MINIMUM VARIANCE INTER-DEPARTURE PROCESS OF M/G/1 QUEUES WITH OPTIMAL STATE DEPENDENT SERVICE RATES

A Project Report

submitted by

KUMAR SUMAN
(EE09B019)

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS

JUNE 2014

PROJECT CERTIFICATE

This is to certify that the project titled **MINIMUM VARIANCE INTER-DEPARTURE PROCESS OF M/G/1 QUEUES WITH OPTIMAL STATE DEPENDENT SERVICE RATES**, submitted by **KUMAR SUMAN**, to the Indian Institute of Technology, Madras, for the award of the degree of **BACHELOR OF TECHNOLOGY**, is a bona fide record of the project work done by him under our supervision. The contents of this project, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. R.MANIVASAKAN
Assistant Professor
Project Guide
Dept. of ELECTRICAL ENGINEERING
IIT-Madras, 600 036

Place: Chennai
Date: 24th June, 2014

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who helped me in one way or the other with regard to the project. I would especially like to extend my appreciation towards the following.

I thank my guide Dr. R. Manivasakan, for his support and guidance during the course of the project.

I would like to thank all my friends who supported me throughout the project.

Last but not the least I would like to thank my family for their immense support and motivation during the four years and before that.

ABSTRACT

In a TDM (Time-Division Multiplexing) over PSN (Packet Switched Network), the inter-departure process of the jitter-buffer at the receiver should have minimum variance to comply to the jitter performance as specified in the IEEE 1588 TDM Standard. Thus minimizing the variance of the inter-departure process of a suitable modeling queue is of paramount importance. This project proposes two queuing-models which can help us in reducing the variance of the inter-departure process. In addition, it also proposes a queuing model to reduce the buffer-size.

In the first model, we consider a modification of the standard M/G/1 queue (queues with markovian arrival process, general service time distribution and single server) with unlimited waiting space and FIFO-discipline in which the service times depend linearly and randomly on the waiting times. In this model the waiting times satisfy a modified version of the classical Lindley's recursion. We determine when the waiting times distribution converge to a proper limit, and we develop approximations for this steady state limit, then based on these approximations we try to schedule the successive services in order to reduce the variance of the inter-departure process.

In the second model we consider an offline algorithm where service times depend linearly on queue-length (number of customers in the queue). A mathematical programming representation for the sample path dynamics of a state dependent queue is presented. Also, some simulation results have been presented.

In third model, we are trying to implement the Mansour's online rate-jitter control algorithm and simulate the algorithm.

In additional literature survey, we are trying to implement the known BRAVO-effect (Balancing Reduces Asymptotic Variance of Outputs) for correlated queuing model.

KEYWORDS: State Dependent M/G/1 Queue; Variance of the Inter Departure Process; TDM over PSN; Lindley's Equation; BRAVO Effect.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

FCFS	First Come First Serve
SDQ	State Dependent Queues
MPR	Mathematical Programming Representation

NOTATION

W_n	Waiting Time of customer n
S_n	Service Time of customer n
X_n	Inter-arrival Time between customer n and $n+1$
Var	Variance

CHAPTER 1

INTRODUCTION TO TDM over PSN

Time-division multiplexing (TDM) is a method of transmitting and receiving independent signals over a common signal path by means of synchronized switches at each end of the transmission line so that each signal appears on the line only a fraction of time. TDM circuits have been the backbone of communications over the past several decades. It is a hard partitioned circuit switched technology and provides reliable and low delay services for real time interactive digital telephony as well as data and video transport. However, these circuits are migrating towards Internet Protocol (IP) based Packet Switched Networks (PSN) (9.Keyur and Junius, 2007). It is so because bandwidth is used inefficiently in TDM. For efficient bandwidth utilization and hence to reduce the cost of transport and management, there has evolved a packet based converged network for all services. In such a network, digitized signals are carried over packet switched network. Due to the sheer magnitude of the installed legacy TDM equipment, this migration to end-to-end IP will go through a transitional phase where some services will continue to use legacy equipment, while the core network moves towards PSN. In this transitional phase, there is a need for technology allowing seamless transmission of TDM services across the packet switch networks.

TDM over PSN is a technology for emulating TDM circuits over packet switched networks. In this technology a logical circuit is realized in a PSN, which links two TDM islands. The TDM traffic at the transmitter is packetized into constant bit sized frames and transmitted across a PSN. When the packet carrying the TDM payload traverses the PSN, it experiences random delay due to the queuing at the intermediate routers. Because of this reason, the packets at the receiver arrive randomly. The received packets are said to possess jitter or packet delay variation. Since outgoing stream is TDM stream, which should comply to a minimum jitter, the packet delay variation of incoming TDM packet stream has to be minimized. Hence, to meet this purpose, a buffer (called jitter buffer) of suitable size is used. A mismatch between the read and the write clocks at the input and the output of this buffer, due to large delay variation will cause overflow or underflow of the jitter buffer. Such clock mismatch can lead to observable defects on the end service. Synchronization in the data link layer of the ISO stack is therefore an important issue in such networks (8.R Manivasakan and Usharani, 2012).

For achieving synchronization at the data link layer of the ISO stack, in this project, we use a queuing model, where frames arriving from the transmitter are queued in the jitter buffer and served such that the variance of the inter-departure process of the outgoing frame stream is minimum. Minimizing variance of the inter-packet time at either of the two layers, physical or data link of the ISO stack will reduce jitter in the outgoing stream.

CHAPTER 2

BACKGROUND MATERIAL

The engineering problem addressed in this project can be well studied using mathematical model of queuing theory. We review basic and necessary information about queuing systems. We specially focus on two types of systems: one having general arrival process, general service time distribution and are called *GI/GI/1*; another having markovian arrival process, general service time distribution and are called *M/G/1*.

2.1 Queuing Theory

The word *queue* comes, via French, from the Latin *cauda*, meaning tail.

Queuing theory is the mathematical study of waiting-lines or queues. In queuing theory a model is constructed so that *queue-lengths(Q)* and *waiting-times(W)* can be predicted.

A *queuing system* can be described as customers *arriving* for service, *waiting* for service if it is not immediate, and if having waited for *service*, leaving *(departure)* the system after being served.

The main characteristics of a queuing system are *arrival process*, *service process*, and *number of servers*.

Now, we will see some basic terms encountered in queuing theory.

Inter-arrival time(X_n) is the time between arrival of customers n and $n + 1$ to a queuing system.

Waiting time(W_n) is the amount of time spent by customer n waiting in the queue before the start of it's service or before entering into the server for service.

Service time(S_n) is the amount of time spent by customer n in the server while getting service or the difference of time between initiation of it's service and and it's departure from the system.

Figure 2.1: We encounter queue on daily basis.

