UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Department of Electrical, Electronic and Computer Engineering

ERP 420 - Practical 1
Queuing Theory
Investigation and Simulation v1.3

Jaco Marais
Adapted by: Llewellyn Strydom

Last updated: August 2, 2019

Contents

1 Change log

Part V — Changed the submission instructions. (29 July 2019)
Part V — Updated the submission instructions. (1 August 2019)

Part I
Overview

2 Objective

The objective of this practical is to improve the student's understanding of queuing theory with regard to design and application. The student will be expected to analyse and compare queuing theory models and conclude on their findings. The student will also simulate a model, and verify and discuss practical results using theoretical analysis.

3 Structure

This practical will consist of two parts. The first part will focus on the theoretical approach where models need to be investigated, designed and evaluated. This will form part of the experiments and data analysis learning outcome stipulated by ECSA ELO 5. The second part will require the use of knowledge gained in the first part to design and implement a queuing simulator. Both of these parts will be used to evaluate ECSA ELO 4 and 5.

Part II
Practical Questions

4 Analysis and Comparison of Queuing Models

4.1 Description

This question will focus on understanding how queuing theory works and how each of the different parameters have an influence on the model. There are three primary characteristics that need to be considered for this task: (1) arrival process, (2) service process and (3) number of servers. These characteristics are used to describe a simple queuing system using the following notation

- **A/S/n**.

Here **A** denotes the arrival process, **S** the service process and **n** the number of servers. A and S can be classified as either **M** (Markov) which indicates an exponential distribution, **D** (Deterministic) where all processes have the same value or **G** (General) which can be any arbitrary probability distribution. For simplicity, this practical will assume exponential distributions for both the arrival process and the service process. The influence that the number of servers (n) has on the model will be evaluated. The different queuing models that will be investigated and evaluated are

- *M/M/1*,
- *M/M/s*, and
- *M/M/∞*.

It is expected of the student to find and study information relevant to understanding these models. This includes resources provided in ERP 420, as well as any other useful information. Focus should specifically be placed on finding and understanding mathematical equations used to define each model since this will be helpful when comparing the models with each other.

4.2 What to Deliver

To keep things simple and help develop a clear understanding of each queuing model, use $\boldsymbol{\lambda}$ (average arrival rate) = 50 customers per minute and $\boldsymbol{\mu}$ (average service rate) = 65 customers per minute. These parameters will be used to calculate other model parameters defining the model's state. The following aspects should be calculated analytically for each of the three models:

Table 1: Queuing System Parameters.

Symbol	**Description**
U	Queuing utilization
ρ	Queuing intensity
P_0	Probability of an idle server
P_n	Probability of n customers in system
W_q	Average time customer spends in line waiting (Delay in queue)
W	Average time customer spends in the system (Delay in system)
L_q	Average number of customers waiting in line (Queue length in queue)
L	Average number of customers waiting in system (Queue length in system)

Once all the parameters have been calculated for each of the 3 queuing systems, the models need to be compared with each other. It is important to investigate and report on the influences of changing specific parameters such as the number of servers (which is the main investigation). The focus should be on optimization and justification of which models will work in which situations, this needs to be discussed in detail in the report.

Certain graphs are required for evaluation. Leaving any of these required graphs out will result in the loss of marks. The required graphs are

- Utilization (x-axis) vs Response time (y-axis) vs (single and multi-server comparison),
- Number of customers vs Throughput,
- Number of customers vs Response Time, and
- Number of customers vs Utilization.

Note: These graphs are to be generated for each of the models and is to be used in a discussion regarding the different models. ***For the graphing of the models you no longer need to use the fixed arrival rate. Think about how varying the arrival rate will influences the different parameters.***

The graphs must have all the models on one plot for easier comparison. The student should include at least 3 different values, excluding 1 and ∞, for the number of servers. Any other figures or graphs may be included for better demonstration of the student's interpretation of the models. Please note that each figure/graph must have a description or else they will be penalised.

