LEBANESE AMERICAN UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS

CSC 430 – Computer Networks

Homework III

Lara Tawbeh
ID#: 202102927

Date: 03-22-2024

Modified: 03-28-2024

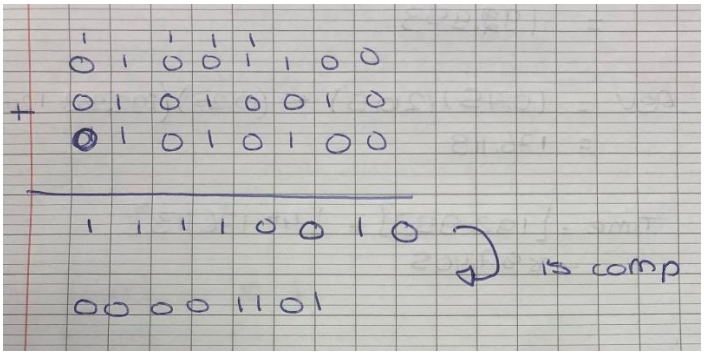
Table of Contents

Problem 1:

What is the 1s complement of the sum of these 8-bit bytes. Show all work. Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum? With the 1s complement scheme, how does the receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2-bit?

LRT: 01001100 01010010 01010100

Sum: 11110010 1s complement of the sum: 00001101



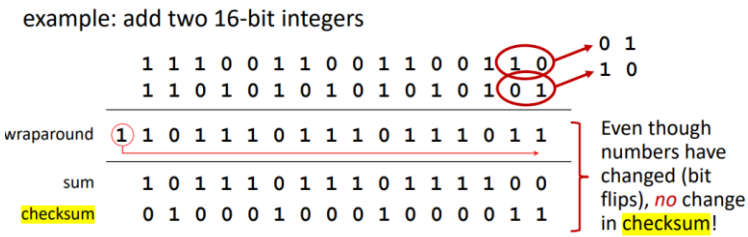
Finding out why UDP uses 1compliment was confusing, different sources say different things. However, I found an interesting historical analysis:

“In the past, computational resources were scarce, and every machine cycle was valuable. The UDP protocol utilizes one's complement arithmetic for checksum calculation to ensure accuracy and efficiency, independent of system endianness.”

Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum? - Quora

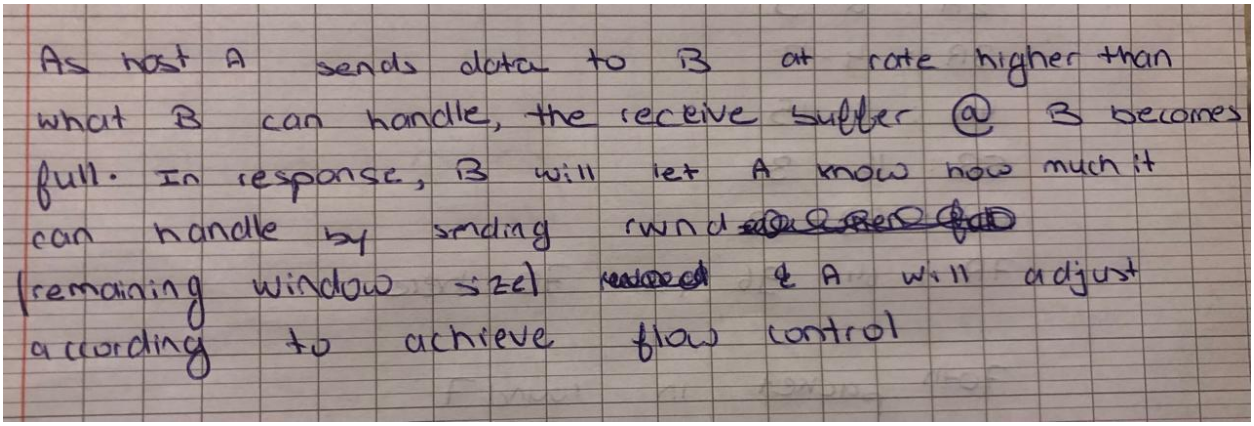
Errors will be detected by doing the same calculation and the sender and receiver side and if they don’t match then an error is present.

It is not possible that a 1-bit error will go undetected, but it is possible a 2-bit error will go undetected if they result in the same answer in the checksum, example:



Taken from book: Kurose, J. and Ross, K. (2020) Computer Networking: A Top-Down Approach. Pearson, 8th Edition.

