TELCOM 2310 Fall 2023
Project 1

As we discussed in class, routers use *queues* (also called *buffers*) to temporarily store packets when they cannot immediately send them on an outgoing link. As we've seen, these queues are finite. Once the queue is filled with packets, any new packets that arrive before a packet transmission makes space in the queue again are dropped and lost.

Recall that the details of this phenomenon are tightly related to the rates at which packets arrive at the queue and can be transmitted on the outgoing link, as well as the size of the buffer. We will use λ to denote the packet arrival rate (in packets per second), μ to denote the packet departure (or transmission) rate (in packets per second), and n to denote the size of the buffer (in packets). See Figure 1 for an illustration. If you are interested in more details for queuing theory and its application to computer networks, you can see [1] and [2].

In this project, you will write a software tool that will simulate the arrival and departure of packets to/from the router. You will then use this tool examine scenarios with different arrival / departure rates and queue sizes.

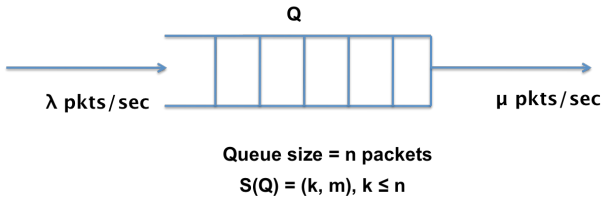

Figure 1: A router buffer representation.

1 Discrete event simulator

Every arrival or departure of a packet to/from the router is a **discrete event**. Whenever a new event takes place, the *state* of the queue is altered. We can define the state S of the queue Q as the following tuple:

$$S(Q) = (\#\ of\ packets\ in\ the\ queue, \#\ of\ packets\ dropped) \tag{1}$$

For example if $S(Q) = (143, 4)$, there are currently 143 packets in the buffer, and 4 packets have been dropped so far in total.

For this project, we have two different possible events: (i) packet arrival and (ii) packet departure. In order to decide if a simulated event is a packet arrival or departure we have to calculate the probability of each event. Since we have an arrival rate of λ packets/sec, and a departure rate of μ packets/sec, the corresponding probabilities P are:

$$P_{arrival} = \frac{\lambda}{\mu + \lambda} \tag{2}$$

$$P_{departure} = 1 - P_{arrival} = \frac{\mu}{\mu + \lambda} \tag{3}$$

After every event, the state $S(Q)$ is changed, that is, the number of packets in the queue and/or the number of dropped packets are changed. Our objective in this project is to keep track of the state changes for a number of discrete events and understand the reasons behind the observed changes.

2 Programming assignment

You can use any programming language you are familiar with to complete this project. The **basic steps** you need to take to complete the assignment are given below in pseudocode.

2.1 Initialization phase

Initialize variables/counters related to the queue status.

- `pkt_in_q = 0`
- `pkt_dropped = 0`

2.2 Simulating the events

Repeat for x times (that is, the number of events we want to simulate) Algorithm 1.

Data: λ (arrival rate), μ (departure rate), n (buffer size)
Result: `pkt_in_q`, `pkt_dropped`
begin

```
1   y=rand([0,1))
    if y < λ/(μ+λ) then                                  // arrival event
2       if pkt_in_q < n then
3           pkt_in_q++
        end
4       else
5           pkt_dropped++
        end
    end
6   else                                                 // departure event
7       if pkt_in_q > 0 then
8           pkt_in_q- -
        end
9   Write pkt_in_q and pkt_dropped in a file
10  end
```

Algorithm 1: Pseudo code for *event*

In line 1 you need to call a random number generator routine that returns a random real number in the range $[0, 1)$. Note that this is must a real number between 0 and 1 (including 0 but excluding 1); it is NOT an integer (i.e. you need to be able to generate numbers between 0 and 1, not just either 0 or 1). For example, Python's `random.random()` can be used here, but `random.randint()` will not give the correct results (https://docs.python.org/3/library/random.html).

3 Analyzing performance

After implementing the simulator you will experiment with it using different values for the input parameters (λ, μ, n). Therefore, you should not hardcode the parameters within the simulator's code. Instead, you should pass them to the program as command line arguments. For each combination of input parameters, you will be required to simulate at least 1,000,000 total events (i.e., $x = 1,000,000$).