To Summarise

Please investigate each of the models and through comparison explain which models work best in the relevant situations.

- Analytical calculations of each model (including the relevant equations)
- Required graphs
- Compare models
- Give scenarios where models would perform best/worst
- Focus on optimization in comparison (e.g. no. of servers vs no. of customers)

Note: The number of servers in the M/M/s model, as well as, the number of customers are not defined and can be whatever the student deems sufficient in order to ensure a better evaluation.

5 M/M/1 Queuing Simulator

5.1 Description

In this question you will simulate a lossless *M/M/1* queuing system. This question assumes that you understand the basic concepts of a queuing system and will depend on knowledge obtained from the previous question. This question will focus on the actual application of queuing theory, making use of packets and a communication link with a limited, predefined bandwidth. To recap, we know there are two important factors influencing queuing systems, to put these into context they are described as follows.

5.1.1 Arrival Rate

This denotes the average number of packets coming from the incoming link in a unit time. For example, if there are 1 000 packets, on average, arriving in a timespan of 1 s, then the average inter-arrival time is 1 ms. However, the inter-arrival time is not always uniform. In this case the inter-arrival time is assumed to follow some exponential distribution denoted by the first M in the Kendall notation for this system.

5.1.2 Service Rate

The second factor is the service rate, which is dependent on the packet size distribution and the capacity of the outgoing link. It is defined by

$$\text{service rate} = \frac{\text{capacity}}{\text{average packet size}}. \tag{1}$$

Example: If the outgoing link has a transmission capacity of 10 Mbps and the average packet size is 1 000 bits, then the average service rate will be 10 000 packets per second. Here we also assume that the service rate has an exponential distribution, as indicated by the second M in the Kendall notation.

5.2 The Simulator

Firstly, you will need to develop a simulator, which must be done in the Python programming language. The simulator should be capable of reading a trace file containing the inter-arrival times and packet sizes for several packets. Each line of the trace file will contain a specific packet's inter-arrival time and its size. An example of the contents of the trace file is shown in Table 2 and described below for a system with a transmission capacity of 10 Mbps.

Table 2: Trace File Format.

Packet inter-arrival time (in μs)	**Packet size (in bits)**
300	500
20	200
100	900

The first row: The first packet arrives at the system at time 300 μs, and its size is 500 bits. Since the outgoing link is idle when the packet arrives, the packet is transmitted immediately, with the transmission time calculated as 500 bits/10 Mbps = 50 μs. Thus, the first packet has queuing delay 0 μs, and its transmission is completed after time 350 μs.

Note: It can be assumed that time starts at 0 s.

The second row: The second packet's inter-arrival time (compared to the previous packet) is 20 μs. Its absolute arrival time is thus 320 μs (300 μs + 20 μs). It has to **WAIT** in the queue while packet 1 is being transmitted. The transmission of packet 2 starts at 350 μs (as soon as packet 1's transmission is completed). Packet 2's queuing delay is therefore 350 μs - 320 μs = 30 μs. Its transmission time is 200 bits/10 Mbps = 20 μs. Thus, packet 2's transmission is completed at time 370 μs (350 μs + 20 μs). Its queuing delay is 30 μs and its response time is 50 μs (30 μs queuing delay + 20 μs transmission time).

The third row: The third packet has an inter-arrival time of 100 μs. Its absolute arrival time is 320 μs + 100 μs = 420 μs. At that time, the link is idle (packet 2's transmission is completed at time 370 μs). Thus, it does not incur any queuing delay. Its transmission time is 900 bits/10 Mbps = 90 μs. Packet 3's transmission is completed at time 510 μs (420 + 90).

A trace file will be provided on ClickUP and can be used to test whether your simulator is implemented successfully. **Assume a link capacity of 1 Mbps.** It is expected that the simulator use the data extracted from the trace file to calculate the average arrival time (λ) and average service time (μ), as well as, the average queuing delay of all the packets.

