Prof. Athina Markopoulou
Spring 2016

Intro to Computer Networks
EECS148/CS132

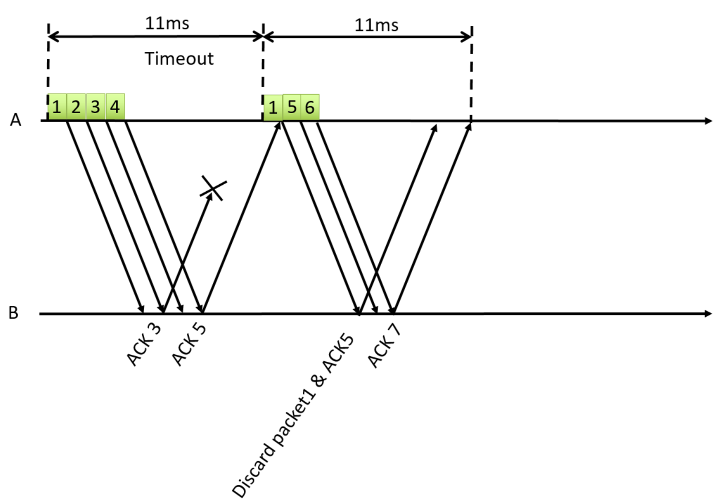
Homework 3 Solution Sketch

Topic: TCP

1. *(30 Points)* **TCP Reliability.**

   Consider the following scenario. *(Yes, the same as in the midterm, but now it is TCP instead of GBN or SR. This is also an idealized TCP in that you should ignore congestion and flow control.)* For simplicity, assume that each TCP packet contains only 1Byte of data, and that the initial TCP sequence number is 1. All parameters are the same as in the midterm problem: 6 packets to send, window size = 4 bytes (ignore TCP congestion and flow control), timeout=10ms (starting *after* a packet is sent), transmission delay = 1 ms for one packet, 0 for the ACK) and one-way propagation delay = 4ms. Suppose that ACK No. 2 is lost and no other packet is lost. The beginning of the space-time diagram is provided below.

   - Fill out the rest of the diagrams until all 6 packets are sent and acknowledged. Show all packets (transmissions, retransmissions and acknowledgements), their sequence numbers and the times they were sent/received.
   - What is the total time until all 6 packets are sent and acknowledged?



Answer:

The answer is shown on the above figure, considering the full TCP sender and receiver as described in Section 3.5 (except for flow control) but not 3.6 (congestion control). Upon receiving segment 1, the receiver does not send ACK2, because of delayed ACK. Upon reception of segment 2, the receiver sends a cumulative ACK3. However, ACK3 is lost mid way. Upon reception of segment 3, the receiver delays the ACK. Upon reception of segment 4, the receiver sends a cumulative ACK5 (at time 8ms), which is received at the sender at time 12ms. Meanwhile, the TCP sender starts a timer after sending segment 1 (at time 1ms), which expires at time 1ms+10ms=11ms, i.e, before receiving any packet. Therefore, at time 11ms, the TCP sender starts retransmitting the un-ACKed packets, starting from segment

1. However, at time 12ms, ACK 5 is received and cumulatively acknowledges all previous packets. The sending window slides forward, thus packets 5 and 6 can be transmitted. The retransmitted segment 1 is ignored at the receiver, as it is out of sequence, and the a delayed ACK7 is sent, upon receiving segment 6. The total time until all 6 packets are sent and acknowledged is 22ms, as shown on the figure.

2. *(40 Points)* **TCP Congestion Control.**

   (a) *(20 Points)* Chapter 3, Problem 40 *(Yes, the same as in the discussion!)*

   a) TCP slowstart is operating in the intervals [1,6] and [23,26]

   b) TCP congestion advoidance is operating in the intervals [6,16] and [17,22]

   c) After the 16th transmission round, packet loss is recognized by a triple duplicate ACK. If there was a timeout, the congestion window size would have dropped to 1.

   d) After the 22nd transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.

   e) The threshold is initially 32, since it is at this window size that slowtart stops and congestion avoidance begins.

   f) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.

   g) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 22, the congestion windows size is 29. Hence the threshold is floor of 29/2=14 during the 24th transmission round(if you got roof of 29/2=15 that would be OK too).