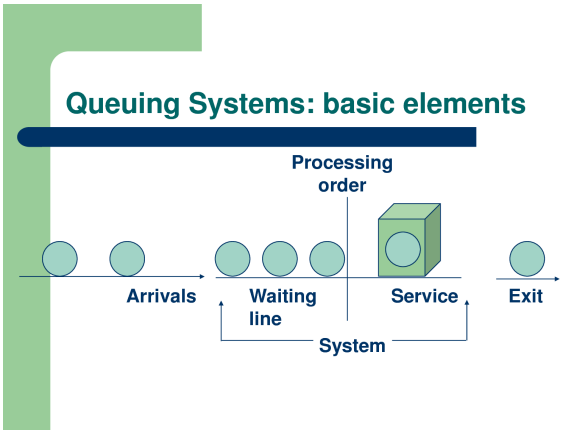

Figure 2.2: Basic elements of a Queuing system are arrivals, waiting-line, server and departures.

Now, we will see some basic relations in queuing theory, which are among *inter-arrival times(X_n)*, *service times(S_n)* and *waiting times(W_n)* of customer n.

Little's Law The long-term average number of customers in a stable system Q is equal to the long-term average effective arrival rate, λ, multiplied by the average time a customer spends in the system W; or expressed algebraically:

$$Q = \lambda \times W \tag{2.1}$$

2.2 M/G/1 Queue

This section provides a simple overview of $M/G/1$ queue and its various properties. An $M/G/1$ is a queue model where arrivals are Markovian (modulated by a Poisson process), service times have a general distribution and there is a single server.

Model Definition: A queue represented by a *M/G/1* queue is a stochastic process whose state space is the set $\{0, 1, 2, 3...\}$, where the value corresponds to the number of customers in the queue, including any being served. Transitions from state i to $i + 1$ represent the arrival of a new customer: the times between such arrivals have an exponential distribution with parameter λ. Transitions from state i to $i - 1$ represent a customer who has been served, finishing being served and departing: the length of time required for serving an individual customer has a general distribution function. The lengths of times between arrivals and of service periods are random variables which are assumed to be statistically independent.

Mean Queue Length : The Pollaczek-Khinchine formula states a relationship between the queue length and service time distribution Laplace transforms for an $M/G/1$ queue. The term is also used to refer to the relationships between the mean queue length and mean waiting/service time in such a model.

The formula states that *mean queue length* is given by:

$$L = \rho + \frac{\rho^2 + \lambda^2 \times Var(S)}{2(1-\rho)} \tag{2.2}$$

where λ is the arrival rate of the process, $(1/\mu)$ is the mean of the service time distribution S, ρ=(λ/μ) is the utilization, *Var(S)* is the variance of the service time distribution S.

Mean Waiting Time: Using *Little's Formula* we can calculate W:

$$W = \frac{\rho + \lambda\mu \times Var(S)}{2(\mu - \lambda)} \tag{2.3}$$

2.3 GI/GI/1 Queue

This section provides a simple overview of standard (uncorrelated) $GI/GI/1$ queue and its various properties. The notation $GI/GI/1$ queue is usually referred to a single server queue with first-in-first-out discipline and with a general distribution of the sequences of inter-arrivals and service-times (which are the "driving sequences" of the system). In studying the single server queue $GI/GI/1$, it is usually assumed that all inter-arrival times and service requirements are independent.

We will see some relations for $G/G/1$ queue, which will be helpful in further analysis of our queuing model.

Lindley's Recursion Let W_n= Waiting time of customer n;

W_{n+1}= Waiting time of customer $n+1$;

S_n= Service time of customer n; and,

X_n=Inter-arrival time between customer n and $n+1$.

Then, Lindley's Recursion tells us that:

$$W_{n+1} = [W_n + S_n - X_n]^+ \tag{2.4}$$

Average Waiting Time Kingman's formula gives an approximation for the mean waiting time in a $GI/GI/1$. The formula is the product of three terms which depend on utilization, variability and service time. It was first published by John Kingman. It is known to be generally very accurate, especially for a system operating close to saturation.

Kingman's approximation states

$$E[W] = \frac{\rho}{\rho - 1} \times \frac{C_a^2 + C_s^2}{2} \times \frac{1}{\mu} \tag{2.5}$$

Where,

μ= Mean service rate

λ = Mean arrival rate

ρ = λ/μ = utilization or normal traffic intensity

C_a = Coefficient of variation for arrivals (that is the standard deviation of arrival times divided by the mean arrival time)

C_s = Coefficient of variation for service times (that is the standard deviation of service times divided by the mean service time)

2.4 State dependent GI/G/1 Queue

This section provides a simple overview of a class of queues in which *service rate* or *arrival rate* or both depends upon the *state* of the queue,i.e., number of customers in the queue or may be waiting time of the customer in the queue.

However much analytic results are not established for this class of queues, we will see how server behaves based on the state of the system. Lindley's Recursion which is stated in the previous paragraph are still valid.

CHAPTER 3

M/G/1 QUEUES WITH SERVICE TIMES DEPENDING LINEARLY AND RANDOMLY UPON WAITING TIMES

The work of this chapter has been inspired from the paper (Ward WHITT, 1990). In this chapter, we will consider an extension of the standard $M/G/1$ queue with unlimited waiting space and FCFS discipline in which the service rates depend linearly and randomly on the waiting times. In this model the waiting times satisfy a modified version of the classical Lindley recursion. We determine when the waiting-time distributions converge to a proper limit. Then, we develop approximations for this steady state limit primarily by applying previous results for the unrestricted recursion of type $Y_{n+1} = C_n \times Y_n + R_n$ where Y_n, C_n and R_n are random variables. We'll consider a normal approximation for the stationary waiting time distribution in the case when queue rarely becomes empty. We also consider the problem of scheduling successive service-times, with the objective of achieving nearly optimal throughput with nearly bounded waiting times, while making the service-time sequence relatively smooth. We identifies policies depending linearly and deterministically upon the work in the system which meet these objectives very well; with these policies the waiting times are approximately contained in a specified interval a specified fraction of time. We will use this scheduling of service times to obtain the expression for inter-departure time and will try to minimize that which is our objective.

3.1 Introduction

We consider a modification of standard $G/G/1$ queue with unlimited waiting space and FCFS discipline, in which the service times depends linearly and randomly upon the waiting times. We study the sequence $\{W_n : n \geq 0\}$, which is defined recursively by

$$W_{n+1} = [W_n + \bar{S}_n - X_n]^+, \qquad n \geq 0; \tag{3.1}$$

where $[x]^+ = max\{x, 0\}$;

$$\bar{S}_n = S_n + G_n \times W_n, \tag{3.2}$$

Here, equation (3.2) is one of the many possible linear or random dependency relation between service times and waiting times. Also, in the above equation, we interpret W_n as the waiting-time, G_n as some random variable and $\bar{S}_n$ as the service-time of the n^{th} customer. We call X_n as the *inter-arrival time* between customers n and $n+1$. We call S_n as the *nominal service time* of customer n; because this would be the actual service-time if the state dependent behavior were omitted i.e. if G_n=0 with probability 1. We assume that $0 < E[X_0] < \infty$ and $0 < E[S_0] < \infty$, and define the *nominal traffic intensity* in the usual way as ρ =E[X_0]/$E[S_0]$.

