COMP 5416 Assignment 2

Due: 2 November 2018, 23:59.

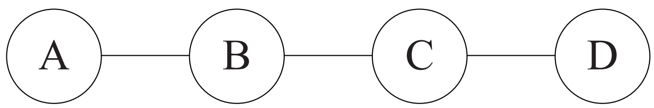
Question 1 (Go-Back-N, 20%)**.** In the following network, node A transmits packets that pass through B and C, and arrive at the destination D. The bit rate of all links is $R = 1$ Mbit/sec. The maximum packet size in the network is 500 Bytes. Ignore the header size. The one-way propagation delay on each link is 4 msec.

(1) How long does it take to transmit a file of size 80000 Bytes if A and D use the Go-Back-6 ARQ and B and C are Store-and-Forward? Assume that there is no error in transmission in the network, and the size of ACK packets is negligible.

(2) Repeat (1) if we use Go-Back-N, where $N = 8$, 10, and 20.

(3) Sketch the overall time vs. N (N is the value of Go-Back-N) curve. N ranges from 1 to 20.

(4) Repeat (1) if the second packet is lost (all other packets are successfully received). The timeout duration is assumed to be 50 ms.



Question 2 (Queueing Theorem: Simulation, Analysis, and Application, 40%)**.** In this task, you need to simulate and analyze an $M/M/m/n$ queue with arbitrary m and n. You can reuse the codes in Week 7 Lab to simulate an $M/M/m/n$ queue. You Python code must be submitted as supplementary material. You also need to theoretically compute the stationary distribution to verify your simulation. Finally, you will understand how queueing theorem will help design and install real-world telephone and cellular networks.

Throughout this question, let the arrival rate be λ. Let the service rate of each server be μ.

(1) Let $\lambda = 1$ and $\mu = 2$. What is the stationary distribution of an $M/M/2/6$ queue. Please figure out this result by both simulation and analysis.

(2) Let $\lambda = 1$ and $\mu = 1$. What is the stationary distribution of an $M/M/5/10$ queue. Please figure out this result by both simulation and analysis.

(3) Use theoretical analysis only. What is the stationary distribution of an $M/M/m/m$ queue (now $n = m$). What is the probability that a new arrival is blocked (dropped) by the system? This probability is defined as p_b.

(4) Follow (3). Let $\mu = 1$ and $m = 10$. If the blocking probability p_b must be limited to 0.001, what is the maximum value of λ?

(5) Follow (3). Let $\lambda = 10$ and $\mu = 1$. If the blocking probability p_b must be limited to 0.001, what is the minimum value of m, i.e., the number of servers?

(6) Telecom/cellular companies usually use methods in (4) and (5) to design their telephone/cellular networks. In (4), given the required blocking probability p_b and the number of telephone lines m, they want to know the maximum possible traffic intensity λ they can accommodate. In (5), given the required blocking probability p_b and the traffic intensity λ, they want to know how many telephone lines they shall install. The method to compute p_b is also well known as *Erlang Formula*. Google *Erlang Formula* and answer the following questions.

