Applying Queueing Theory for the Optimization of a Banking Model

Kevin W. G. Cowdrey, Jaco de Lange, Reza Malekian, Johan Wanneburg, Arun Cyril Jose

Department of Electrical, Electronic and Computer Engineering,
University of Pretoria, South Africa
u12066754@tuks.co.za, u12007146@tuks.co.za, reza.malekian@ieee.org,
johan.wannenburg@gmail.com, poonjarian@outlook.com

Abstract

The purpose of this paper is to investigate waiting times at banking firms to design a system to optimise the overall banking experience. Queueing analysis and queuing theory will be investigated and applied in order to improve customer experience whilst maximising profits. Different queueing strategies will be implemented using waiting time as a performance measurement. In order to find the most efficient solution the following queuing methods will be investigated: First in First out (FIFO), Last in First out (LIFO), Shortest Job First (SJF), Longest Job First (LJF), most profitable job first and priority queues. After investigation, design and simulation, queuing strategies will be implemented on a real world banking scenario whilst introducing banking traffic at different times of the day as well as disturbances such as tellers going offline. A practical design includes a Field Programmable Gate Array (FPGA) in order to simulate the banking scenario. The FPGA approach will provide a mobile solution for the optimization of queueing in a banking firm which could later be implemented in a banking firm for a better banking experience. Based on results obtained, the SJF method produces the best customer satisfaction, whilst most profitable job first queues maximizes profits.

Keywords: Field programmable gate arrays, Optimization, Profitability, Queueing analysis, Scheduling algorithms

1 Introduction

Queuing is a very important study in modern day society and is presented within various fields from telecommunications to normal queuing at toll gates, dentist offices, banks and much more. A queue is formed when arrivals to a system exceed the number of requests the system can service per unit of time, thus when a system is congested it is said to contain a queue. Within this paper the system to be addressed will be in the context of a banking firm, where arrivals to the system is customers which need to be serviced in a timely manner to reduce waiting time whilst maximizing profits for the banking firm. The main research question to answer is if the time spent at banks could be improved if a queuing system is implemented. Important elements that needs to be defined in queueing theory is: arrival rate (λ), service rate (μ) and utilization (ρ) [1-2], where in the context of the banking scenario the arrival rate is the number of customers that arrive at the bank per unit of time. The service rate is the number of customers that is serviced per unit of time and the utilization is the efficiency of work performed by the banking system or in other words the arrivals divided by departures or serviced clients [3-4]. The system will not be effective when the utilization is more than one, since a utilization more than one indicates that the number of arrivals is greater than the number of clients serviced per unit of time.

When implementing queueing in an intelligent manner waiting times and customer satisfaction will be improved [5]. In order to improve queueing systems it must be understood in-depth and queuing theory is used to do this, which leads to a Queueing Management System (QMS). A QMS can be implemented to manage the efficiency of a queuing system, which is the objective researched within this paper. The relationship between customer service, efficiency and quality within a queueing system can be seen within Figure 1. A QMS will analyse system parameters and adapt the system accordingly.

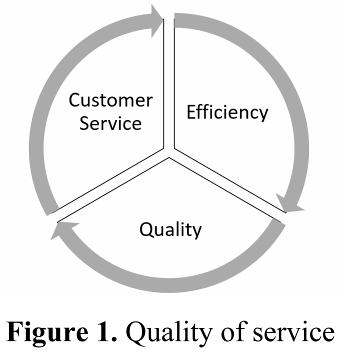

Figure 1. Quality of service

[*]Corresponding Author: Reza Malekian; E-mail: reza.malekian@ieee.org

DOI: 10.3966/160792642018031902007

The paper is structured as follows. In Section 2, an overview of current methods and existing research is given. Section 3 describes the theoretical method used. In Section 4, the simulation results are discussed. Section 5 outlines a real world implementation as well as theoretical simulations. In Section 6, a practical design is introduced using an FPGA. Section 7 outlines the discussion of results whilst conclusions are given in Section 8.

2 Existing Research

Previous implemented methods has been researched to improve specific aspects of the banking environment. Using a remote and local service could provide the QMS with better performance [6]. Using an M/M/1 queue simplifies the modelling process [7]. The round-robin method of [8] will not be implemented, as customers need to be serviced in order, until successfully helped. It is known that queueing delay could have a negative impact on the performance of a system [9] thus different servicing models will be introduced into the system to measure performance improvements. There are implementations to improve service quality [10], efficiency of the tellers [10], the service time [11], and more recently the queue length and waiting time of the banking scenario [12]. In [10] and [11], software simulations are performed, but this paper will improve by adding hardware in the form of an FPGA as local element as in [6].

3 Method

A good every day example of a queueing system is a bank. It is thus chosen to investigate how using different servicing and scheduling methods will affect the efficiency of the overall waiting time and throughput of the customers arriving and leaving the bank [13]. One important factor to form the basis of the implemented QMS is the fact that every customer will arrive at the bank requesting a specific banking service. Each service will have a different service time thus knowing the chosen banking service an informed decision can be made regarding the queue for the customer to follow. The main objective of this investigation is to see how a QMS can improve the efficiency of a bank when using queueing and scheduling techniques. The arrival times of the queueing system is dependent on the time of the day. This is chosen as a design parameter as it is known that banks are busier at certain times of the day as seen in Figure 2.