We analyze this model by recognizing that the waiting times satisfy the generalized Lindley recursion

$$W_{n+1} = [C_n \times W_n + R_n]^+, \qquad n \geq 0; \tag{3.3}$$

where,

$$C_n = 1 + G_n \qquad \text{and} \qquad R_n = S_n - X_n, \qquad n \geq 0. \tag{3.4}$$

Equation(3.3) reduces to classical Lindley recursion when P(C_0=0)=1. Similar to the classical case, our analysis of the queuing model depends on the equation(3.3) and the sequence$\{(C_n, R_n)\}$, and not on the specific way (C_n, R_n) is defined in terms of (X_n, S_n, G_n) in equation(3.4). Recursion(3.3) is a special case of more general recursions that have been analyzed earlier. However, we will get stronger results for (3.3) by analyzing special structure in detail.

Our analysis of (3.3) is primarily based on relating it to unrestricted recursion

$$Y_{n+1} = C_n \times Y_n + R_n, \qquad n \geq 0, \tag{3.5}$$

which has been studied by Vervaat and Brandt (A. Brandt, 1986) in detail.

The system studied here has different stability conditions then the nominal system in which $C_n = 1$ with probability 1 in (3.3). In the nominal system the stability condition is well known which is $\rho < 1$. However, in our system, when $P(C_n = 1) < 1$, stability depends on multiplicative factor C_n instead of ρ. Moreover, for this model, the concept of stability is only a limited partial characterization. It is possible to have instability, even though the time required to reach a high level, from which the process can diverge to $+\infty$, can be very large with high probability. On the other hand, it is possible to have stability, even though the limiting distribution can concentrate on very high values.

We also focus on stable systems with $\rho > 1$. Having $\rho > 1$ can tend to keep the process $\{W_n\}$ in (3.3) away from origin, so that $\{W_n\}$ behaves much like $\{Y_n\}$ in (3.5). We'll also show that a normal approximation for $\{Y_n\}$ developed by Vervaat also applies to $\{W_n\}$ when $\rho > 1$ under appropriate conditions. We'll apply this approximation to determine specific policies for scheduling service-times under which the waiting times are approximately contained in a specified interval a specified fraction of time. These policies have the property that service-times change smoothly, which is desirable. We'll use these policies to determine the *inter-departure time* and try to reduce it's *variance*.

3.2 Stability of Queues

To obtain the stability conditions of our model, we'll begin with two preliminary lemmas as mentioned in (Ward WHITT, 1990). The first Lemma relates recursion (3.3) to an associated unrestricted recursion of the form (3.5). We say that a sequence $\{W_n : n \geq 0\}$ is *stochastically bounded* if for all $\gamma > 0$ there exists a constant *K* such that $P(|W_n| > K) < \gamma$ for all n. A sequence is stochastically bounded if and only if every sub-sequence has a sub-sequence converging to a proper limit. Let $\Rightarrow$ denote the convergence in the distribution.

In equation (3.3), if we replace C_n by $[C_n]^+$ and R_n by $[R_n]^+$; then the waiting-times will be at least as large and the positive-part operator in (3.3) becomes unnecessary.

Lemma 1:(Ward WHITT, 1990) If W_n satisfies (3.3), then $W_n \leq Y_n$ for all n with probability 1, where

$$Y_{n+1} = [C_n]^+ \times Y_n + [R_n]^+, \qquad n \geq 0, \tag{3.6}$$

and $Y_0 = W_0 \geq 0$.

Corollary: (Ward WHITT, 1990) If W_n satisfies (3.3), Y_n satisfies (3.6) and $Y_n \Rightarrow Y$ as $n \to \infty$ where Y is proper then $\{W_n\}$ is stochastically bounded and $P(W_n > t) \leq P(Y_n > t)$ for all t, where W is the limit of any convergent subsequence of $\{W_n\}$.

We'll now get an explicit expression for W_n for the condition $P(C_0 \geq 0) = 1$. Without loss of generality, we assume that the stationary sequence $\{(C_n, R_n)\} : n \geq 0\}$ has been extended to $-\infty < n < \infty$. Let $=^d$ denotes equality in distribution. We say that a sequence $\{W_n : n \geq 0\}$ is *stochastically increasing* if $W_n \leq_d W_{n+1}$ for all n, where $\leq_d$ denotes stochastic order.

Lemma 2:(Ward WHITT, 1990) If $P(C_0 \geq 0) = 1$, then

$$W_{n+1} = max(0, R_n, R_n + C_n \times R_{n-1}, R_n + C_n \times R_{n-1} + C_n \times C_{n-1} \times R_{n-2},$$

$$\ldots\ldots\ldots\ldots\ldots, \mathbf{R}_n + C_n \times R_{n-1} + \ldots + C_{n2} \times R_1 + C_{n1} \times R_0 + C_{n0} \times W_0) \tag{3.7}$$

$$=^d M_{n+1} \equiv max\{0, R_0, R_0 + C_0 \times R_{-1}, R_0 + C_0 \times R_{-1} \times C_0 \times C_{-1} \times R_{-2},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \mathbf{R}_0 + C_0 \times R_{-1} + \ldots + C_{0-(n-1)} \times r{-}1 + C_{0-n} \times W_0. \tag{3.8}$$

If $W_0 = 0$, then W_n is stochastically non-decreasing in n, so that $W_n \Rightarrow W =^d M$ as $n \to \infty$, where $M_n \Rightarrow M$ as $n \to \infty$, with M possibly being improper. It is easy to see that non-negativity condition on C_0 is needed in *lemma 2*

Now, we will discuss about the *conditions of stability* with respect to random sequence C_n and in particular C_0 (Ward WHITT, 1990). There are five possible cases:

(a) If $P(C_0 < 0) > 0$, then W_n is stochastically bounded for all ρ and W_0. If, in addition , $\{(C_n, R_n)\}$ is a sequence of independent vectors with $P(C_0 \leq 0, R_0 \leq 0) > 0$, then the events $\{W_{n+1} = 0\}$ are regeneration points with finite mean time and $W_n \Rightarrow W$ as $n \to \infty$, where W is proper for all ρ and W_0. (A. Brandt, 1986)

(b) If $P(C_0 \geq 0) = 1$ and $P(C_0 = 0) > 0$, then $W_n \Rightarrow W$ as $n \to \infty$, where W is proper for all ρ and W_0.