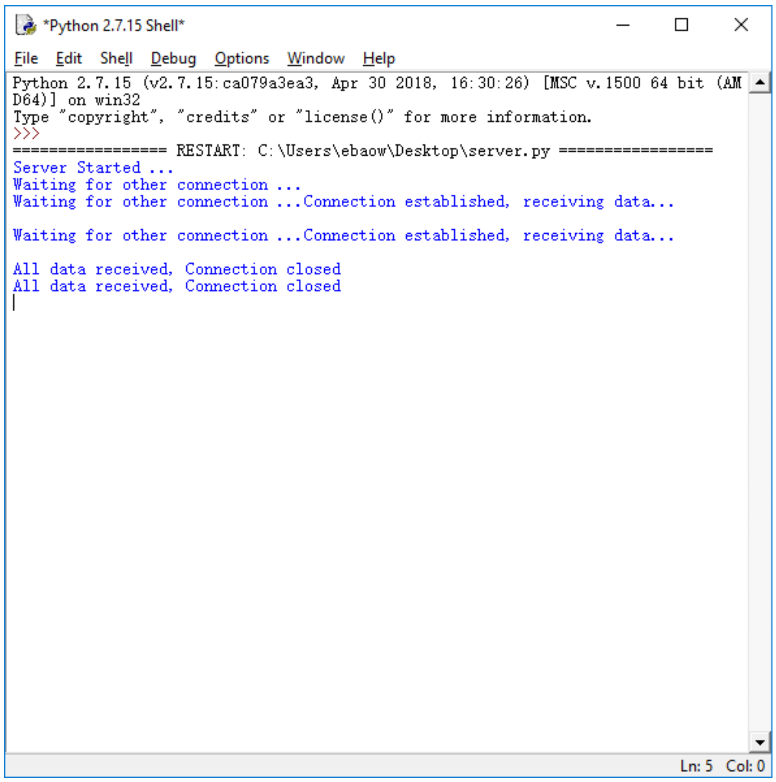
(a) What are Erlang-B and Erlang-C Formulas? When are they used? What is the difference between them?

(b) In (4) and (5), do you use Erlang-B or Erlang-C Formula? Why?

Question 3 (Multi-thread Server: Implementation, 20%)**.** You are given the complete code for the client in Lab in Week 11. Your task is to write the TCP server. The client code is in `client.py`. You must not modify this code. (However, you are allowed to change `ServerName` and `ServerPort`).

Different from the server in the lab, the new server must be able to serve multiple clients simultaneously. Please note that the server code in Week 11 can only accept one client! In order to serve multiple clients simultaneously. The server should run multiple threads. The server will establish a new connection socket to communicate with one new client, and each new connection socket will be managed by a new thread. You should self-study the following function: `thread.start_new_thread()`.

The following figure shows an example of server when two clients are sending images at the same time. The two connections are closed in the end, demonstrating that there are two concurrent transmissions before the first "Connection closed".



You also need to capture the concurrent data transmission by Wireshark. In the example in the next page, we can see that the two clients are running at 192.168.0.2, with port numbers 54766 and 54769. The server is running at 192.168.0.8, with port number 12010. The throughputs of the two connections are both positive at around 12th second.

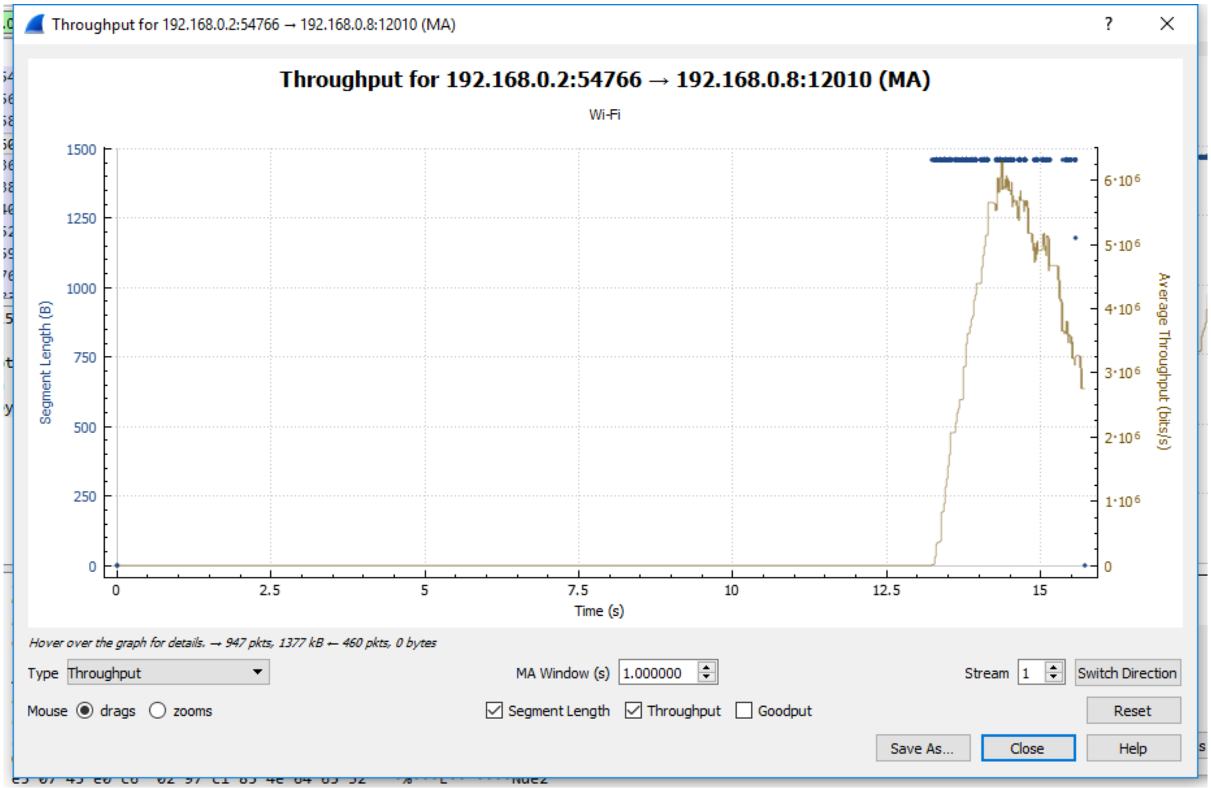
Tasks and submissions:

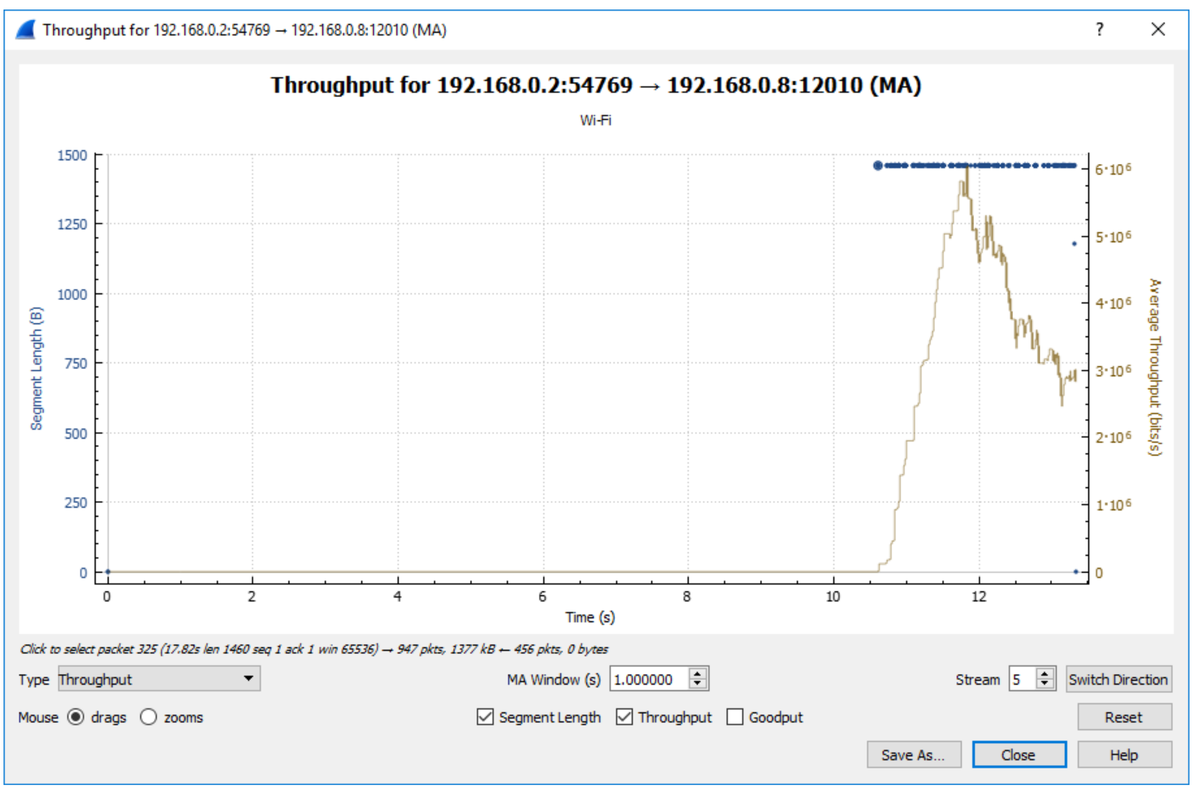
(1) Build up a multi-thread server which can serve multiple clients **at the same time**. Submit your server-side Python code. Submit your server code as `Lastname_Firstname_Server.py`. We will use the client in Week 11 to test against your server.