   h) During the 1st transmission round, packet 1 is sent; packet 2-3 are sent in the 2nd transmission round; packets 4-7 are sent in the 3rd transmission round; packets 8-15 are sent in the 4th transmission round; packets 16-31 are sent in the 5th transmission round; packets 32-63 are sent in the 6th transmission round; packets 64 96 are sent in the 7th transmission round. Thus packet 70 is sent in the 7th transmission round.

   i) The congestion window and threshold will be set to half the current value of the congestion window (8) when the loss occurred. Thus the new values of the threshold and window will be 4.

   j) Threshold is 21, and congestion window size is 4.

   k) Round 17, 1 packet; round 18, 2 packets; round 19, 4 packets; round 20, 8 packets; round 21, 16 packets; round 22, 21 packets. So, the total number is 52.

   (b) *(20 Points)* Chapter 3, Problem 45. *(You will prove the formula about average throughput of TCP for long-lived connections.)*

Answer:

a) Let's consider the steady state in congestion avoidance. The TCP sender window will increase linearly between $\frac{W}{2}$ and W, at which point, one segment will be lost and the window will decrease down to $\frac{W}{2}$ again, and the cycle will repeat again.
The total number of segments sent, during one cycle of additive increase and multiplicative decrease, is as follows:

$$S = \frac{W}{2} + (\frac{W}{2} + 1) + ... + W = \sum_{n=0}^{\frac{W}{2}} (\frac{W}{2} + n)$$

$$= (\frac{W}{2}+1)\frac{W}{2} + \sum_{n=0}^{\frac{W}{2}} n = (\frac{W}{2}+1)\frac{W}{2} + \frac{\frac{W}{2}(\frac{W}{2}+1)}{2}$$

$$= \frac{W^2}{4} + \frac{W}{2} + \frac{W^2}{8} + \frac{W}{4} = \frac{3}{8}W^2 + \frac{3}{4}W$$

Since there is exactly one segment lost every cycle, the loss rate is:

$$L = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

For W large $\frac{3}{8}W^2 >> \frac{3}{4}W$. Therefore $L \approx \frac{3}{8}W^2$.

b) From the text and the slides we know that the average throughput is $\frac{3W \cdot MSS}{4RTT}$ (Bytes/sec). But we know the relation between L and W: $L \approx \frac{3}{8}W^2 \rightarrow W \approx \sqrt{\frac{8}{3L}}$.
Substituting W into the throughput formula, we get:

$$avg\ throughput = \frac{3}{4}\sqrt{\frac{8}{3L}}\frac{MSS}{RTT} = \frac{1.22MSS}{RTT\sqrt{L}}(Bps)$$

3. *(30 Points)* **Chapter 3, Wireshark Lab #2: Exploring TCP.** A summary is provided at the end of the chapter, and the full description is provided on the book's website, and repeated here for your convenience: `https://eee.uci.edu/16s/18105/hws/Wireshark_TCP_v6.0.pdf`. You just need to answer the questions. Including screenshots is optional.

Answer:

1 192.168.1.102 : 1161

2 128.119.245.12 : 80

3 198.168.10.4 : 15245

4 tcp.seq= 0 (sequence number), tcp.flags=0x002 (SYN)

5 tcp.seq= 0 (sequence number), tcp.ack=1, the next byte expected from client. tcp.flags=0x012 (SYN, ACK)

6 tcp.seq= 0

7

Seq. Number	Seg. time	ACK time	RTT	EST. RTT
1	0.026477	0.053937	0.027460	0.027460
566	0.041737	0.077294	0.035557	0.028472
2026	0.054026	0.124085	0.070090	0.033674
3486	0.054690	0.169118	0.114428	0.043768
4946	0.077405	0.217299	0.139894	0.055784
6406	0.078157	0.267802	0.189645	0.072516

8 565, 1460, 1460, 1460, 1460, 1460

9 the minimum amount of available buffer space advertised at the receiver is 5840

10 no, there is no duplicate ACK or duplicate sequence number

11 received segment size

12 file size is the useful data we send through the duration of this connection. So, the throughput is

$$\frac{152,138(B)}{5.61141(sec)} = 277112.2(Bps)$$

13 This depends on the trace you collected. You should look for when the increase switches from exponential (doubling) to linear (+1).