(c) If $P(C_0 > 0) = 1$ and $E[logC_0] < 0$, then $W_n \Rightarrow W$ as $n \to \infty$, where W is proper for all ρ and W_0.

(d) If $P(C_0 > 0) = 1$ and $E[logC_0] > 0$, then $W_n/(C_0 \ldots C_{n-1}) \to \hat{W}$ with probability 1 as $n \to \infty$ where $(C_0 \ldots C_{n-1})^{1/n} \to e^{E[logC_0]} > 1$, with probability 1 as $n \to \infty$ and $\hat{W}$ is proper for all ρ and W_0.

(e) If $P(C_0 > 0) = 1$ and $E[logC_0] = 0$, then $W_n \Rightarrow W$ when $W_0 = 0$, where W may be proper or improper. If $P(C_0 = 0) = 1$, then $W_n \Rightarrow W$ for all W_0, where W is proper(improper) for $\rho < 1 (\rho > 1)$.

3.3 Normal Approximations when $\rho > 1$

In this section we assume that $\{(R_n, C_n)\}$ is a set of i.i.d random vectors with $E[R_0^2] < \infty$, $P(C_0 > 0) = 1$, $E[(logC_0)^2] < \infty$ and $E[logC_0] < 0$, so that $W_n \Rightarrow W$, as $n \to \infty$, where W is proper. Using the work of Vervaat and Brandt , we show that if $E[R_0] > 0$, which corresponds to $\rho > 1$, and $|E[logC_0]|$ is suitably small, then W is approximately distributed with mean (Ward WHITT, 1990)

$$E[W] = \frac{E[R_0]}{|E[logC_0]|} \tag{3.9}$$

and variance

$$Var[W] = \frac{(E[R_0])^2 \times Var[logC_0]}{2 \times |E[logC_0]|^3} + \frac{Var[R_0]}{2 \times |E[logC_0]|} + \frac{E[R_0] \times Cov[R_0, logC_0]}{(E[logC_0])^2} \tag{3.10}$$

Since W is non-negative, one test for reasonableness of this approximation is that the mean $E[W]$ should be sufficiently far away from 0 in the scale of the standard deviation $(Var[W])^{1/2}$.

Assuming that the process $\{W_n\}$ tend to not to be near the origin (which is what happening in this case, asymptotically), we should have

$$E[W] = E[C_0 \times W + R_0] \tag{3.11}$$

as a reasonable approximation, which yields (Ward WHITT, 1990)

$$E[W] = \frac{E[R_0]}{1 - E[C_0]} \tag{3.12}$$

given that $E[C_0] < 1$. We can see that equation(3.12) is consistent with equation(3.9) when C_0 tend to be slightly less than 1, i.e, if $C_0 = 1 - \epsilon \times Z_0$ for some random variable Z_0, because then

$$logC_0 = log(1 - \epsilon \times Z_0) = -\epsilon \times Z_0 = C_0 - 1. \tag{3.13}$$

and,

$Var[W] = \{2 \times E[R_0] \times E[R_0 \times C_0] \times (1 - E[C_0] + E[R_0^2] \times (1 - E[C_0])^2 - (1 - E[C_0^2]) \times (E[R_0^2])\}/\{(1 - E[C_0^2]) \times (1 - E[C_0^2])^2\}$

3.4 Scheduling Services

In this section, our focus is to be able to formulate some policies to reduce the fluctuations in the service times of successive customers. We will try to schedule the service-times using these policies. For this reason, in our model we assume that inter-arrival times come from a given sequence $\{X_n\}$ of i.i.d. random variables not subject to control. At each departure we must select the next service time depending on the history upto that point of time. By history, we mean all the previous service times and all the

inter-arrival times upto that time. Now, we want to determine $\bar{S_n}$, where

$$\bar{S_n} = S_n + H_n \tag{3.14}$$

with

$$H_n = f(S_{i-1}, X_i : i \leq n). \tag{3.15}$$

If $H_n = G_n \times W_n$, which is one of the possibilities in (3.14), then (3.14) reduces to (3.2).

Our proposition or general idea is that something like what we have previously considered should be an appropriate policy in some circumstances. We want to choose a suitable H_n. Here, we mention three general criteria for choosing H_n.

First criterion is quite obvious that waiting-time should not be large i.e., as small as possible. To achieve that we might want to control the expectation of some increasing function of waiting time, such as the mean $E[W]$.

Then, secondly, we want the throughput to be high as well as optimal, so that we might want the probability of emptiness after each departure small.

Third, we want to reduce the fluctuations in the successive service times. This should certainly help us in making inter-departure process relatively more smooth which is required for various applications. So, in effect we want to control $|\bar{S_{n+1}} - \bar{S_n}|$ or its distribution. As mentioned earlier in this section, an appropriate policy is required to fulfill our third requirement. This third criterion is our most important criterion for TDM over PSN.

We will now consider two cases of different lower bounds over service times (Ward WHITT, 1990). In first case, let the lower bound is zero, i.e., $P(S_0 = 0) = 1$. In this case, we can obviously have high and optimal service rate and $W_n = 0$ for all n (and thus satisfy the first two criteria above which is to minimize the waiting-time and obtain the optimal service time) by setting

$$\bar{S_n} = H_n = X_n - W_n, \qquad\qquad n \geq 0, \tag{3.16}$$

To me it seems like that we start from boundary condition assuming ideal cases like $W_n = 0$. However, this policy is not optimal as the successive service time will fluctuate substantially as much as the inter-arrival times will fluctuate. In particular , for $n \geq 1$, this policy gives

$$\bar{S}_{n+1} - \bar{S_n} = X_{n+1} - X_n, \tag{3.17}$$

Analyzing the above equation, we get

$$E[\bar{S}_{n+1} - \bar{S_n}] = 0 \tag{3.18}$$

and

$$Var[\bar{S}_{n+1} - \bar{S_n}] = 2 \times Var[X_n]. \tag{3.19}$$

As we can see the resultant variance of successive service times which is twice the variance of inter-arrival times is relatively larger. A natural alternative to (3.16), if smoothing of $\bar{S_n}$ is of concern, is a more smoothed response which we'll discuss about now.