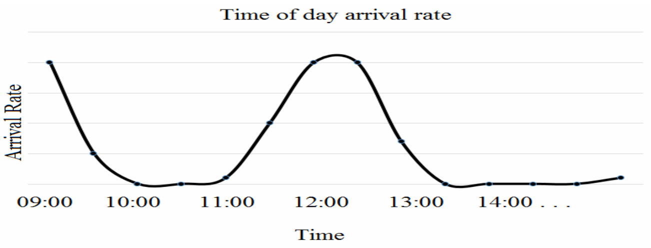

Figure 2. Arrivals as the day progresses

The average service time will depend on the type of the service that the customer wants to be completed. The customers will walk in to the bank and select their reason for visiting the bank on a machine using Table 1 [14]. The customer will then receive a tag which will indicate if it is the customer's turn to be serviced. The jobs that would have the smallest service time is completing an account statement and the largest services will be completing a loan or creating a new account.

Table 1. Banking activities

Class	Banking activity	Time (min)
1	Banking statements	0-2
2	Deposits and withdrawals	2-5
3	General banking activities	5-15
4	New accounts	15-30
5	Loan or complicated banking duties	> 30

The servers in this scenario will be the cashiers in the bank. The maximum number of servers (m = 5) will not be available all the time. In Figure 2 the arrival rate is slow during certain times of the day thus there will be a minimum number of cashiers to serve the customers. A bank will have the highest arrival rate in the morning and at lunch time. To achieve optimal efficiency the maximum number of cashiers should be serving customers. As seen in Table 2.

Table 2. Number of servers throughout the day

Time of the day	09:00-10:00	10:00-11:00	11:00-13:30	13:30-15:30
Number of servers	5	3	5	3

Changing the number of servers (m) will have an impact on the utilization as seen in (1).

$$\rho = \lambda / m\mu \tag{1}$$

The average waiting time (W) including the probability of a number of (Q) customers in the system (P_Q) can be seen within (2).

$$W = P_Q \rho / \lambda(1-\rho) \tag{2}$$

Different scheduling methods will be implemented on the banking model. The effects of the scheduling methods will be measured by comparing the utilization, service rate and the average waiting time. The different methods are: FIFO, LIFO, SJF, most profitable job first and priority queues [15-17].

4 Simulation Results

In order to create a simulation to compare different scheduling techniques a trace file containing clients to visit the bank needs to be created. The trace file is created using an algorithm to generate random arrival and service times as seen in Figure 3 and Figure 4 and then grouping each customer into a specific banking activity using the classes defined in Table 1 [14].

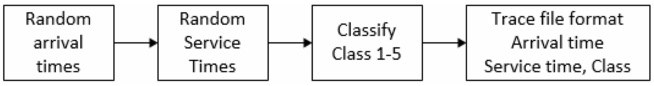

Figure 3. Process of creating the trace file

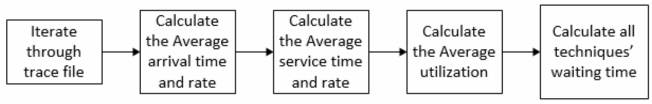
Iterate through trace file
Calculate the Average arrival time and rate
Calculate the Average service time and rate
Calculate the Average utilization
Calculate all techniques' waiting time

Figure 4. Process to read data from trace file

The results will be obtained by averaging over a large number of random arrival iterations in order to get statistical valid data. 10000 iterations will be executed to collect the data. The results in Table 4, Table 5, Table 6 and Table 7, demonstrates the difference between the different scheduling techniques. These tables are assumed as optimal conditions, thus no faults or offline servers (cashiers). Within Table 3 the general performance measurements for the generated trace file can be seen. It should be noted that these results are also averaged results.

Table 3. General performance measurements

Property	Fast arrivals	Slow arrivals
Arrival time between customers (min)	3	10
Arrival rate (customers/min)	0.3364	0.0985
Arrival rate (customers/hour)	20	6
Service time (min)	15	20
Service rate (customers/min)	0.0666	0.0505
Service rate (customers/hour)	4	3
Utilization (%)	100	39

Within Table 3, the performance measurements for fast and slow arrivals are defined to be 3 and 10 minutes between arrivals respectively. This is used to define the other variables within Table 3. Implementation of a FIFO queue will lead to a method where the customers are served in the order of arrival. This discipline is a normal queue with the results obtained in the simulation of the FIFO queue shown in Table 4 [15-17].

Table 4. Simulation results for the FIFO queue

Property	Fast arrivals	Slow arrivals
Waiting time (min)	39.238	0.0279

A LIFO queue functions on the principle that the newest customer to arrive in the queue will be served first. This discipline serves customers from the back of the queue. The results for this method can be seen within Table 5 [15-17]. This will not be a viable solution for the banking scenario, because this will dissatisfy customers that entered the queue first [18].

