Performance Modelling and Analysis of Weighted Fair Queueing for Scheduling in Communication Networks

An investigation into the Development of New Scheduling Algorithms for Weighted Fair Queueing System with Finite Buffer

Amina Said Mohammed Alsawaai

Submitted for the Degree of
Doctor of Philosophy

Department of Computing
School of Computing, Informatics and Media
University of Bradford

2010

Abstract

Analytical modelling and characterization of Weighted Fair Queueing (WFQ) have recently received considerable attention by several researches since WFQ offers the minimum delay and optimal fairness guarantee. However, all previous work on WFQ has focused on developing approximations of the scheduler with an infinite buffer because of supposed scalability problems in the WFQ computation.

The main aims of this thesis are to study WFQ system, by providing an analytical WFQ model which is a theoretical construct based on a form of processor sharing for finite capacity. Furthermore, the solutions for classes with Poisson arrivals and exponential service are derived and verified against global balance solution.

This thesis shows that the analytical models proposed can give very good results under particular conditions which are very close to WFQ algorithms, where accuracy of the models is verified by simulations of WFQ model. Simulations were performed with QNAP-2 simulator. In addition, the thesis presents several performance studies signifying the power of the proposed analytical model in providing an accurate delay bounds to a large number of classes.

These results are not able to cover all unsolved issues in the WFQ system. They represent a starting point for the research activities that the Author will conduct in the future. The author believes that the most promising research activities exist in the scheduler method to provide statistical guarantees to multi-class services. The author is convinced that alternative software, for example, on the three class model buffer case, is able to satisfy the large number of buffer because of the software limitation in this thesis. While they can be a good topic for long-term research, the short-medium term will show an increasing interest in the modification of the WFQ models to provide differentiated services.

Dedicated to my parents, my husband and my lovely son ...
keep inspiring me!

Acknowledgements

My deep thanks, from beginning to end, goes to God, who created me, helps me and has given me the ability to reach this stage.

I would especially like to express my sincere gratitude and appreciation to my project supervisors Prof. Irfan Awan and Dr. Rod Fretwell for their helpful guidance, advice and valuable comments. Without their help, I would not have been able to complete my project.

I would also like to extend my deep gratitude to the Computing department who provided me the needed knowledge to undertake this study. I would like also to thank the members of the Computing Department: Miss Rona Wilson and Mrs. Bev Yates.

I give loving thanks to my husband, Hussain Ali, my lovely son, Ali, and my mother. It would not have been possible to complete my PhD research without their encouragement and support throughout my personal life and career.

My gratitude also goes to all my friends Dr. Iman Alansari, Rasha Osman, Ibtihal Nafea, Rasha Fares, Hamda Al.Shihi and Ruqayya Abdulrahman and colleagues who supported me in my personal life and career. I am greatly indebted to the Ministry of Higher Education for providing me the opportunity and encouragement to obtain a higher degree in my field. Last, I wish to extend my heartfelt to my father and my family, who encouraged and continued supporting me until the end.

Publications

1. Amina Al-Sawaai, Irfan Awan, Rod Fretwell, "Validation of Analytical Model of WFQ System ", *proceedings of 26rd Annual UK Performance Engineering Workshop (UKPEW2010).* July 08-09, Warwick University, UK, pp. 1-8, 2010.

2. Amina Al-Sawaai, Irfan Awan, Rod Fretwell, "Performance of weighted fair queuing system with multi-class jobs", *proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 50-57, 2010.

3. Amina Al-Sawaai, Irfan Awan, Rod Fretwell, "Performance Evaluation of Weighted Fair Queuing System Using Matrix Geometric Method", proceedings of International Conferences on Networking (IFIP2009) *Lecture Notes in Computer Science*, vol. 5550/2009, pp.66-78, 2009.

4. Amina Al-Sawaai, Irfan Awan, Rod Fretwell, "Analysis of the weighted fair queuing system with two classes of customers with finite buffer", *proceedings of 2009 International Conference on Advanced Information Networking and Applications Workshops*, pp. 218-223, 2009.

5. Amina Al-Sawaai, Irfan Awan, Rod Fretwell, Stationary queue length distribution for M/M/1/K queue with preemptive priorities . *proceedings of 23rd Annual UK Performance Engineering Workshop (UKPEW2007).* July 09-10, Edge Hill University, UK, pp. 1-5, 2007.

6. Amina Al-Sawaai, Irfan Awan, Rod Fretwell, Stationary queue length distribution for M/M/1/K queue with non-preemptive priorities. *proceedings of 8th Annual Postgraduate Symposium on the Convergence of Telecommunication, Networking and Broadcasting (PG NET 2007).* June 28 -29, School of Computing and Mathematical Sciences, John Morse University, Liverpool, UK, pp. 15-20, 2007 .

7. Amina Al-Sawaai, Irfan Awan, Rod Fretwell, "Stationary queue length distribution for M/M/1/K queue with non-preemptive and preemptive priorities", *proceedings of 8th Informatics Workshop for Research Student*, 28 June, School of Informatics, University of Bradford, UK, pp. 25-32, 2007.

List of Symbols

$\mathbf{K}$	Number of classes
λ_i	Arrival rate of class-i
μ_i	Service rate of class-i
$\frac{\mathbf{1}}{\mu_\mathbf{i}}$	Mean service time of class-i
$\mathbf{N}$	Number of jobs in the system
$\mathbf{n}_i$	Number of jobs in the class-i
$\mathbf{w}_i$	Weight of class-i
$\vec{\pi}_0 \ldots \vec{\pi}_N$	Steady state probabilities
$\mathbf{L}_i$	Mean queue length of class-i
$\mathbf{S}_i$	Mean response time of class-i
$\lambda_{effective_i}$	The rate of jobs not dropped of class-i
$\mathbf{T}_i$	Throughput of class-i
$\prod_{j=2}^{N} R_j$	Product rate matrices of N jobs in K class

List of Abbreviations

FIFO	First In First Out
PS	Processor Sharing
RR	Round Robin
WRR	Weighted Round Robin
GPS	Generalized Processor Sharing
QoS	Quality of Service
QBD	Quasi Birth Death
FQ	Fair Queueing
SFF	Smallest Finish-time First
SSF	Smallest Start-time First
WFQ	Weighted Fair Queueing
CBWFQ	Class Based Weighted Fair Queueing
SCFQ	Self-Clocking Fair Queueing
SFQ	Start-time Fair Queueing
VC	Virtual Clock
$\mathbf{WF^2Q}$	Worst-case Fair Queueing

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

A Weighted Fair Queueing (WFQ) service discipline is used frequently within the Internet and seems well suited for use within different services [1] [2]. The main reason for its importance comes from the fact that it overcomes some of the limitations of the First In First Out (FIFO) and priority service disciplines by providing service in a fair manner for packets of different classes.

To the best of our knowledge the constraints of the weighted fair queueing (WFQ) algorithms make it difficult to provide exact analytical models for WFQ systems [3]. Therefore, the contribution of this thesis is to provide an analytical WFQ model which is a theoretical construct based on a form of processor sharing. In this work, the analytical models proposed give good results which are very close to measured results from WFQ systems [4] [5] [6].

1.2 Motivation

In a modern computer and telecommunication system jobs can be grouped into classes according to Quality-of-Service (QoS) requirements. Jobs belonging to different classes are expected to be served with different Quality of Service requirements. Packet scheduling discipline is a tool that can be used to achieve service differentiation in order to provide Quality of Service (QoS) to streams in a network. Such a discipline can be implemented in the switches or routers of a network in order to accommodate the various bandwidth requirements of incoming flows that share the same departing link.

CBWFQ is an important discipline in packet-switched networks where multiple traffic types, such as voice, video, and data, compete for the same network resources. Each of these traffic types has its own quality-of-service (QoS) measures that must be met. CBWFQ provides, in some sense, a compromise that attempts to meet such requirements. On the one hand, it avoids the extreme of isolating each traffic class on its own network (which is better for meeting QoS requirements, but wastes unused network resources) and avoids the other extreme of letting all classes compete in a first-come-first-served manner for the same resources (in which case, high demand from one class can degrade performance for the other classes). CBWFQ is currently implemented in combination with priority queueing as part of the low latency queueing discipline in some Cisco routers [7].

In most servers a First In First Out (FIFO) discipline is used (Keshav [8]). This strategy is fair in the sense that it serves the jobs in the sequence of their arrival times and is starvation free. A great improvement to response time can be achieved by using a Priority Queueing (PQ) strategy (Keshav [8]) which improves the mean response time. However, PQ can lead to starvation. Sometimes this makes the request for large jobs wait a very long time before being served. Hence, my main aim is to find a scheduling strategy that reduces the mean response time, without losing the property of fairness. To the best of our knowledge the weighted fair queueing (WFQ) system combines the advantages of both FIFO and PQ for mean response, mean queue length and throughput.

It is important to investigate the impact of the weighted fair queueing (WFQ) scheduling approach on queueing network models; this will require considerable attention. A persistent problem in queueing systems is how to choose the most appropriate model to support a WFQ scheduling strategy, especially under optimal conditions. This thesis ultimately aims to identify a tractable analytical model of a queue under WFQ.

1.3 Aims and Objectives

This thesis aims to:

- Develop an analytical queueing model finite buffer with multiple classes which approximates to WFQ based on a form of processor sharing (PS).
- Provide a general solution for a PS approximation to the WFQ model with a multi-class finite buffer.
- Analyse the PS approximation to the WFQ model by using several traffic load and weights.
- Prove that the proposed analytical model can be a good approximation to queue under WFQ.

These aims are achieved by developing an analytical model and validating it through a simulation model.

1.4 Research Methodology

Given the objective of the research which is to model a system for WFQ scheduling, the approach initially was to review what is known in the area of

- scheduling discipline.
- simulation model in queueing system.
- analytical model in queueing system.
- Matrix analytic methods.

That review presented in Chapter 2 exposes two important main points. Firstly, it presents a survey of the most common scheduling techniques, in particular the Weighted Fair Queueing (WFQ). Secondly, a brief description of previously related works which prove that the constraints of weighted fair queueing (WFQ) algorithms make it difficult to provide exact analytical models for WFQ systems. Consequently, the next step was to propose an analytical model, which approximates to WFQ system. A new Theorem was presented, proved and evaluated as shown in Chapter 3. As a result, it is used to calculate the steady state of probability for any buffer and class. Experimental results based on the implementation of the proposed analytical model are presented where simulation results are used to validate the proposed mathematical model and to prove that the behaviour of the system is similar to that in the simulation model (see Chapter 4). Finally, the WFQ simulation model that had been proposed and the simulation results in [9] proves that the analytical model in this thesis is a good approximation to WFQ model.

1.5 Thesis Organization

- **Chapter 1** provides a context for the research area; it provides a brief introduction and outline of the problem domain. The structure of the thesis is also presented.

- **Chapter 2** presents the procedure, benefits, and limitations of a different number of common queue scheduling disciplines together with the literature review.

- **Chapter 3** introduces a proposed analytical model of WFQ system that could generally be applied to support any number of classes.

- **Chapter 4** presents the experimental results based on the implementation of the proposed analytical model.

- **Chapter 5** concludes the thesis with a summary of the contributions and possible future directions for the work.

Chapter 2

Literature Review

2.1 Introduction

In a modern computer and telecommunication system, customers can be grouped into classes according to Quality-of-Service (QoS). The customers belonging to different classes are expected to be served with different service. To provide the Quality of Service to streams in a network, a tool called the packet scheduling discipline is used to achieve service differentiation. The main aim of these scheduling disciplines is to share the bandwidth between the all classes. This chapter has two parts: a review of scheduling disciplines for fair queueing and related research work.

To meet the needs aforementioned, our attention is turned to service discipline which provides a different service for individual classes. A non-preemptive scheduling (also referred to as head of line priority scheduling, HOL-PS) occurs as follows: a high priority customer can move ahead of all the low priority customers waiting in the queue, but low priority customers in service are not interrupted by high priority customers. Among the simplest time-priority scheduling schemes, the non-preemptive HOL priority scheduling discipline has been proposed to provide differentiated services Miller [10]. However, PQ can lead to starvation. Sometimes this makes the request for large jobs wait a very long time before being served. Hence, the main aim in this work is to find a scheduling strategy that reduces the mean response time, without losing the property of fairness. To the best of our knowledge weighted fair queueing (WFQ) system combines the advantages of both FIFO and PQ. A detailed study of the weighted fair queueing algorithms is described in Section 2.2, with a particular focus on the procedure, benefits, and limitations of fair queueing.

There are different types of Quasi Birth Death (QBD) solution methods available. However, next chapter refers to the most known solution method, the matrix-geometric solution proposed by Neuts in [11] and [12]. It is difficult to choose the best solution method, since each method has its advantages and disadvantages. A good comparison discussion has been seen in Tran [13] and [14]. The previous WFQ system solutions, and the solutions limitations are discussed in Section 2.3.

2.2 Review of Scheduling Disciplines for Fair Queueing

Fair scheduling disciplines are so-called because they are designed to share the bandwidth between all the classes. The idea of FQ was proposed by Nagel [15]. This section has three parts: Processor-Sharing (PS), Generalized Processor Sharing (GPS) and Approximating GPS.

2.2.1 Processor-Sharing (PS)

The processor-sharing (PS) queueing system, has a considerable value and is used widely to study computer and communication [16]. In PS, each customer receives an equal share of the processor; if there are n customers at some time, then each customer is serviced at $\frac{1}{n}$ times the speed of the processor Kleinrock [17].

In spite of its simple description, PS is difficult to analyse compared to other queueing systems, regarding to analyse PS in Kleinrock [16], Yashkov [18], Ott [19] and Schassberger [20].

Therefore, a theoretical construct of the approximation model in this thesis based on the idea of processor sharing, which is serviced all classes at the same times.

Hence, we turn our attention to a scheduling strategy that provides differentiated services for each class, without losing the property of PS.

2.2.2 Generalized Processor Sharing (GPS)

The Generalized Processor Sharing (GPS) [21] is an ideal scheduler that distributes the bandwidth of a shared link fairly among packet flows in proportion to their reserved bitrates. GPS attains its bandwidth guarantees by serving an infinitesimal amount from each backlogged class in proportion to each class's reservation. However, GPS is unrealizable in practice because it services a small part of each packet at a time.

The GPS works by assigning a separate queue to each flow (or class); then services an infinitesimal amount from each flow according to a weighted cyclic schedule. In GPS [22], it is assumed that traffic satisfies the fluid model and that every packet is infinitely divisible. Assume that there are N classes sharing an outgoing link of capacity C. The share of bandwidth reserved by class j is represented by a real number w_i . The ws are selected such that the fraction,

$$\frac{w_j}{\sum_{i=1}^{N} w_i} \tag{2.1}$$

corresponds to the required bandwidth reservation of the class. If α_i is the required bandwidth reservation of class j, then

$$\frac{w_j}{\sum_{i=1}^{N} w_i} C \geq \alpha_i. \tag{2.2}$$

The amount

$$r_i = \frac{w_j}{\sum_{i=1}^{N} w_i} C \tag{2.3}$$

is called the guaranteed rate for class j and is the minimum available bandwidth to class j at any known instance of time.

Let $B(\tau, t)$ be the group of classes that are backlogged in the interval (τ, t). Under GPS, the service $S_j(\tau, t)$ offered to class j that belongs to $B(\tau, t)$ is proportional to w_i according to

$$S_j(\tau, t) = \frac{w_j C}{\sum_{i \in B(\tau, t)} w_i} (t - \tau). \tag{2.4}$$

2.2.3 Application of GPS in a Packet System

A real scheduler must complete the service of a whole packet from a class before it moves to the next class. It has been observed that the GPS scheduler is inapplicable because no packet can be partitioned into infinitesimal amounts. As a result, the suggested researches take the idea of the service order of packets in GPS to schedule packets in a packet system. This leads to two packet selection policies, the Smallest Finish-time First(SFF) [23] and the Smallest Start-time First (SSF) [24]. The SFF technique services the packet according to the finishing order under GPS. On the other hand, in the SSF techniques, packets are serviced in the starting order under GPS. Start-time Fair Queueing (SFQ) is an example of a SSF scheduler while Weighted Fair Queueing (WFQ) is an example of a SFF scheduler [25].

There is more than one packet to finish at the same time even if they arrive at different times in GPS [26]. As a result, in [27] it is assumed that a GPS system is inapplicable since it demands knowledge of most of the N events at a certain moment of time. As a result, it was stated that the application complexity of any GPS emulation is $O(N)$. However, Tayyar proved in [26] that this is not correct since the system can be reduced to $O(1)$, even when as many as N events are happening simultaneously. This is reached by creating a priority queue data structure that tracks the finish order of packets in the system .

2.2.4 Approximating GPS

There are different ways of approximating GPS service; the most popular GPS approximation methods are presented in the following subsections.

2.2.4.1 Weighted Round-Robin (WRR)

Packet round robin is the foundation for the Weighted Round Robin (WRR) [28]. We take into consideration the Round-Robin (RR) [8], which is one of the GPS emulations that serves a packet from each nonempty queues, instead of infinitesimal amount. The RR is considered a good approximation of GPS when all queues have equal weight and all packets have equal size. However, if queues have different weights, then RR is-called the WRR queueing discipline [8].

We can see from Figure 2.2 the basic behaviour of WRR. The packets in the WRR are classified into different service classes and then the scheduler accesses each queue in a

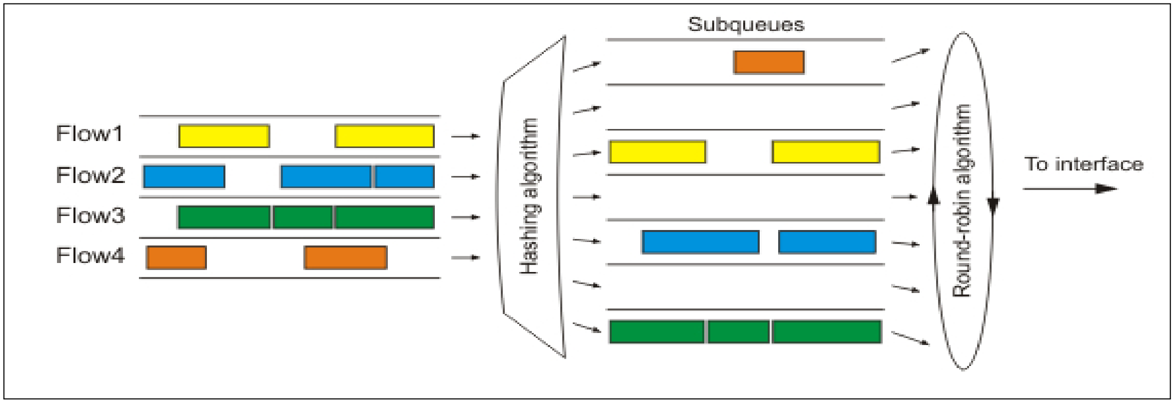

Figure 2.1: Round Robin (Fair Queueing)

round-robin approach. Each queue has a weight that allows it to service a certain number of packets before giving service to the next queue. The weight is typically predetermined as a percentage of the whole bandwidth.

The quality of the Weighted Round Robin is its ability to manage packets instead of bytes. Customized class is calculated taking into consideration the weight assigned to each class, so that it can send an exact number of bytes each round. This means that when a packet is sent then class allocation is decreased by an amount equal to the packet length. In the case when packets from different queues have different sizes, a WRR scheduler divides each queue's weight by its mean packet size to get a normalized group of weights. However, if a source cannot predict its mean packet size, a weighted round robin server will not be able to allocate bandwidth fairly. In addition, the weighted round robin is fair only over time scales longer than a round time because, at the shorter time scale some classes may receive more service than others [8]. Furthermore, WRR has the benefit of providing only $O(1)$ processing complexity per packet, but its delay becomes a problem as the number of classes sharing the link increases [29].

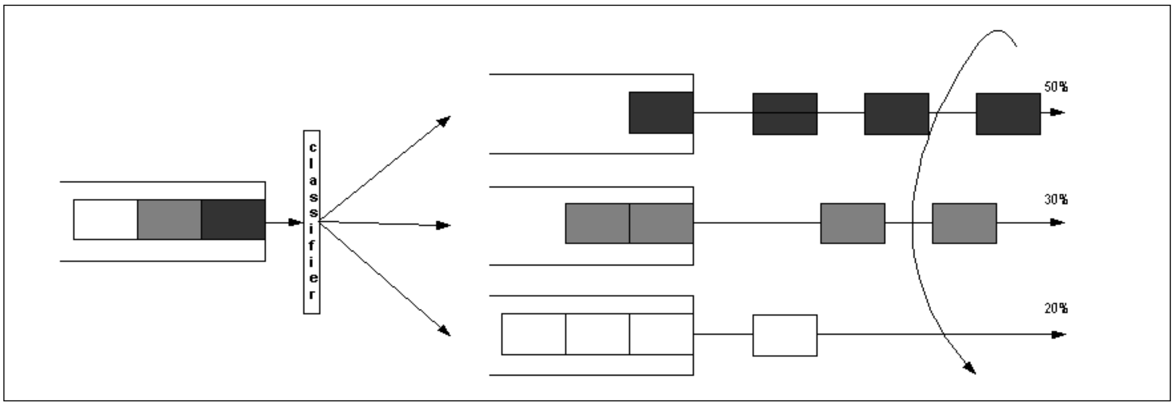

Figure 2.2: Weighted Round Robin (WRR)

2.2.4.2 Weighted Fair Queueing (WFQ)

Packet-by-packet GPS or WFQ as it is known, is an approximation of GPS scheduling that transmits packets according to their finishing order under GPS [27]. In WFQ a GPS fluid-model system is simulated in parallel with the actual packet-based system in order to classify the group of classes that are backlogged at each instant of time and their service rates. On the basis of this information, a timestamp is calculated for each arriving packet, and the incoming packets are inserted into a priority queue based on their timestamp values in order of increasing timestamp. The timestamp determines the finishing number of the packet. To determine the finishing number, WFQ keeps track of a virtual time function $V(t)$ which is a piecewise linear function of real time t , and its slope changes depending on the number of busy classes in GPS and their service rates. More specifically, if $B(t)$ represents the set of backlogged classes in the GPS Scheduler at a time t is given by

$$\frac{\sum_{i=1}^{N} \gamma_i}{\sum_{j \in B(t)} \gamma_j} \tag{2.5}$$

where N is the number of active classes. At the arrival of a new packet, the virtual time must first be calculated. Then, the timestamp TS_i^k connected with the k th packet of class j which arrives at time, t is calculated as

$$TS_j^k = max(TS_j^{k-1}, V(t)) + \frac{L_j^k}{r_j}, \tag{2.6}$$

where L_j^k is the length of the arrived packet and r_j is the guaranteed link share of class j. The basic behavior of WFQ is shown in Figure 2.3.

