Discrete event traffic simulation

Laboratory Report 1

RTEL 2024/2025 **Class: 2MEEC_T01**

António Lopes, Pedro Duarte

1 Introduction

This report outlines the development of a Traffic Simulation model designed to replicate the behavior of a call-handling system through discrete events. It was created for the Telecommunications Networks course at FEUP, using Python as the programming language due to its simplicity and flexibility. The simulation relies on two primary statistical methods: the Exponential Distribution and the Poisson Arrival Process.

The Exponential Distribution approach calculates call arrival times by applying the formula:

$$t = \frac{-\ln(1-U)}{\lambda}$$

where U is a random value between 0 and 1, and λ represents the call arrival rate. The Poisson Arrival process, on the other hand, tracks time in fixed intervals and determines call arrivals based on λ and random values.

2 Methodology

The developed traffic simulation model consists of a python class, *Simulation*, which has the following parameters:

- λ - arrival rate (default: 5)
- μ - service rate (default: 2)
- *samples_nr* - desired number of samples (default: 1000)
- T - simulation time period (default: 1000)
- *max_resources* - maximum number of system resources (default: 500)
- *stop_condition* - stopping condition: no. of samples, time period
- *mode* - exponential, poisson

The class functions are used to implement an event-driven simulation, which generate call arrival and departure events, based on the simulation parameters. This events can be generated either by an Exponential Distribution process or by a Poisson Arrival process. The simulation was then run, both in exponential and poisson mode, using the default parameter values and using as the stopping condition different numbers of samples: 50, 100, 1000, 10000. The desired outputs of the simulation are:

a) the histogram of the interval between the arrival of consecutive calls.

b) e - the estimator of the average value of the interval between consecutive calls.

The source code of this simulation can be find in GitHub repository [1].

The next sections will present the results of these simulations and the main conclusions of this project.

3 Simulation Results

3.1 Exponential Distribution Results

In Figure 1 is shown the Exponential distribution mode histogram output results. For this simulations the resulting value of e was 0.182, 0.225, 0.200, 0.197, for simulations (a), (b), (c) and (d), respectively, being the theoretical expected value $1/\lambda = 0.2$.

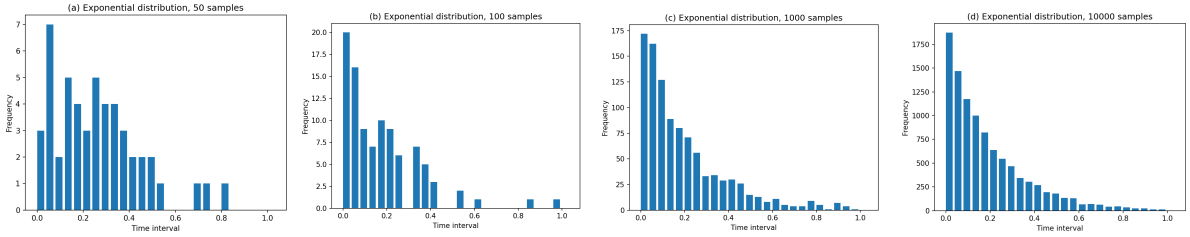

Figura 1: Exponential Distribution histogram outputs: Default parameters and number of samples: 50 (a), 100 (b), 1000 (c), 10000 (d)

3.2 Poisson Process Results

In Figure 2 is shown the Poisson process mode histogram output results. For this simulations the resulting value of e was 0.181, 0.224, 0.193, 0.196, for simulations (a), (b), (c) and (d), respectively, being the theoretical expected value $1/\lambda = 0.2$.

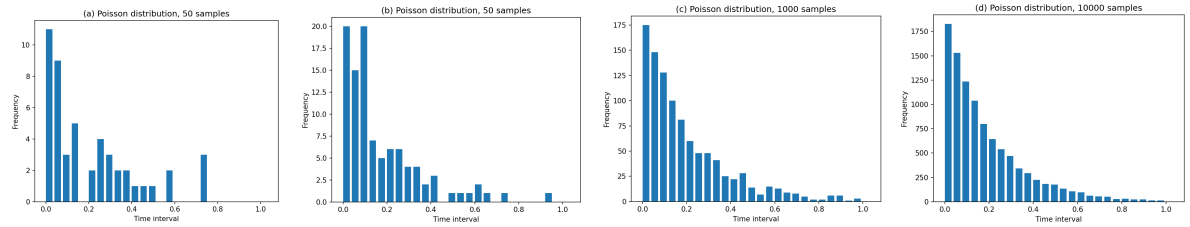

Figura 2: Poisson Process histogram outputs: Default parameters and number of samples: 50 (a), 100 (b), 1000 (c), 10000 (d)

4 Conclusions

The simulation showed that both the Exponential Distribution and Poisson Arrival processes produced similar results when the total number of calls was consistent. As the number of call attempts increased, the actual arrival times better matched the expected distribution, both expressed by the histogram function form and the estimator of the average time intervals. This means that an Exponential Distribution process can be used to simulate a Poisson Arrival process, which is beneficial since the Exponential mode run faster and consume less computational power than the Poisson mode.

In summary, these simulations confirm basic engineering principles, offering insights into network behavior under different conditions and showing that simple models can effectively simulate complex systems.

Referências

[1] P. Duarte and A. Lopes. GitHub Repository: `https://github.com/Pektro/RTEL`. Last accessed: October 2024.