CS 3873: Net-Centric Computing

Assignment #: Assignment Title (if any)

Student Name: __Adrian Freeman___ Student Number: _3661616__

[Mandatory] Declaration: “I warrant that this is my own work.”

Signed by _Adrian Freeman_

(You can type in your name as your signature.)

From the UNB Undergraduate Calendar, available at the website: http://www.unb.ca/academics/calendar/undergraduate/current/regulations/universitywideacademicregulations/viii-academicoffences/index.html.

“Plagiarism includes:

1. quoting verbatim or almost verbatim from any source, regardless of format, without acknowledgement;
2. adopting someone else's line of thought, argument, arrangement, or supporting evidence (such as, statistics, bibliographies, etc.) without indicating such dependence;
3. submitting someone else's work, in whatever form (essay, film, workbook, artwork, computer materials, etc.) without acknowledgement;
4. knowingly representing as one's own work any idea of another.”

Penalties for plagiarism and other academic offences can be found at the above website.

1. True or False?

    a. Host A is sending Host B a large file over a TCP connection. Assume Host B has no data to send to Host A. Host B will not send acknowledgements to Host A because Host B cannot piggyback the acknowledgements on data.

       False, if there is no data to send, Host B will just sent the acknowledgement.

    b. Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receive buffer.

       True, the unacknowledged bytes cannot exceed the receiving host's receive buffer.

    c. Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is m, then the sequence number of the subsequent segment will necessarily be m+1.

       False, it will be m + the # of bytes in the current segments payload.

    d. Suppose that Host A sends one segment with sequence number 38 and 4 bytes of data over a TCP connection to Host B. In this same segment the acknowledgement number is not necessarily 42.

       True, if not, we would have to assume that host B hast already received all data up to 41 before the ack.

2. A process in Host C has a UDP socket with port number 5120. Host A and B each send a UDP segment to Host C with destination port number 5120. Will both of these segments be directed to the same socket at Host C? How will the process at Host C know that these two segments originated from two different hosts?

   Yes they will be directed to the same socket, since both host A and B are sending to Host C port number 5120. Host C can differentiate the two segments' hosts by checking the source IP and source port.

3. Complete the following:

   a. Suppose you have the following packet with the bits: 01111001 11001111 and 11110011 10011000. What is the Internet checksum of this packet?

      01111001 11001111

      <u>+11110011 10011000</u>

      101101101 01100111 -- carry the one

      01101101 01101000 – flip the bits

      <u>**10010010 10010111**</u> This is the internet checksum of this packet

   b. Suppose you have the following packet with the bits: 11000011 10101010, 10010001 01000101 and 11000011 10011000. What is the Internet checksum of this packet?

      11000011 10101010

      <u>+10010001 01000101</u>

      101010100 11101111 –carry the one

      01010100 11110000

      <u>+11000011 10011000</u>

      10001100010001000 – carry the one

      00011000 10001001 – flip the bits

      <u>**11100111 01110110**</u> This is the internet checksum of this packet

   c. For the packet in a), give an example where one bit is flipped in each of the two 16-bit words and yet the checksum doesn't change.

      Original words = 01111001 11001111 and 11110011 10011000

      original checksum = 10010010 10010111

New words (1 flipped bit) 01111001 11001110 and 11110011 10011001 (flipped the last bit)

01111001 11001110

+11110011 10011001

101101101 01100111 – carry the one

01101101 01101000 – flip the bits

10010010 10010111 is the internet checksum, which is the same as the previous

4. Consider a cross-country example where a host in East Coast is connected with another host in West Coast by a channel with a transmission rate of 100 Mbps. The round-trip delay between these two end systems is approximately 70 ms. How big would the window size have to be for the utilization over this channel to be greater than 95%? Suppose that the size of a packet is 2 KB, including both header fields and data.

U	0.95
L	$2*10^3*8$ = 16000bits
R	$100*10^6 = (10*10^7)$bps
RTT	70ms
d_{trans}	$d_{trans} = L/R$ d_{trans} = 16000bits/$(10*10^7)$bps d_{trans} = 0.00016s d_{trans} = 0.16ms
Window Size	$N = U*((d_{trans} + RTT)/d_{trans})$ N = 0.95((0.16ms + 70ms)/0.16ms) N = 0.95(70.16ms/0.16ms) N =0.95*438.5 N =416.575
	The Window Size would have to be 417 packets

5. Suppose that three measured SampleRTT values (see Section 3.5.3) are 110 ms, 120 ms, and 90 ms. Compute the EstimatedRTT after each of these SampleRTT values is obtained, using a value of **α = 0.125** and assuming that the value of EstimatedRTT was 100 ms just before the first sample was obtained. Compute also the DevRTT after each sample is obtained, assuming a value of **β = 0.25** and assuming the value of DevRTT was 5 ms just before the first sample was obtained. Last, compute the TCP TimeoutInterval after each of these samples is obtained.