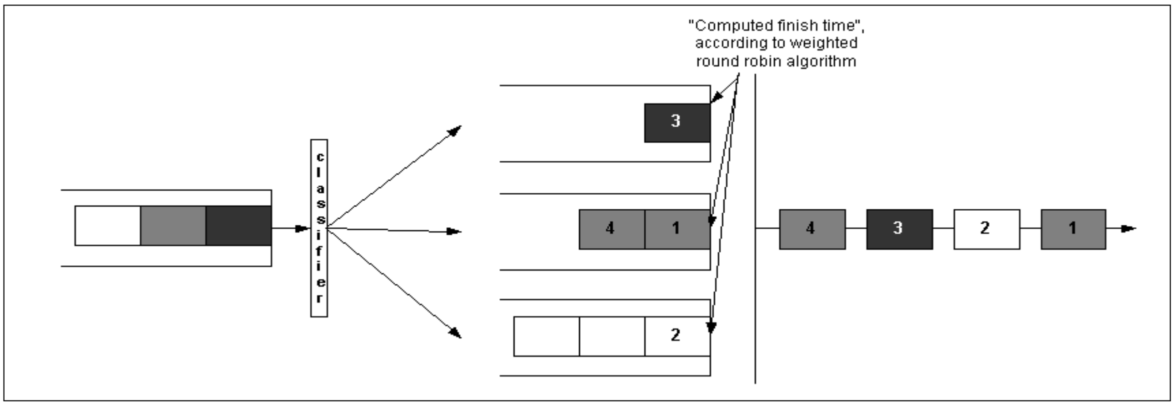

Figure 2.3: Weighted Fair Queueing (WFQ)

The benefit of using WFQ queueing is that it gives the same latency bound as the GPS, with a maximum difference equal to the transmission time of one maximum length packet. It has been suggested, that the WFQ is complex and the computational complexity is increasing from the solution method to the GPS scheduler [27]. Another source of complicity belongs to the queue management operations. In [30] it is proved that this operation is accomplished with $O(log(N))$ complexity. Although the guaranteed delay bounds supported by a weighted fair queue are commonly better than for other fair queue scheduling disciplines, the bounds will still be quite large [2].

2.2.4.3 Enhancements to WFQ

Different enhancements have been proposed to the basic WFQ to solve the problem of computational complexity. These will be discussed. Among the popular WFQ variants are these:

1. A Self-Clocking Fair Queueing (SCFQ) is an enhancement to WFQ that reduces the complexity of calculating the virtual time corresponding GPS system and the virtual time function is computed using timestamp of the packet at present in service [23]. So, the virtual time is defined as the timestamp of the packet as currently in service TS_{cur}. A SCFQ computes the timestamp of an arriving packet as

$$TS_j^k = max(TS_j^{k-1}, TS_{cur}) + \frac{L_j^k}{r_j}, \tag{2.7}$$

   This approach decreases the complexity of the algorithm in a larger worst-case delay which increases with the number of service classes [2].

2. Virtual Clock (VC) scheduling presents the same end-to-end delay bound as WFQ with a simple timestamp computation algorithm. A VC computes the timestamp of an arriving packet as :

$$TS_j^k = max(TS_j^{k-1}, t) + \frac{L_j^k}{r_j}, \tag{2.8}$$

   where t is real time.

   The disadvantage of this scheduling is that a backlogged class can be starved for an arbitrary period of time as a result of more bandwidth it obtained from the server when other classes were idle [31].

3. Start-time Fair Queueing (SFQ) is another enhancement of WFQ. This scheduling uses to schedule packets according to start time in GPS. The virtual time is approximated by the virtual start-time of the packet presently in service. They can calculate the virtual finish time of packet as the sum of the virtual start time plus the ratio of length to class share. The virtual start time of packet when it arrives in class equals to the virtual finish time of the previous packet of that class. A SFQ is easy to implement, but it has a delay bound well above that of WFQ [24].

4. Worst-case Fair Queueing (WF^2Q) was proposed in [22] to overcome the problem of the departure process resulting from packet in WFQ server which could be bursty. The idea of WF^2Q like WFQ using the virtual time concept and the virtual finish time is defined as the time packet amount if it is sent under the GPS discipline. A WF^2Q focus for the packet has the smallest virtual finishing time between packets waiting in the system that have started service under GPS instead of looks for the smallest virtual finish time for all the packets waiting in the system. The service offered by WF^2Q is similar to that of GPS in spite of the difference by not more than one maximum size packet. A Worst-case Fair Queueing +(WF^2Q+) is the simpler implementation of WF^2Q with a relatively low complexity of $O(log(N))$ [32].

5. Class-based WFQ (CBWFQ) extends the regular WFQ functionally to give support for user -defined traffic classes. In the CBWFQ mechanism, the bandwidth is shared in traffic classes according to the weights assigned to the CBWFQ traffic classes. These weights are considered to guarantee the traffic class which gets a certain amount of the available bandwidth. CBWFQ extends the weighted fair queueing (WFQ) by assigning weights to classes of traffic rather than individual flow of traffic known by origination/destination pairs. The CBWFQ discipline is being used often within the Internet with these possible Quality of Service (QoS) problems for the

multiple traffic classes. For example, the base CBWFQ system has two classes of packets. There is no service interruption of packet by another. Upon service completion of a packet, the next packet be served is randomly chosen based on the class weights [2].

2.3 Related Research Work

To the best of our knowledge the constraints of weighted fair queueing (WFQ) algorithms make it difficult to provide exact analytical models for WFQ systems [3]. The assumption of Poisson packet arrivals and exponential service times is used in this thesis, which are the most lenient assumptions one could make and get analytical results. If the customer arrivals are according to Poisson processes, and service times are exponentially distributed, the system can be modelled by a multi-dimensional Markov chain [9].

There is a large body of research on Weighted Fair Queueing (WFQ) in general. However, most of the literature deals with simulation solutions for example [33] [34], [35] and [34]. All papers that deal with cyclic polling models in which the server visits each queue in ordered rotation, rather than in a random fashion Takagi [36].

A related concept of an analytical model in this thesis is processor sharing in which jobs from different queues are served simultaneously and the server services classes in a random fashion. There were many works proposed to analyse the WFQ.

Most authors that deal with approximation models to WFQ model with infinite buffer. The work in [37] studies the fluid weighted fair queueing which does not account for the discrete nature of packets, which assigns to each flow a weight, w say. Therefore a minimum bandwidth equal to wt is to be guaranteed to the flow, where t is the transmission capacity of the link. In the WFQ model in this work, customers of class $i \in \{0, 1\}$, arrive according to a Poisson process with λ_i and require service times with arbitrary distribution L_i and with mean service times $\frac{1}{\mu_i}$. It has assumed that the fluid WFQ system is stable, since the WFQ discipline is work conserving under this condition $(\rho_0 + \rho_1) < 1$, where $\rho_i = \frac{\lambda_i}{\mu_i}$ is the load offered by customers of class i. The weight of class i is w_i and, assumes that $w_0 + w_1 = 1$. It has supposed that Z_t^i denotes the workload in queue i and $\pi(z_0, z_1)$ and denotes the joint probability density function of (Z_t^0, Z_t^1) in the stationary system, which exists under the stable condition. In this work the authors provide an analytical solution for fluid WFQ system with two classes of customers by using Laplace transform. The Laplace transform $\pi^*(s_0, s_1)$ of the density distribution function $\pi(x_0, x_1)$ can be expressed by means of some unknown Laplace transforms and the proof is presented. Therefore, the result is a two variable complex function, making this approach difficult.

Guillemin and Pinchon [38] consider a weighted fair queueing system with two classes of customers with an infinite buffer. The customers of class i, $i = 0, 1$ arriving according to Poisson processes with rate λ_i require exponential service times with mean $\frac{1}{\mu_i}$. The load of queue i is $\rho_i = \frac{\lambda_i}{\mu_i}$. In this work, assume each class is assigned a virtual queue and incoming customers enter the virtual queue related to their class and served in FIFO order. The queue i is served at a rate w_i for some $w_i > 0$ when the queue $1 - i$ is not empty and at rate unity when queue is empty. It supposes that the system is stable, which amounts to assuming that the load ρ is $\rho = \rho_0 + \rho_1 = \frac{\lambda_0}{\mu_0} + \frac{\lambda_1}{\mu_1} < 1$. It assumed $N_i(t)$ is the number of customers in queue i, $i = 0, 1$ at time t. The infinitesimal generator matrix of this work is given by:

$$Q = \begin{pmatrix} B_0 & A_1 & & & \\ C_0 & B_1 & A_2 & & \\ & C_1 & B_2 & A_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & \ddots \end{pmatrix}.$$

The function $t \to \mathbf{p}(t)$ is the solution to the backward Chapman-Kolmogorov equation:

$$\frac{\partial \mathbf{P}}{\partial t}(i,j;t) = Q\mathbf{P}. \tag{2.9}$$

The authors provide an analytical solution for this system by using Laplace transform in the generating function of two dimensional Markov chain. The asymptotic behaviour of the approximation model only presents a logarithmic limit and the approximation model is applicable to two classes and, thus it is impossible to apply if there are more than two queues. Because, they obtained a two-variable complex function, and making this approach difficult.

Horváth and Telek presented another solution method, called Matrix-Geometric solution, which is detailed in Section 3.5. In this work [9], they consider a two class system and, the inter arrival times are given in two moments. They provide an approximation for the first and second moments of the waiting time. The main concept in this work is to approximate the two queue systems as the queues were separated and a service process for each that approximately follows the behaviour of the original server. The arrival process assumes a Phase-type process and requires Phase-type (PH) distributed service times with an infinite buffer. Horv´ath and Telek use the PH representation to use the matrix geometric methods as in Latouche and Ramaswami [39]. The steady state probabilities of block tri-diagonal structured Markov chains are obtained from the well known formula $\vec{\pi}\ Q = \mathbf{0}$, where $\vec{\pi}$ consists of vectors $\vec{r}_n : \vec{\pi} = [\vec{r}_0, \vec{r}_1, \ldots]$ and the block tri-diagonal generator matrix as:

$$Q = \begin{pmatrix} B_0 & A_0 & & & \\ C_1 & B & A & & \\ & C & B & A & \\ & & C & B & A \\ & & & \ddots & \ddots & \ddots \end{pmatrix}.$$

The authors use the matrix-geometric solution method to get the performance measures of the QBDs. Therefore, it evaluate two numerical examples to show the algorithm. It provide a simple formula to calculate the mean waiting time and the mean queue length of PH/PH/1 type for two classes. But this formula has limits and usability and when the system overloaded, it gets higher errors.

Many methods were investigated to give a solution to the CBWFQ Markov chain with infinite buffer. First we mention the analytical solutions. In trying to solve the problem analytically, Gross and Harris [40], faced extreme difficulty. So, they began with the simplest assumption with an infinite buffer, two service classes, exponential service times, and Poisson arrivals. The investigated method has been successfully applied to analytically solve similar queues, such as the priority queue. The two-class priority queue is a special case of the two-class CBWFQ when the weights of one class or the other is zero Fischer, Masi and Shortle [41].

The generating function approach has been applied in the two-class priority queue as two-dimensional generating function $H(z_1, z_2)$. We can see in Gross and Harris [40], a suitable summation of state equations, $H(z_1, z_2)$ as a function of known variables and an unknown one-dimensional generating function $P_{02}(z_2)$. Gross and Harris [40] drew attention to the equal service rates and the fact that the whole form for $P_{02}(z_2)$ is not required to calculate the partial derivative of $H(z_1, z_2)$ with respect to z_1, where only the value of $P_{02}(1)$ is needed and this value can be easily obtained. Thus, it get all standard performance measures for the priority queue without calculating the exact functional form for $H(z_1, z_2)$.

On the other hand, we can not apply the generating function approach for the CBWFQ form $H(z_1, z_2)$ which contains two unknown generating functions, $P_{02}(z_2)$ and $Q_{01}(z_1)$. It means that taking the partial derivative of $H(z_1, z_2)$ with respect to either z_1 or z_2 needs knowing the complete form of at least one of these two functions [40]. Therefore, the standard performance measures cannot be calculated without knowing the whole form of these two functions.

The step was to solve the equations numerically. The author in Blance [42] consider the same problem. However, they call this type of system Coupled Processor Model. They describe the steady state probabilities of the two dimensional Markov chain as a power series of the load, and by using a recursive way they can calculate the coefficients of the powers. This approach can be applied on two number of classes, and as the load approaches to 1 a large number of coefficients have to be computed to achieve a given accuracy. According to [43], Servi uses another approach to approximate the infinite model, and apply a type of Gaussian elimination to solve the infinite Markov chain. Throughout the Gaussian elimination the structure of the system is exploited to get reasonable results. However, it doesn't give good results when the load is high.

Shortle and Fischer [41] present and provide the critical role of simulation which has given great development performance analysis tools for the CBWFQ discipline. In Shortle and Fischer an extensive literature review is given for the CBWFQ system where it relates to two-class $M/M/1$ systems. In that paper they develop an approximation for the expected delay in the buffer when the load is less than one. This approximation may be classified as a system level approximation in that it is related to a Non Preemption Priority Queueing system.

2.4 Summary

This chapter gives a detailed study of the weighted fair queueing algorithms, with a particular focus on the procedure, benefits, and limitations of the fair queueing. In addition, the previous WFQ system solutions and their limitations have been studied.

Chapter 3

Mathematical Model

3.1 Introduction

The modelling of multi class systems is more challenging than the modelling of single class systems. Even when we consider the simplest case of exponentially distributed interarrival times and service times the resultant model is a multi-dimensional Markov chain, which usually does not have a closed form steady state solution.

In this chapter we introduce the mathematical models which approximate the behaviour of a multi-class systems with a WFQ service discipline, and present a derivation of the most important performance measures such as the mean queue length distribution, throughput and mean response time. In our approximate model the arrival process is assumed to be a Poisson process, and service times are assumed to be exponentially distributed. We use the Matrix-Geometric solution technique to obtain the performance measures of the Quasi Birth Death (QBDs). Section 3.5 summarizes the matrix geometric solution for computing the steady state probabilities of QBDs processes. Finally we give a summary in Section 3.8.

3.2 Simulation Model

The class based WFQ system has K classes of packets. We assume that the arrival process for each class is an independent Poisson process with rate λ_i and that the service times are exponentially distributed with mean $\frac{1}{\mu_i}$. In addition, the buffer N is shared between K classes.

In the WFQ system, there is no service interruption of packets by another. Upon service completion of a packet, the next packet to be served is randomly chosen based on the class weights. Specifically, in two-class systems when both classes are present, a class-1 packet is chosen with probability w_1 and a class-2 packet is chosen with probability $(1 - w_1)$. Packets of the same class are served in First-In-First-Out (FIFO) order.

For Markov process to model the system state must incorporate not only the number of packets of each class but also the class of the packet in service. Thus, for a 2-class system the state is $s = 0, 1, 2$ where s is a variable denoting the class of packet in service: $(s = 1, 2)$ or $(s = 0)$ when the system is empty.

Whereas in the approximation model is processor sharing in which customers from different classes are served simultaneously.

3.3 K-Class Queueing Mathematical Model

The Processor Sharing approximation to the WFQ model proposed in this work operates as follows: Consider a single-server system. Let $K > 1$ classes of jobs arrive to the multiple queueing system as K independent Poisson processes as shown in Figure 3.1. Jobs arrive according to a Poisson process at a rate λ_i and have exponential service rates with mean $\frac{1}{\mu_i}$ for class-i, $i = 1, 2, \ldots, K$. Jobs of the same class are served in First-In-

First-Out (FIFO) order.

Each class-i, is assigned a weight $0 \leq w_i \leq 1$ for $i = 1, 2, \ldots, K$. The PS approximation of WFQ model consists of a single server distributing service between K classes in proportion to their assigned weights. If all classes are nonempty, the job at head of class-i will be served with w_i. However, if some classes are empty, the head job of class-i will be served with $\hat{w}_i$:

$$\hat{w}_i = \frac{w_i}{\sum_j I_{\{n_j \neq 0\}} w_j} \tag{3.1}$$

The buffer N is shared between K classes, hence, the maximum number of jobs in the system at any time will not exceed N and any additional arriving customers will be refused entry to the system and will depart immediately without service. Only when all the K classes are empty can the server be in idle state. In the system described in Figure 3.1, the PS approximation to WFQ is applied in the following manner and the analytical models proposed give good results which are very close to WFQ algorithms [4] [5] [6], as shown in Chapter 4.

3.4 Analysis

The state of the system is given by $(n_1, n_2, \ldots, n_K)$ when there are n_i jobs of class-i, $i = 1,\ 2,\ \ldots,\ K$.

The state space can be partitioned on the total number $n = n_1 + n_2 + \ldots + n_K$ where $n = 0, 1, \ldots, N$ of jobs in the system. Within each partition the states can be partitioned further to the number of jobs of classes 1 to $K - 1$ and, within that, according to the

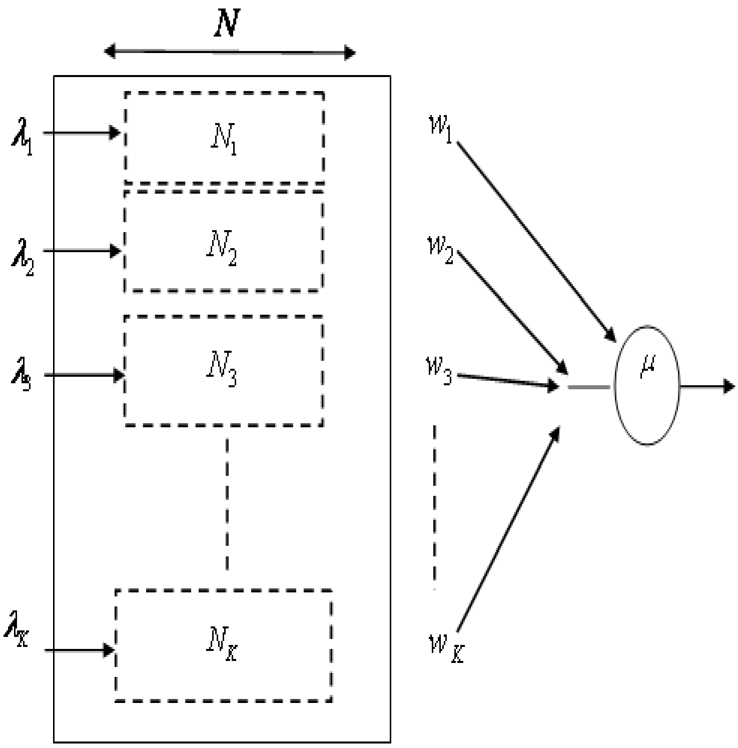

Figure 3.1: An K-class PS approximation to WFQ model

number of classes 1 to $K-2$, and so on. This ordering of the states leads to the transition rate matrix of block tri-diagonal form:

$$Q = \begin{pmatrix} B_0 & A_1 & & & & \\ C_0 & B_1 & A_2 & & & \\ & C_1 & B_2 & A_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & C_{N-2} & B_{N-1} & A_N \\ & & & & C_{N-1} & B_N \end{pmatrix}.$$

The Q is the infinitesimal generator matrix of the process. The Q is a square matrix of dimension $\frac{(K+N)!}{N!\ K!}$ and has the general structure which is quite large even for a small buffer. A_n has the rates of transitions from states with $n-1$ jobs in total to those with

n jobs. C_n has the rates of transitions from states with $n+1$ jobs in the total to those with n, and B_n is diagonal with each entry being the negative of sum of the rates in the corresponding rows of C_{n-1} and A_{n+1}.

In the K class system C_n is $(W_{k\,n})^{-1}C_{k,\,n}$, where,

$$C_{k,\,n} = \begin{pmatrix} w_k\mu_k & & & & \\ C_{k-1,1} & w_k\mu_k I & & & \\ & C_{k-1,2} & w_k\mu_k I & & \\ & & C_{k-1,3} & \ddots & \\ & & & \ddots & w_k\mu_k I \\ & & & & C_{k-1,n} \end{pmatrix},$$

and $C_{1,n} = (w_1\mu_1)$

$W_{k,\,n} =$ diag$(w_k,\,(w_kI + W_{k-1,1}),\ldots,(w_kI + W_{k-1,n-1}), W_{k-1,n})$ and $W_{1,n} = w_1$.

$A_n = A_{k,n}$ where

$$A_{k,n} = \begin{pmatrix} \lambda_k & A_{k-1,1} & & & & \\ & \lambda_k I & A_{k-1,2} & & & \\ & & \lambda_k I & A_{k-1,3} & & \\ & & & \ddots & \ddots & \\ & & & & \lambda_k I & A_{k-1,n} \end{pmatrix},$$

and $A_{1,n} = (\lambda_1)$

3.4.1 State Equilibrium Equations

The steady state probability vector $\vec{\pi}$ for Q is generally partitioned as $\vec{\pi} = [\vec{\pi}_0, \vec{\pi}_1, \ldots, \vec{\pi}_N]$, where row vectors $\vec{\pi}_0, \vec{\pi}_n, 1 \leq n < N$.

