horizontal line

DES PROGRAM REPORT

**Yu-Ho Hsieh, Cheng Zhang**

# **SECTIONS**

1. Overview
2. Introduction of queueing network
3. Goals of study
4. Implementation of simulation library
5. Implementation of the configuration program
6. APIs
7. Testing
8. Results and analysis of experiments
9. M/M/1 Survey
10. Reference

# **OVERVIEW**

This is a study to simulate the queueing network through a discrete event simulation program. The report contains three parts, introduction to the study, method/implementation of study and analysis of results. The simulation library is implemented by Cheng Zhang and the configuration program is implemented by Yu-Ho Hsieh.

# **INTRODUCTION OF QUEUEING NETWORK**

Queueing network is a system model to simulate the life cycle of a set of service centers provide services to a collection of customers (S Balsamo, 2007). The life cycle includes customer generation, customers distribution to different service stations to enjoy services, line up, and exit system.

# **GOAL OF STUDY**

The goal of this study is to experiment different inter-arrival time of customers on a food court system to collect the results of the time for each customer to stay and wait in the system, as well as the waiting time of each queue station. After the experiment, we can have a rough idea regarding how fast we should allow customers to get into the food court system and which queue station may be the bottleneck to cause long waiting time.

# **IMPLEMENTATION OF SIMULATION LIBRARY**

This simulation library is intended to implement the functions as follow:

1. Build and return entities for network.

2. Implement the function for each entity.

3. Make event Handlers to manage the entities created.

4. Maintain variables to collect statistics and create function to expose them.

5. Build the simulation engine, executing simulation and destroying simulation.

# **IMPLEMENTATION OF CONFIGURATION FILE**

The configuration file can be separated into three parts. The first part is read and validate the configuration file provided by users; the second part is to build the queueing network using simulation library; the third part is to store the statistics results to the output file. The validation message can be found in the log file in txt folder.

The part which worth mention is the configuration validation part. We list a bunch of configuration file test.

Valid component parameters:

1. The number of components should be aligned with the given number. The type should be known, and the ID should be unique.
2. Parameters in components should be either positive integer or double.
3. The correct number of parameter of the corresponding different components, also the type of them.

Valid connection

1. G should not be any predecessor component of anyone
2. G should not directly connect to X

# **APIs**

Below is the code block for simulation library API. Details explanation can be found in folder src. Generally speaking, the simulation API contains two parts. One part is for creating a queue network component and run the simulation. The other part is to extract the statistics results of the experiment.

|  |
| --- |
| //statistic functions void get\_sys\_data(double\* avg\_wait, double\* max\_wait, double\* min\_wait); void get\_queue\_data(int queue\_id, double\* avg\_wait, double\* max\_wait, double\* min\_wait); void get\_customer\_data\_count(int\* number\_enter\_system, int\* number\_exit\_system); void get\_customer\_data\_time(double\* avg\_time\_stay\_in\_system, double\* max\_time\_stay\_in\_system, double\* min\_time\_stay\_in\_system); void get\_customer\_data\_wait(double\* avg\_time\_wait\_in\_system);  //simulation functions int create\_generator(int generator\_id,double P,int D); int create\_queue(int queue\_id, double P,int D); int create\_fork(int fork\_id, int K,double\* P, int\* D); int create\_exit(int exit\_id); int init\_simulation(int component\_num); void destroy\_simulation(void); void run\_simulation(double end\_time); |

