**LAB 11**

1. Lab 19 Networking (Lab19\_Manual.pdf)

a. Exercise 1

1. Start the TCP/IP app.
2. In the text field; type:

Computer networking: Packet City!

Then press the Send a message button.

1. Watch the entire sequence. How many DAT packets were sent? How many ACK were sent?

ANS: 4 DAT and 4 ACK packets were sent.

1. If each character or blank in a packet header counts for one and DAT packets except the last carry 10 characters and ACK packets have no data characters, count up how many characters were sent in total.

ANS: 33 characters were sent in total

In each packet, we have 11 characters for DAT and 9 characters for ACK and we have total 4 packets for each.

So, it becomes 4\*11 + 4\*9 = 80

Then total number of characters sent = 80 +33 = 113

1. There are 52 characters in Example message, including blanks and punctuation. Sub 52 from total no. of characters sent in both directions in all packets. Divide this number by total number of characters to get overhead, expressed as percentage.

ANS: We have 33 characters in the message

Total characters sent = 113

Overhead characters = total – useful = 113-33 = 80

Overhead = 80/113 = 70.8 % (approx.)

1. Imagine you have a million-character message to send, perhaps a large file. How many characters total would be sent in all packets necessary to move it from node 0 to node 1?

ANS: In case we have a million-character message to send then the total characters sent will no. of characters in file (say 1 million ) + (Total number of DAT packets sent )\*20

1. What would be an obvious way to decrease the overhead? Why might this solution backfire? Under what conditions?

ANS:

b. Exercise 3

1. Run the app, using a message. See if you can compute the checksum as TCP/IP does. Select a data packet, but don’t select the last packet because it might be too short – less than 10 characters.

{Message used is “Computer networking: Packet City!”}

1. Count up the characters, including blanks. If they do not equal 10, assume there are blanks at the end so that the characters count is 10.
2. Using the text translator into ASCII app from a previous lab, type in the characters from the data packet and click on transfer text.
3. Add up the values for each of the characters. Then take modulus of this number using 256. The modulus operator is available on some computer calculations and is represented by mod, but you can easily compute it by dividing the total by 256 and saving only the remainder. For instance, suppose the total is 7452. Since 7452/256 = 29.109375, but we only want the remainder, multiply 0.109375 by 256. The result is 28.

ANS: Checksum for first DAT header is:

C o m p u t e r n

67+ 111+ 109+ 112+ 117+ 116 +101+ 114+ 32+ 110 = 989

CHECKSUM = 989 % 256 = **221**

1. Compare your computed checksum against what the app shows for the packet. Do they agree?

ANS: Yes, it is the same.

1. Now damage your packet by altering one character. Recompute the checksum. Do you see how TCP/IP can spot errors?

ANS: It shows errors like:

“Received damaged packet

Got NAK 0; resending packet

Timed out, resending packet”

In this case checksum remained same but the data header changed.

1. Think of a way that a packet can be damaged and still have the same checksum as the undamaged version.

2. (TCP/IP) Start the TCP/IP applet. Send the following message: Computer Science Illuminated CPSC 1050

a. How many packets of DAT and ACK are transmitted?

ANS: 4 packets of DAT and 4 packets of ACK are transmitted.

b. Write down the headers for each DAT and ACK packet, e.g.: DAT 1 0 144 .

ANS: Headers for each DAT packets are:

DAT 1 0 194 ACK 0 0 0

DAT 1 1 168 ACK 0 1 0

DAT 1 2 186 ACK 0 2 0

DAT 1 3 204 ACK 0 3 0

c. Calculate the checksum for the first DAT header.

ANS: Checksum for first DAT header is:

C o m p u t e r S

67+ 111+ 109+ 112+ 117+ 116 +101+ 114+ 32+ 83 = 962

CHECKSUM = 962 % 256 = **194**

d. If each character or space in header counts for one character, and all DAT packets carry 10 characters (except possibly for the last one), and ACK packets have no data characters (just the header), calculate the total number of characters transmitted both ways.

ANS: No. of DAT and ACK packets are 4 each

Characters in each DAT packet = 11

Characters in each ACK packet = 9

Total characters in the message = 3\*10 +8 = 38

Total number of characters sent = 4\*(11+9) + 38 = **118**

e. How many characters are in the above message? How many extra characters are transmitted according to your answer to part d? Divide this number by the total number of characters to get the "overhead". What is the ratio of "useful data" to the total data transmitted over the network for the given message?

ANS: Total characters sent = 118

No. of Useful characters = 38

Overhead characters = 118-38 = 80

Overhead = 80/118 = 67.8% (approx. by rounding off)

Useful data/ Total data = 38/118 = 32%

3. (TCP/IP) The message "Packet is a unit of data sent across the network" is to be sent over a reliable connection from node 0 to node 1. Here is a diagram of the first packet being sent:

a. Presuming that each packet can carry 10 characters of data, how many packets of DAT and ACK are required?

ANS: There are 5 DAT and 5 ACK packets required.

b. Presuming that the data header of each packet and the ACK packet is of the same size as in the diagram above (11 characters and 9 characters respectively), what is the ratio of "useful data" to the total data transmitted over the network for the given message? What is the "overhead"?