Table 5. Simulation results for the LIFO queue

Property	Fast arrivals	Slow arrivals
Waiting time (min)	39.48	0.0253

When comparing Table 4 and Table 5 it can be seen that FIFO and LIFO converges to the same results if enough averaging is applied. Waiting times for both methods converges to 39 minutes for fast arrivals, and 0.02 minutes for slow arrivals.

In the SJF queueing method the customers with the shortest job will be served first e.g. account statements. This discipline decreases the average service time but each customer's reason for being in the bank has to be known in order to now their time within the bank. The results for the SJF method can be seen within Table 6 [15-17]. SJF is more efficient than FIFO and LIFO since the shortest jobs are serviced first saving overall time and improving overall customer satisfaction.

Table 6. Simulation results for Shortest Job First method

Property	Fast arrivals	Slow arrivals
Waiting time (min)	24	0.025

The LJF queueing method functions that the customers with the longest job will be served first e.g. loans and new accounts. Each customer's reason for being in the bank has to be known in order to now their time within the bank and the results for the LJF method is shown within Table 7 [15-17].

Table 7. Simulation results for Longest Job First method

Property	Fast arrivals	Slow arrivals
Waiting time (min)	68	0.0264

The most profitable job first method functions on the concept that jobs that are profitable for the bank are serviced first. The activities and waiting times can be seen in Table 1. The order from most to least profitable activates in Table 1 is: 5, 4, 2, 1 and 3. The simulation results for the most profitable job first scheme is shown in Table 8 [15-17].

Table 8. Simulation results for most profitable job first

Property	Fast arrivals	Slow arrivals
Waiting time (min)	59	0.0263

The Priority queueing method functions that each service will receive a priority rating. The order of priorities from Table 1 is: 3, 4, 5, 1 and 2 but each customer’s reason for being in the bank has to be known. The simulation results for priority queues can be seen in Table 9 [11-12, 14].

Table 9. Simulation results for priority queues

Property	Fast arrivals	Slow arrivals
Waiting time (min)	66	0.0264

When comparing the results from Table 4, Table 5, Table 6, Table 7, Table 8 and Table 9, it can be observed that for optimal conditions the SJF method will have the shortest waiting times for fast and slow arrivals. The shortest waiting time in a queue will provide the highest customer satisfaction [18]. The most profitable job method will yield the most profit for the bank but sacrifices the waiting times as well as the customer satisfaction.

5 Real World Implementation and Simulation

Using the results from Table 4, Table 5, Table 6, Table 7, Table 8 and Table 9, a theoretical comparison can be made between the different queueing strategies [19-20], however a real world simulation should include disturbances in the banking model including: cashiers going offline (Table 2) and banking traffic deviations as described in Figure 2. The QMS should adapt to disturbances as in Figure 5.

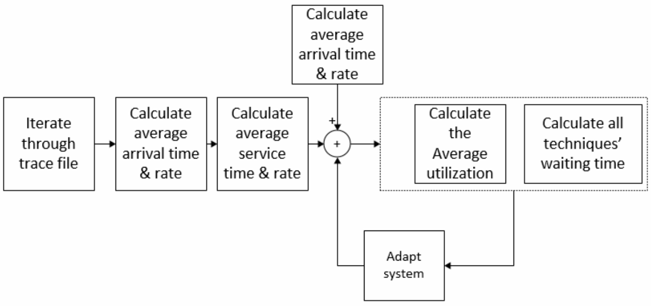
Calculate average arrival time & rate
Iterate through trace file
Calculate average arrival time & rate
Calculate average service time & rate
+
+
Calculate the Average utilization
Calculate all techniques' waiting time
Adapt system

Figure 5. Method for QMS to dynamically adapt to disturbances

The first disturbance introduced into the system is cashiers going offline as described in Figure 6. Fast and slow arrivals for different number of cashiers will be compared for all of the queueing models. It should be noted that cashiers and servers are an interchangeable term in the banking scenario.

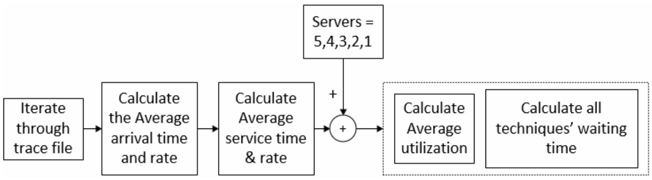
Servers = 5,4,3,2,1
Iterate through trace file
Calculate the Average arrival time and rate
Calculate Average service time & rate
+
+
Calculate Average utilization
Calculate all techniques' waiting time

Figure 6. Performance measurements for the banking model by iterating through the number of cashiers

Within Table 10 the performance of the FIFO, LIFO and the SJF queueing methods are compared to each other by varying the number of servers between 1 and 5 for slow and fast arrival rates. It can be observed that the SJF method is performing the best, this is due to the fact that the customers with the fastest banking activity is given a priority over the customers with longer banking activities to increase customer satisfaction. Within Table 11 the performance of the LJF, most profitable job first and priority queueing methods are compared to each other by varying the number of active servers between 1 and 5. The best customer satisfaction will be achieved if the bank implements the SJF method. Customer satisfaction is of utmost importance. At quiet hours the most profitable job first method can be used in conjunction with the priority queue method to boost profit, but the SJF method should be used if the arrival rate of customers are fast.