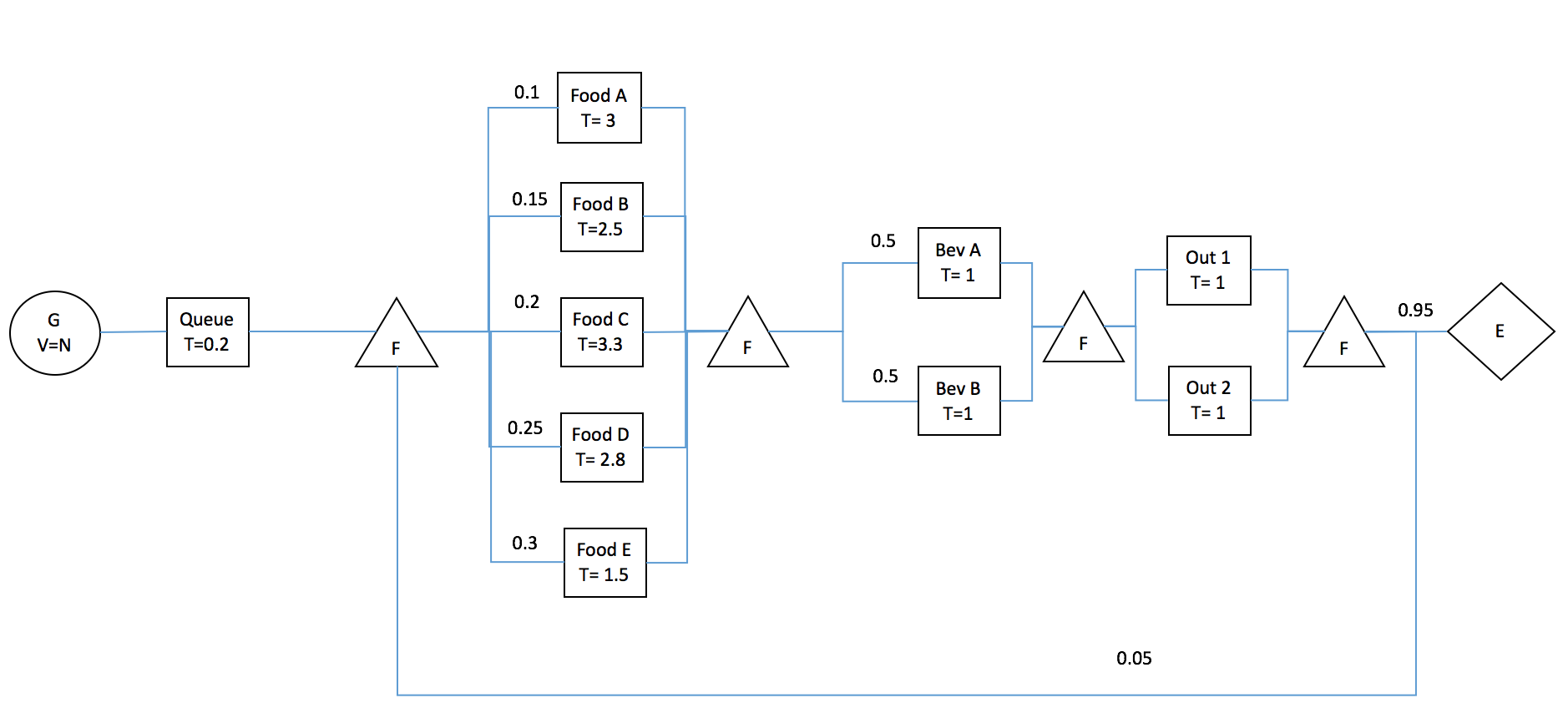
# **TESTING**

The simulation library is tested with a self-made queueing network test.c to see whether all the functions can work normally.

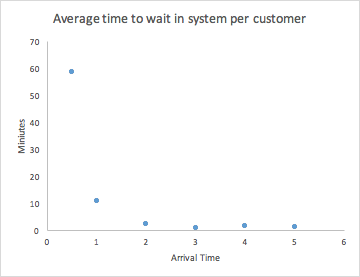
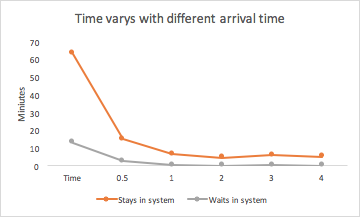
The configuration program is tested using different configuration files and the outputs are logged into a log file.