ANS:

Useful data = 48

Total data transmitted= 48+100 = 148

Ratio of useful data to total data: 48/148 = 32%

Overhead characters: total data – useful = 148 – 48 = 100

Overhead = 100/148 = 67%

4. Lab 23 Limits of Computing (Lab23\_Manual.pdf)

a. Exercise 1

1. Start the comparison app and click on example 1 button. The app runs through a bunch of N values and put the corresponding (N) in the text areas. Write down what value N has when the N^N text areas first says “too big!” Do the same for the N! and 2^N text areas.

ANS: Values of N are:

When N^N is too big, N is 15

When N! is too big, N is 21

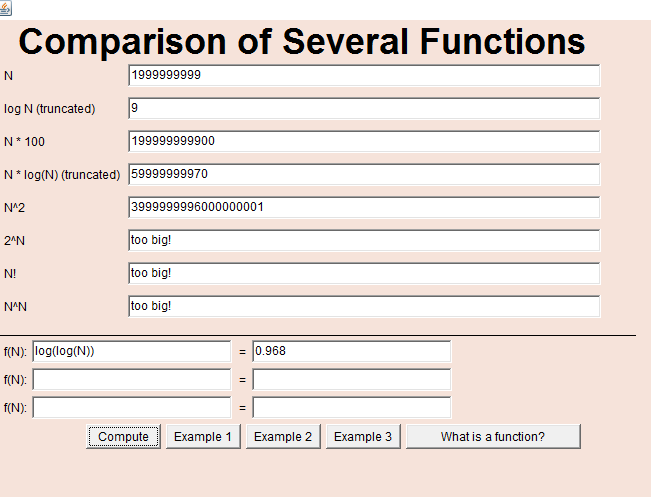
When 2^N is too big, N is 65

1. Type the function into first f(N) text space:

log (log N)

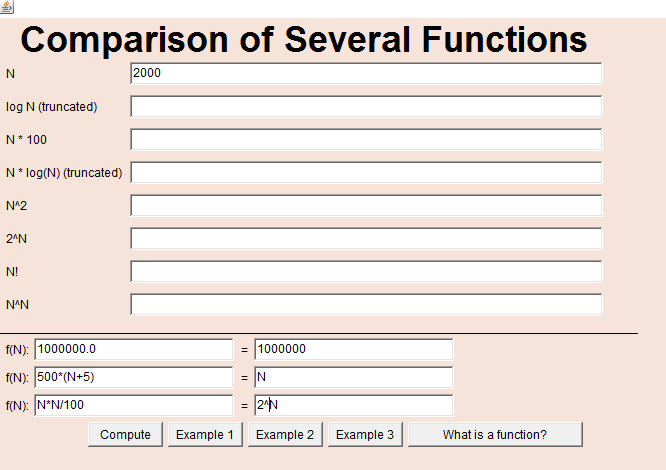
Then, type in several values of N, one at a time, click on the compute button after each one. Now see if you can get the value of “1.0” to appear in the result area by finding a value of N such that log(log(N)). Take a screenshot.

ANS: It does not becomes equal to 1.0 but it has highest value “0.968” when N = 1999999999



1. Lets experiment with three functions. Click on example 3, and three functions will be inserted into three f(N) text areas.
2. Take screenshot, print I, and write the order of magnitude in big-O notation next to each of three functions.

ANS:



Big-O notation for the three functions is:

Function1: 1000000

Function2: N

Function3: N^2 or N\*N

1. Type1 into the N text area and press compute. Write down the values of three function. Which function gives the biggest answer?

ANS: Values of 3 functions are:

1. f(N)= 1000000
2. f(N) =3000
3. f(N) = 0.01

f(N): 1000000.0 that is function 1 has the biggest answer for N=1

1. Type “3000” into N text area and press compute. Write down the values of the three functions. Which function now gives the biggest answer?

ANS: Values of 3 functions are:

1. f(N)= 1000000

2. f(N) =1502500

3. f(N) = 90000

f(N): 500\*(N + 5) that is function 2 has the biggest answer for N=3000

1. Finally, find a value for N that causes the third function to give the biggest answer.

ANS: For N = 100000 and bigger, third function can give the biggest answer.

1. Write down your observations as to the values that these three functions produce for given ranges of N. Is one function always larger than the others? Why or why not?

ANS: As per my observations, for N =1 to 1994, 1st function is biggest

For N = 1996 to 50004(approx.), 2nd function is biggest

For N>50005, 3rd function is the biggest

At N= 1995, value of function 1 and 2 are same.

At N=50005, value of function2 and 3 are approx. same.

5. (Big-O-Notation) what is the order of growth of the following functions in Big-O notation?

a. f(N) = (N3 + 100N2 + 10N + 50)

ANS: f(N) is of the order N3

b. f(N) = (100N2 + 10N + 50) / N2

ANS: f(N) = 100 + 10/N +50/N2

f(N) is of the order N0 (100N2/N2) because 100N0 will always remain greater than N-1 and N-2

c. f(N) = 10N + 50Nlog2(N)

ANS: f(N) is of the order Nlog2(N) because Nlog2(N) is greater than N

d. f(N) = 50N2 log(N) / N

ANS: f(N) = 50Nlog(N)

f(N) is of the order Nlog(N)