3.1 Constant rates

In the first set of experiments, you will use constant input rates. Table 1 summarizes the different values of λ, μ and n that you are going to use[1]. Note that you will have to use all the different combinations (27 in total).

λ(pkt/sec)	30	80	120
μ (pkt/sec)	50	100	120
n (packets)	50	100	150

Table 1: Simulation parameters.

3.2 Variable input rate

In the second set of simulations, the input rate λ will vary over the events, while the departure rate stays constant at $\mu = 120$pkt/sec and the buffer size $n = 100$ packets. You will again have to simulate at least 1,000,000 events, and during the simulation λ will vary according to Table 2.

Events (%)	0-10	10-70	70-80	80-90	90-100
λ(pkt/sec)	70	200	130	120	70

Table 2: Variations of λ as the simulation progresses.

Table 2 should be interpreted as follows: "For the first 10% of the events the input rate will be 70 pkt/sec, for the next 60% of the events $\lambda = 200$pkt/sec" and so on.

4 Deliverables

You are required to deliver the following:

[1]Note that these number are just representative for the purposes of our simulations. If you are interested in more realistic numbers for the buffer sizing you are referred to [4].

- *Clean* event simulator code (i.e., structured and with comments) that will compile and run without modifications, and a text README file that clearly explains exactly how to compile (if needed) and run the code (**35%** of the total project's points).

- A report (**65%** of the total project's points) that will, (i) briefly describe the implementation of your simulator, and (ii) analyze the simulation results. In particular, for the second part, you are asked to provide figures for every scenario, for the number of packets in the queue and the number of packets dropped from the queue, versus the number of simulated events. Additionally, you will need to provide a (brief) explanation of the behaviors observed based on our discussions in class. Figure 2 shows a sample evaluation plot. We have 10 simulated events and the queue size is $n = 5$. The relation between the incoming and outgoing rate is such that the queue gets filled up quickly and packet drops start to be observed after the 6^{th} event (but, you will have to provide a more detailed explanation, and show results for 1 million events, not 10).

  To help you create your figures, we are providing a Python program that you can use/modify for this purpose (although you are also free to use a tool of your choice to create the figures). Note that in some cases, it may be helpful to create "zoomed-in" plots that focus on the initial behavior (e.g. the first 2000 events rather than all 1 million) to better understand the effects of the different parameters.

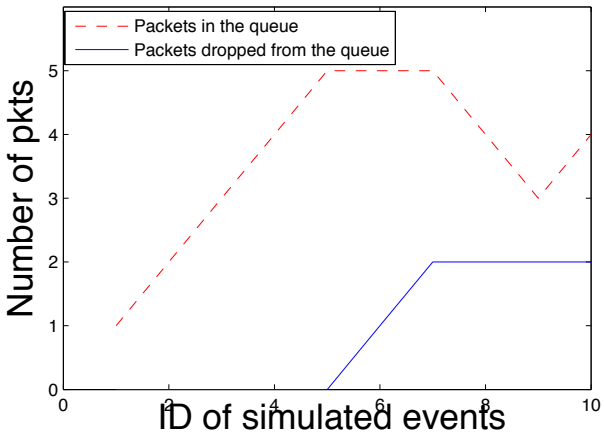

Figure 2: Sample evaluation plot.

The deadline is **Monday October 2, 2023 11:59pm**. Projects must be submitted on Canvas. As with the homework assignments, you may use your textbook and notes to complete this assignment, and may discuss *concepts* with other students or use online resources that help you better understand the concepts involved. You are NOT permitted to look at other students' solutions (including code), or online solutions/code for substantially similar problems. Similarly, you are not permitted to share your solutions with other students, or post your solutions online. Ask the instructor if you have any questions about this policy.

References

[1] L. Kleinrock, "Queueing Systems. Volume 1: Theory", Wiley-Interscience, 1975, ISBN 0471491101.

[2] L. Kleinrock, "Computer Applications, Volume 2, Queueing Systems", Wiley-Interscience, 1976, ISBN 978-0-471-49111-8.

[3] A. Law and W.D. Kelton, "Simulation Modeling and Analysis", McGraw-Hill College, 1999, ISBN 9780070592926.

[4] N. Buckshot, Y. Ganjali, M. Ghobadi, N. McKeown and G. Salmon, "Experimental Study of Router Buffer Size", in *ACM IMC*, 2008.