Exercice 1. Waiting queue

The goal of this exercise is to implement the communication channel seen on TD2.

We recall that the model is a communication channel with a buffer.
- Traffic Input: packets of random length (according to uniform distribution on \{1, \ldots, M\}), where \(T_n\) arrival date of the \(n\)-th packet follows a Poisson process with intensity \(\lambda\), i.e. \(T_0 = 0\) and inter-arrival \((T_n - T_{n-1})_{n \in \mathbb{N}}\) are i.i.d. of law \(\text{Exp}(\lambda)\).
- Server: FIFO service of rate 1 as long as there is work (transmission time of a packet = packet length).
- Queue: storage with \(\infty\) memory.

1. Implement a function \(\text{rexp2}(n, \lambda)\) simulating \(n\) experiment with exponential random variable with parameter \(\lambda\). You only have the right to use the distribution \(\text{runif}(n, \text{min}=0, \text{max}=1)\) of R.

2. Implement the full model and plot the number of packets and the total size of the buffer in function of the time.

Can you find a distribution it follows?

Exercice 2. Double

For this second part, we complexify a bit the model, by considering two identical servers, one sending the results of his computations to the other one.

1. Implement this model in the case where the output package length of the first server is uniformly random. Analyze the number of packets and the size of each queues.

2. Same question but with the output package length of the first server half of the size of its input.

3. Suppose now that we have the following setup:
   - the server \(s_1\) is as described in Exercise 1;
   - the server \(s_1\) sends its output data to an identical server \(s_2\), that also receives random data following the same law as \(s_1\).

Implement this model and analyze the number of packets in the buffers and the size of the queues.

4. Suppose now that \(s_1\) caches its output in a cache of size \(N\) (a parameter). Plot the times where the cache of \(s_1\) is flushed before the messages are treated by \(s_2\).

5. Same question if \(s_2\) treats the inputs from \(s_1\) with a higher priority than the other inputs.
6. The server $s_2$ is able to detect transmission errors (for instance, with a hash) from $s_1$. The transmission errors happen with a probability $\sigma$ for each bit of the message. When $s_2$ detects an error, it asks $s_1$ to send again the incriminated message, if $s_1$ has it in its cache, else the message is lost forever. Plot the number of messages lost in functions of all the parameters.

7. Suppose that all transmissions are subject to errors, with a probability of error depending on the channel.

The only channel that has a correction feature is the channel between $s_1$ and $s_2$ (described above). For which (interesting) values of the parameters is it better to send a message through $s_1$ for it to reach $s_2$?