The equation $\vec{\pi}\,Q = 0$ satisfied by the invariant vector $\vec{\pi}$ can be rewritten in the form:

$$\vec{\pi}_0 B_0 + \vec{\pi}_1 C_0 = 0 \tag{3.2}$$

$$\vec{\pi}_0 A_1 + \vec{\pi}_1 B_1 + \vec{\pi}_2 C_1 = 0 \tag{3.3}$$

$$\vec{\pi}_{n-1} A_n + \vec{\pi}_k B_n + \vec{\pi}_{n+1} C_n = 0,\ 1 < n < N-1 \tag{3.4}$$

$$\vec{\pi}_{N-2} A_{N-1} + \vec{\pi}_{N-1} B_{N-1} + \vec{\pi}_N C_{N-1} = 0 \tag{3.5}$$

$$\vec{\pi}_{N-1} A_N + \vec{\pi}_N B_N = 0 \tag{3.6}$$

and $\vec{\pi}\,e = 1$ is the normalizing equation.

3.5 The Analytical Solution

The Matrix geometric technique is a method to solve stationary state probability for vector state Morkov processes. The theory of the matrix geometric solution was developed by Neuts [11], [12], [44] and [45]. It is applied in two parts, boundary set and repetitive set. The repetition of state transitions to $M/M/1$ queue considered in [46] implied a geometric solution; the repetition of the state transitions for vector processes implies a geometric form where scalars are replaced by matrices. Such Markov processes are called matrix geometric processes. Furthermore, Matrix geometric methods are applicable to both continuous and discrete time Markov processes. In our model, if we get a matrix R such that

$$\vec{P}_n = \vec{P}_{n-1}R \ \forall n \geq 1 \tag{3.7}$$

By successive substitutions into the state equilibrium equations, we obtain that

$$\vec{P}_n = \vec{P}_0 R^n \ \forall n \geq 0 \tag{3.8}$$

where
$\vec{P}_n$ is the vector state of the Markov process.
Then the solution of the form (3.8) is called the matrix geometric solution [47]. The explanation for solving a matrix geometric system is to state the matrix R, which is called the rate matrix and which we will discuss below.

3.5.1 Computation of the Rate Matrices

By simple algebraic manipulation of the state equilibrium equations we describe the calculation of R_n, $1 \leq n \leq N$ as follows:

From Equation (3.2) taking into account that B_0 is non-singular, we obtain $\vec{\pi}_0 B_0 = -\vec{\pi}_1 C_0$ Multiply both sides by B_0^{-1} results in

$\vec{\pi}_0 = -\vec{\pi}_1 C_0 B_0^{-1}$

Let $R_0 = -C_0 B_0^{-1}$ this gives

$$\vec{\pi}_0 = \vec{\pi}_1 R_0 \tag{3.9}$$

Multiplying both sides of Equation (3.6) by B_N^{-1} (B_N is non-singular) and reorganizing the result gives:

$\vec{\pi}_N = -\vec{\pi}_{N-1} A_N B_N^{-1}$

Letting $R_N = -A_N B_N^{-1}$ gives the following equation (3.10):

$$\vec{\pi}_N = \vec{\pi}_{N-1} R_N \tag{3.10}$$

Substituting Equation (3.10) in Equation (3.5) and reorganizing, we get:

$\vec{\pi}_{N-1} B_{N-1} + \vec{\pi}_{N-1} R_N C_{N-1} = -\vec{\pi}_{N-2} A_{N-1}$

By taking out the common factor in the left-hand side, we get:

$\vec{\pi}_{N-1}(B_{N-1} + R_N C_{N-1}) = -\vec{\pi}_{N-2} A_{N-1}$

Multiply both sides by $(B_{N-1} + R_N C_{N-1})^{-1}$ gives as:

$\vec{\pi}_{N-1} = -\vec{\pi}_{N-2} A_{N-1} (B_{N-1} + R_N C_{N-1})^{-1}$

Let $R_{N-1} = -A_{N-1}(B_{N-1} + R_N C_{N-1})^{-1}$, then

$$\vec{\pi}_{N-1} = \vec{\pi}_{N-2} R_{N-1} \tag{3.11}$$

Finally from Equation (3.4) and using Equation (3.11), we get a general relation between $\vec{\pi}_{n-1}, \vec{\pi}_n$ and R_n,

$\vec{\pi}_n B_n + \vec{\pi}_n R_{n+1} C_n = -\vec{\pi}_{n-1} A_n$

By taking out the common factor in the left-hand side, we obtain:

$\vec{\pi}_n(B_n + R_{n+1}C_n) = -\vec{\pi}_{n-1}A_n$

Multiply both sides by $(B_n + R_{n+1}C_n)^{-1}$:

$\vec{\pi}_n = -\vec{\pi}_{n-1}A_n(B_n + R_{n+1}C_n)^{-1}$

Simplify by letting $R_n = -A_n(B_n + R_{n+1}C_n)^{-1}$, we get

$$\vec{\pi}_n = \vec{\pi}_{n-1}R_n \tag{3.12}$$

R_n can be calculated from Algorithm 1.

Algorithm 1 Calculation of R_n

$R_N \leftarrow -A_N B_N^{-1}$
if $N \geq 1$ **then**
  for $j = N - 1 \rightarrow 1$ **do**
    $R_j \leftarrow -A_j\,(B_j + R_{j+1}C_j)^{-1}$
  end for
  return $R_n \leftarrow -A_n\,(B_n + R_{n+1}C_n)^{-1}$
end if
if $n = 0$ **then**
  return $R_0 \leftarrow -C_0 B_0^{-1}$
end if

3.5.2 Stationary Probabilities

Theorem 3.1 *For Processor-Sharing approximation of Weighted Fair Queueing analytical model process with finite state space, having an infinitesimal generator matrix Q, the* *stationary probabilities are given in matrix-geometric form by*

$$\vec{\pi}_n = \vec{\pi}_1 R_n^* \tag{3.13}$$

where $R_n^* = \prod_{j=2}^{N} R_j$ *and* R_j *is computed using Algorithm* 2 *and* $\vec{\pi}_n$ *is steady state probability at state* n*.*

Proof. We solve the system of the linear equations for $\vec{\pi}_1$, and from Equation (3.10), we obtain $\vec{\pi}_n$ as,

$$\begin{aligned}
\vec{\pi}_n &= \vec{\pi}_{n-1} R_n \\
&= \vec{\pi}_{n-2} R_{n-1} R_n \\
&\vdots \\
&= \vec{\pi}_1 R_2 \dots R_{n-2} R_{n-1} R_n \\
&= \vec{\pi}_1 \prod_{j=2}^{n} R_j = \vec{\pi}_1 R_n^*
\end{aligned}$$

□

Solving Equations (3.2) and (3.3) for $\vec{\pi}_1$ and using $\vec{\pi}_2 = \vec{\pi}_1 R_2$ we get,

$$\vec{\pi}_1 (R_0 A_1 + B_1 + R_2 C_1) = 0 \tag{3.14}$$

We use the solution (3.13) with the normalizing condition $\sum_{n=0}^{N} \vec{\pi}_n e = 1$ and Equation (3.9) to obtain:

$(\vec{\pi}_1 R_0 e + \vec{\pi}_1 \sum_{n=1}^{N} R_n^* e) = 1$

By taking out the common factor in the left-hand side, we get,

$$\vec{\pi}_1(R_0 e + \sum_{n=1}^{N} R_n^* e) = 1 \tag{3.15}$$

It is worth noting that Theorem 3.1 only gives the structure of the vector $\vec{\pi}_n$ and that the vector $\vec{\pi}$ still needs to be determined. Substituting solution (3.13) into Equations((3.3)-(3.5)), hence the vector $\vec{\pi}_1$ could be computed from either one of the equations

$$\vec{\pi}_1(R_0 A_1 + B_1 + R_2 C_1) = 0, \tag{3.16}$$

$$\vec{\pi}_1(R_{n-1}^* A_n + R_n^* B_n + R_{n+1}^* C_n) = 0, \; 1 < n < N-1, \tag{3.17}$$

$$\vec{\pi}_1(R_{N-1}^* A_N + R_N^* B_N) = 0. \tag{3.18}$$

and

$\vec{\pi}_1(R_0 e + \sum_{n=1}^{N} R_n^* e) = 1$

Consider the linear system of one of the equations ((3.16), (3.17) and (3.18)) with (3.15), this can be written as

$$A \; \vec{\pi}_1 = \vec{b} \tag{3.19}$$

where A is the number of classes-by-number of classes matrix, $\vec{\pi}_1$ and $\vec{b}$ denote the number of classes-by-one vectors. The inverse of A exists because the determinant of the matrix

A does not equal zero and A is a square matrix, thus the solution of this equation is then

$$\vec{\pi}_1 = A^{-1}\,\vec{b} \tag{3.20}$$

Algorithm 2 is used to compute the stationary probabilities $\vec{\pi}_k$.

Algorithm 2 Calculation of $\vec{\pi}$

1: $j = 2 \rightarrow n$ **do**
2: $\vec{\pi}_j \leftarrow \vec{\pi}_1 \prod_{j=2}^{n} R_j$
3: **end for**

3.6 Application of the Solution Method to Queues with More Complex Arrival and Service Process

The method in this work can be extended to cope with bursty arrivals Markov-Modulated Poisson Process (MMPP), Neuts Process and Hyper-exponential arrival and service distributions.

Everything can be represented as $M/M/1/N$ queues in a randomly changing environment. For example, we consider a single server queueing system as shown in Figure (3.1) when two classes of traffic wait for service. Assume that inter-arrival times for each class is phase renewal process. When the continuous time Markov chain (CTMC) is phase s, $(s = 1, 2)$, arrivals occur according to a Poisson process of rate λ_{si} for class-i and the intensity of transition from phase 1 to phase 2 is ϕ_{12} and phase 2 to phase 1 is ϕ_{21}. The service time of class-i, $(i = 1, 2)$ traffic is exponentially distributed with mean $\frac{1}{\mu_{si}}$.

Randomly changing environment infinitesimal generator matrix is given by:

$$\Phi = \begin{pmatrix} -\phi_{12} & \phi_{12} \\ \phi_{21} & -\phi_{21} \end{pmatrix}$$

Consider the Markov chain that corresponds to the queueing process. This chain has states (n_1, n_2, s). The n_1 signifies the number of the class-1, the n_2 signifies the number of the class-2 and s is the phase of the service process.

Let $\vec{\pi}$ be the steady state probability vector of this Markov chain:

$$\vec{\pi} = (\vec{\pi}_{0,1}, \vec{\pi}_{1,1}, \ldots, \vec{\pi}_{N,1},\ \vec{\pi}_{0,2}, \vec{\pi}_{1,2}, \ldots, \vec{\pi}_{N,2})$$

The infinitesimal generator matrix of this Markov chain is $\mathbf{Q}$ and the steady-state probability vector $\vec{\pi}$ satisfies the following equations

$$\vec{\pi}_1 \, Q_1 - \phi_{12}\vec{\pi}_1 + \phi_{21}\vec{\pi}_2 = 0 \tag{3.21}$$

$$\vec{\pi}_2 \, Q_2 - \phi_{21}\vec{\pi}_2 + \phi_{12}\vec{\pi}_1 = 0 \tag{3.22}$$

From equations (3.21) and (3.22), the $\vec{\pi}_s \, Q_s$ for $s = 1, 2$ is the same of the case of exponentially distributed interarrival times and service times.

3.7 Derivations of Performance Metrics

In this section we derive the analytical expressions of the performance metrics for the multi-class jobs, under PS approximate to WFQ system, specifically the mean queue length, mean response time and throughput. The derivation of these performance metrics is adapted from the derivations of performance metrics of the non-preemptive $M/M/1/N$ system with FIFO scheduling found in [48]. It is possible to use the same derivations because the effect of the weights in the PS approximate to WFQ scheduling is explicit in the stationary probabilities.

3.7.1 Mean Queue Length

The value for L, the expected number of jobs in the PS approximate to WFQ system, is done by summing the weighted probabilities having n jobs in the system. The mean queue length L is calculated from the $M/M/1/N$ model as follows:

$$L = \sum_{i=0}^{N} n \;\; \pi_n \tag{3.23}$$

Where π_n is the steady state probabilities of n jobs in the system, $n = 0,\ 1,\ \ldots, N$.

The following equations are derived from Equation (3.23) to find $L_1,\ L_2,\ \ldots,\ L_K$ the expected number of jobs in class-1, class-2, ..., class-K, respectively:

$$L_1 = \sum_{n_1=0}^{N} \sum_{n_2=0}^{N-n_1} \sum_{n_3=0}^{N-n_1-n_2} \cdots \sum_{n_N=0}^{N-n_1-n_2-n_3-\ldots-(n_{N-1})} n_1\, \pi(n_1,\ n_2,\ \ldots,\ n_K) \tag{3.24}$$

$$L_2 = \sum_{n_1=0}^{N} \sum_{n_2=0}^{N-n_1} \sum_{n_3=0}^{N-n_1-n_2} \cdots \sum_{n_N=0}^{N-n_1-n_2-n_3-\ldots-(n_{N-1})} n_2\, \pi(n_1,\ n_2, \ldots,\ n_K) \tag{3.25}$$

$$\vdots$$

$$\vdots$$

$$L_K = \sum_{n_1=0}^{N} \sum_{n_2=0}^{N-n_1} \sum_{n_3=0}^{N-n_1-n_2} \cdots \sum_{n_N=0}^{N-n_1-n_2-n_3-\ldots-(n_{N-1})} n_K\, \pi(n_1,\ n_2, \ldots,\ n_K) \tag{3.26}$$

Where $\pi_{(n_1, n_2, \ldots, n_K)}$ is the steady state probability at state $(n_1,\, n_2,\, \ldots,\, n_K)$.

3.7.2 Mean Response Time

Using Little's law, the mean response time $S_1,\, S_2,\, \ldots,\, S_K$ for class-1, class-2, class-K is obtained as:

$$
\begin{aligned}
S_1 &= \frac{L_1}{\lambda_{effective_1}} + \frac{1}{\mu_1} && (3.27)\\
S_2 &= \frac{L_2}{\lambda_{effective_2}} + \frac{1}{\mu_2} && (3.28)\\
&\vdots\\
&\vdots\\
S_K &= \frac{L_K}{\lambda_{effective_K}} + \frac{1}{\mu_K} && (3.29)
\end{aligned}
$$

where $\lambda_{effective}$ is the net arrival rate, i.e. the rate of jobs not dropped, and is given by

$$
\begin{aligned}
\lambda_{effective_1} &= \lambda_1(1 - \pi_N) && (3.30)\\
\lambda_{effective_2} &= \lambda_2(1 - \pi_N) && (3.31)\\
&\vdots\\
&\vdots\\
\lambda_{effective_K} &= \lambda_N(1 - \pi_N) && (3.32)
\end{aligned}
$$

where π_N is the probability that an arriving job finds the queue full (at state N).

3.7.3 Throughput

An expected number of jobs being served (throughput T) for the $M/M/1/N$ queue from the analytical model [17] is

$$T = \lambda_{effective} \tag{3.33}$$

Equation (3.33) is used to obtain the throughput $T_1,\ T_2,\ \ldots, T_K$ for class-1, class-2, class-K, respectively:

$$\begin{aligned} T_1 &= \lambda_{effective_1} && (3.34)\\ T_2 &= \lambda_{effective_2} && (3.35)\\ &\vdots \\ &\vdots \\ T_K &= \lambda_{effective_K} && (3.36) \end{aligned}$$

3.7.4 Job Loss

Under steady-state assumptions the functional form of Job loss $Loos_i$, $i = 1, 2$ for class-1 and class-2, respectively, gives as

$$Loos_i = \frac{\lambda_1}{\lambda_1 + \lambda_2} \pi_N. \tag{3.37}$$

where π_N is the probability that an arriving job finds the queue full (at state N).

3.7.5 Jitter

Packets from the source will reach the destination with different delays. A packet's delay varies with its position in the queues of the routers along the path between source and destination and this position can vary unpredictably. This variation in delay is known as jitter and can seriously affect the quality of streaming audio and video [49].

Jitter, a measure of variance in delay, is becoming important in data networks. Low Jitter can be achieved with isochronous network or with a protocol that handles the transmission of real-time audio and video; the Internet uses the protocol approach. Also, Jitter (variance of delay) is an important measure of performance, especially the variation in transmission delays of delay-sensitive telecommunications traffic, such as voice and real-time video). By controlling or managing queue length and hence delay also expect to manage jitter [50].

Effectively jitter is an important issue and is an area for further work that is a realtively straightforward extension of the analytical model but is not considered in this work.

3.8 Summary

In this chapter the mathematical model has been presented. We found that the steady state holds for any number of classes and any number of buffers as shown in equations (3.13), thus instead of having to solve a huge number of equations to calculate the steady state probabilities, we can use the steady state equation. That is the steady state holds where we have Poisson input arrivals and exponential service time distribution. This result will be used in the mathematical models in numerical results Chapter (4).

Chapter 4

Numerical Results

4.1 Introduction

The approximation models verification has a special importance. The influence of all the parameters on the accuracy of the results has been checked by comparing them to simulation results for the WFQ the model.

The mathematical models presented in Chapter 3 have been validated by comparing the results obtained from the mathematical models and the results obtained from simulation (simulation results) using the QNAP-2 simulator and in comparison with the results by [9]. The simulation is run until a steady state is reached.

A number of cases have been studied using the analytical model (described in Chapter 3) and give results which are very close to those from simulation at 95% confidence interval [4] [5] [6].

The models presented in Chapter 3 are based on Poisson arrivals and exponential service time. In this Chapter, the numerical results of the performance metrics for the two and three class jobs have been calculated, specifically the mean queue length, mean response time and throughput. The proposed approach has been illustrated through examples; and investigated the effect of the weight on the performance of the system. Representative cases are presented in as follows:

- Section 4.2 presents the results for the $M_1{+}M_2/M/1/N$ class. The capacity of the system is set to be $N_1{+}N_2 = 50$; hence this model has 1326 states. Five representative cases are presented in this Section.
- Section 4.3 presents the results for the $M_1{+}M_2{+}M_3/M/1/N$ class. The capacity of the system is set to be $N_1{+}N_2{+}N_3 = 28$; hence this model has 4495 states. Three representative cases are presented in this Section.

4.2 Two-class Model

The table gives the parameters for the five cases that are presented in this section 4.2.

Table 4.1: The parameters used in the Two-class Model

Case	λ_1	μ_1	w_1	λ_2	μ_2	w_2
1	$[0.1, 1]$	1	0.4	$[0.1, 1]$	1	0.6
2	0.2	1	0.4	$[0.1, 1]$	1	0.6
3a	0.2	1	$[0, 1]$	0.3	1	$[1, 0]$
3b	0.6	1	$[0, 1]$	0.7	1	$[1, 0]$
4	0.1	2	0.9	$[0.1, 1]$	2	0.1
5a	0.7	1	0.9	$[0.1, 1]$	10	0.1
5b	0.7	1	0.9	$[1, 10]$	10	0.1
5c	0.7	1	0.9	0.1	$[1, 10]$	0.1

Case 1 is to examine the effect of the weight on the system, when the service demand for both classes are equal, i.e. $\mu_1 = \mu_2 = 1$, and $\lambda_1 = \lambda_2$. The effects of changing the assigned parameters on the mean queue length, mean response time and throughput have been tested.

Case 2 is to study the effect of changing system load on the mean queue length, mean response time and throughput. The results in case 1 and case 2 are illustrated in the Figures (4.1)–(4.10). The relative error is 1% at most. The parameters are the same as in Horváth and Telek [9]. The results of the mathematical model match the simulation results better than that of [9].

The third case shows the effect of changing the assigned weight on the mean queue length, the mean response time and throughput. The third case is studied under both low load (in case 3a) and high load (in case 3b). The impact of the weight of class-1 and the weight of class-2 on its own performance metrics has been analysed.

a) The mean queue length, the mean response time and throughput values using the fixed arrival rate for class-1 and class-2 as recommended in [9] when traffic is light ($\rho < 0.5$). Numerical results as well as simulation results are shown in Figures (4.15– 4.17). **b)** The effect when traffic is heavy ($\rho > 1$) has been examined. The results in Figures (4.18– 4.20) have been proved that the comparisons between the analytical results and the simulation results are good even with high load.

Case 4 presents the effect of changing class-2 load with low weight and fixed arrival rate for class-1 with high weight. Graphs of the mean response time of the analytical and simulation models for a range of buffer sizes $N = 50$ and $N = 5$ are shown in Figures (4.21) and (4.22), respectively.

Figures (4.27– 4.26) show the mean queue length and the mean response time, for the case 5 in Table 4.1, respectively, when the service times are unequal. The comparisons with simulation results are good when $\mu_i = 2, i = 1, 2$ in case 4. For the case 5 when $\mu_1 = 1, \mu_2 = 10$ they are not good. In addition, to explain the behaviour of the WFQ and

PS models are very similar when the service rates are very similar and poorer when they are different.

In all five cases, there is a slight difference between analytical and simulation results in both class-1 and class-2. It is mostly visible for value $\rho \approx 1$, when the entire buffer becomes a shared region both streams, and become homogeneous. Because the mathematical model depends on serving two classes at the same time with probability w_1 for class-1 and w_2 for class-2 [4], the steady state birth and death equations are presented in Appendix A. However, the simulation model serves one class in different time as shown in Algorithm 3.

To show the difference in the steady state equations between two models consider the case in which $n_1, n_2 > 0$ and $n_1 + n_2 < N$ in the two-class model.

