Coursework 1: Experimental Verification of Little's Theorem

Kyeong Soo (Joseph) Kim
Department of Electrical and Electronic Engineering
Xi'an Jiaotong-Liverpool University
26/10/2018

I. Introduction

In this assignment, you will run a series of simulations for an M/M/1 queueing system, where you are to measure the average waiting time of the queueing system via two different methods.

One way is a direct measurement where the waiting times of packets are accumulated as they finish waiting in the queue; when the simulation completes, this sum is divided by the number of packets to obtain the average waiting time. Note that this approach is already implemented in the sample M/M/1 simulation program based on SimPy, which is available on the module homage page at the ICE.

The other way is an indirect measurement using the average queue length (i.e., the average number of packets in the queue) and Little's theorem; looking back at the graphical proof of Little's theorem shown in the slide #9 of "Basic Queueing Theory and Network Simulation", we can see that the average queue length is the area under the second graph divided by the period of simulation. This area is the sum of the rectangles; the height of each rectangle is the number of packets in the queue, and the width is the time interval between changes in this number. This suggests the following algorithm for computing the average queue length:

- Define variables sum, queue_length, and status_change_time.
- At the beginning of the simulation, initialise them, i.e.,
  - sum $\rightarrow 0$
  - queue_length $\rightarrow 0$
  - status_change_time $\rightarrow 0$
- These variables are updated as follows during the simulation:
  - At each packet arrival:
    - sum $\rightarrow$ sum + queue_length$\times$(env.now[1]-status_change_time)
    - queue_length $\rightarrow$ queue_length + 1
    - status_change_time $\rightarrow$ env.now
  - At the end of each packet's waiting in the queue:
    - sum $\rightarrow$ sum + queue_length$\times$(env.now-status_change_time)
    - queue_length $\rightarrow$ queue_length - 1
    - status_change_time $\rightarrow$ env.now
- At the end of the simulation, the average queue length is computed by dividing sum by the simulation period.

II. Task: Comparison of Two Different Measurements of Average Waiting and Service Time

- (10 points) First, you need to create a SimPy simulation program based on the sample one, where you implement the suggested algorithm for the computation of the average queue length and measure the average waiting time using Little's theorem. Then you need to run a series of simulations for the

[1]The current simulation time.

parameter values given below and compare the results for the average waiting time with those from the sample program by plotting a chart.

Below are the parameter values for the simulation of M/M/1 queueing system:

- Arrival rate[2]: 5, 10, 15, ...95
- Service rate[3]: 100 (fixed)

The chart should show clearly (* you may need to set y range properly in this regard *) the average waiting times from both measurements.

- (5 points) Second, you need to modify the simulation program from the subtask #1 and measure the average service time based on the average number of packets under service and Little's theorem. Again, you need to run a series of simulations for the same parameter values as in the subtask #1 and compare the results with the inverse of the service rate (i.e., the given average service time) by plotting a chart; unlike the subtask #1, you don't have to run another set of simulations for direct measurements this time.

You need to submit the Lab report in person to TA (Mr Xintao Huan) or to the assignment box in front of EB306 by 5 pm, Friday, 23/11/2018.

III. Deliverables

- Hard copy of a coursework report.
- Soft copy of source code for the two simulation programs (*via e-mail*).
- Soft copy of scripts for batch execution and plotting (e.g., matplotlib, gnuplot, and MATLAB) (*via e-mail*).

As part of the report, you must provide detailed instructions how to run the simulations so that TA can reproduce your results on his computer (as in the slide #64 of "Network Simulation Based on Python/SimPy").

[2]This is the inverse of the mean packet interarrival time.

[3]This is the inverse of the mean packet service time.