For that we consider the second case where the lower bound is not zero. (Ward

WHITT, 1990) In this policy, we will relax the assumption we made previously which was $P(S_n = 0) = 1$ and reconsider equation (3.14). We shall find a policy of the form $H_n = d+\epsilon\times(X_n-W_n)$ that is more general than our previous policy and tends to keep the process W_n in a prescribed interval [a,b] (Here, ϵ is a small positive number and d is an appropriate positive number; also a and b and positive numbers such that $a < b$). To do this we will apply the normal approximation to produce control parameters d and ϵ such that

$$p(W < a) \approx P(W > b) \approx \pi \tag{3.20}$$

for any specified probability π. Our solution will require that $E[S_0] \geq E[X_0]$, i.e $\rho > 1$. Since, $\bar{S}_{n+1} = S_n + d + \epsilon \times (X_n - W_n)$, in this case we have $C_n = 1 - \epsilon$, $R_n = (\epsilon - 1) \times X_n + S_n + d$.

Now, using the previous equations for mean and variance of waiting times

$$E[W] = \frac{(\epsilon - 1) \times E[X_0] + E[S_0] + d}{\epsilon} \tag{3.21}$$

and,

$$Var[W] = \frac{(\epsilon - 1)^2 \times (Var[X_0] + Var[S_0])}{\epsilon \times (2 - \epsilon)} \tag{3.22}$$

we first use the desired range $r \equiv b - a$ to specify ϵ. Since

$$r \equiv b - a = 2 \times \beta \times (Var[W])^{1/2}; \tag{3.23}$$

where, $P(N(0,1) > \beta) = \pi$, we can apply (3.22) to obtain

$$\epsilon = 1 - \left| 1 - \frac{(Var[S_0] + Var[X_0])^2}{(Var[S_0] + [r/(2 \times \beta)]^2)^2} \right|^{1/2} \tag{3.24}$$

which has a solution provided that $Var[S_0] < (r/(2 * \beta))^2$. We can see that ϵ varies with r and β.

Next we use the intended mean $E[W] \approx (a + b)/2$ to solve for d. We apply (3.22) to get

$$E[W] = (a + b)/2 = \frac{(\epsilon - 1) \times E[X_0] + E[S_0] + d}{\epsilon} \tag{3.25}$$

so that

$$d = \epsilon \times (a + b)/2 - (\epsilon - 1) \times E[X_0] - E[S_0] \tag{3.26}$$

Of course a feasible solution requires that $d > 0$ in (3.26). A necessary condition is $E[S_0] > E[X_0]$, but ϵ determined by (3.24) must also be appropriate.

As noted above, a primary motivation for considering policies of this form is to control the fluctuations in the successive service times. We have done this in two ways. first, given that $a \leq W_n \leq b$, we have overall bounds on the final service-times, i.e.,

$$S_n + d + \epsilon \times (X_n - b) \leq \bar{S}_n \leq S_n + d + \epsilon \times (X_n - a). \tag{3.27}$$

Second, we have controlled the short run fluctuations in $\{\bar{S}_n\}$, i.e.,

$$\bar{S}_{n+1} - \bar{S}_n = S_{n+1} - (1+\epsilon) \times S_n + \epsilon \times X_{n+1} - (\epsilon)^2 \times W_n - (\epsilon)^2 \times X_n - \epsilon \times d, \tag{3.28}$$

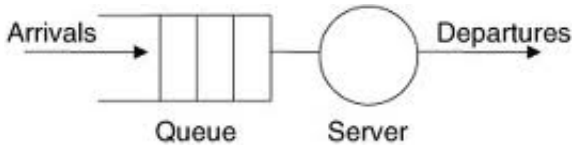

Figure 3.1: A departure process.

so that $E[\bar{S}_n] = E[X_n]$ and $E[\bar{S}_{n+1} - \bar{S}_n] \approx 0$ for large n, and

$$Var[\bar{S}_{n+1} - \bar{S}_n] = \frac{2 \times (2+\epsilon)}{2-\epsilon} \times Var[S_0] + \frac{2 \times (\epsilon)^2}{2-\epsilon} \times Var[X_0]. \tag{3.29}$$

3.5 Inter-Departure Process

The study of departure process in queuing system is primarily motivated by need to analyze queuing network model, in which the departure process of one queue is arrival process of another queue. There are very few exact results for departure processes if one considers general arbitrary distributions.

Here, we will use general definition of *inter-departure time* (11. Dimitris J Bertsimas and Daisuke , 1990)

$$ID_n = X_n + W_{n+1} - W_n + \bar{S}_{n+1} - \bar{S}_n \tag{3.30}$$

where ID_n denotes the inter-departure time between n^{th} and $\{n+1\}^{th}$ customer. Now we will replace terms $\bar{S}_{n+1}$ and $\bar{S}_n$ in the above equation by their respective definition involving waiting times i.e., W_n and W_{n+1}.

$$ID_n = X_n + W_{n+1} - W_n + S_{n+1} - S_n + \epsilon \times (X_{n+1} - X_n) - \epsilon \times (W_{n+1} - W_n) \tag{3.31}$$

Now, we will try to figure out the variance of the inter-departure process:

$$Var(ID_n) = Var(X_n + \epsilon \times (X_{n+1} - X_n)) + Var(S_{n+1} - S_n) + Var(W_{n+1} - W_n - \epsilon \times (W_{n+1} - W_n)) \tag{3.32}$$

which gets manipulated to:

$$Var((1-\epsilon) \times X_n + \epsilon \times X_{n+1}) + Var(S_{n+1} - S_n) + Var((1-\epsilon) \times (W_{n+1} - W_n))$$

We can see that to calculate the variance of inter-departure time we need to calculate the variance of $W_{n+1} - W_n$. So, we will try to calculate that variance using following equation,

$$Var(\bar{S}_{n+1} - \bar{S}_n) = Var(S_{n+1} - S_n + \epsilon \times (X_{n+1} - X_n) - \epsilon \times (W_{n+1} - W_n)) \tag{3.33}$$

So, as we can see,

$$\epsilon^2 \times Var(W_{n+1} - W_n) = (-2 \times Var(S_n) - 2\epsilon^2 \times Var(X_n) + Var(\bar{S}_{n+1} - \bar{S_n}))$$

From the above equation we will get $Var(W_{n+1} - W_n)$, and using this we find $Var(ID_n)$:

$$Var(ID_n) = \left[((1-\epsilon)^2 + (\epsilon)^2) \times \frac{1}{\lambda^2}\right] + \left[\frac{2}{\mu^2}\right] + \left[\frac{(1-\epsilon)^2}{\epsilon^2} \times (-\frac{2}{\mu^2} - \frac{2\times\epsilon^2}{\lambda^2} + (\frac{2\times(2+\epsilon)}{2-\epsilon} \times (\frac{1}{\mu^2}) + \frac{2\times(\epsilon)^2}{2-\epsilon} \times (\frac{1}{\lambda^2})))\right] \quad (3.34)$$

3.6 Results

Here, we are showing the results for the inter-departure time which is obtained by plotting the equation in MATLab.