In WFQ simulation model:

$$(\lambda_1+\lambda_2+\mu_1)\pi_{n_1,n_2,1} = \lambda_1\pi_{n_1-1,n_2,1}+\lambda_2\pi_{n_1,n_2-1,1}+w_1\mu_1\pi_{n_1+1,\,n_2,1}+w_1\mu_2\pi_{n_1,n_2+1,2} \quad (4.1)$$

and

$$(\lambda_1+\lambda_2+\mu_2)\pi_{n_1,n_2,2} = \lambda_1\pi_{n_1-1,n_2,2}+\lambda_2\pi_{n_1,n_2-1,2}+w_2\mu_1\pi_{n_1+1,\,n_2,1}+w_2\mu_2\pi_{n_1,n_2+1,2} \quad (4.2)$$

and, in mathematical approximation to WFQ model:

$$(\lambda_1+\lambda_2+w_1\mu_1+w_2\mu_2)\pi_{n_1,n_2} = \lambda_1\pi_{n_1-1,n_2}+\lambda_2\pi_{n_1,n_2-1}+w_1\mu_1\pi_{n_1+1,\,n_2}+w_2\mu_2\pi_{n_1,n_2+1} \quad (4.3)$$

Multiply Equation (4.1) by w_1 in and Equation (4.2) by w_2, and add.

$$\begin{aligned}(\lambda_1+\lambda_2+w_1\mu_1+w_2\mu_2)\pi_{n_1,n_2} &= \lambda_1\pi_{n_1-1,n_2}+\lambda_2\pi_{n_1,n_2-1} \\ &+ (w_1^2+w_2^2)\mu_1\pi_{n_1+1,n_2}+(w_1^2+w_2^2)\mu_2\pi_{n_1,n_2+1} \quad (4.4)\end{aligned}$$

Algorithm 4 Two Class Model (class-i, class-j)

w_i = weight of class-i
w_j = weight of class-j
n_i = number of jobs in class-i
n_j = number of jobs in class-j
if wran$_{i,j} \leq w_i$ **then**
  if $n_i > 0$ **then**
    serve class-i
  else serve class-j
  end if
else if $n_j > 0$ **then**
    serve class-j
  else serve class-i
end if

4.2.1 Case 1

In Figure (4.1), the overall mean response time of class-1 and class-2 increases with the increase of the arrival rate. Because of the consequent increase in the mean queue length given that the queue capacity is finite, with low load for $\rho \leq 0.4$ the mean response time is similar but when $(\lambda_1 + \lambda_2) \approx \mu$. The increase in the overall mean response time is significantly higher in the class-1. Because, the class-1 has additional delay of low weight waiting for high weight of class-2 to be served and the drop in class-2 happened later than class-1 at $\lambda_2 = 0.5$. The mean queue length confirms this in Figure (4.2).

Figures (4.1) and (4.2) show that by increasing the arrival rate value, the mean queue length and the mean response time for both the analytical results and the simulation results increases with almost the same percentage. However, there is a slight difference between analytical and simulation results in both class-1 and class-2. It is visible for value $\rho \geq 1$, when the entire job becomes a shared region both streams, being homogeneous. Because the mathematical model depends on serves two classes at the same time with probability w_1 for class-1 and w_2 for class-2, but in the simulation model serve one class in a different time.

Figure (4.3) shows the throughput. The system throughput increases with the increase of arriving traffic until the total arrival rate exceeds service rate; that is when $\lambda_1 + \lambda_2 = \mu_1 = \mu_2$. From that point after dropping, the system throughput will not exceed the service rate (in this example: $\mu_1 = \mu_2 = 1$).

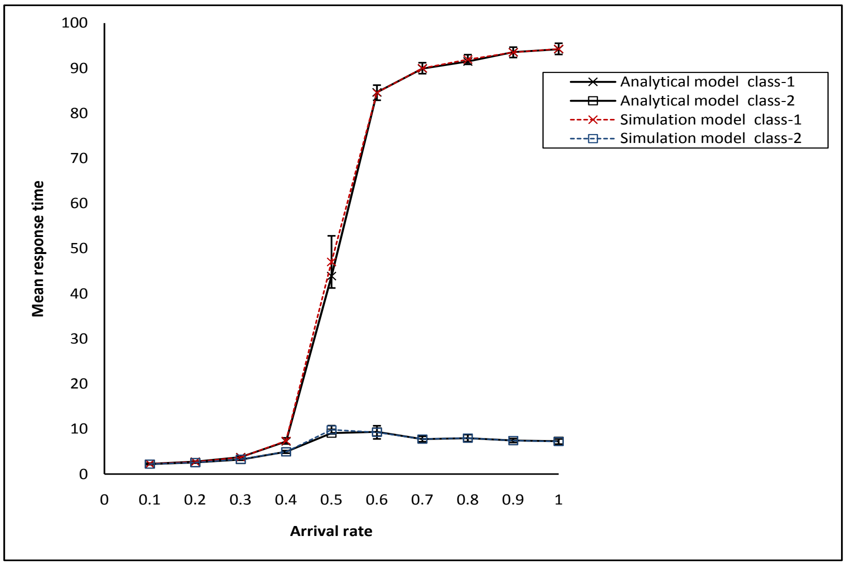

Figure 4.1: Case 1, Mean response time.

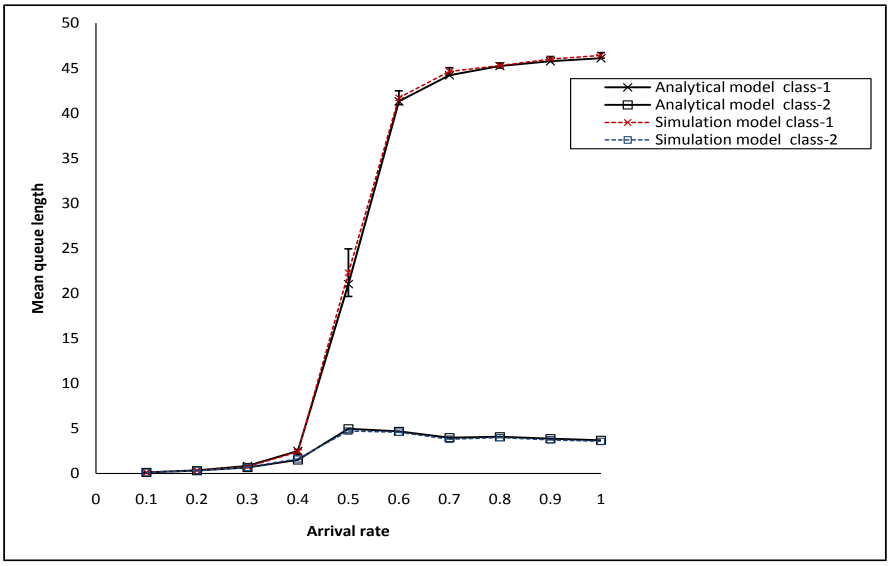

Figure 4.2: Case 1, Mean queue length.

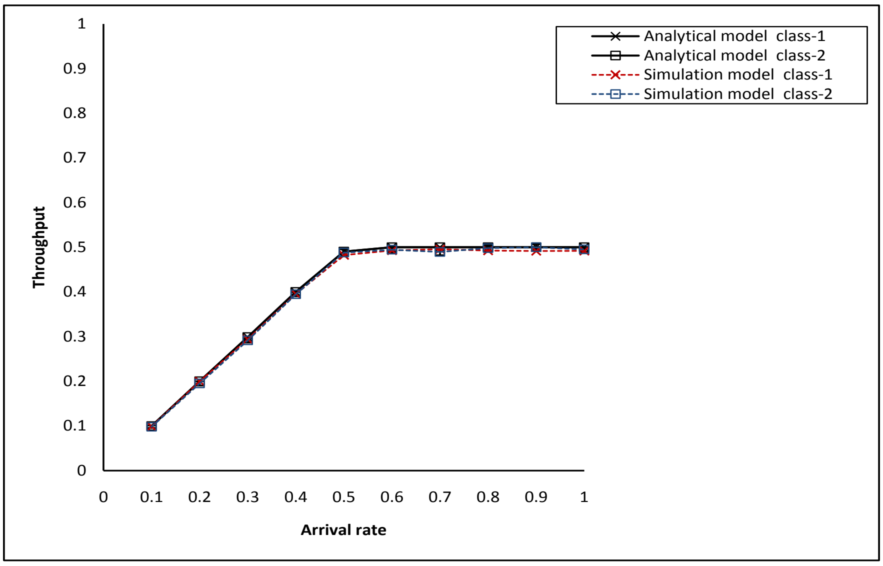

Figure 4.3: Case 1, Throughput.

Figures (4.4) to (4.7) illustrate the comparison of the analytical and the simulation results with exponentially distributed inter-arrival times and service times, and simulation results with generalized exponential (GE) inter-arrival times and exponentially distributed service times.

In the case of GE distribution of inter-arrival times for class-1 and class-2, since $\lambda_1 = \lambda_2$ and the squared coefficient of variance inter-arrival time for class-1 and class-2, respectively, $C^2_{a_1} = C^2_{a_2} = 5$. Since the mean arrival rates for both classes are equal, as well as the higher burstiress of the arrival jobs of both classes, both classes are built up the job loss will be almost the same for low load. However, for high load the job loss for class-1 is greater than the job loss for class-2. Regarding the mean response time, since class-2 will be served with $w_2 = 0.6$, its mean response time will be lower than that of class-1.

Figures (4.4) to (4.7) show the effect of inter-arrival and service times that are more variable than exponential. Figures (4.4) to (4.6) are the same as Figures (4.1) to (4.4) but with the addition of results from simulation where the inter-arrival times and service times have generalized exponential (GE) distributions withe same same mean rates as previously but squared coefficients of variance equal to five. Figure (4.7) shows the loss rates.

The results show quite clearly the earlier onset of congestion in the region below saturation (at arrival rates of 0.5). Near saturation and when the system is overloaded the effect of the finite buffer dominations behaviour: the so-called “heavy traffic conditions”.

In this experiment class-2 customers were given a favourable service weight, 50% greater than class-1, but both have the same arrival rates. Under heavy load, when the buffer is almost always full, almost all the customers queueing will be of class-1. In the case of exponential inter-arrival and service times, the ratio of the two classes stabilized at about 40% overload (both classes arrival rate of 0.7) whereas for the most variable GE inter-arrival and service times the ratio of class-1 to class two continued to increase with degree of overload.

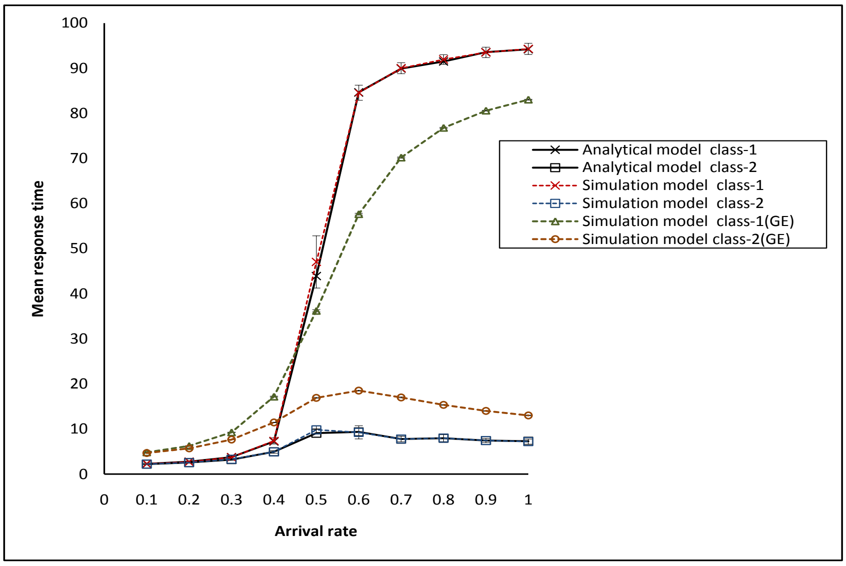

Figure 4.4: Case 1, Comparison of the mean response time with the different arrival distribution

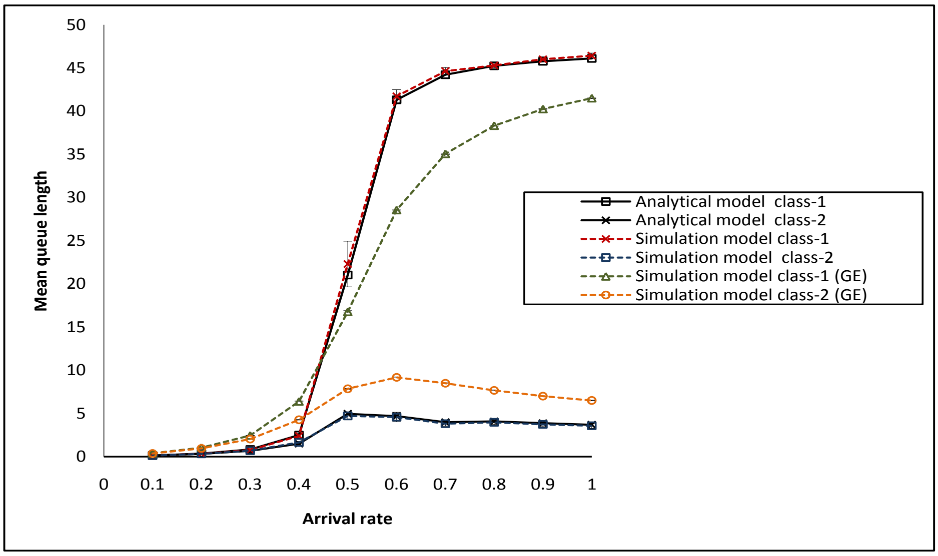

Figure 4.5: Case 1, Comparison of the mean queue length time with the different arrival distribution

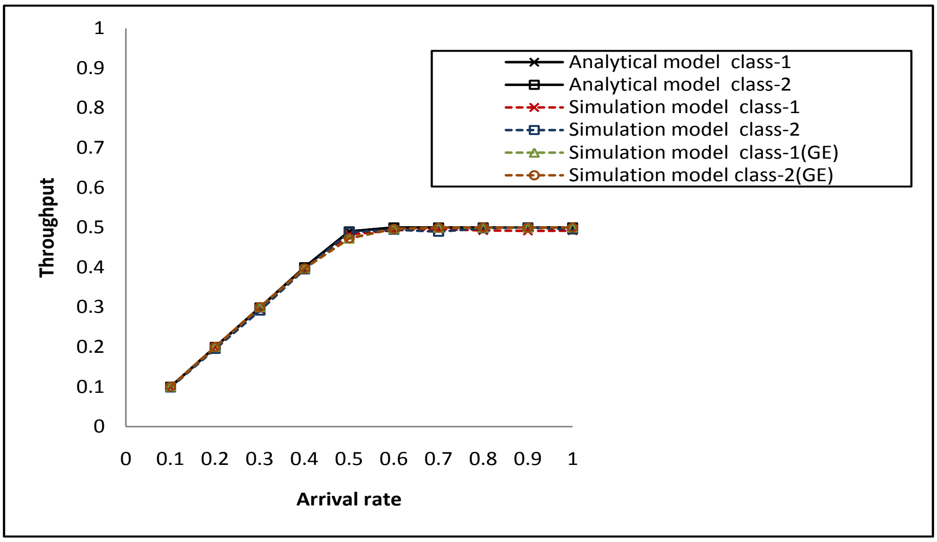

Figure 4.6: Case 1, Comparison of the throughput with the different arrival distribution

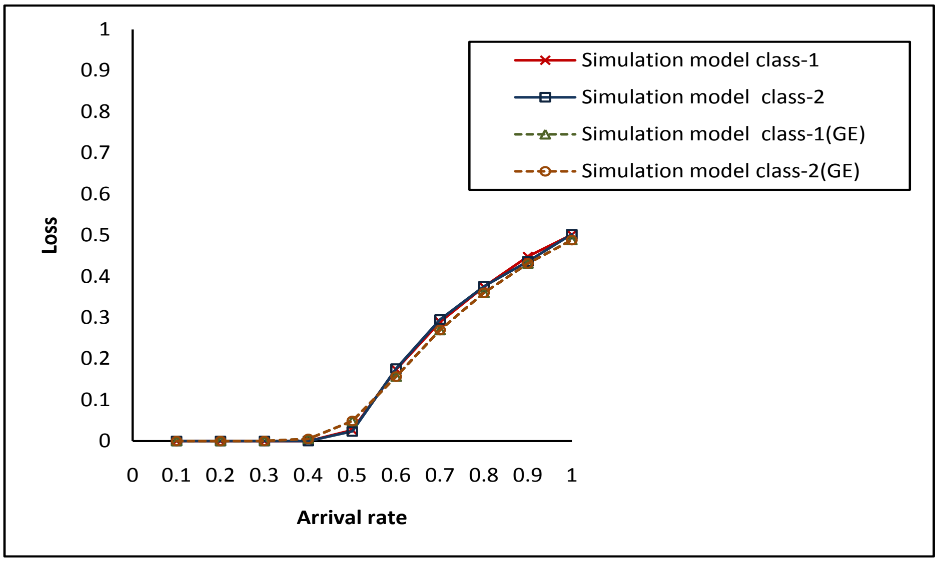

Figure 4.7: Case 1, Comparison of the loss with the different arrival distribution.

4.2.2 Case 2

From Figures (4.8) and (4.9), when traffic is light, (ρ_1+ρ_2 $<$ 1) has been observed that, the response time of class-1 was higher than the response time of class-2 because $w_1\mu_1 < w_2\mu_2$. The higher arrival lead to longer queue length for class-1. When traffic is heavy, (ρ_1+$\rho_2 > 1$) the arrival of class-2 was greater than class-1, hence the queue length was longer for class-2. However, when $\lambda_2 = 0.8$ the buffer was full, with more jobs from class-2 than class-1 because of the higher arrival of class-2.

For class-2, even though the probability of being served was higher than for class-1, the long queue leads to longer waiting times. However, for class-1, the full buffer is filled with more jobs from class-2 and whenever a job was served the probability of new a job being from class-1 was higher than from class-2 due to class-2 higher arrival. This lead to an ever decreasing proportion of class-1 jobs in the buffer, the saturation lead to shorter response times for the class-1 jobs.

As a result, we note that from Figure (4.10) was that as λ_2 increased, and then T_1 and T_2 increase until the system reaches a saturation point at $\lambda_2 = 0.8$. However, the throughput of class-1 increases until $\lambda_2 = 0.6$ and then slightly decreases.

An interesting observation from Figure (4.10) is that the class-2 with higher weight had a higher throughput, which means a weight can significantly the throughput under good condition such as arrival rate and service rate. For the analytical model and simulation model results, the comparisons were quite good with finite buffer for both classes.

Comparing the analysis and simulation results, the accuracy of the approximation was good, except on the mean response time of class-2 around $\lambda_2 = 0.6$ in first case. This is the point from where queue size of class-2 is overloaded. The results of the six plots are reasonably accurate with error $\leq 1\%$ in second case at all points.

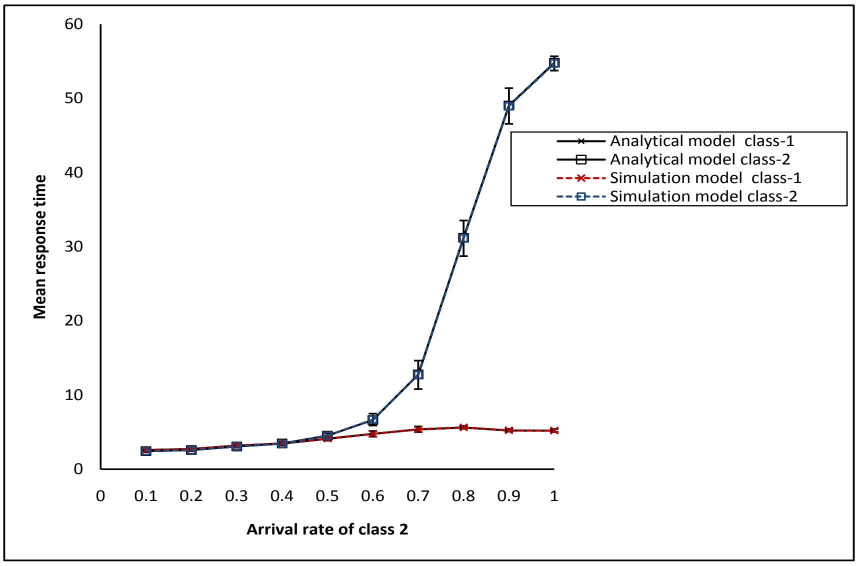

Figure 4.8: Case 2, Mean response time.

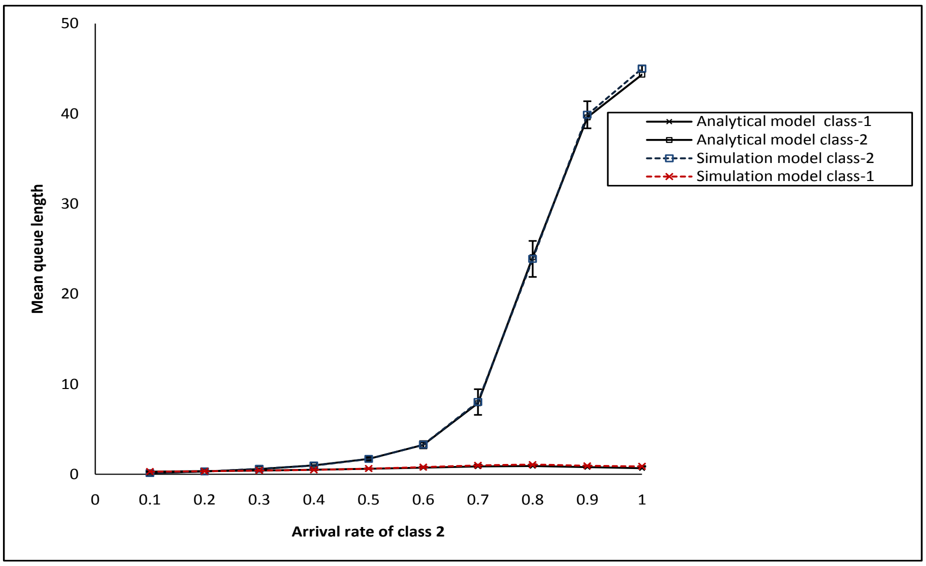

Figure 4.9: Case 2, Mean queue length.