Table 10. Queueing methods performance measurements

Method	Servers	Waiting time (min)	ρ	λ	μ
FIFO slow	1	87	1.00	0.10	0.10
	2	15	0.91	0.10	0.05
	3	1.5	0.65	0.33	0.10
	4	0.2	0.49	0.09	0.05
	5	0.02	0.39	0.09	0.05
FIFO fast	1	156	1.00	0.33	0.32
	2	122	1.00	0.33	0.16
	3	91	1.00	0.33	0.01
	4	63	1.00	0.33	0.08
	5	39	1.00	0.33	0.07
LIFO slow	1	70	1.00	0.09	0.09
	2	13	0.87	0.09	0.05
	3	1.66	0.65	0.09	0.05
	4	0.21	0.49	0.09	0.05
	5	0.02	0.39	0.09	0.05
LIFO fast	1	155	1.00	0.33	0.32
	2	122	1.00	0.33	0.16
	3	92	1.00	0.33	0.10
	4	64	1.00	0.33	0.08
	5	39	1.00	0.33	0.07
SJF slow	1	61	1.00	0.09	0.09
	2	11	0.91	0.09	0.05
	3	1	0.65	0.09	0.05
	4	0.19	0.49	0.09	0.05
	5	0.02	0.39	0.09	0.05
SJF fast	1	112	1.00	0.33	0.32
	2	82	1.00	0.33	0.16
	3	58	1.00	0.33	0.11
	4	38	1.00	0.33	0.08
	5	24	1.00	0.33	0.07

Table 11. Queueing methods performance measurements

Method	Servers	Waiting time (min)	ρ	λ	μ
LJF slow	1	113	1.00	0.33	0.33
	2	22	0.91	0.10	0.05
	3	2	0.65	0.09	0.05
	4	0	0.49	0.09	0.05
	5	0	0.39	0.09	0.05
LJF fast	1	170	1.00	0.33	0.32
	2	147	1.00	0.33	0.16
	3	122	1.00	0.33	0.01
	4	95	1.00	0.33	0.08
	5	68	1.00	0.33	0.07
Most profit slow	1	100	1.00	0.09	0.09
	2	19	0.91	0.09	0.05
	3	2	0.65	0.09	0.05
	4	0	0.49	0.09	0.05
	5	0	0.39	0.09	0.05
Most profit fast	1	160	1.00	0.33	0.32
	2	129	1.00	0.33	0.16
	3	103	1.00	0.33	0.11
	4	82	1.00	0.33	0.08
	5	39	1.00	0.33	0.07
Priority slow	1	112	1.00	0.09	0.09
	2	21	0.91	0.09	0.05
	3	2	0.65	0.09	0.05
	4	0	0.49	0.09	0.05
	5	0	0.39	0.09	0.05
Priority fast	1	170	1.00	0.33	0.32
	2	147	1.00	0.33	0.16
	3	122	1.00	0.33	0.11
	4	95	1.00	0.33	0.08
	5	66	1.00	0.33	0.07

Figure 7 compares the waiting time for a slow arrival rate of the customers versus the scheduling method for the different number of servers. Comparing the results, it can be observed that LJF, most profitable job first and priority queues suffer with very long waiting times when the number of available servers are limited. If the number of servers are more than three the results converges to each other as no congestion occurs.

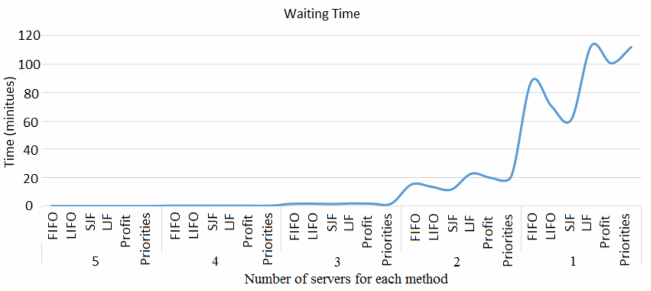

Figure 7. Waiting times for different queueing methods.

A graphical representation for waiting time is given in Figure 8, comparing different scheduling methods and number of servers. It can be seen that LJF, most profitable job first and priority queues will have the longest waiting time compared to SJF.

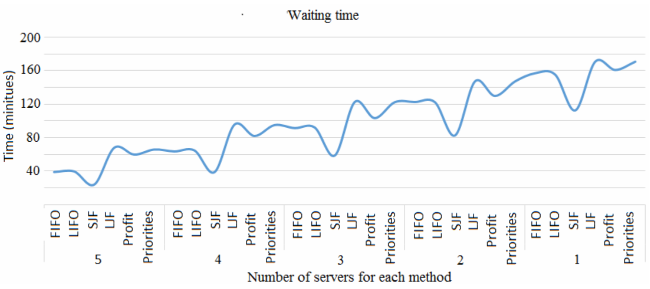

Figure 8. Waiting times for different queueing methods

Another disturbance is introduced in Figure 9, where the number of clients arriving changes at different times of the day.