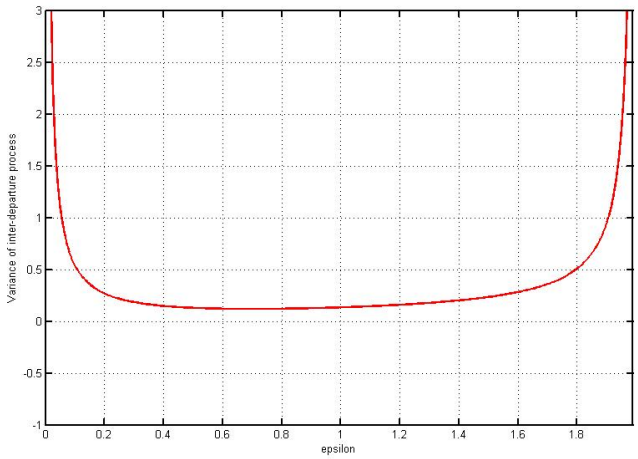

Figure 3.2: Variance of inter-departure process v/s ϵ.

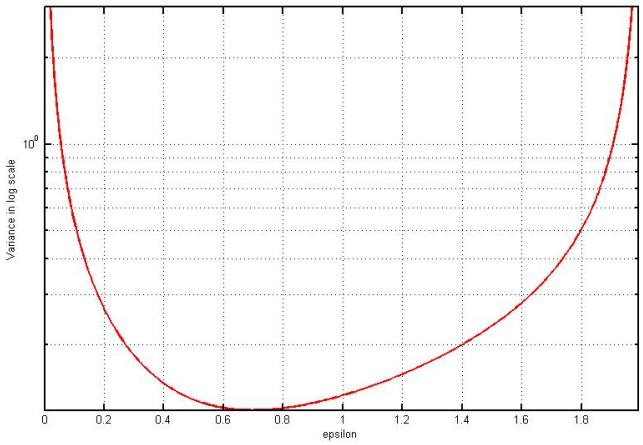

Figure 3.3: Log of variance of inter-departure process v/s ϵ.

CHAPTER 4

MATHEMATICAL PROGRAMMING PRESENTATIONS for STATE DEPENDENT QUEUES

The work of this chapter has been inspired from the paper (Chan and Schruben, 2008). In his paper, Victor Chan has developed a LP (Linear Programming) optimization model for state dependent queues where service rate decreases with increase in queue length. This is kind of an offline algorithm, where future information of arrivals and all the necessary data are already given. With minor modification, we have developed a LP optimization model for state dependent queues, where service rate increases with increase in queue length. We have tried to simulate a simpler state dependent queue and draw some inferences.

In this chapter, we are trying to model the state dependent discrete-event dynamic systems. However these systems are difficult to model due to uncertainties and dependencies of system performance on the system state. For example queue-length dependent service rate of a state dependent queue can change during service. We try to obtain a mathematical programming representation (MRP) for the sample path dynamics of a state-dependent queue.

Before we could go any further, we would like to review some concepts of *Mathematical Programming*. Mathematical Programming (MP) is the use of mathematical models, particularly optimizing models, to assist in taking decisions. The term "programming" antedates computers and means preparing "a schedule for activities". Mathematical Programming is one of a number of OR techniques. Its particular characteristic is that the best solution to a model is found automatically by optimization software. An MP model answers the question "What's best?" rather than "What happened?" (statistics), "What if?" (simulation), "What will happen?" (forecasting) or "What would an expert do and why?" (expert systems).

Mathematical Programming is more restrictive in what it can represent than other techniques. Nor should it be imagined that it really does find the best solution to the real-world problem. It finds the best solution to the problem as modeled. If the model has been built well, this solution should translate back into the real world as a good solution to the real-world problem. If it does not, analysis of why it is no good leads to greater understanding of the real-world problem.

One special case of Mathematical Programming which has been enormously successful is Linear Programming (LP). In an LP model all the relationships are linear, hence the name.

Talking about MRP, it has been recently used to describe the behavior of discrete event systems as well as their formal properties. The main advantage of such models is the rapidity of searching for the optimal solutions, given the explicit knowledge of the objective function and constraints. Here, in this chapter, an appropriate LP optimization model, for optimizing SDQ, has been proposed.

4.1 Introduction

We consider a queue with single-server whose service rate depends on the queue length called as state dependent queue (SDQ). Inter-arrival times and service requirements are generally distributed. We use a notation $G/G(Q)/1$ for the queue. SDQs are realistic models for discrete-event dynamic systems. We want to develop a mathematical programming representation (MRP) for SDQs. MRP is a mathematical programming based technique for modeling discrete-event systems (DES) dynamics as the solutions to the optimization models. A DES changes its state in accordance with occurrence of events. The trajectory of a discrete-event system, therefore, consists of a series of (state) marked event-occurrence times. Simulating such a system will give a realization of its state trajectory. Modeling system state trajectories as the solutions to an optimization problem is another way of observing the system dynamics. We want to get insights into the behaviors of state-dependent queues depending on how the server responds to the changes in line-length. It may further help us to optimize the inter-departure process of the state-dependent $G/G/1$ queue.

We will derive MRP for state dependent $G/G/1$ queue and call it as SDQ-LP. We'll use two steps to derive the SDQ-LP optimization model. At first, we'll develop set of equations for the service time in a SDQ and establish the convexity property of the service time. Then, we'll derive the constraints from a $G/G/1$ queue simulation model and from the equations for the service times, making use of the convexity property of the service time.

4.2 Linear Dependence of Service Rates on Queue Lengths

In this section we'll obtain set of equations for the service time in a SDQ and establish convexity property of the service time. Consider a queuing system with general independent arrival process where each arriving customer has a general i.i.d. service requirement. The $G/G(Q)/1$ queue follow FCFS-discipline and has infinite waiting space. The service-speed and hence the service-time depends on the queue-length i.e., the server may increase or decrease its speed when there are more or less jobs in the queue.

We say that speed of server is according to a deterministic rate function:

$$\mu(t) = f(Q(t)), \tag{4.1}$$

where $\mu(t)$ denotes the speed of sever at time t and $Q(t)$ denotes the length of queue at time t.

However, in DES, queue size changes only at discrete times, above equation can be rewritten as:

$$\mu(t) = f(Q_t), \qquad t = t_1, t_2, ... \tag{4.2}$$

where, Q_t is the queue length and $t_1, t_2, ...$ denotes the times when queue size changes due to arrival or departure of customers.

Now, let us define μ_k as the service rate when there are k customers in the queue. We will also assume that service rate increases linearly as the no of customers increases

or queue length increases. This we call that service process has increasing service rate. So, the service rate can be defined as

$$\mu_{Qt} = (Q_t + 1) \times \mu_0, \qquad t = t_1, t_2, ... \tag{4.3}$$

In the above equation, μ_0 indicates the base service rate or minimum service rate, which is the case when there are no customers waiting in the queue i.e. $Q_t = 0$.

