# >>> Assignment #3 for Computer Networks (CNT 4004) <<<

(due on Thursday, October 8th at the start of class)

This assignment covers material from chapter 3 of the textbook and from roughly the third two weeks of class lecture.

## Problem #1

Answer the following questions about TCP and UDP checksums.

a) TCP and UDP use 16-bit checksums. The checksum code was given to you in class. Give the checksum for the following three 16-bit words 0x3346, 0x7766, 0x71AB.

b) Suppose that a UDP receiver computes the checksum for a received UDP segment and finds that it matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? If your answer is "no", then give an example of bit errors (using the three words in (a) above for your example) that could not be detected.

## Problem #2

Design a protocol to do the following. Provide a service where a sender will broadcast a message every 5 seconds and a receiver will acknowledge each message. Each successive message that is acknowledged should contain an incrementing counter value ("1" for the first message sent after system reset of the sender, "2" for the second message sent, and so on). If a sent message is not acknowledged (that is, no ACK is received since the last message sent) then the next sent message should have the same sequence number as the previously sent message. It is possible that sent messages and/or ACKs may be lost. You will need to give (reasonable and necessary) assumptions and provide a list of messages, the syntax of each message, the semantics of each message, and the FSMs for the server and receiver. Any timers, variables, constants, etc. used in the FSMs must be defined.

## Problem #3

Do problem P6 (page 289) in your textbook.

## Problem #4

Consider the RDT protocol developed in the book using RDT 2.2 receiver and RDT 3.0 sender.

a) If the time-out is less than the RTT between a sender and receiver sketch a timing diagram of the packet and ACK flows between the sender and receiver.

b) If the time-out is much greater than the RTT between a sender and receiver sketch the time diagram for the case of a packet loss.

c) Comment on the trade-off between time-out less than RTT and time-out much greater than RTT. Which condition could cause the most harm to other senders and receivers (that is, to other users of the network)? Explain why.

## Problem #5

Do problem P13 (page 290) in your textbook.

## Problem #6

Consider the following scenario.

- Distance from sender to receiver is 3000 miles
- Data rate between sender and receiver is 1 Mbps

- Data packet length is 1250 bytes
- ACK packet length is 64 bytes

a) What is the link utilization if an SAW protocol is used and you assume no bit errors (i.e., lost packets or lost ACKs never occur)?

b) What is the link utilization if an SAW protocol is used and you assume a bit error rate of $10^{-5}$ (you may assume that only data packets will have errors and that all errors are detectable at the receiver, ACK packets never contain bit errors).

c) What window size would be needed (given an SW protocol) to achieve 100% link utilization assuming no bit errors?

## Problem #7

When decoding an Internet packet how can the decoder (be it you doing it by hand or a program doing it automatically) know if the packet is a TCP segment, a UDP segment, or something else? What might the "something else" be? "Guessing" is not the right answer even if this is more-or-less what we did when decoding the packet in class.

## Problem #8

Do problem P25 (page 293) in your textbook.

## Problem #9

There are two ways to terminate a TCP connection, what are they? What are the implications (that is, what happens?) of each way?

## Problem #10

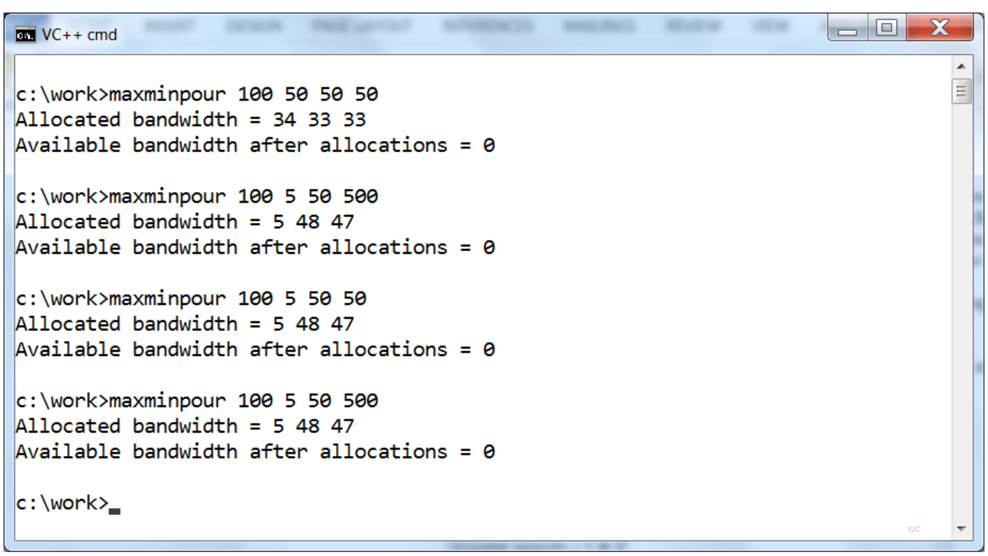
Write a program that will implement max-min scheduling by simulating the pouring process. The program should take as command line input the available bandwidth and up to 10 requested data rates (this for up to 10 sources). The bandwidth and data rates are all in integer increments of 1 Mb/s and the final allocation is also to be an integer value (that is, no allocations of less than 1 Mb/s allowed). Here below is an execution of the program written for the solution. You should give a screen shot showing the same inputs and outputs as in the below screen shot. Also, run your program for the following two cases:

- Available is 100 Mb/s - Source A requests 50 Mb/s, source B requests 70 Mb/s, and source C requests 10 Mb/s

And…

- Available is 100 Mb/s - Source A requests 50 Mb/s, source B requests 700 Mb/s, and source C requests 10 Mb/

Submit also your source code.




```
c:\work>maxminpour 100 50 50 50
Allocated bandwidth = 34 33 33
Available bandwidth after allocations = 0

c:\work>maxminpour 100 5 50 500
Allocated bandwidth = 5 48 47
Available bandwidth after allocations = 0

c:\work>maxminpour 100 5 50 50
Allocated bandwidth = 5 48 47
Available bandwidth after allocations = 0

c:\work>maxminpour 100 5 50 500
Allocated bandwidth = 5 48 47
Available bandwidth after allocations = 0

c:\work>
```