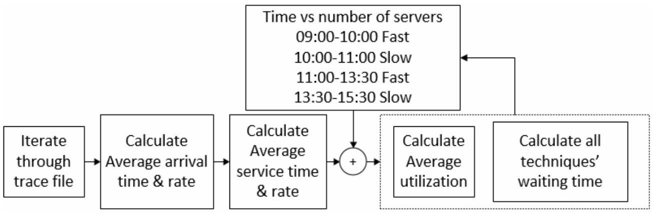
Time vs number of servers
09:00-10:00 Fast
10:00-11:00 Slow
11:00-13:30 Fast
13:30-15:30 Slow
Iterate through trace file
Calculate Average arrival time & rate
Calculate Average service time & rate
+
Calculate Average utilization
Calculate all techniques' waiting time

Figure 9. The number of cashiers change as the time of day progresses

Within Figure 10 it can be seen that for any situation where the arrival rate changes the SJF queueing method is the most optimal method for a constant number of servers.

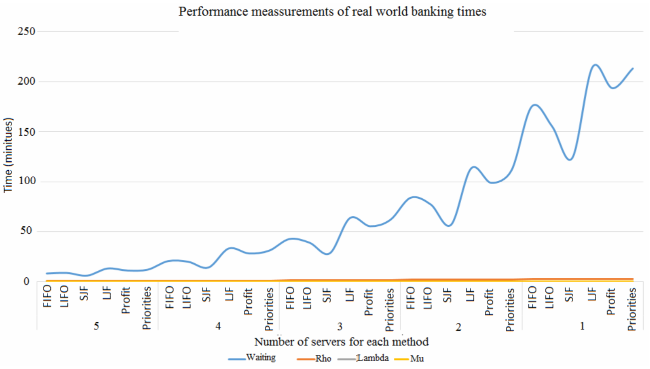

Figure 10. The time disturbance for each queueing method whilst also iterating the number of servers

The number of servers vary as the day progresses as can be seen within Figure 11. This disturbance is introduced as it is not necessary for full bank staff to be present at all times of the day. If such a scheme is used the bank will save money by not having staff work if they won't be able to serve any customers.

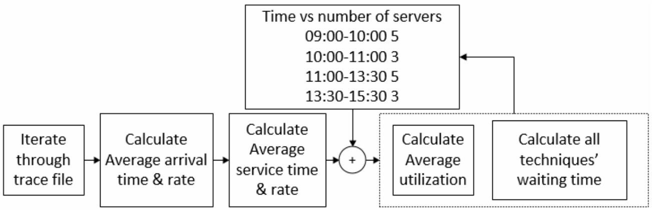
Time vs number of servers
09:00-10:00 5
10:00-11:00 3
11:00-13:30 5
13:30-15:30 3
Iterate through trace file
Calculate Average arrival time & rate
Calculate Average service time & rate
+
Calculate Average utilization
Calculate all techniques' waiting time

Figure 11. Varying the number of cashiers

Results obtained from simulating the performance measurements can be seen in Figure 12. The SJF method will have the shortest waiting time, whilst the LJF method will be the longest. Using the most profitable job first method produces roughly the same waiting time as other methods, but will adversely affect the customer satisfaction. Refer to Table 12 for the number of clients.

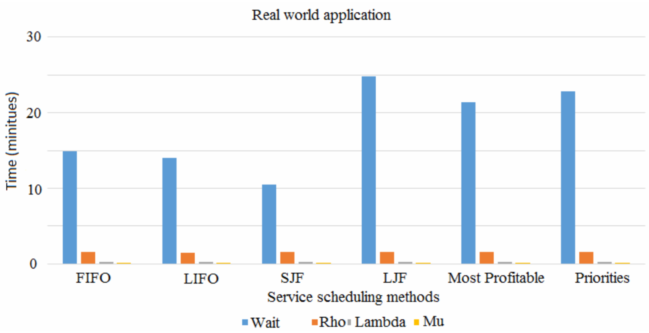

Figure 12. Comparing the different queueing methods to each other for real world implementation

Table 12. The customers served for scheduling methods

Method	Number of Customers
FIFO	41.86
LIFO	39.79
SJF	39.84
LJF	36.91
Profit	37.75
Priorities	37.62

Figure 13 compares the waiting time and the number of customers served for each scheduling method. The FIFO and SJF methods serves the same amount of customers. The SJF method will serve the most customers.