Once the simulator is capable of successfully reading a trace file and calculating the relevant model parameters, it should be expanded so that it can generate the inter-arrival time and packet size distributions by itself. These distributions should be for the *M/M/1* queuing model. The simulator can be altered so that it accepts 2 input parameters, the link capacity, and the average packet size. This means that the simulator should be able to provide the required outputs no matter what the inputs are (any link capacity and service rate).

Note: Inter-arrival times need to be converted to absolute arrival times for scheduling.

The simulator is expected to measure the average queuing delay when the outgoing link experiences certain workloads, say at 10%, 20%, ..., to 90%. The workload can be varied by varying the arrival rate, while the link capacity should stay fixed. Make sure that the queue is large enough so that no packet will be dropped. Since the inter-arrivals and packet sizes are randomly generated, for each workload, the simulator needs to be run multiple times (at least 100 times). Each run will consist of 100 seconds of traffic (this does not mean your simulator has to run 100 seconds) in order to calculate a meaningful average queuing delay.

5.3 What to Deliver

To summarise the expected outputs of this question:

- The simulator should be designed so that it can read data from the trace file and calculate the average queuing delay, as well as λ and μ. These values should be included in the final report.
- Plot a queue length vs time graph for the given trace file.
- Generate a trace file with $\sim$100 seconds of traffic.
- From the generated trace file use a given link capacity and average packet size to calculate the average queuing delay at the respective workloads. **Use a link capacity of 100 Mbps and an average packet size of 1000 bits.**
- Plot an average queue length vs time graph for the simulated runs.
- Compare different workloads and discuss how this affects λ and μ. Explain the relationship between λ, μ and the workload.

Part III
Evaluation

There will be a practical demonstration for this practical on **Friday, 16 August 2019** to demonstrate that your simulator is functional. You will be provided with a trace file on the morning of the demonstration to test your simulator's calculation capabilities. You will also be required to generate your own trace file.

The formulae and calculations for the first part of this practical assignment must also be shown at the demonstration.

Part IV
Report

6 Format

The report should be a clear and concise document that highlights the theoretical aspects of the system, as well as describing the actual implementation. Marks will be deducted for poor language or spelling errors. The EPR 400 report writing standards should be followed, figures and tables must be numbered with captions (use the built in LaTeX methods to achieve this).

The report should be no longer than **12** pages. You will lose 10 % if your report exceeds the page limit. Do not start new sections on the next page but rather below the previous section.

7 Content

The report should include the following:

- The cover page must use the template provided under ERP 420 on ClickUP. This cover page must be signed and fully completed otherwise the assignment will not be graded or regarded as submitted.
- Introduction that describes the practical.
- Background and understanding of theoretical concepts, focus on the various parameters to be calculated and evaluated.
- Details regarding the design and implementation of each functional part in the system.
- Analytical comparison in question 1 with the relevant details discussed in the question.
- Simulator design and relevant details discussed in the question.
- Conclusion.

Don't have a table of contents or a list of figures or tables

Part V
Additional Information

Everyone is required to submit 2 soft copies of the practical report and their source code: one on ClickUp and one on EPS (`https://eps.ee.up.ac.za/`).

Please upload a single PDF that contains your practical report, as well as your source code as an appendix, to TurnItIn (done through ClickUP). ***DO NOT*** *include the cover page/plagiarism statement in this submission.*

Please upload your report (with the cover page/plagiarism statement) and all the relevant code files to the EPS.

Both submissions needs to take place before **Friday, 30 August 2019 at 23h59**.

It is the responsibility of each student to ensure that both submissions are successful (make sure you receive both TurnItIn and EPS email receipts as proof of submission). If the student only submits on a single platform (TurnItIn or EPS), it will be treated as a non-submission.

Please feel free to email me at *up.erp420@gmail.com* if you have any questions.