Homework 4

Simulation and Performance Evaluation – University of Trento

DEADLINE: June 14, 2023

You can solve the following assignments using any programming language. In doing so, you are allowed to use *any* utility functions you wish to use.

Exercise 1

Consider the network scenario of Fig. 1. A source S wants to transmit a packet to destination D. A multihop network separates S from D. Specifically, there are r stages, each of which contains N relays. The source is connected to all nodes of stage 1. Each node of stage 1 is connected to all nodes of stage 2; each node of stage 2 is connected to all nodes of stage 3, and so on. Finally, all nodes of stage r are connected to D. The probability of error over every link in the whole network is equal to p.

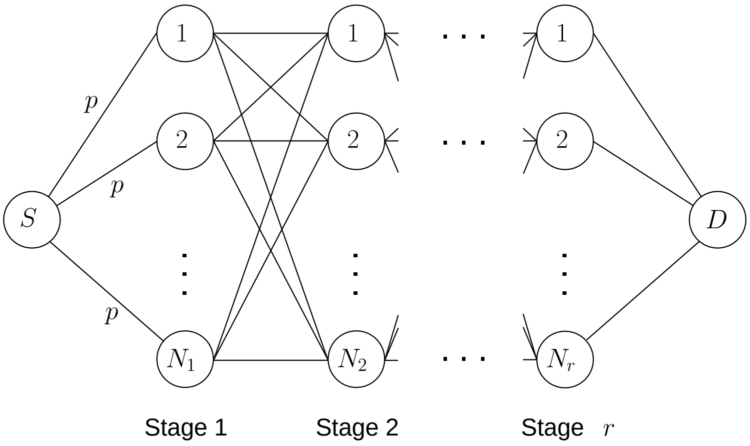

Figure 1: Reference scenario to simulate the flooding of a packet through a multihop network.

S employs a flooding policy to send its packet through the network. This means that every node that receives the packet correctly will re-forward it exactly once.

For example, at relay stage 1, the probability that any node will fail to receive the packet from S is p. However, say that k nodes at stage i receive the packet correctly: because of the flooding policy, all k nodes will retransmit the packet. Therefore, the probability that a node at stage $i+1$ fails to receive the packet is not p, but rather p^k (i.e., the probability that *no* transmissions from any of the k relays at stage i is received by the node at stage $i+1$).

1. Use Monte-Carlo simulation to estimate the probability that a packet transmitted by the source S *fails to reach* the destination D. Consider two different cases: $r = 2, N = 2$, and $r = 5$, $N = 5$. For each Monte-Carlo trial, simulate the transmission of the packet by S, the correct or incorrect reception by the relays at stage 1, the retransmission of the packet towards the next stages, and so forth until the packet reaches D or is lost in the process.
(*Hint*: remember that the probability to fail the reception of a packet is p^k, where k is the number of nodes that hold a copy of the packet at the previous stage.)
2. Repeat the above process for different values of the link error probability p. Plot the probability of error at D against p for the two cases $\{r = 2, N = 2\}$, and $\{r = 5, N = 5\}$. Plot also the 95%-confidence intervals (e.g., as error bars) for each simulation point.
3. Compare your results against the theoretical error curves provided in the file `theory_ex_flooding.csv` (column 1: values of p; column 2: probability of error at D for $\{r = 2, N = 2\}$; column 3: probability of error at D for $\{r = 5, N = 5\}$).
4. Draw conclusions on the behavior of the network for the chosen values of r and N.

5. Plot the average number of successful nodes at each stage, and the corresponding confidence intervals. What can you say about the relationship between the number of successful nodes and the probability of error at D?

6. *Facultative*: Repeat point 1 by applying post-stratification on your computed average probability of error. You can choose the number of relays that get the packet at Stage 1 as the stratum variable (i.e., you have $N+1$ strata, as the number of relays that get the packet correctly from the source can be $0, 1, \ldots, N$). How does your precision improve?

Exercise 2 (facultative)

Compare different Monte-Carlo methods for the estimation of π. In particular compare the following:

1. The naive estimator that computes $\pi/4$ as the ratio of points within a circle of radius equal to 1 centered on $(0, 0)$ to the total number of points drawn at random within a square of side length 2, also centered on $(0, 0)$;

2. The improved estimator seen in class that exploits conditioning and antithetic random numbers;

3. The improved estimator seen in class that exploits conditioning, antithetic random numbers, and stratification.

Set your preferred stopping rules in terms of

- the number of correct digits of π estimated by your method;
- the size of the 95% confidence interval for the estimated value of π.

Show how many iterations the above methods need in each case.

For each of the three above Monte-Carlo methods, it may also be curious to check the histogram of the values you compute the sample average of.