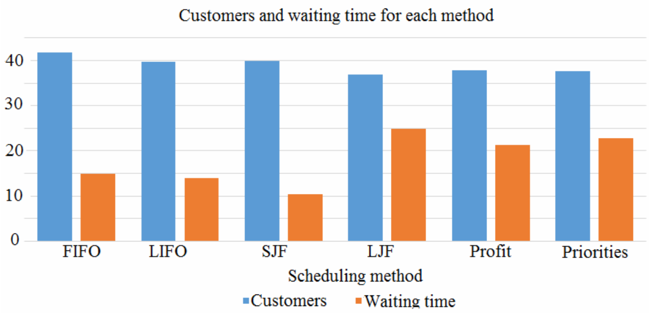

Figure 13. Waiting time and number of customers for each method

As shown within Figure 5, the system should adapt if there is a disturbance to the effectiveness of the system. The QMS will provide adequate customer satisfaction whilst still making a good amount of profit, by taking into consideration the time of the day as well as the arrival rate of new customers to the bank.

The QMS should analyse the customer arrival rate and number of servers to optimise customer satisfaction and profits.

If the arrival rate is slow, leading to a low system utilization, the QMS should focus on optimising profits for the bank. This could be accomplished by serving high priority jobs first using the most profitable job first strategy. When the system utilization reaches a point where no balance exist between customer satisfaction and profits, the QMS should focus on customer flow.

Within Figure 14 it can be seen that the QMS will optimise the banking experience for customers while also trying to increase the profit for the bank. When looking at region A within Figure 14 it can be observed that customer waiting times are really short. This indicates a slow arrival time and the system optimises for most profit. When the waiting time rises as seen within part B in Figure 14 the QMS will detect that customer satisfaction may be at risk. Within part C the QMS adapts the system to reduce the customer waiting times. This will in turn decrease the profit but a balance between waiting time and profit will be found.

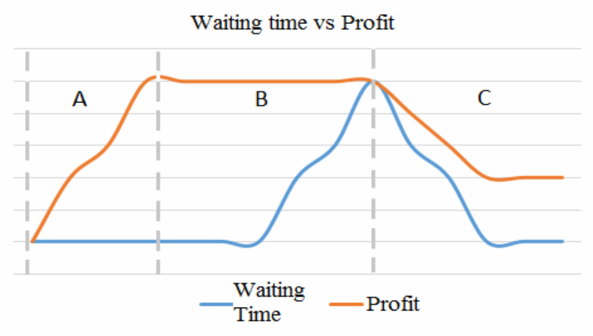

Figure 14. Waiting time vs. profit

When the maximum number of cashiers are not available, a disturbance is crated. The system should adapt using different scheduling algorithms to find the ideal solution for the bank.

The results are shown in Figure 10. The waiting time will exponentially increase as the number of cashiers is decreases but using the correct scheduling method the bank can still serve the customers or make a profit.

6 Design

A practical implementation to improve banking systems will be implemented. For mobility and to design a QMS, an FPGA will be used. An FPGA is ideal since it offers performance and reliability which is ideal for banking scenarios. It is chosen to use the Altera DE1 development board as simulation platform.

The FPGA system will communicate with the user through a terminal program on a computer. The FPGA communicates to the computer using the RS-232 interface. A trace file containing simulation data will be generated in order to produce results as seen in Figure 15.

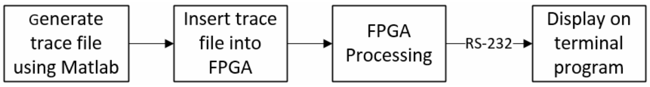
Generate trace file using Matlab
Insert trace file into FPGA
FPGA Processing
RS-232
Display on terminal program

Figure 15. FPGA implementation of QMS

Random data is generated and will be copied into the block diagram as shown within Figure 16. The random data will be converted to binary and transmitted over the data busses. Transmission only occurs when Trigger is active high. The contents of the data busses are given in Table 13.

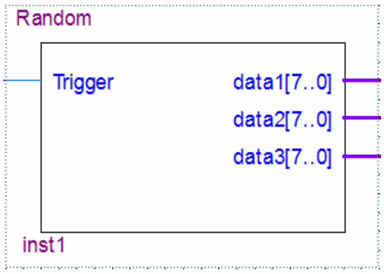

Figure 16. Random generator data import

Table 13. Data bus descriptions

Bus	Content
Data1	Arrival time
Data2	Service time
Data3	Banking activity

The block diagram in Figure 17 is used to select the scheduling method to use for a specific simulation. This is done using switches on the FPGA development board. The "S1_data_out" to "S5_data_out" is the service time remaining for each of the five servers. The "Queue_out" and "Queue_data_out" contains the data for each of the customers waiting in the queue.

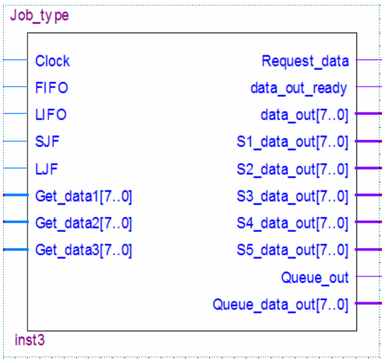

Figure 17. Simulation selection block diagram

The block diagram within Figure 18 is used to convert the data from the block diagram within Figure 17 into American Standard Code for Information Interchange (ASCII) values that will be sent through the RS-232 serial interface to be displayed on a terminal program on a computer.