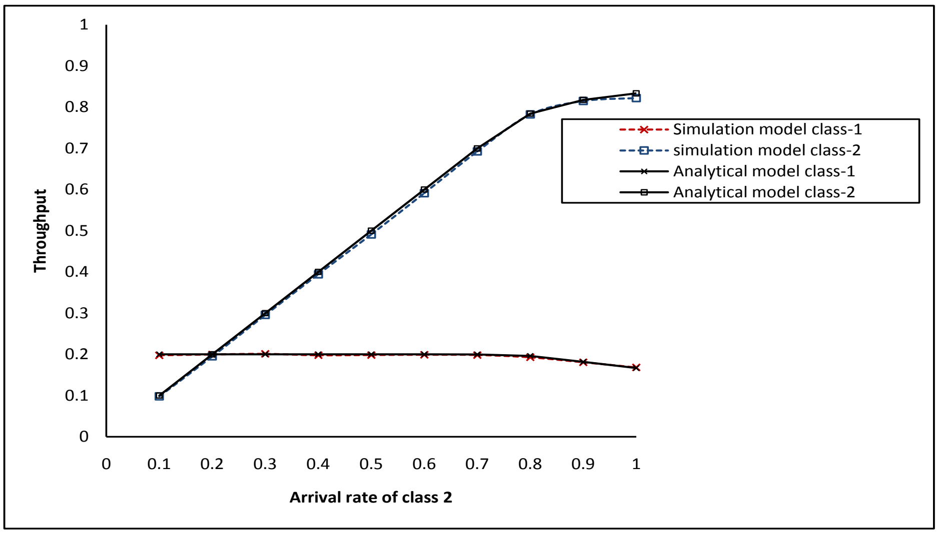

Figure 4.10: Case 2, Throughput.

Figures (4.11) to (4.14) illustrate the comparison of the analytical and the simulation results with exponentially distributed inter-arrival times and service times, and simulation results with exponentially distributed inter-arrival times and 2 phase balanced hyper-exponential distributed service times. In the case of balance hyper-exponential distribution for service times for class-1 and class-2, the squared coefficient of variance service time for class-1 and class-2, respectively, $C^2_{s_1} = C^2_{s_2} = 5$.

The job loss of hyper-exponential distribution for service was higher than exponential distribution for service. As a result, compared with exponential service times, the mean response time for class-1 in the case of balance hyper-exponential distribution for service times was higher because the mean queue length was higher. However, for class-2 the mean response time and mean queue length for balance hyper-exponential distribution for service was higher than that of exponential under low load because of low job loss. On the contrary, the class-2 mean queue length was lower than that for exponential as there was high job loss under high load.

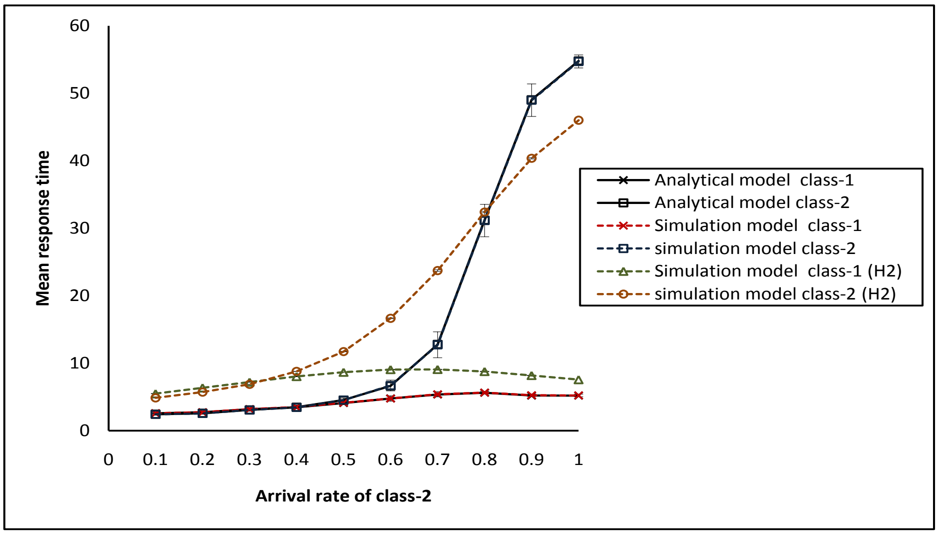

Figure 4.11: Case 2, Comparison of the mean response time with the different service distribution.

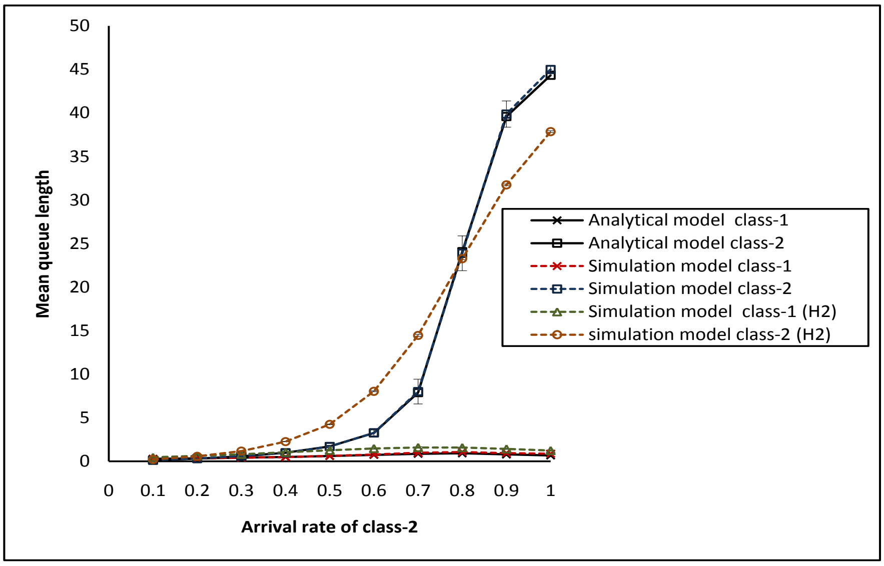

Figure 4.12: Case 2, Comparison of the mean queue length with the different service distribution.

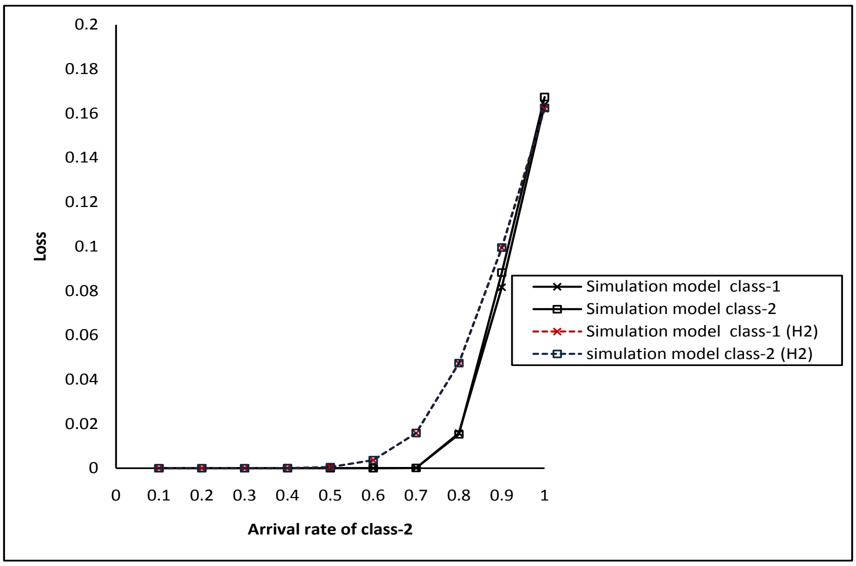

Figure 4.13: Case 2, Comparison of the job loss with the different service distribution

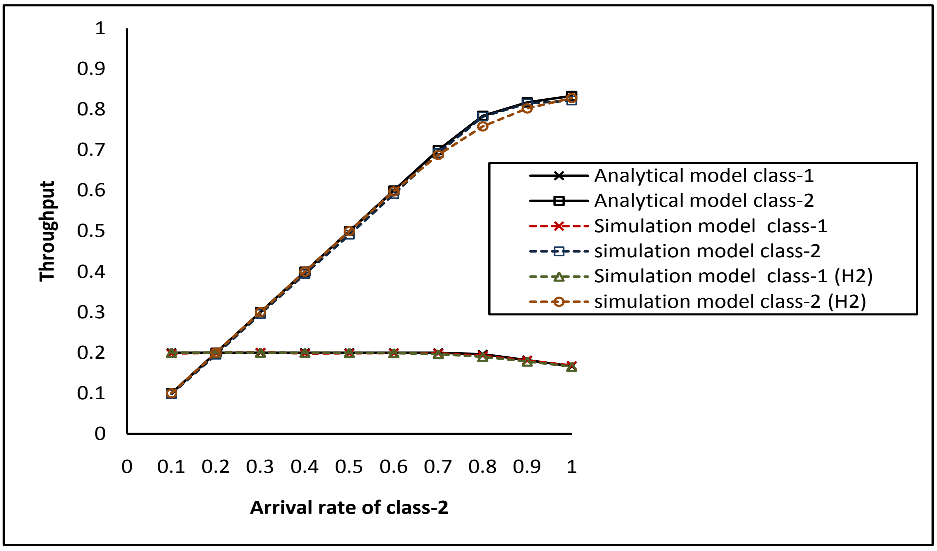

Figure 4.14: Case 2, Comparison of the throughput with the different service distribution.

4.2.3 Case 3

Figure (4.15) shows that as the weight increased the mean response time of class-1 jobs tended to decrease, whilst that of class-2 jobs increased. This is because the mean service rate of class-1 traffic increased. On the other hand, the mean service rate of class-2 traffic reduced. The remarkable increase in the queueing delay of class-2 traffic is clearly shown in Figure (4.15).

In Figure (4.16), the mean queue length is depicted as a function of weight. Class-1 having weight 0 means that class-2 has priority. The results reflect this behavior.

Figure (4.17) reveals that the effects of traffic input on the throughput for both classes are not significant. This is because the PS approximate to WFQ scheduling scheme provides a guaranteed service of the class-1 and class-2 and, hence the throughput of two classes equals with the increase in the weight of class-1 and the throughput of class-1 $= 0.2 = \lambda_1$

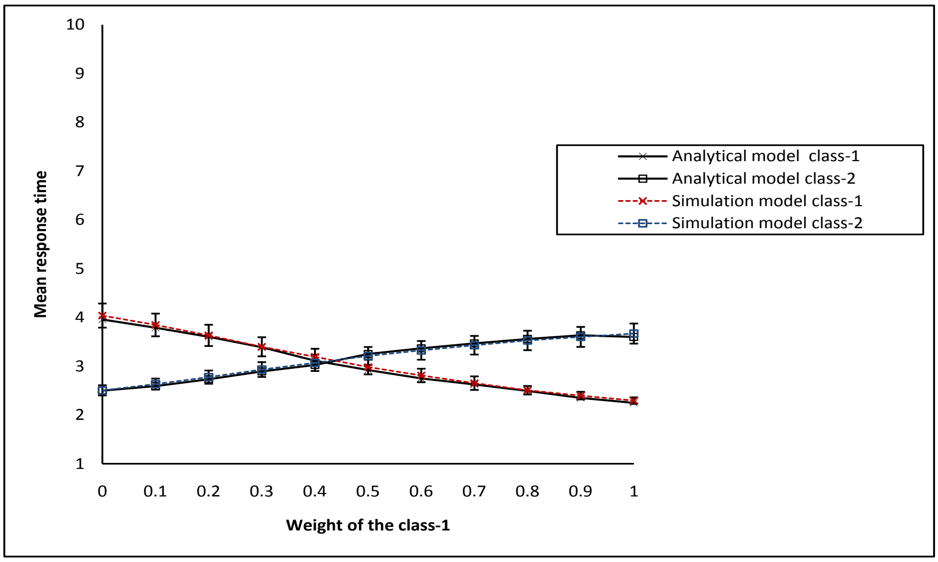

Figure 4.15: Case 3a, Mean response time.

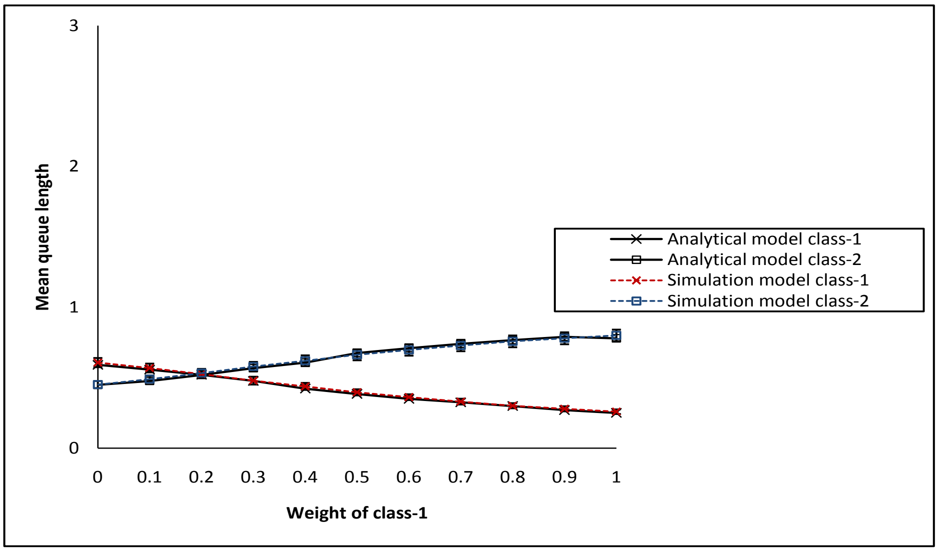

Figure 4.16: Case 3a, Mean queue length.

and class-2 $= 0.3 = \lambda_2$. There are not many jobs in the classes. For all considered the throughput of class-1 and class-2, obtained the calculated analytical model results show an exact match with simulation results with finite case.

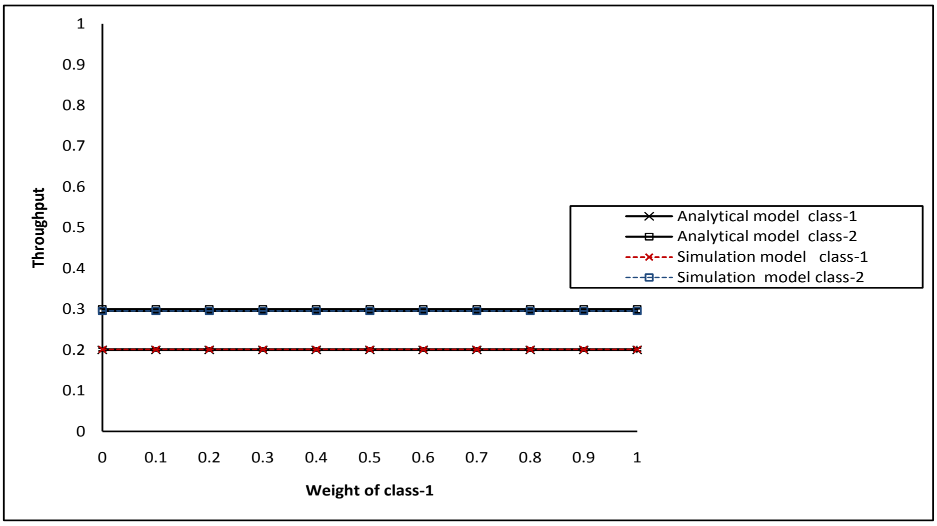

Figure 4.17: Case 3a, Throughput.

The results are depicted on (4.15– 4.17). Investigating the accuracy of the approximation, in this case we can draw the same conclusion as in case $3b$.

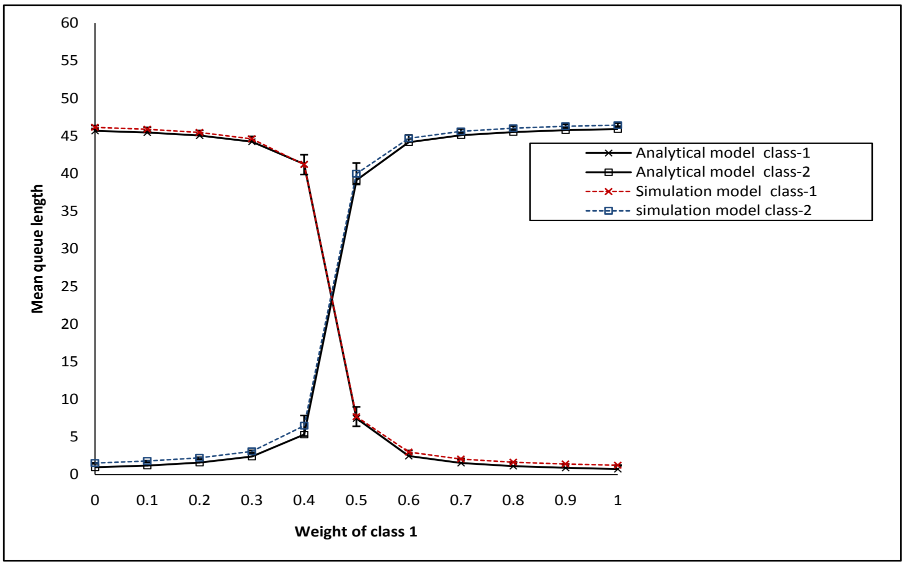

Figure 4.18: Case 3b, Mean queue length.

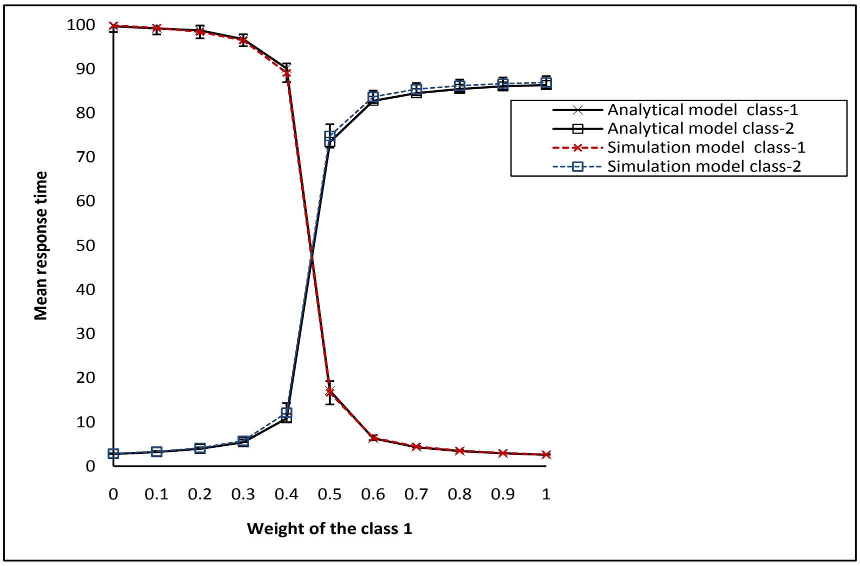

Figure 4.19: Case 3b, Mean response time.

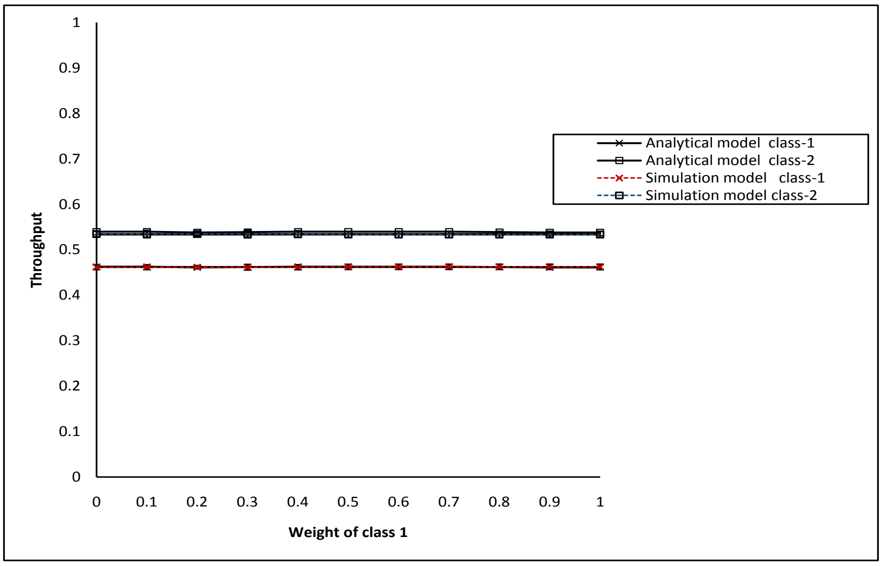

Figure 4.20: Case 3b, Throughput.

4.2.4 Case 4

In Figures (4.21) and (4.22), significant detrimental impact of increasing the arrival rate of class-2 on the mean response time was detected. In addition, the characteristic of the analytical model as the load increases, the significant increase in the delay for the low weight class was shown. The slight difference between the analytical and simulation results is likely due to all classes sharing processes at the same time in the analytical model (see Section 4.2 introduction).

Nevertheless, these graphs show excellent agreement between the analytical model and the simulation, the error being less than 2% at all points even at $\lambda_2 = 0.9$. Figures (4.21) and (4.22) illustrate offered throughput to the buffer, where dark curves represents the analytical results and the dotted lines show simulation results.