Problem 2:


As host A sends data to B at rate higher than what B can handle, the receive buffer @ B becomes full. In response, B will let A know how much it can handle by sending rwnd (remaining window size) & A will adjust according to achieve flow control

Problem 3: X= 927 ms

$$EstimatedRTT = 0.125 * 927 + 0.875 * 100 = 203.375\, ms$$

$$DevRTT = 0.25 * |100 - 203.375| + 0.75 * 5 = 29.5\, ms$$

$$TimeoutInterval = 203.375 + 4 * 29.5 = 354.393\, ms$$

$$Estimated\ RTT = 0.125 * 120 + 0.875 * 203.375 = 192.953\, ms$$

$$DevRTT = 0.25 * |120 - 192.953| + 0.75 * 29.5 = 40.36\, ms$$

$$TimeoutInterval = 192.953 + 4 * 40.36 = 354.393\, ms$$

$$Estimated\ RTT = 0.125 * 140 + 0.875 * 192.953 = 186.3\, ms$$

$$DevRTT = 0.25 * |140 - 186.3| + 0.75 * 40.36 = 41.845\, ms$$

$$TimeoutInterval = 186.3 + 4 * 41.845 = 353.68\ ms$$

$$Estimated\ RTT = 0.125 * 90 + 0.875 * 186.3 = 174.26\, ms$$

$$DevRTT = 0.25 * |90 - 174.26| + 0.75 * 41.845 = 52.44\, ms$$

$$TimeoutInterval = 174.26 + 4 * 52.44 = 384.02\ ms$$

$$Estimated\ RTT = 0.125 * 115 + 0.875 * 174.26 = 166.85\, ms$$

$$DevRTT = 0.25 * |115 - 166.85| + 0.75 * 52.44 = 52.29\, ms$$

$$TimeoutInterval = 166.85 + 4 * 52.29 = 376.01\, ms$$

Problem 4:

a) slow => before it starts inc by 1 only
=> $[1, \bar{6}] \cup [23, 26]$

b) inc by 1
$[6^{+}, 16^{-}] \cup [17 \cup 22]$

c) Triple dup ACK as window size drops to 1/2 cwnd.

d) dropped to 1 HSS → timeout

e) 32 (where slow start stopsh)

f) 42/2 = 21

g) 29/2 = 14.5

h) 1R : 1
2R : 2 - 3
3R : 4 - 7
4R : 8 - 15
5R : 16 - 31
6R : 32 - 63
7R : 64 - 96 <= 70 here