(2) Test your server with three clients (using the code in Week 11) sending images at the same time. Capture the packets by Wireshark at the server side. You are allowed to run the server and clients in one computer using `localhost`. Submit your Wireshark capture. Your capture file must be smaller than 10MB. Your capture will be ignored and will not be marked if it is greater than 10MB. Submit your capture as `Lastname_Firstname_Capture.pcapng` (or `.pcap`).

(3) In the main submission file, based on your capture in (2), plot the throughput vs. time of the three connections (similar to the figures in the next page). Show that they are operated parallel. In the main file, you also need to give the three clients' IP addresses, clients' port numbers, server's IP address, and server's port number.

You overall mark will be zero if you do not submit code in (1), no matter if you submit (2) or (3).





You submission in (2) and (3) will be ignored and will not be marked if your submission in (1) does not work.

You overall mark will be zero if your Wireshark capture in (2) does not match the throughput plots in (3).

Question 4 (BER vs SNR, 20%)**.** Following the tutorial in Week 12, question 2 (CDMA with Noise), we now study how the noise level will influence the bit error rate (BER), i.e., the probability that 1 is decoded as -1. The physical meaning of δ^2 is *the power of noise* and 1 is *the power of signal*. So the SNR is $1/\delta^2$.

Redo the question when $\delta^2 = 10^i$, where $i = 0, -1, -2, -3, -4, -5$. When computing the Q function, do not use the table in the tutorial since it does not cover enough range. Use the following website at http://137.193.149.100/labalive/erfc/.

Plot the BER vs SNR curve based on your result. The x-axis should be in dB, and the y-axis should be in powers of 10. See week 10's slide (wireless and mobile network 1) page 15.

Note: A similar question will be discussed in Week 12.

Submission Instructions: You should submit one main file and several supplementary files. You should include your answers to Q1–Q4 and explanations of your answers in the main file. You should submit your main file at “main file submission”. The main file is in the format of pdf. For Q2, you need to submit your queue simulator at “Q2 code submission”. For Q3, you should submit your Python code at “Q3 code submission” and Wireshark capture at “Q3 capture submission”. Your code and capture will be examined against your answers in the main file. Penalty would be incurred if your code/capture does not match your answer in the main file. All your submissions will be checked by plagiarism examination tools.

This is one assignment with multiple pieces to submit. Your submission time is equal to the submission time of the *last* piece. For example, suppose you submit the main file, Q2 code, and Q3 code on time, but your Q3 capture is late for 26 hours. The whole assignment is regarded as a 26-hour late submission. If your answers can get 75% in the whole assignment, you will finally get 75%-40%=35%.

Warning: If you want to give up Q2 code/Q3 code/Q3 capture. Leave the submission blank. If there is no submission, it is regarded as “no submission.” If there is a delayed submission, the delayed submission is regarded as the submission time and this may disadvantage you. For example, suppose you submit the main file, Q2 code, and Q3 code on time and you would get 70%. 25 hours after the deadline you find that you solve the Q3 capture and you submit it (25 hours late). Then, your whole assignment will be regarded as 25 hours late. Even if your new answers can now get 75%, you will finally get 35% due to late submission.