Sample RTT	EstimatedRTT	DevRTT	TimeoutInterval
N/A	100ms	5ms	N/A
110ms	((1-0.125)*100ms) +0.125*110ms = 101.25ms	((1-0.25)*5ms +0.25*|110ms -100ms| = 6.25ms	101.25ms + 4*6.25ms = 126.25ms
120ms	((1-0.125)*101.25ms) +0.125*120ms = 103.59ms	((1-0.25)*6.25ms +0.25*|120ms - 101.25| = 23.69ms	103.59ms + 4*23.69ms = 198.35ms
90ms	((1-0.125)*103.59ms) +0.125*90ms = 101.89ms	((1-0.25)*23.69ms +0.25*|90ms - 103.59ms| = 21.165ms	101.89ms + 4*21.165ms = 186.55ms

6. Host A and B are communicating over a TCP connection, and host B has already received from A all bytes up through byte #126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgement whenever it receives a segment from Host A.

a. In the second segment sent from A to B, what are the sequence number, the source port number, and the destination port number?

   127 + 80 = 207, the sequence number is #207

   the source port # is 302 and the destination port # is 80

b. If the first segment arrives before the second segment, in the acknowledgement of the first arriving segment, what are the acknowledgement number, the source port number, and the destination port number?

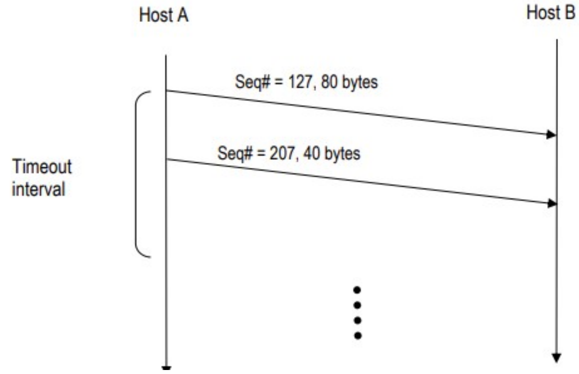
   Because the first segment ends with byte #206, the acknowledgement number is 207

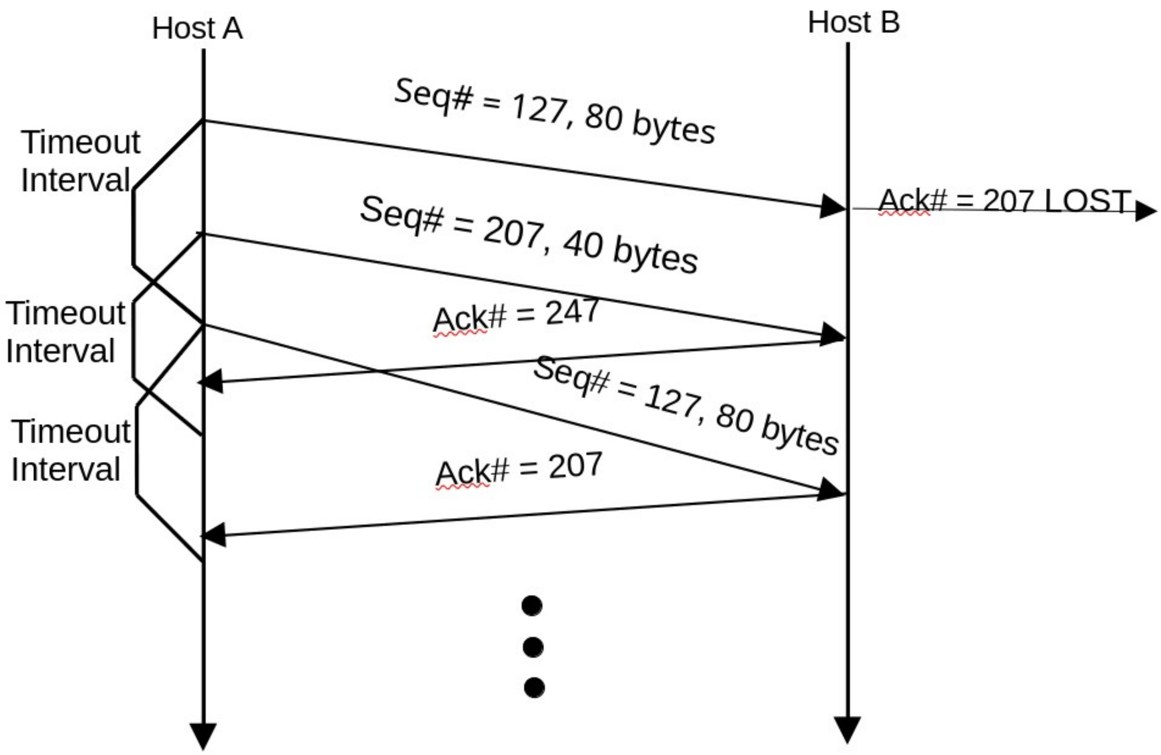
   The source port # is 80, the destination port # is 302

c. If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, what is the acknowledgement number?

   It will still be expecting byte 127, from the first segment. So the ack number is 127

d. Suppose the two segments sent by A arrive in order at B. The 1st acknowledgement is lost and the 2nd acknowledgement arrives after the 1st timeout interval. Draw a timing diagram showing these segments and all other segments and acknowledgements sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgement that you add, provide the acknowledgement number. For example:




Host A
Host B
Seq# = 127, 80 bytes
Ack# = 207 LOST
Timeout Interval
Seq# = 207, 40 bytes
Timeout Interval
Ack# = 247
Seq# = 127, 80 bytes
Timeout Interval
Ack# = 207