70th packet in round 7

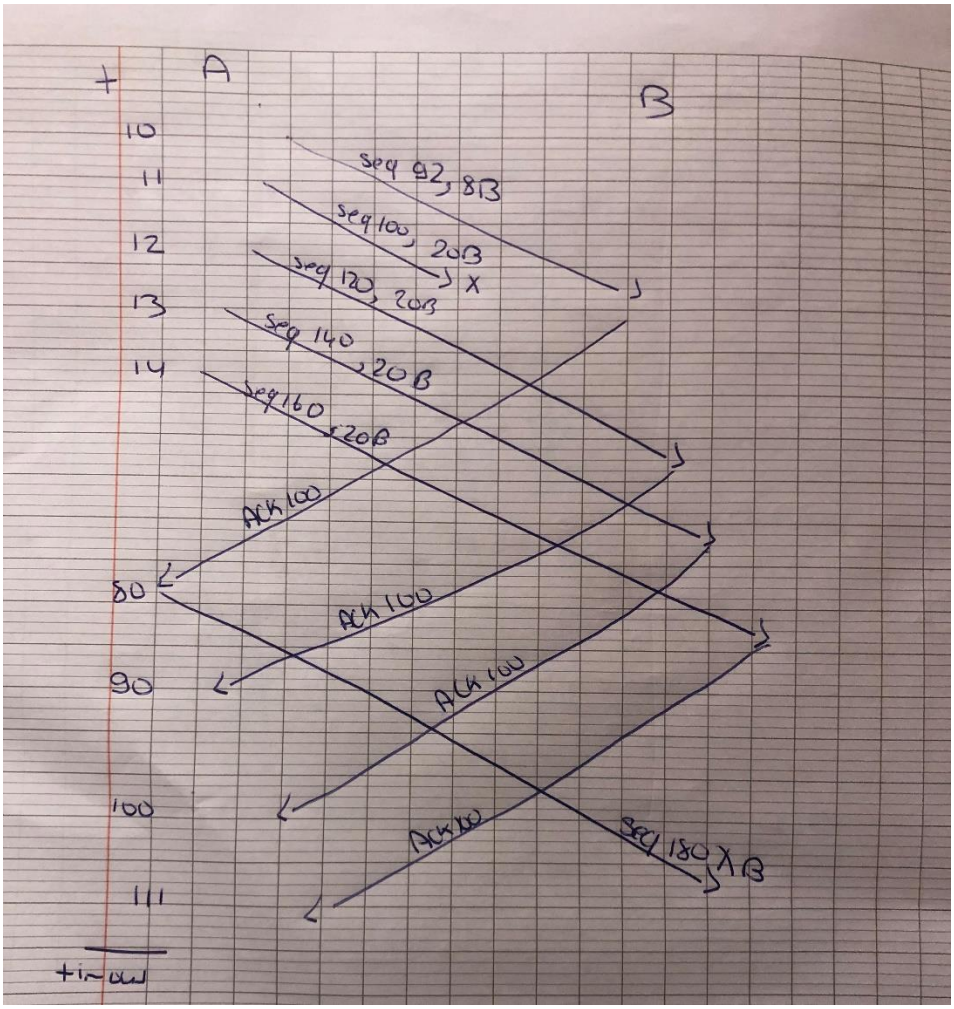
i) cwnd : 8/2 = 4
ssh = 4

j) ssh = 21 cwnd 8.
@ 16 cwnd : 1
@ 17 cwnd : 2
@ 18 : 4 => @ 19 : cwnd 8

k) 17 : 1
18 : 2
19 : 4
20 : 8
21 : 16
22 : 21 (ssh)

total = 52 packets

Problem 5:


t
A
B
10
11
12
13
14
80
90
100
111
time
seq 92, 8B
seq 100, 20B
X
seq 120, 20B
seq 140, 20B
seq 160, 20B
ACK 100
ACK 100
ACK 100
ACK 120
seq 180
X B

Problem 6:

a)

Go-Back-N:

A sends a total of 9 segments. Initially sent as segments 1 - 5, and then retransmitted as segments 2 - 5. B replies with 8 ACKs: 4 ACKs with ACK number 1 and 4 ACKs with ACK numbers 2 - 5.

Selective Repeat:

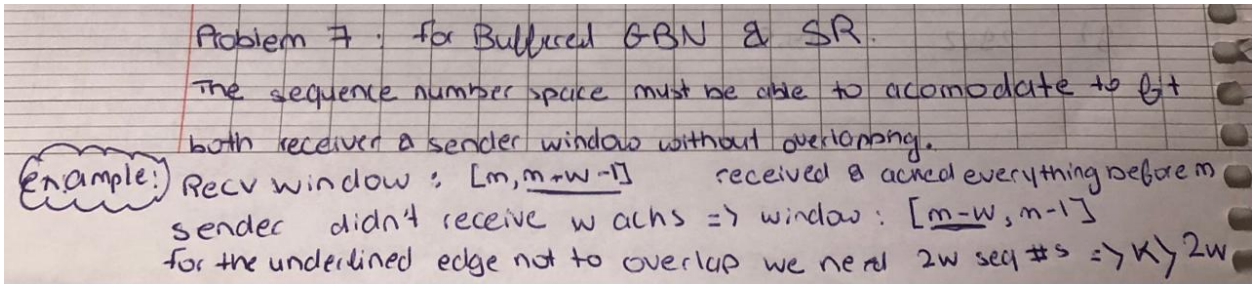
A sends a total of 6 segments. Initially sent as segments 1 – 5 and then segment 2 is retransmitted. B acknowledges with 5 ACKs: 4 ACKs with ACK numbers 1, 3, 4, and 5, and 1 ACK with ACK number 2.

TCP:

A sends a total of 6 segments. Initially sent as segments 1 - 5 and then segment 2 is retransmitted. B acknowledges with 5 ACKs: 4 ACKs with ACK number 2 and 1 ACK with sequence number 6.

b) TCP due to its fast retransmit capability.

Problem 7:



As for non-buffered GBN k-1.