# **RESULTS & ANALYSIS OF EXPERIMENTS**

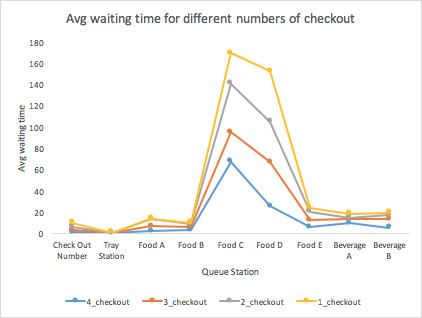
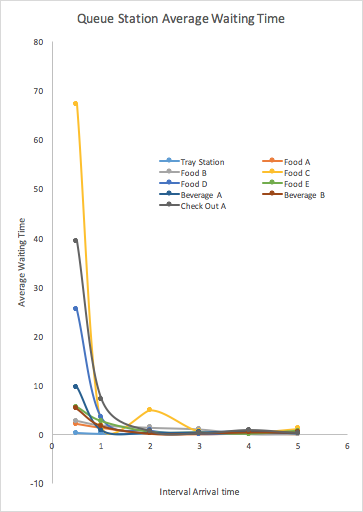
We build the food court experiment according to the description from the assignment hangout. The following chart is the food court illustration.



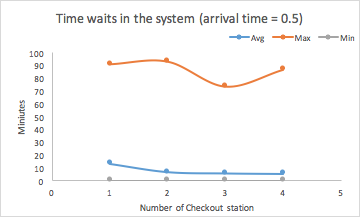
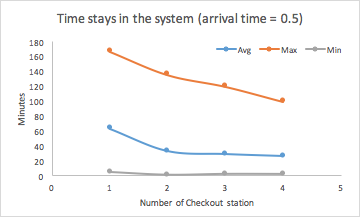
After runnign different parameters, the findings are listed as bullet points



1. The shorter the inter-arrival time, the longer the customer need to stay and wait in the system, which aligns with our expectation. The more customer in the system, the longer they have to wait to be served.

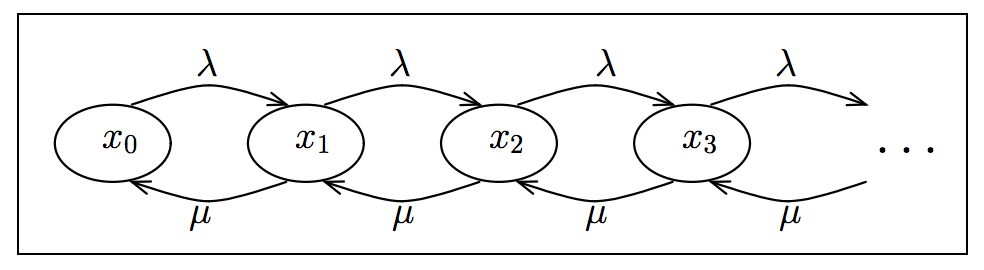


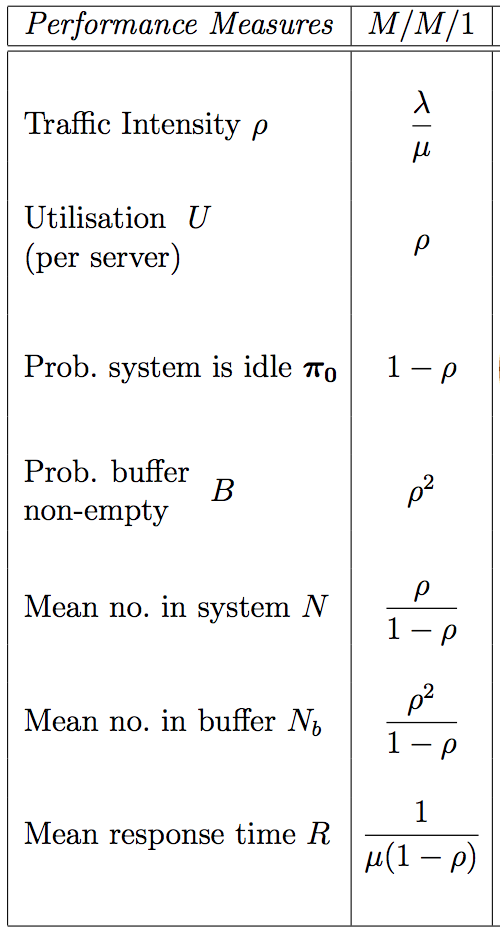
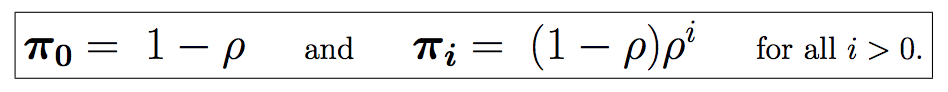
1. The queue station bottleneck is food C station and food D station which align with our config file. They have higher probabilities to serve as the destination to order food.