The job loss was calculated using different values of arrival rate for class-2. Figure (4.25)

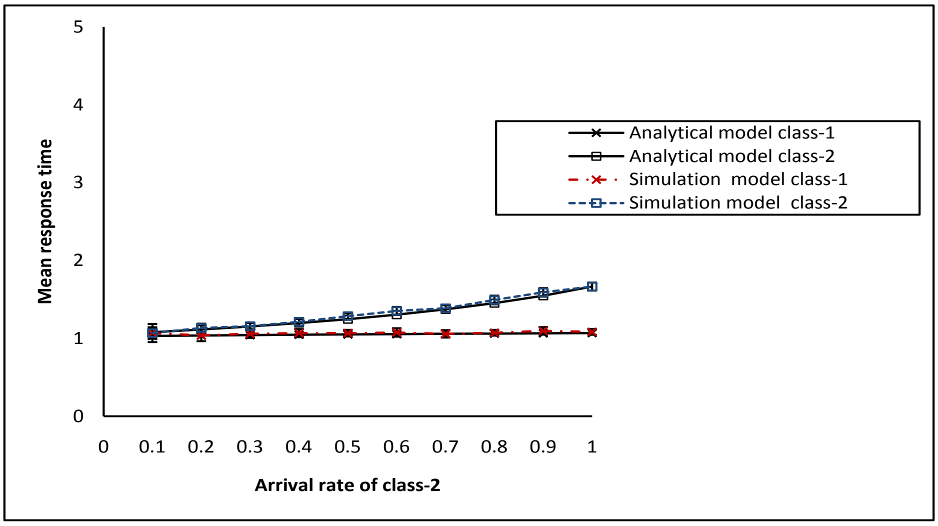

Figure 4.21: Case 4, Mean response time, $N = 50$.

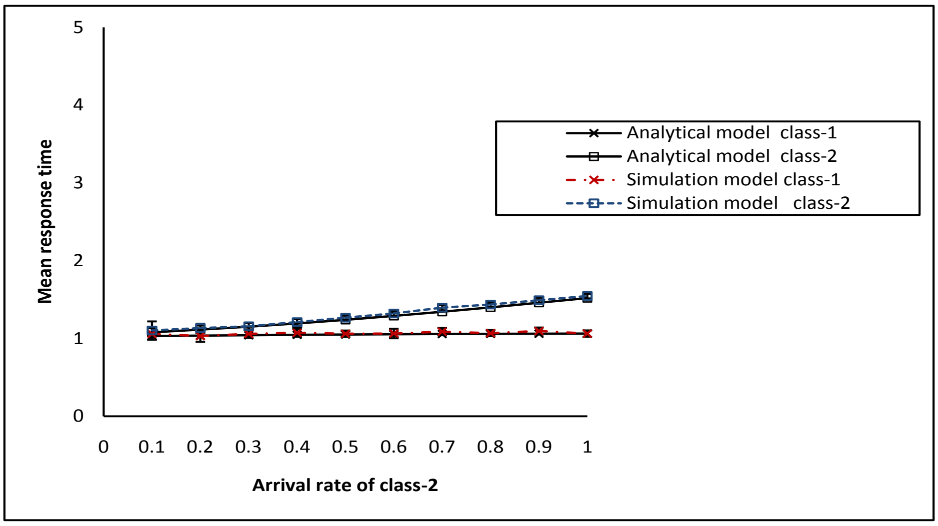

Figure 4.22: Case 4, Mean response time, $N = 5$.

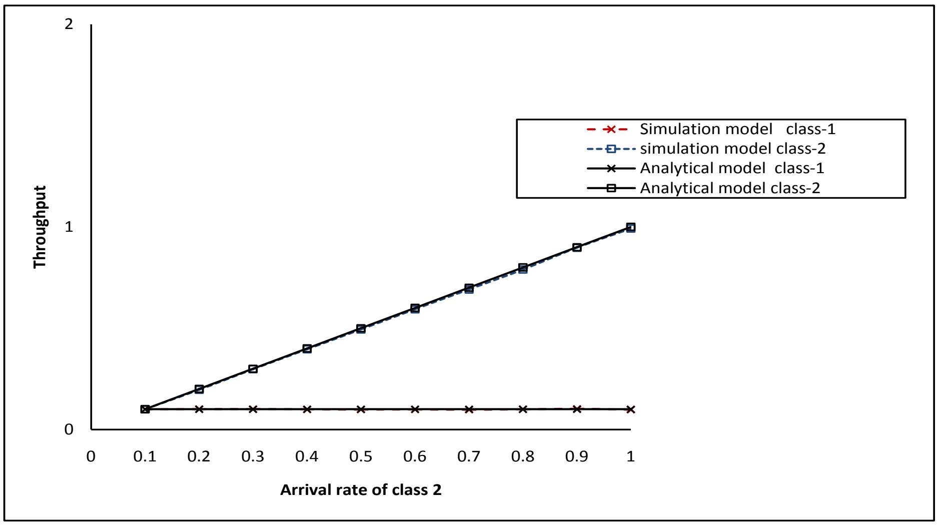

Figure 4.23: Case 4, Throughput, $N = 50$.

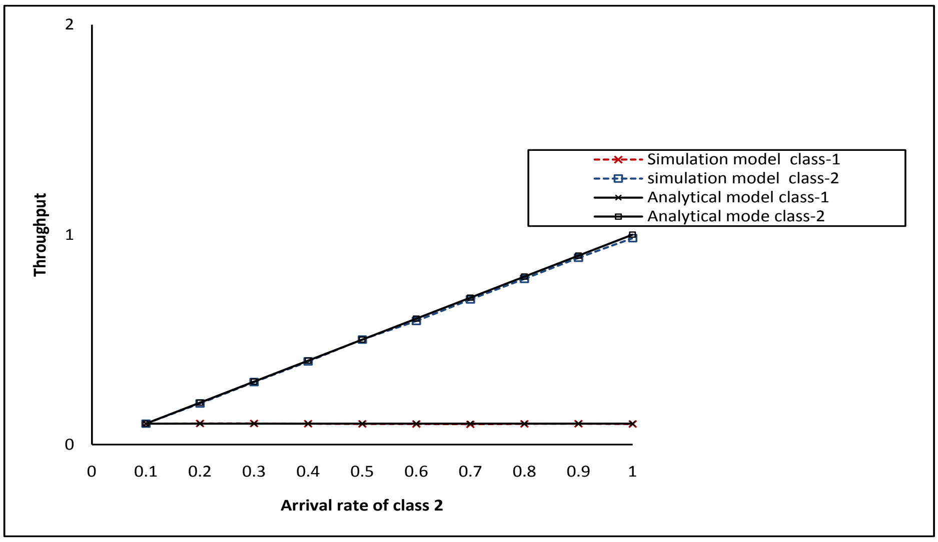

Figure 4.24: Case 4, Throughput, $N = 5$.

plots the resulting curves. The job loss of the class-1 less than job loss of the class-2 because the drop of class-1 happened later than class-2. The job loss of the class-1 and the class-2 gave very very small values < 0.01 of job loss for all class-2 arrival rates. So, the difference of the traffic loss between the case with $N = 50$ when no drop and the case $N = 5$, not big. This explains the difference between the mean response times of case $N = 5$ and the mean response times of case $N = 50$ were very small.

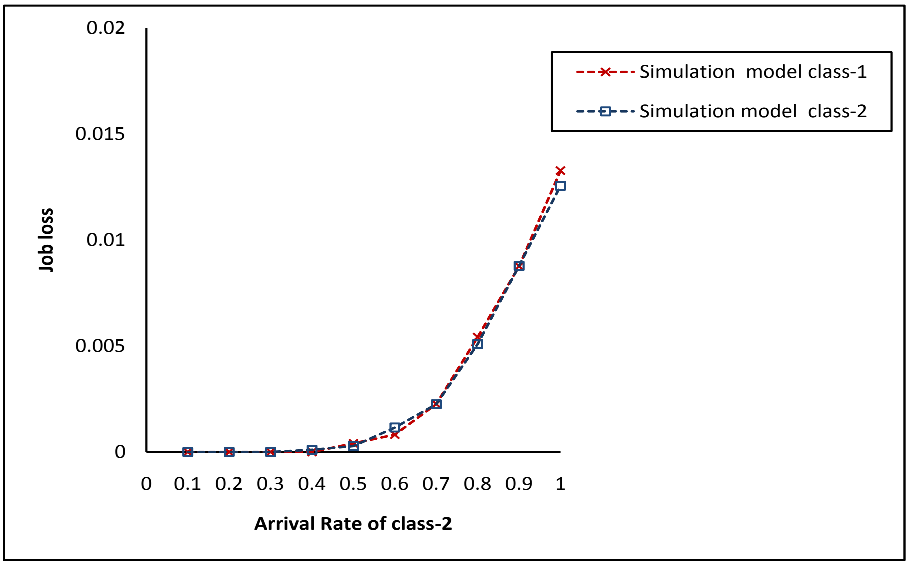

Figure 4.25: Case 4, Job Loss, $N = 5$.

4.2.5 Case 5

Case 5 shows the effect of the weights when service demands for each class are considerably different. From Figure (4.26) when arrival rates are such that the buffer is nearly empty and there are no lost customers, class-1, $\mu_1 = 1$, the simulation and analytical model gave similar behaviour.

For class-2, $\mu_2 = 10$ the analytical model and the simulation model show contradictory results. However, when the arrival rate of class-2 is increased such that the buffer occupancy customer drop rate is similar to case 4, the results of the simulation and analytical models are similar (4.27). Hence, when the system is empty the Processor Sharing approximation to the WFQ model proposed in this thesis does not correctly approximate the WFQ system. The aim of the model is Quality of Service (QoS) which is not a factor in empty systems.

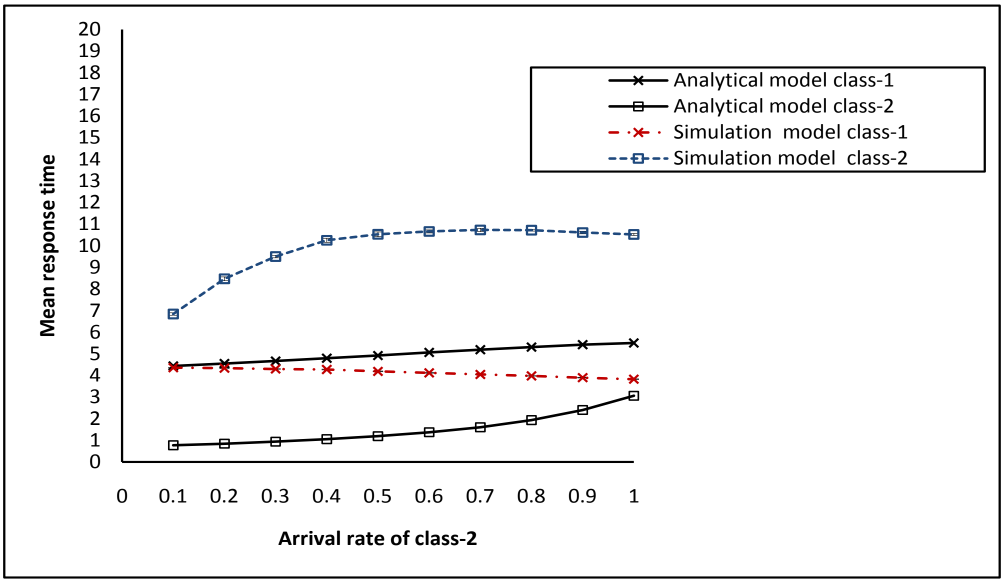

Figure 4.26: Case $5a$, Mean response time, $N = 50$.

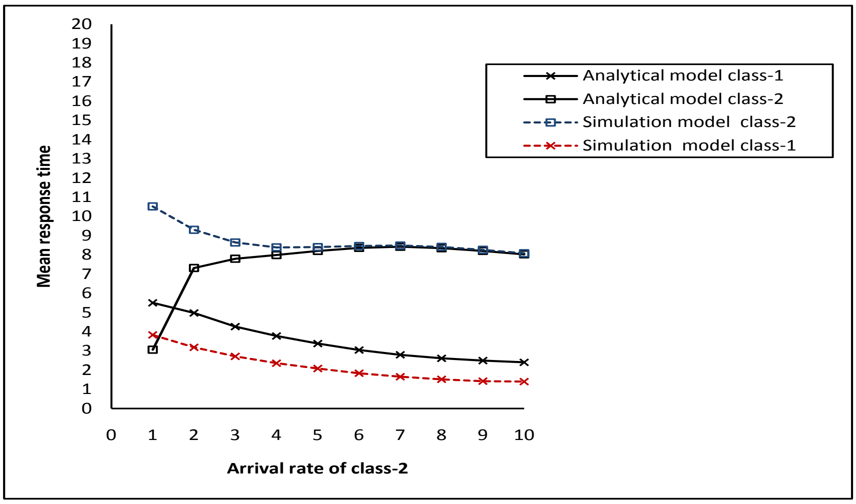

Figure 4.27: Case $5b$, Mean response time, $N = 50$.

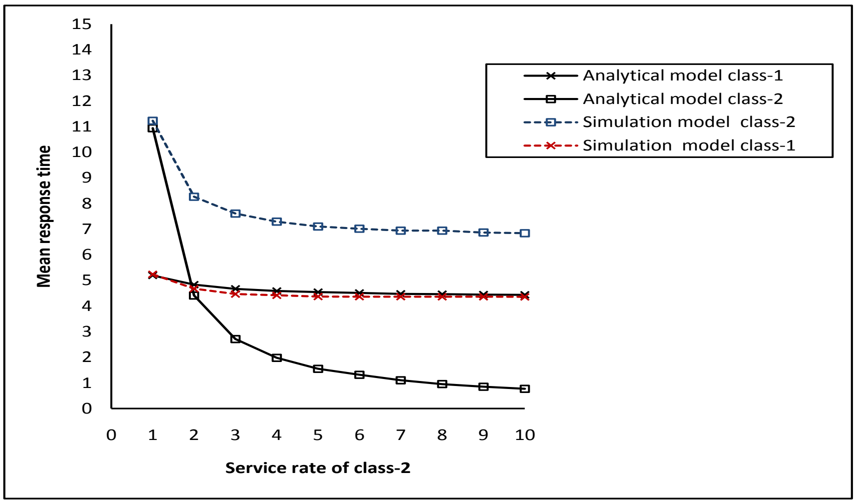

Figure 4.28: Case $5c$, Mean response time, $N = 50$.

4.3 Three-class Model

The system was studied when the service demand for both classes are equal, i.e. $\mu_1 = \mu_2 = \mu_3$, in order to examine the effect of the weight on the system. The effects of changing the assigned parameters on the mean queue length, mean response time and throughput were tested. The three traffic classes were used, which will be referred to as class-1, class-2, and class-3. The simulation model for three class as shown in Algorithm 4. Table 4.2 summarises the input parameters for cases.

Table 4.2: The parameters used in Three-class Model

Case	λ_1	μ_1	w_1	λ_2	μ_2	w_2	λ_3	μ_3	w_3
1	$[0.01, 0.1]$	0.2	0.4	$[0.01, 0.1]$	0.2	0.45	$[0.01, 0.1]$	0.2	0.15
2	0.01	0.1	0.7	0.01	0.1	0.2	$[0.1, 1]$	0.1	0.1
3	0.08	0.2	$[0, 1]$	0.01	0.2	$\frac{1-w_1}{2}$	0.04	0.2	$\frac{1-w_1}{2}$

To study the effect of changing system load on the mean queue length, mean response time and throughput, the system examined when $\lambda_1 = \lambda_2 = \lambda_3$ in case 1. The same parameters were used in [34] in case 1. Therefore, the results of the mathematical models match the simulation results in [34], the relative error went up to 2% with low load points. The results are illustrated in Figures (4.29) and (4.30).

Case 2 presents the effect of changing class-3 load with low weight and fixed arrival rate for class-1 and class-2. Class-1's allocation of the weight is much more than other classes. Graphs of the mean response time of the analytical and simulation models for a range of buffer sizes $N = 28$ are shown in Figures (4.32) and (4.33), respectively.

Algorithm 5 Three Class Model

w_1 = weight of class-1
w_2 = weight of class-2
w_3 = weight of class-3
n_1 = number of jobs in class-1
n_2 = number of jobs in class-2
n_3 = number of jobs in class-3
wran = random integer between $[1, 100]$
wran $_{i,j}$ = random integer between $[1, (w_i + w_j)]$
if wran $\leq w_1$ **then**
  if $n_1 > 0$ **then**
    serve class-1
  else
    Two Class Model (class-2, class-3)
else if wran $\leq w_1 + w_2$ **then**
  if $n_2 > 0$ **then**
    serve class-2
  else
    Two Class Model (class-3, class-1)
else if $n_3 > 0$ **then**
    serve class-3
  else
    Two Class Model (class-1, class-3)

The third case shows the effect of changing the assigned weight on the mean queue length, the mean response time and throughput. Numerical results as well as simulation results are shown in Figure (4.35). The impact of the weight of class-1, class-2 and class-3 on its own performance metrics has been analysed.

4.3.1 Case 1

In Figure (4.29), the overall mean response time of class-1, class-2 and class-3 increased with the increase of the arrival rate. Because of the consequent increase in the mean queue length given that the queue capacity is finite, with low load for $\rho \leq 0.03$ the mean response time was similar but when $(\lambda_1+\lambda_2+\lambda_3) \approx \mu$. The increase in the overall mean response time was significantly higher in the class-3. Because, the class-3 has additional delay of low weight waiting for high weight of class-1 and class-2 to be served. The mean queue length confirms this in Figure (4.30).

Figures (4.29) and (4.30), show that by increasing the arrival rate value, the mean queue length and the mean response time for both the analytical results and the simulation results increases with almost the same percentage. However, there is a slight difference between the analytical and the simulation results in class-1, class-2 and class-3. It is most visible for value $\rho \geq 1$, when the entire job becomes a shared region both streams, being homogeneous, because the mathematical model depends on serve three classes at the same time with probability w_1 for class-1, w_2 for class-2 and w_3 for class-3, but in the simulation model serve one class at one time.

Figure (4.31) shows the throughput. The throughput system increases with the increase of arriving traffic until the total arrival rate exceeds service rate; that is when $\lambda_1+\lambda_2+\lambda_3 = \mu_1 = \mu_2 = \mu_3$. From that point after dropping, the system throughput will not exceed the service rate (in this example: $\mu_1 = \mu_2 = \mu_3 = 0.2$).

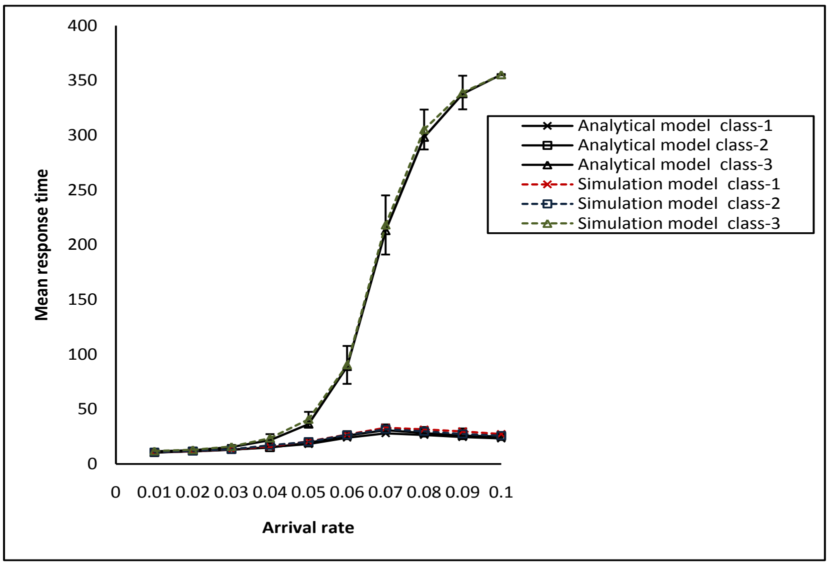

Figure 4.29: Case 1, Mean response time.

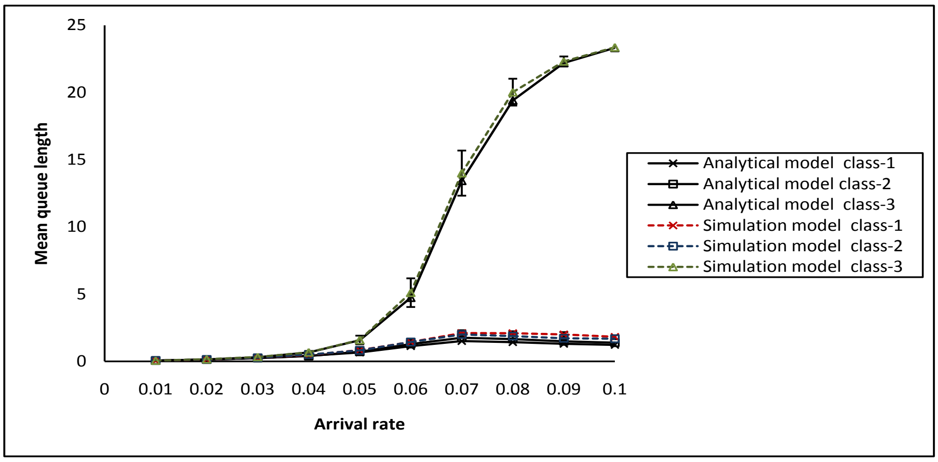

Figure 4.30: Case 1, Mean queue length.