Now, for notation we define d_{kl} as the difference between the service rates μ_k and μ_l:

$$d_{kl} = \mu_k - \mu_l, \qquad k, l \in \{0, 1, 2, \ldots, n | k > l\}. \tag{4.4}$$

Here, as we can see that service rate μ_k is monotonically increasing function and these monotonic properties lead to useful monotonic properties of the service rate.

Our next job is to define a set of sample path equations to model the service times. Here, we will assume that all customers have same service requirements for simplicity of equations. So, the service time of a customer i depends on the service rate which will change with queue length, which in turn is a function of the arrival times and finish times of other customers or jobs.

For example, if the system is empty when a customer i arrives and no other customers arrive during customer i's entire service, then the service time of customer i would simply be $1/s_0$. However, if one and only one customer (say customer $i + 1$) arrives during customer i's service, then the service time of customer i would become $a_{i+1} + (1 - a_{i+1} \times s_0)/s_1$, where a_{i+1} is the inter-arrival time between customer i and customer $i + 1$. The service time of customer $i + 1$ would then depend on the finish time of customer i and also on subsequent customers' arrival times.

Now, let k_i be the number of customers arriving during the service period of customer i. We will denote that service time of customer i as $s_{ij}^{k_i}$, when k_i customers arrive during it's service period and j being the first one to arrive.

If customer i starts a busy-period, then the queue size is zero at its arrival and k_i at the time of its departure. Now, let $sb_i^{k_i}$ gives the service time of customer i, when it initiates the busy period containing k_i jobs. Here, the following definition gives an expression for $sb_i^{k_i}$:

Definition 1 (Chan and Schruben, 2008):Define the following set of formulas

$$sb_i^0 = 1/\mu_0 \tag{4.5}$$

$$sb_i^k = \frac{1 + \sum_{l=0}^{k-1} a_{i+1+l} d_{kl}}{\mu_k} \tag{4.6}$$

where $k \in \{1, 2, \ldots, n-1\}$,

and, sb_i^k denotes the service time for customer i initiating a busy period, consisting of k customers if $k_i = k \in \{1, 2, \ldots, n-i\}$ for $i \in \{1, 2, \ldots, n\}$.

It seems that in order to find the service time $sb_i^{k_i}$, knowledge of k_i is required and it is difficult to get this information. However, our calculation becomes simple by knowing the fact that equation (4.6) is actually convex in k_i for increasing service rate system.

Other important point is that the minimum of these service time values in k_i equals to the true value of $sb_i^{k_i}$. This property will allow us to drop the subscript k_i from $sb_i^{k_i}$ and use sb_i to denote the service time service time of customer i regardless of how many customers arrive during its service period. The next *lemma* formally documents this:

Lemma 1: Given a set of $\{a_{i+1}, a_{i+2}, \ldots, a_n\}$, the formula specified in (4.6) is convex for an increasing service rate system, and the service time for customer i initiating a busy period is

$$sb_i = min\{sb_i^k\} \tag{4.7}$$

for all k and for $i \in \{1, 2, \ldots, n\}$.

Because of this convexity property we can avoid all the equations when finding the minimum.

Now, we will consider the case in which customer i arrives when another job is being served. In this case, the service time of customer i will depend upon the finish time of the customer currently being served i.e., customer $(i-1)$ and also on the customers which will arrive during its service period.

Let $sf_{ij}^{k_i}$ be the service time of customer i which arrives when server is busy and there are k_i jobs arriving during the customer i's service with j being the first customer to arrive. Let A_i and F_i be the arrival time and finish time of customer i. Here is the next definition describing the computation for $sf_{ij}^{k_i}$:

Definition 2 (Chan and Schruben, 2008): Define the following set of formulas:

$$sf_{ij}^0 = 1/\mu_{j-i-1}, \tag{4.8}$$

$$sf_{ij}^1 = \frac{1 + (A_j - F_{i-1})d_{j-i,j-i-1}}{\mu_{j-i}} \tag{4.9}$$

$$sf_{ij}^k = \frac{1 + (Aj - F_{i-1})d_{j-i-1+k,j-i-1} + \sum_{l=1}^{k-1} a_{j+1}d_{j-i-1+k,j-i-1+l}}{\mu_{j-i-1+k}} \tag{4.10}$$

for $i \in \{2, 3, \ldots, n-1\}$ and $j \in \{i+1, \ldots, n\}$.

The service time of the customer i, who arrives at a non-empty queue and seeks k_i arriving customers during its entire service period, with first arriving customer being customer j, can be computed by above formula i.e., sf_{ij}^k where $k_i = k \in \{0, 1, 2, \ldots, n-j+1\}$ and $i \in \{2, 3, \ldots, n-1\}$ and $j \in \{i+1, \ldots, n\}$.

As we saw in *lemma 1*, the formula in *definition 2* also exhibit convexity property. Hence, using this convexity property we can find the service time of customer i without knowledge of number of customers arrived during the service time of customer i. So, we will say that sf_{ij} is the service time of customer i when it arrives at a non-empty system and the first customer to be arrived is j. Here, we define *Lemma 2*

Lemma 2: Given a set of $\{a_j, a_{j+1}, \ldots, a_n\}$, the formula given in Definition 2 is convex in k for the increasing rate system and the service time of customer $i(sf_{ij})$ entering a non-empty system and j being the first customer to arrive during customer i's service period is given by:

$$sf_{ij} = min\{sf_{ij}^k\}, \tag{4.11}$$

for all k and for $i \in \{2, 3, \ldots, n-1\}$ and $j \in \{i+1, \ldots, n\}$.

However, while using *Lemma 2* we require the knowledge of the first customer j to arrive during the service period of customer i. Good news is that sf_{ij} possesses another concavity property that allows us to compute its value without knowing the actual identity of job j.

So, in next lemma, we will define sf_i as the service time of customer i when it arrives at a busy system.

Lemma 3: $sf_{ij}, j = i+1, \ldots, n$ given in *Lemma 2* is concave in j for increasing rate system and the service time of customer $i(sf_i)$, entering a busy system is:

$$sf_i = max\{sf_{ij}\}, \tag{4.12}$$

for all j and for $i \in \{2, 3, \ldots, n-1\}$.

Now, our assumption was that simulation runs for n customers; there is a possibility that all n simulated customers arrived before the service initiation of customer i. However, this possibility is quite thin, even then we will discuss about it. Let sl_i be the service time under this situation. So, lemma 4 defines this particular case of service time of customer i:

Lemma 4: The service time of customer i, when all n customers arrived earlier than its service initiation, can be computed as:

$$sl_i = 1/\mu_{n-i}, \tag{4.13}$$

for $i \in \{2, 3, \ldots, n-1\}$.