1. The more checkout station, the lower time for customers to stay and wait in the system, which also aligns with our expectation. However, there is a weird turning point in the max time waits in the system for checkout station equals to 3 and 4. That may result from the outlier max serve/waiting wait generated from the exponential distribution. However, keep increasing the number of check-out stations cannot decrease the waiting time forever because the waiting bottle neck may still be food station C. Under the conditional that the arrival time equals to 0.5, 2 or 3 check-out stations makes the best trade-off between waiting time and cost.
2. The detail of experimental data can be found in the documentation folder.

# **M/M/1 INTRODUCTION**

The M/M/1 queueing system is a single generator, single queue system with inter-arrival time following Poisson process and service time following exponential process. We can have a state variable which is the number of customer in the system and having the following chart. (Jane Hillston, 2009) X represents state variable and λ represents generation rate, µ represents services rate.

From the chart, we can derive the following state distribution equation. **Π** represents states variable and ρ equals to λ / µ. In the end, we can have this table. The detailed proof and explanation are in the documentation written by Jane. (Jane Hillston, 2009)

# 

# **EXPERIMENT USING M/M/1**

G: 2 (arrival rate 0.5), Q: 1 (serving rate 1)

The time that customer stays in system: Avg: 1.81

The time that customer waits in system: Avg: 0.83,

The waiting time of whole system: Avg: 0.83

The customer count for system: In: 4989, Out: 4989

Queue1 waiting stats Avg: 0.83

Here the arrival rate , and the serving rate .

As is listed above, the average time that customers stay in the system is ; The average time that customers wait in the system is

The comparison of the theoretical results and practical results are listed below:

|  |  |  |
| --- | --- | --- |
|  | average time customers stay in the system | average time customers wait in the system |
| Theoretical value | 1.81 | 0.83 |
| Practical value | 2 | 1 |

G: 2 (arrival rate 0.5), Q:3 (serving rate 0.33), the number of lining up people keeps increasing

The time that customer stays in system: Avg: 1671.54

The time that customer waits in system: Avg: 1668.59,

The waiting time of whole system: Avg: 1668.59

The customer count for system:

In: 5034, Out: 3389

Queue1 waiting stats Avg: 1668.59

The arrival rate , and the serving rate .

We can see that here, which means the serving ability of this system is weaker than the arrival ability. In this case, the amount of the queueing customers in this system may tend to infinity as the time tends to infinity, and the average waiting time or average staying time will accumulate unlimitedly as time grows, thus the system is unstable. The expectances value for the average staying time and the average wait time can’t be calculated in an unstable system.

G:3 (arrival rate 0.33) Q:2 (serving rate 0.5)

The time that customer stays in system: Avg: 6.21

The time that customer waits in system: Avg: 4.14,

The waiting time of whole system: Avg: 4.14

The customer count for system: In: 3325, Out: 3323

Queue1 waiting stats Avg: 4.14

The arrival rate , and the serving rate .

As is listed above, the average time that customers stay in the system is ; The average time that customers wait in the system is

The comparison of the theoretical results and practical results are listed below:

|  |  |  |
| --- | --- | --- |
|  | average time customers stay in the system | average time customers wait in the system |
| Practical value | 6.21 | 4.14 |
| Theoretical value | 6 | 4 |

As we can see in case1 and case3, the theoretical values are close to the practical values when the experiment time is large enough, thus the validity of the code is proved.

# **REFERENCE**

### 

1. Balsamo S., Marin A. (2007) Queueing Networks. In: Bernardo M., Hillston J. (eds) Formal Methods for Performance Evaluation. SFM 2007. Lecture Notes in Computer Science, vol 4486. Springer, Berlin, Heidelberg
2. Jane Hillston hjeh@inf.ed.ac.uki. September 17, 2009.