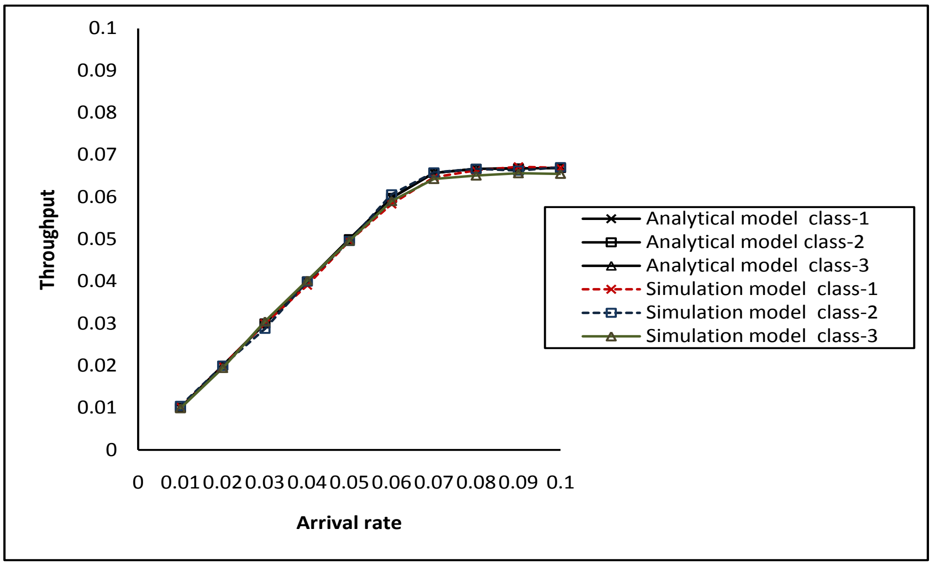

Figure 4.31: Case 1, Throughput.

4.3.2 Case 2

In the Figures (4.32) and (4.33), the significant detrimental impact of increasing the arrival rate of class-3 on the mean response time detected. In addition, the characteristic of analytical model as the load increases, the significant increase in the delay for the low weight class has been shown. The slight difference between the analytical and the simulation results is likely due to all classes sharing processes at the same time in analytical model. Nevertheless, these graphs show excellent agreement between the analytical model and the simulation, the error being less than 5% at all points even at $\lambda_2 = 0.9$. Figure (4.34) illustrates offered throughput to the buffer, where dark curves represent the analytical results and the dotted lines show simulation results.

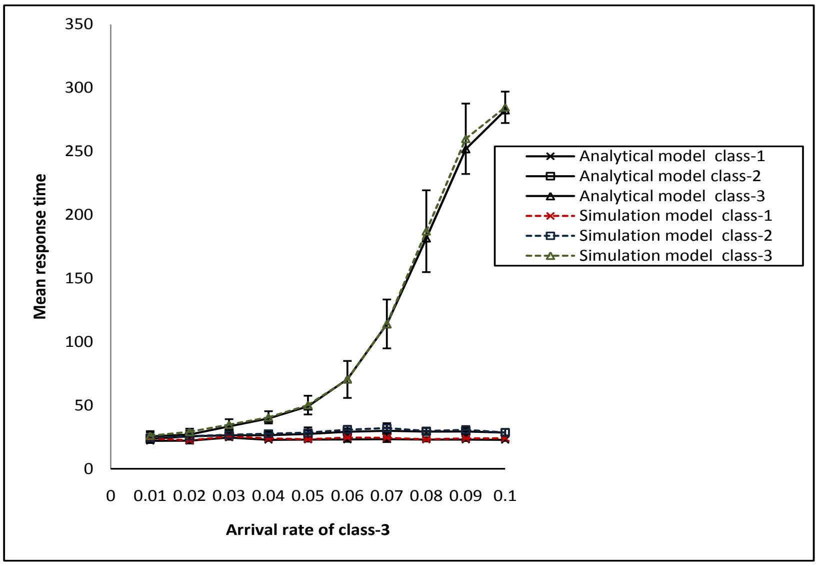

Figure 4.32: Case 2, Mean response time.

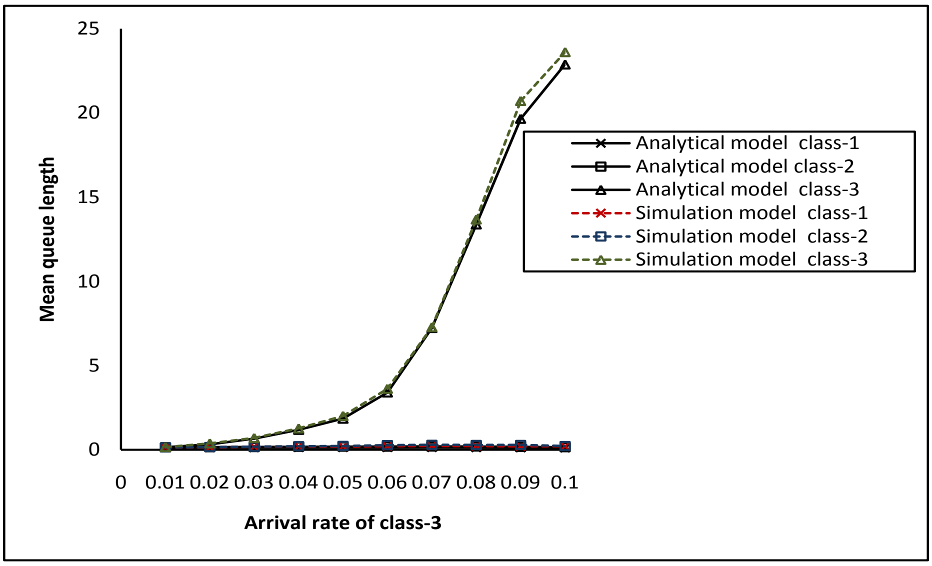

Figure 4.33: Case 2, Mean queue length.

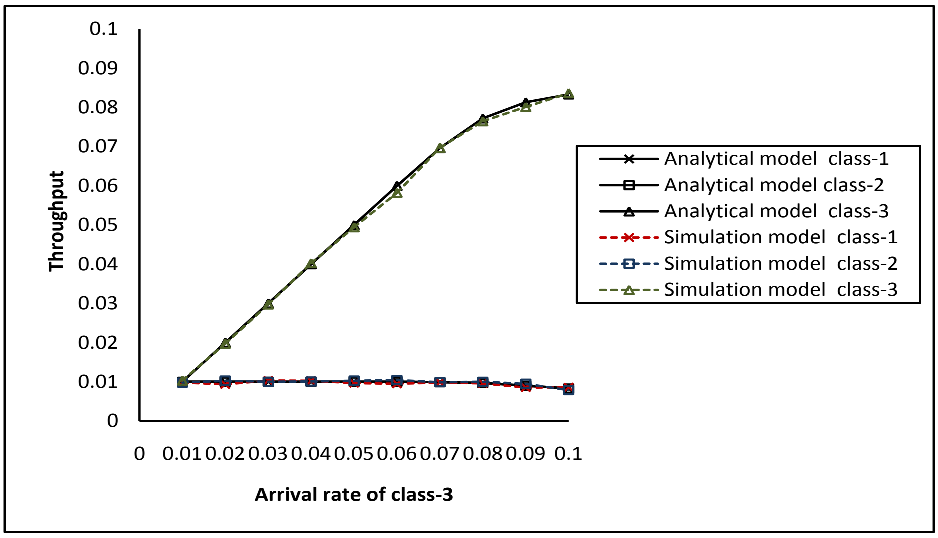

Figure 4.34: Case 2, Throughput.

4.3.3 Case 3

Figure (4.35) shows that as the weight increases the mean response time of class-1 jobs tended to decreased, whilst that of class-2 and class-3 jobs increase. This is because the mean service rate of class-1 traffic increase. On the other hand, the mean service rate of class-2 and class-3 traffic reduced. The remarkable increase in the queueing delay of class-2 and class-3 traffic is clearly shown in Figure (4.35). In Figure (4.35), the mean queue length is depicted as a function of weight. Class-1 having Weight 0 means that class-2 and class-3 have priority. The results reflect this behavior.

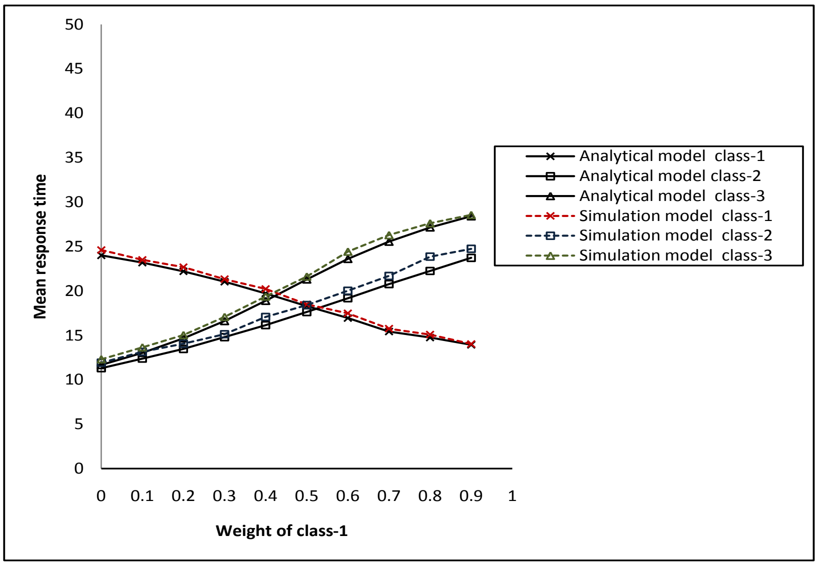

Figure 4.35: Case 3, Mean response time.

4.4 Summary

Experimental results based on the implementation of the proposed analytical model are presented and the simulation results are used to validate the proposed mathematical model. It was proved that the behaviour of the system is similar to that in the simulation model. In this case, the analytical models proposed give good results which were very close to WFQ algorithms. From the two-class model, we found that the closed form (3.13) holds which use Poisson input arrivals and exponential service time distribution.

Chapter 5

Conclusions and Future Work

5.1 Summary

The importance of a WFQ scheduler lies in the fact that it provides minimum delay and guaranteed fairness. However, the constraints of weighted fair queueing (WFQ) algorithms make it difficult to provide exact analytical models for WFQ systems. This thesis shows that analytical models of the approximation to weighted fair queueing (WFQ) based on a form of processor sharing, are a good approximation to WFQ systems. The achievements and contributions of the thesis can be summarised as follows:

- In **chapter** (1), an introduction to Weighted Fair Queueing Algorithms in queueing system was given. First a brief introduction in chapter (1), was presented section (1.1), gives the Background and Motivation of the research. In section (1.3) the Research Aims and Objectives were discussed.
- In **chapter** (2), a brief introduction to the weighted fair queueing algorithms, with a particular focus on the procedure, benefits, and limitations of the fair queueing was given. In addition, the previous WFQ system solutions were studied, and its

limitations.

- In **chapter** (3) the analytical models of Processor Sharing approximation to the WFQ model with multi-class with shared finite buffer was introduced. The Matrix-Geometric method was used to obtain the steady state probabilities for any number of classes and any number of buffers as shown in equation (3.13), thus instead of having to solve a huge number of equations to calculate the steady state probabilities. That is the steady state holds where we have Poisson input arrivals and exponential service time distribution. For example, it can simply be extended to cope with phase type arrival and service distribution for tow-class model.

- In **chapter** (4) the experimental results based on the implementation of the proposed analytical model were presented. The simulation results were used to validate the proposed mathematical model and proved that the behaviour of the system is similar to that in the simulation model.

- The main goal of the thesis was to develop models of Processor Sharing approximation to the WFQ single-server K-class with finite capacity based on the $M/M/1/N$ system. We have obtained a general solution, that calculates the steady state probability any number of classes with any buffer size. The advantage of these solutions is that it can be applied for any number of classes with any buffer size, theoretically without any limit.

- Two models were proposed. The first model was on the $M_1{+}M_2/M/1/N$ queueing system while the second was on $M_1{+}M_2{+}M_3/M/1/N$. Solutions for the model based on the $M_1{+}M_2/M/1/50$ and $M_1{+}M_2{+}M_3/M/1/28$ queueing system can by found by solving a set of equations describing the state probabilities. A method to find those solutions was presented step by step. The accuracy of both models was validated experimentally, compared and discussed as follows:

1. In Two-class model, the analytical model proposed gave good results which were very close to WFQ systems. The relative error was $\leq 1\%$ in all cases except in extreme cases goes up to 2%.
2. A novel idea of developing the analytical results of more than a two-class model was proposed and realized. The relative error went up to 5% in some cases. However, a consequence of the Java implementation for the three-class model was a limitation to small buffer sizes.
3. The aim of the model is Quality of Service (QoS) which is not a factor with low load systems. Hence, in the case of low load system, the weight has no effect on the QoS of the higher weight class, when service for the lower weight class is considerably faster as shown in case 5.

5.2 Future Work

The work presented in this thesis is an investigation into the utility of the class-based processor sharing (PS) approximation to Weighted Fair Queueing (WFQ) under the simplifying assumption of Poisson arrivals processes and exponential service times. That assumption is not realistic; network traffic is not Poisson, indeed some is long-range dependent (LRD), and the distribution of packet sizes is not exponential.

The model should be extended to accommodate more general arrival and service processes. One line of development arises from the observation that queues with arrival and service processes of any the MAP or Neut's type processes is equivalent to a $M/M/1/N$ queue in a randomly changing environment. It might be possible to solve the PS approximation model in a randomly changing environment without substancially increasing the memory requirements of the solver. This is because, in the interesting cases (e.g. high traffic burstiness), the sojourn in each phase of the environment is long relative to interarrival and service times, i.e. the rates of environmental phase change are small. So it could be practical to solve the PS approximation model for each phase separately and then apply an iterative method to take account of phase transitions.

To address LRD traffic it might be possible to develop a PS approximation model with batch renewal arrival processes. The current model could not be adapted readily because it depends upon rate matices (infinitessimal generators) whereas the batch renewal process is determined by two probabilty distributions.

A major limitation of the current solver is the restriction on the buffer sizes (50 for 2-class models and 28 for 3-class models). The current solver is a straight-forward implementation for the stationary vector of the block tri-diagonal matrix (see Q in section 3.4). It does not exploit the replication in and recursive structure of the component matrices (see $C_{k,n}$ and $A_{k,n}$ in section 3.4). It is anticipated that the solver's memory requirement might be reduced by 50% by exploiting that structure and, so, enabling it to handle larger problems.

References

[1] "Cisco systems quality of service solutions con-figuration guide (congestion manage-ment overview)." http://www.cisco.com.

[2] C. Semeria, "Supporting differentiated service classes: queue scheduling disciplines," *Juniper networks*, 2001.

[3] R. Guerin and V. Peris, "Quality-of-service in packet networks: basic mechanisms and directions," *Computer networks and ISDN systems*, vol. 31, no. 3, pp. 169–190, 1999.

[4] A. Al-Sawaai, I. Awan, and R. Fretwell, "Performance evaluation of weighted fair queuing system using matrix geometric method," *Lecture Notes in Computer Science*, vol. 5550/2009, pp. 66–78, 2009.

[5] A. Al-Sawaai, I. Awan, and R. Fretwell, "Analysis of the weighted fair queuing system with two classes of customers with finite buffer," *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 0, pp. 218–223, 2009.

[6] A. Al-Sawaai, I. U. Awan, and R. Fretwell, "Performance of weighted fair queuing system with multi-class jobs," *Advanced Information Networking and Applications, International Conference on*, vol. 0, pp. 50–57, 2010.

[7] "Cisco, low latency queueing." http://www.cisco.com.

[8] S. Keshav, *An engineering approach to computer networking: ATM networks, the Internet, and the telephone network.* 1997.

[9] G. Horváth and M. Telek, "An approximate analysis of two class wfq systems," *PMCCS-6*, pp. 43–45.

[10] D. Miller, "Computation of steady-state probabilities for m/m/1 priority queues," *Operations Research*, vol. 29, no. 5, pp. 945–958, 1981.

[11] M. Neuts, *Matrix-geometric solutions in stochastic models: an algorithmic approach.* Dover Pubns, 1994.

[12] M. Neuts, *Structured stochastic matrices of M/G/1 type and their applications*. CRC, 1989.

[13] T. Tran and T. HV. Do, "Computational aspects for steady state analysis of qbd processes," *periodica polytechnica electrical engineering*, vol. 44, no. 2, pp. 179–200, 2000.

[14] T. Tran and T. HV. Do, "Generalised invariant subspace based method for steady state analysis of qbd-m processes," *periodica polytechnica electrical engineering*, vol. 44, no. 2, pp. 159–178, 2000.

[15] J. Nagle, "On packet switches with infinite storage," *IEEE transactions on communications*, vol. 35, no. 4, pp. 435–438, 1987.

[16] L. Kleinrock, "Time-shared systems: a theoretical treatment," *Journal of the ACM (JACM)*, vol. 14, no. 2, pp. 242–261, 1967.

[17] L. Kleinrock and R. Gail, *Queueing systems*. Wiley New York, 1976.

[18] S. Yashkov, "Derivation of response time distribution for a m/g/1 processor-sharing queue," *PROB. CONTROL INFO. THEORY.*, vol. 12, no. 2, pp. 133–148, 1983.

[19] T. Ott, "The sojourn-time distribution in the m/g/1 queue with processor sharing," *Journal of applied probability*, vol. 21, no. 2, pp. 360–378, 1984.

[20] R. Schassberger, "A new approach to the m/g/1 processor-sharing queue," *Advances in applied probability*, vol. 16, no. 1, pp. 202–213, 1984.

[21] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, 1993.

[22] J. Bennett and H. Zhang, "Wf^ 2q: Worst-case fair weighted fair queueing," in *IEEE INFOCOM*, vol. 96, pp. 120–128, Citeseer, 1996.

[23] S. Golestani, "A self-clocked fair queueing scheme for broadband applications," in *Proceedings of IEEE Infocom*, vol. 94, pp. 636–646, 1994.

[24] P. Goyal, H. Vin, and H. Chen, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks," in *Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*, pp. 157–168, ACM, 1996.

[25] D. Stephens, *Implementing distributed packet fair queueing in a scalable switch architecture*. PhD thesis, 1998.

[26] C. Semeria, "Supporting differentiated service classes: queue scheduling disciplines," *Juniper networks*, 2001.

[27] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *SIGCOMM '89: Symposium proceedings on Communications architectures & protocols*, (New York, NY, USA), pp. 1–12, ACM, 1989.

[28] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose atm switch chip," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1265–1279, 1991.

[29] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," *IEEE/ACM Transactions on Networking (ToN)*, vol. 6, no. 5, pp. 611–624, 1998.

[30] F. Risso and J. Crowcroft, "Quality of service on packet switched networks," *Politechnico Di Torino, Italy, Ph. D Thesis Jan*, 2000.

[31] L. Zhang, "Virtualclock: a new traffic control algorithm for packet-switched networks," *ACM Transactions on Computer Systems (TOCS)*, vol. 9, no. 2, pp. 124–128, 1991.

[32] J. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," *Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp. 143–156, 1996.

[33] R. Khayari, "Class-based weighted fair queueing: validation and comparison by trace-driven simulation," *International Journal of Communication Systems*, vol. 18, no. 10, pp. 975–994, 2005.

[34] B. Masi, M. Denise, M. Fischer, and D. Garbin, "Modeling the performance of low latency queueing for emergency telecommunications," in *Proceedings of the 39th*

conference on Winter simulation: 40 years! The best is yet to come, pp. 2266–2275, IEEE Press, 2007.

[35] "Assurance of video conference services with combination of weighted fair queueing scheduling discipline and low latency queue,"

[36] H. Takagi, *Analysis of polling systems*. 1986.

[37] F. Guillemin, R. Mazumdar, A. Dupuis, and J. Boyer, "Analysis of the fluid weighted fair queueing system," *Journal of Applied Probability*, pp. 180–199, 2003.

[38] F. Guillemin and D. Pinchon, "Analysis of the weighted fair queueing system with two classes of customers with exponential service times," *Journal of Applied Probability*, vol. 41, pp. 832–858, 2004.

[39] G. Latouche and V. Ramaswami, *Introduction to matrix analytic methods in stochastic modeling*. Society for Industrial Mathematics, 1999.

[40] D. Cross and C. Harris, "Fundamentals of queuing theory," *John Wily and Sons*, 1998.

[41] M. Fischer, D. Masi, and J. Shortle, "Simulating the performance of a class-based weighted fair queueing system," in *Proceedings of the 40th Conference on Winter Simulation*, pp. 2901–2908, Winter Simulation Conference, 2008.

[42] J. Blanc, "A numerical study of a coupled processor model," in *Computer Performance and Reliability: Proceedings of the Second International McPr Workshop Held in Rome, Italy, May 25-29, 1987*, p. 289, North-Holland, 1988.

[43] L. Servi, "Algorithmic solutions to two-dimensional birth–death processes with application to capacity planning," *Telecommunication Systems*, vol. 21, no. 2, pp. 205–212, 2002.

[44] M. Neuts, “The probabilistic significance of the rate matrix in matrix-geometric invariant vectors,” *Journal of Applied Probability*, pp. 291–296, 1980.

[45] M. Neuts, “Markov chains with applications in queueing theory, which have a matrix-geometric invariant probability vector,” *Advances in Applied Probability*, vol. 10, no. 1, pp. 185–212, 1978.

[46] W. Grassmann and D. Heyman, “Equilibrium distribution of block-structured markov chains with repeating rows,” *Journal of applied probability*, vol. 27, no. 3, pp. 557–576, 1990.

[47] J. Daigle, *Queueing theory for telecommunications*. Addison-Wesley, 1992.

[48] D. Gross, *Fundamentals of queueing theory*. Wiley India Pvt. Ltd., 2008.

[49] J. Dobrowolski, *Introduction to computer methods for microwave circuit analysis and design*. Artech House on Demand, 1991.

[50] D. Comer, *Computer networks and internets*. Pearson Prentice Hall, 2009.

[51] D. Miller, “Computation of steady-state probabilities for M/M/1 priority queues,” *Operations Research*, vol. 29, no. 5, pp. 945–958, 1981.

