 is lost once (i.e., it sees segment loss during its first transmission). No other segments are lost during transmissions.

What are the values for cwnd and ssthreshold when the sender receives the TCP ACK with number 15? Show your intermediate steps or your diagram in your solution.

**Note:** For simplicity, we assume that TCP sender does NOTHING when receiving 1st and 2nd duplicate ACKs, following the convention in the textbook, rather than the optimized behaviors described in RFC 5681, Section 3.2.

Problem 5 continued on next page...