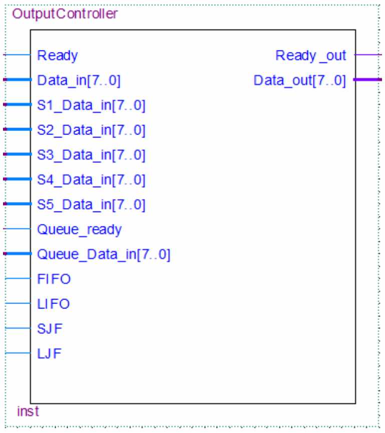

Figure 18. Output converter block diagram

The contents from the data bus in Figure 18 will be transmitted using the transmitter block in Figure 19. The RS-232 pin is connected to the "TxD" pin and the baud rate is set as a parameter.

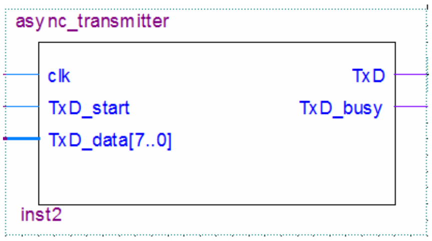

Figure 19. Output converter block diagram

The different block diagrams must be connected together to produce the final hardware simulation platform on the FPGA development board. This can be seen within Figure 20.

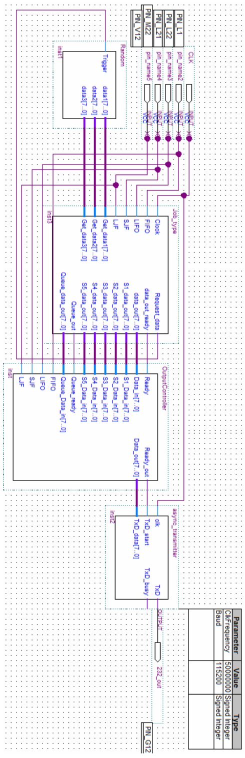

Figure 20. Full simulation system block diagram

The current design allows for the FPGA implementation to simulate the QMS for a banking environment.

7 Discussion

The results within this paper is obtained by averaging over 10000 iterations of random numbers. The results could be improved if more iterations are presented, also the current implementation takes into account a single day. In the reality banking traffic should be monitored for each day of the week in order to produce a better simulation. The implications of this study could be used to improve banking models if a real-time FPGA implementation should be designed. Limitations presented with the current simulation model include that only a few queueing strategies are implemented. This gives an opportunity for future research of how other queueing methods would affect the waiting time. Given the practical limitation to only simulate the banking model, future work allows for a practical implementation that could be designed that take into account users as they arrive at the bank, sorting them into the correct queue, regarding the appropriate queuing strategy relating to their reason for visiting the bank.

When implementing this in a bank, different scheduling methods may be used according to the requirements of the bank. Previous implemented methods focussed on service quality, service time and efficiency of the tellers.

The main focus for a bank is to achieve customer satisfaction and obtain a profit. To be able to implement the scheduling methods for a banking scenario, the bank has to be remodelled by the management to include a machine to allocate tickets to customers and remove the normal queueing system for the bank. When the bank has been remodelled, different scheduling methods can be implemented to optimise the bank.

8 Conclusion

Using different scheduling methods it is concluded that a queueing system for a bank can be improved. The different scheduling methods implemented in this paper, each has its own functionality. The FIFO method serves the highest amount of customers, the LIFO method will have the shortest waiting time for slow arrivals, but dissatisfy customers. The SJF method has the shortest waiting time and highest customer satisfaction, priority queueing and most profitable job first can be implemented to boost profits of the bank.

The SJF method should be implemented at peak hours, while the most profitable job first scheme should be implemented at off-peak hours.

The overall conclusion is that using the queuing methods listed in this paper will reduce waiting time and improve customer satisfaction, whilst improving profits for a banking firm. Future research includes different queueing strategies and a real-time practical implementation.

Acknowledgement

Kevin William George Cowdrey, Jaco de Lange, and Reza Malekian contributed equally to this work.

References

[1] A. O. Allen, Queueing Models of Computer Systems, *Computer*, Vol. 4, pp. 13-24, April, 1980.

[2] D. Gross, *Fundamentals of Queueing Theory*, John Wiley & Sons, 2008.

[3] B. S. Blanchard, W. J. Fabrycky, *Systems Engineering and Analysis*, Prentice Hall, 1990.

[4] G. Bolch, G. S. Greiner, H. De Meer, *Queuing Networks and Markov Chains: Modelling and Performance Evaluation with Computer Science Applications*, John Wiley & Sons, 2006.

[5] R. A. Nosek, J. P. Wilson, Queuing Theory and Customer Satisfaction: A Review of Terminology, Trends, and Applications to Pharmacy Practice, *Hospital Pharmacy*, Vol. 36, No. 3, pp. 275-279, March, 2001.