Appendices

Appendix A

Two Class Model

A.1 Analytical Model Description

The PS approximation to WFQ system is a single finite capacity FIFO queue with a non-preemptive service center, hence, the maximum number of jobs in the system at any time will not exceed K and any additional arriving customers will be refused entry to the system and will depart immediately without service.

For our PS approximation to WFQ model, we assume two classes of jobs; jobs of class-1 and of class-2. Jobs arrive according to a Poisson process with rate λ_i, $i = 1, 2$ and have exponential service rates with mean $\frac{1}{\mu_i}$, $i = 1, 2$. The buffer K is shared finite buffer divided into two classes. Each class is served in FIFO order. The server is work conserving, i.e., it serves jobs if at least one class is not empty. Figure A.1 depicts the PS approximation to WFQ system with two classes.

Let $w_i > 0$, $i = 1, 2$, be the weight assigned to class-i and $w_1 + w_2 = 1$. From the Equation 3.1, we have that class-i will be served at rate $\hat{w}_i = w_i$, when other class is not empty and at rate unity when other class is empty. When both classes are present, class-1 jobs are selected to be serviced with probability w_1 and class-2 jobs are serviced with probability w_2. At boundaries, when $w_1 = 0$ or $w_1 = 1$, the PS approximation to WFQ system is reduced to the $M/M/1/K$ non-preemptive system, which has been extensively studied in the past [51].

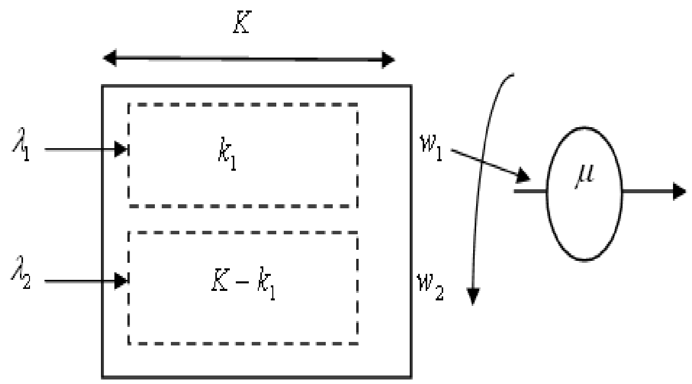

Figure A.1: The PS approximation to WFQ model

Figure A.2, illustrates the state-transition-rate diagram of the PS approximation to WFQ system where each state denotes the number of jobs in the system. A generalized Markov model can described by a two dimensional Markov chain with state $(n_1,\, n_2)$, where n_1 and n_2 are the number of jobs in class-1 and class-2 at each state, respectively.

For the case at the boundary states the transitions do not depend on weight. When the process is in state(n_1, n_2), it can transfer to one of the states $(n_1 + 1,\, n_2)$, $(n_1,\, n_2 + 1)$, $(n_1 - 1,\, n_2)$ and $(n_1,\, n_2 - 1)$.

The transition rate from state (n_1, n_2) to $(n_1 + 1, n_2)$ where $(0 \leq n_1 \leq K - 1)$ is the arrival rate of class-1, i.e. λ_1 of the Poisson process. A transition out of state (n_1, n_2) to $(n_1, n_2 + 1)$ where $(0 \leq n_2 \leq K - 1)$ is the arrival rate of class-2, i.e.λ_2, of the Poisson process. When no customers of class-1 are in the system, the transition rate from $(0, n_2)$ to $(0, n_2 - 1)$ is the service rates of class-2, i.e.μ_2. The change from state $(n_1, 0)$ to $(n_1 - 1, 0)$ is the service rate of class-1, i.e.μ_1.

However, the transition rate from state (n_1, n_2) to $(n_1 - 1, n_2)$ is the service rate of class-1 multiplied by the weight of class-1, because class-1 jobs are selected to be serviced with probability w_1, i.e.$w_1\mu_1$. A transition rate from (n_1, n_2) to $(n_1, n_2 - 1)$ is the service rate of class-2 multiplied by the the weight of class-2, for the reason that class-2 jobs are serviced with probability w_2, i.e.$w_2\mu_2$.

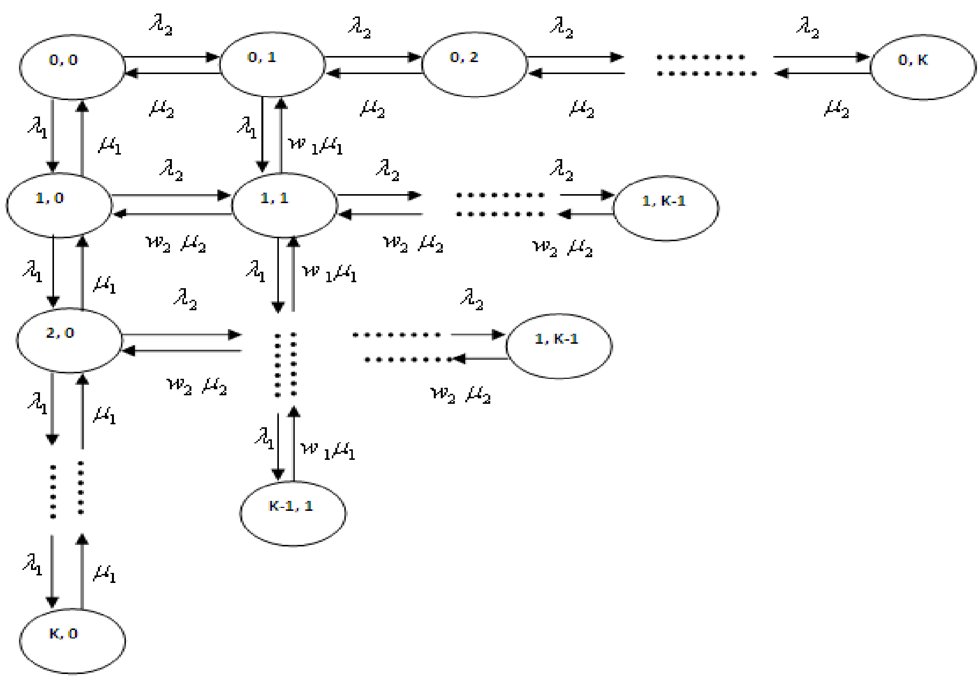

Figure A.2: The State Transition Diagram at State (n_1, n_2).

The infinitesimal generator of this process is given by:

$Q_{(n_1,0),(n_1-1,0)} = \mu_1,\ n_1 \geq 1$

$Q_{(0,n_2),(0,n_2-1)} = \mu_2,\ n_2 \geq 1$

$Q_{(n_1,n_2),(n_1-1,n_2)} = w_1\mu_1$

$Q_{(n_1,n_2),(n_1,n_2-1)} = w_2\mu_2,\ n_1 \geq 1,\, n_2 \geq 1$

$Q_{(n_1,n_2),(n_1+1,n_2)} = \lambda_1,\ n_1 \geq 0,\, n_2 \geq 0$

and

$Q_{(n_1,n_2),(n_1,n_2+1)} = \lambda_2,\ n_1 \geq 0,\, n_2 \geq 0.$

A.2 Steady State Probability Calculation of Two-Class Model

The steady state birth and death equations of this system for the shared finite buffer $(K_1 + K_2 \leq K)$ case are :

$$(\lambda_1+\lambda_2)\pi_{0,0} = \mu_1\pi_{1,0} + \mu_2\pi_{0,1} \tag{A.1}$$

$$(\lambda_1+\lambda_2+\mu_2)\pi_{0,1} = \mu_2\pi_{0,2} + w_1\mu_1\pi_{1,1} + \lambda_2\pi_{0,0} \tag{A.2}$$

$$(\lambda_1+\lambda_2+\mu_1)\pi_{1,0} = \mu_1\pi_{2,0} + w_2\mu_2\pi_{1,1} + \lambda_1\pi_{0,0} \tag{A.3}$$

and for $n_1 = 1,\, 2,\, \ldots,\, K$ and $n_2 = 1,\, 2,\, \ldots,\, K$,

$$(\lambda_1+\lambda_2+w_1\mu_1+w_2\mu_2)\pi_{n_1,n_2} = \lambda_1\pi_{n_1-1,n_2}+\lambda_2\pi_{n_1,n_2-1}+w_1\mu_1\pi_{n_1+1,n_2}+w_2\mu_2\pi_{n_1,n_2+1}$$

(A.4)

When $n_1 = 1$ and $n_2 \geq 1$ (the first expression in the right hand side of equation (A.4)), we have $\pi_{0,n_2} = 0$, and, similarly in equation (A.4), if $n_2 = 1$ and $n_1 \geq 1$ (the second expression on the right hand side of equation (A.4)), we have $\pi_{n_1,0} = 0$.

For class-2,

$$(\lambda_1 + \lambda_2 + \mu_2)\pi_{0,n_2} = \mu_2\pi_{0,\,n_2+1} + w_1\mu_1\pi_{1,n_2} + \lambda_2\pi_{0,n_2-1} \tag{A.5}$$

and

$\pi_{0,\,n_2} = 0.$

For class-1,

$$(\lambda_1 + \lambda_2 + \mu_1)\pi_{n_1,0} = \mu_1\pi_{n_1+1,0} + w_2\mu_2\pi_{n_1,1} + \lambda_1\pi_{n_1-1,0} \tag{A.6}$$

and

$\pi_{n_1,\,0} = 0.$

Appendix B

Three Class Model

B.1 Analytical Model Description

We consider the same model in Section 3.3; however instead of three classes, where an analytical solution is provided by using Matrix-Geometric method; where use a finite buffer is shared with size $K_1 + K_2 + K_3 = K$. The maximum buffer is number of jobs which can be in the system at any time is K and any additional without service. We drive a general single-server system $M_1+M_2+M_3/M/1/K$ with the following characteristics:

- The service discipline is PS approximation to WFQ, where a weight is assigned to each class. The ratio of server capacity available for a class is given by the ratio of the weights of the classes that are active. An absolute service rate can be simply achieved by assigning a fixed weight. So the importance of the jobs is regulated by the weight assigned to their class.
- The capacity $K > 0$ of the PS approximation to WFQ system is finite and comprises one server while $K - 1 > 0$ is waiting places.

- The work of the server is conserving, i.e., it serves jobs only if at least one class is not empty.
- The class-i is served at rate w_i for some $w_i > 0$ when other two classes are not empty and at rate unity when that classes are empty. The coefficients w_i are such that: $w_1 + w_2 + w_3 = 1$.
- If one of three classes is empty when class-i is served, the head job of class-i will be served with probability $\hat{w}_i$ is determined from Equation B.1as:

$$\hat{w}_{i \in BC} = \frac{w_i}{\sum_{j \in BC} w_j} \tag{B.1}$$

  The coefficients w_i are such that: $\hat{w}_1 + \hat{w}_2 + \hat{w}_3 = 1$.

- The arrival streams of the jobs of each class-i, $i = 1,\ 2,\ 3$ are assumed to be independent Poisson process with rates $\lambda_i > 0,\ i = 1, 2, 3$.
- The service times of jobs are independent exponentially distributed with mean service time $\frac{1}{\mu_i} > 0,\ i = 1, 2, 3$.

Figure B.1 depicts the PS approximation to WFQ queueing system with three classes and Figure B.2, illustrates state transition diagram of the PS approximation to WFQ system where each state denotes the number of jobs in the system. A generalized Markov model can be described by three dimensional Markov chain with state $(n_1,\ n_2,\ n_3)$, where $n_1,\ n_2$ and n_3 are the number of jobs in class-1, class-2 and class-3 at each state, respectively. When the process is in state $(n_1,\ n_2,\ n_3)$, it can be transferred to one the states $(n_1,\ n_2,\ n_3 + 1)$, $(n_1,\ n_2 + 1,\ n_3)$, $(n_1 + 1,\ n_2,\ n_3)$, $(n_1 - 1,\ n_2,\ n_3)$, $(n_1,\ n_2 - 1,\ n_3)$ and $(n_1,\ n_2,\ n_3 - 1)$ as in Figure B.2. The total number of each state does not exceed the buffer K i.e. $(n_1 + n_2 + n_3) \leq K$.

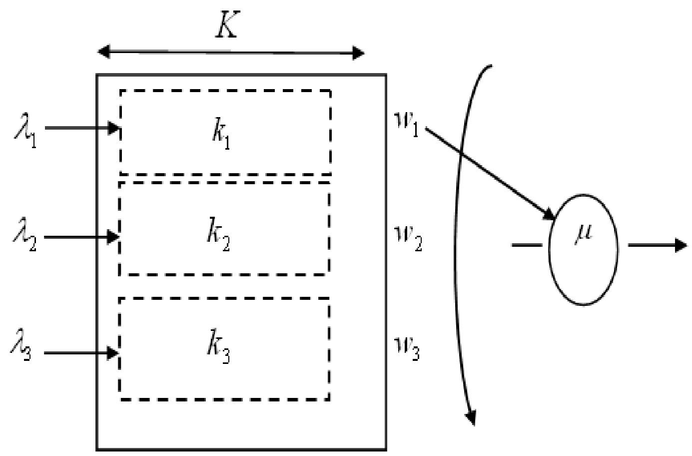

Figure B.1: The PS approximate to WFQ model with Three Classes of Traffic.

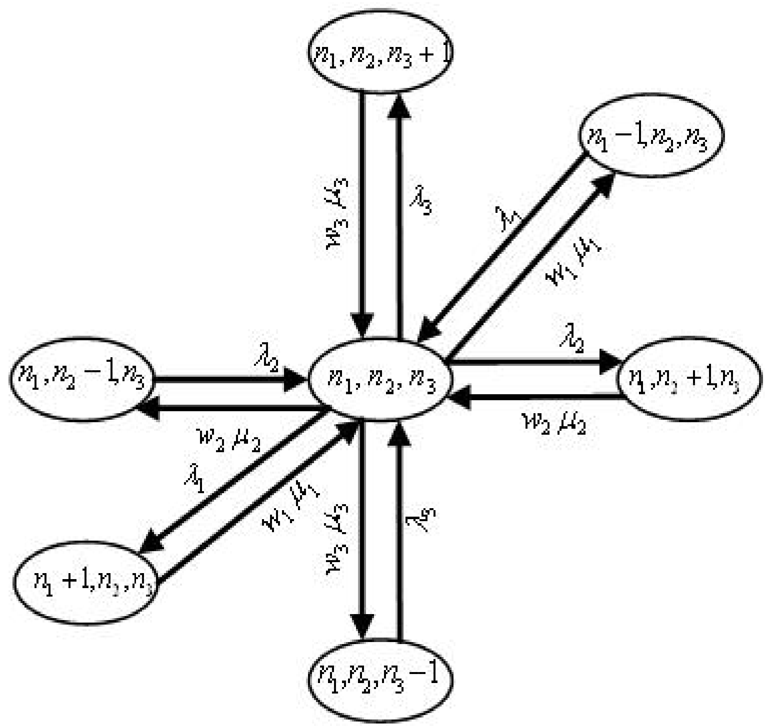

Figure B.2: The State Transition Diagram at State $(n_1,\, n_2,\, n_3)$.

B.2 Steady State Probability Calculation for Three-Class Model

To calculate $\pi_(n_1,\, n_2,\, n_3)$ the probability at state $(n_1,\, n_2,\, n_3)$, we need to solve the three dimensional Markov chain shown in Figure B.3.

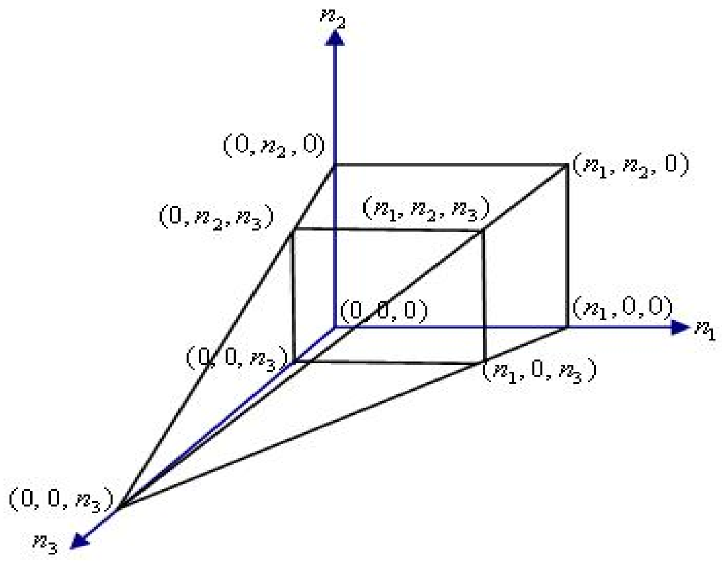

Figure B.3: A three-dimensional Markov chain Shaped by states $n_1,\, n_2$ and n_3.

We can do that by setting up a set of global balance equations by equating the rates of flow out of and into each state Figure B.2 and solve these balance equations by the Matrix-Geometric method. The steady states of birth and death equations of this system for shared finite buffer $(K_1 + K_2 + K_3 = K)$ case are:

$$(\lambda_1 + \lambda_2 + \lambda_3)\pi_{(0,\,0,\,0)} = \mu_3\pi_{(0,\,0,\,1)} + \mu_2\pi_{(0,\,1,\,0)} + \mu_1\pi_{(1,\,0,\,0)} \tag{B.2}$$

$$(\lambda_1+\lambda_2+\lambda_3+\mu_3)\pi_{(0,0,1)} = \hat{w}_2\mu_2\pi_{(0,1,1)}+\mu_3\pi_{(0,0,2)}+\lambda_3\pi_{(0,0,0)}+\hat{w}_1\mu_1\pi_{(1,0,1)} \quad \text{(B.3)}$$

$$(\lambda_1+\lambda_2+\lambda_3+\mu_2)\pi_{(0,1,0)} = \hat{w}_3\mu_3\pi_{(0,1,1)}+\mu_2\pi_{(0,2,0)}+\lambda_2\pi_{(0,0,0)}+\hat{w}_1\mu_1\pi_{(1,1,0)} \quad \text{(B.4)}$$

$$(\lambda_1+\lambda_2+\lambda_3+\mu_1)\pi_{(1,0,0)} = \hat{w}_2\mu_2\pi_{(1,1,0)}+\mu_1\pi_{(2,0,0)}+\lambda_1\pi_{(0,0,0)}+\hat{w}_3\mu_3\pi_{(1,0,1)} \quad \text{(B.5)}$$

and for $n_1 = 1, 2, \ldots, K, n_2 = 1, 2, \ldots, K$ and $n_3 = 0$

$$\begin{aligned}(\lambda_1 + \lambda_2 + \lambda_3 + \hat{w}_1\mu_1 + \hat{w}_2\mu_2)\pi_{(n_1,n_2,0)} &= \lambda_1\pi_{(0,n_2,0)} + \hat{w}_2\mu_2\pi_{(n_1,n_2+1,0)} \\ &\quad +\hat{w}_3\mu_3\pi_{(n_1,n_2,1)} + \hat{w}_1\mu_1\pi_{(n_1+1,n_2,0)} \\ &\quad +\lambda_2\pi_{(n_1,0,0)} \qquad \text{(B.6)}\end{aligned}$$

for $n_1 = 1, 2, \ldots, K, n_2 = 0$ and $n_3 = 1, 2, \ldots, K$

$$\begin{aligned}(\lambda_1 + \lambda_2 + \lambda_3 + \hat{w}_1\mu_1 + \hat{w}_3\mu_3)\pi_{(n_1,0,n_3)} &= \lambda_1\pi_{(n_1-1,0,n_3)} + \hat{w}_3\mu_3\pi_{(n_1,0,n_3+1)} \\ &\quad +\hat{w}_2\mu_2\pi_{(n_1,1,n_3)} + \hat{w}_1\mu_1\pi_{(n_1+1,0,n_3)} \\ &\quad +\lambda_3\pi_{(n_1,0,0)} \qquad \text{(B.7)}\end{aligned}$$

and for $n_1 = 0, n_2 = 1,\ 2,\ \ldots,\ K$ and $n_3 = 1,\ 2,\ \ldots,\ K$

$$\begin{aligned}(\lambda_1 + \lambda_2 + \lambda_3 + \hat{w}_2\mu_2 + \hat{w}_3\mu_3)\pi_{(0,\,n_2,\,n_3)} &= \lambda_2\pi_{(0,\,n_2-1,\,n_3)} + \hat{w}_3\mu_3\pi_{(0,\,n_2,\,n_3+1)} \\ &\quad + \hat{w}_1\mu_1\pi_{(0,\,n_2+1,\,n_3)} + \hat{w}_1\mu_1\pi_{(1,\,0,\,n_3)} \\ &\quad + \lambda_3\pi_{(0,\,n_2,\,n_3-1)} \qquad \text{(B.8)}\end{aligned}$$

From Figure B.2, when $n_1 = 1,\ 2,\ \ldots, K$, $n_2 = 1,\ 2,\ \ldots, K$ and $n_3 = 1,\ 2,\ \ldots, K$, we have $\pi_{(n_1,n_2,n_3)}$ as

$$\begin{aligned}(\lambda_1 + \lambda_2 + \lambda_3 + w_1\mu_1 + w_2\mu_2 + w_3\mu_3)\pi_{(n_1,\,n_2,\,n_3)} &= \lambda_1\pi_{(n_1-1,\,n_2,\,n_3)} + w_3\mu_3\pi_{(n_1,\,n_2,\,n_3+1)} \\ &\quad + w_2\mu_2\pi_{(n_1,\,n_2+1,\,n_3)} + w_1\mu_1\pi_{(n_1+1,\,n_2,\,n_3)} \\ &\quad + \lambda_3\pi_{(n_1,\,n_2,\,n_3-1)} \\ &\quad + \lambda_2\pi_{(n_1,\,n_2-1,\,n_3)} \qquad \text{(B.9)}\end{aligned}$$