[6] S. Bagchi, Analyzing Distributed Remote Process Execution Using Queuing Model, *Journal of Internet Technology*, Vol. 16, No. 1, pp. 163-170, January, 2015.

[7] K. Mahmood, A. Chilwan, O. Sterb, Modelling of Openflow-basedsoftware-defined Networks: The Multiple Node Case, *IET Networks*, Vol. 4, No. 5, pp. 278-284, May, 2015.

[8] A. Rasmussen, H. Yu, S. Ruepp, M. S. Berger, L. Dittmann, Efficient Round-robinmulticast Scheduling for Input-queued Switches, *IET Networks*, Vol. 3, No. 4, pp. 275-283, December, 2014.

[9] A. Al-Mogren, M. Iftikhar, M. Imran, N. Xiong, S. Guizani, Performance Analysis of Hybrid Polling Schemes with Multiple Classes of Self-Similar and Long-Range Dependent Traffic Input, *Journal of Internet Technology*, Vol. 16, No. 4, pp. 615-627, July, 2015.

[10] D. Hammond, S. Mahesh, A Simulation and Analysis of Bank Teller Manning, *Winter Simulation Conference Proceedings*, Arlington, VA, 1995, pp. 1077-1080.

[11] A. Sarkar, A. R. Mukhopadhyay, S. K. Ghosh, Improvement of Service Quality by Reducing Waiting Time for Service, *Simulation Modelling Practice and Theory*, Vol. 19, No. 7, pp. 1689-1698, August, 2011.

[12] N. Madadi, A. H. Roudsari, K. Y. Wong, M. R. Galankashi, Modeling and Simulation of a Bank Queuing System, *2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation*, Seoul, South Korea, 2013, pp. 209-215.

[13] M. S. Olusola, S. Okolie, A. K. Adesina, Queue Management Systems for Congestion Control: Case Study of First Bank,

Nigeria, *International Journal of Advanced Studies in Computers, Science and Engineering*, Vol. 2, No. 5, pp. 54-58, October, 2013.

[14] C. H. Sauer, E. A. MacNair, J. F. Kurose, Queueing Network Simulations of Computer Communication, *Selected Areas in Communications, IEEE Journal on*, Vol. 2, No. 1, pp. 203-220, January, 1984.

[15] S. Suranauwarat, A CPU Scheduling Algorithm Simulator, *2007 37th Annual Frontiers in Education Conference-global Engineering: Knowledge without Borders, Opportunities without Passports*, Milwaukee, WI, 2007, pp. F2H-19-F-2H-24.

[16] M. Chen, Z. Gao, W. Wang, Research on Multi-objective Dynamic Priority Requestscheduling of Business Web System, *Wireless Communications, Networking and Mobile Computing*, New York, NY, 2007, pp. 1954-1957.

[17] B. T. Doshi, H. Heffes, Overload Performance of Several Processor Queueing Disciplines for the M/M/1 Queue, *IEEE Transactions on Communications*, Vol. 34, No. 6, pp. 538-546, June, 1986.

[18] H. Xiao, G. Zhang, The Queuing Theory Application in Bank Service Optimization, *2010 International Conference on Logistics Systems and Intelligent Management*, Harbin, China, 2010, pp. 1097-1100.

[19] R. Malekian, A. Abdullah, N. Ye, Novel Packet Queuing Algorithm on Packet Delivery in Mobile Internet Protocol Version 6 Networks, *Applied Mathematics and Information Sciences*, Vol. 7, No. 3, pp. 881-887, May, 2013.

[20] R. Malekian, A. H. Abdullah, Traffic Engineering Based on Effective Envelope Algorithm on Novel Resource Reservation Method over Mobile Internet Protocol Version 6, *International Journal of Innovative Computing, Information and Control*, Vol. 8, No. 9, pp. 6445-6459, September, 2012.

Biographies

Kevin W. G. Cowdrey is currently persuing the B.Eng. degree in computer engineering at the University of Pretoria, South Africa. His research interests include artificial intelligence, Internet of Things, advanced sensor networks and computer vision.

Jaco de Lange is currently pursuing the B.Eng. degree in computer engineering at the University of Pretoria, South Africa. His research interests include artificial intelligence, computer vision, network security and Internet of Things.

Reza Malekian is an associate professor with the Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, Pretoria, South Africa. His current research interests include advanced sensor networks, Internet of Things, and mobile communications. Dr. Malekian is also a Chartered Engineer and an associate editor for the IEEE Internet of Things Journal.

Johan Wannenburg is currently pursuing the M.Eng. degree in Electronic Engineering. He received the B.Eng. degree in Electronic Engineering from the University of Pretoria, Pretoria, South Africa in 2014. His research interests include e-health, biosensors, body area networks, and finding novel ways of improving health monitoring through concepts such as context awareness.

Arun Cyril Jose received Engineering in Computer Science from Mahatma Gandhi University, India in 2009 and Master’s in Forensic Computing from Coventry University, UK in 2011. He is currently a Ph.D. student in the Department of Electrical Electronic and Computer Engineering at University of Pretoria, South Africa.