Now we are equipped with all the necessary equations to find out the service time of customer i. So we will define a theorem for service time s_i of job i.

Theorem 1 (Chan and Schruben, 2008): The service time of a customer in $G(Q)$ queue is given as

$$s_i = min\{min(\forall k)\{sb_i^k\}, max\{max(\forall j)\{min(\forall k)\{sf_{ij}^k\}\}, sl_i\}\} \tag{4.14}$$

4.3 Formulation of SDQ-LP

Theorem 1 gives an equation which is *max-plus recursion*, which can be mapped as linear constrains. In this section, we will use the stated equation to come up with a derivation of a LP(Linear Programming) for SDQ (state dependent queue).

First of all, we will start with a simulation model of $G/G/1$ queue where the service rate is constant. Figure shows this simple simulation model, which is one of the many simulation models for $G/G/1$ queue. We use ERG (event relationship graphs) to define the system dynamics. ERGs are a general, minimalist means of explicitly representing or expressing all the dynamic causal relationship between events in a discrete event dynamic model system.

The ERG in the given figure can be interpreted simply by following the arrows. We define two events: which are *Arrivals*(A) of customers and *Finish* of services. There

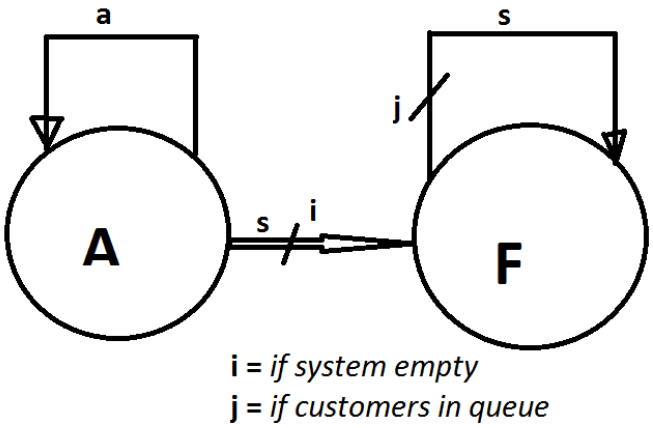

Figure 4.1: ERG simulation model of G/G/1 Queue.

are three arcs given in the figure. The first arc (A, A), which is *unconditional*, indicates that once a customer arrives, the next one is scheduled to arrive after a delay of a which is a random variable with realization a_i, called as inter-arrival time.

Second arc (A, F), which is *conditional*, indicates that when a customer arrives, and if the server is idle, it can start it's service and will leave the system after a service delay of s, which is a random variable with its realization s_i, being the service time of i^{th} customer.

Third arc (F, F), which also is *conditional*, makes sure that once the server becomes available due to departure of a customer, it will serve the next customer immediately, provided that there is at least one customer is waiting in the line. And, this job will also leave after it's service delay of s, which is the same random variable as discussed in the above paragraph for conditional arc (A, F).

The Mathematical Programming Representation (MPR) for this ERG that generates the same sample path for a given data set of $\{(a_i, s_i), i = 1, 2, ..., n\}$ for n customers is the Linear Programming (LP) shown in $GG1 - LP$. We know that arrival event times are determined by data, the sample path is simply the one which finishes all the jobs (departing of all the customers) in the minimum time i.e., as early as possible.

GG1-LP (Chan and Schruben, 2008)

$$min \sum_{\forall i} F_i \tag{4.15}$$

subjected to the constrains:

$F_i - A_i \geq s_i, \qquad i = 1, ..., n$

$F_i - F_{i-1} \geq s_i, \qquad i = 2, 3, ..., n$

and all variables are positive.

Now, we have seen a linear programming for normal $G/G/1$ queue. Our goal is to modify the $G/G/1$ ERG by replacing the independent service times $s_i, \forall i$ with the service times specified in Theorem 1. By doing this we are actually extending the $G/G/1$

ERG to a pseudo SDQ-ERG. We call this modified ERG pseudo SDQ, because it can't actually be simulated in the usual "next event scheduling" manner because it requires future information (i.e., future arrival times) to compute the service times. However, the requirement of future information is never a problem for the corresponding Linear Programming model because all the necessary data is available to LP.

Now, talking about the constrains for this pseudo ERG model for the SDQ, we can easily say that, since it has same structure as the $G/G/1$ ERG model, the two constrains in the GG1-LP model will also be needed in the SDQ-LP. Apart from these two constrains, we need to incorporate the equations in Theorem 1 into additional constrains on the service times. From here we will derive the resulting SDQ-LP.

To come up with the desired LP, we add slack variables to express all constrains as equalities. All the notations for slack variables follow the same format: y^{α}_{β}, where α represents the original variable in the corresponding constraint and β is the subscript (these can be multiple subscripts) of the original variable and the corresponding constraint. For example in a slack variable y^{F}_{i1}, where F represents the original variable F_i in the constraint and the subscript $i1$ represents that this is the first ("1") constraint with the original variable F_i.

We will try to derive the constraints for service times now. From *Lemma 1* we can say that sb_i is the minimum of all sb^k_i. From here we can draw an important conclusion that

$$sb_i \leq sb^k_i \tag{4.16}$$

$i = 1, ..., n; k = 0, ..., n - i.$

Now, we will focus on sf_{ij}. From *Lemma 2* we can conclude that sf_{ij} is the minimum of all sf^k_{ij}, which means

$$sf_{ij} \leq sf^k_{ij} \tag{4.17}$$

$i = 2, ..., n - 1; j = i + 1, ...n; k = 0, ..., n - j + 1.$

Now, we will focus on sf_i . From *Lemma 3* we can say that sf_i is the maximum of all sf_{ij}. So,

$$sf_i \geq sf_{ij} \tag{4.18}$$

$i = 2, ..., n - 1; j = i + 1, ..., n.$

Now, we will look into Theorem 1 for more constraints. In the expression for service time in Theorem 1, the second argument for the first minimum function is the maximum between sf_i and sl_i. Now, we will define sfl_i as the maximum and then:

$$sfl_i \geq sf_i, \tag{4.19}$$

and

$$sfl_i \geq sl_i \tag{4.20}$$

$i = 2, ..., n - 1$. Finally, again from Theorem 1, the service time s_i is the minimum of sb_i and sfl_i. Hence,

$$s_i \leq sb_i, \tag{4.21}$$

$i = 1, ..., n, and$

$$s_i \leq sfl_i, \tag{4.22}$$

$i = 2, ..., n - 1.$

From the inequalities (4.18),(4.19) and (4.20); we can say that the objective function of LP should act as an mechanism to push sf_i and sfl_i down to the maximum of their corresponding right hand side. At the same time we can conclude from inequalities (4.16),(4.17), (4.21) and (4.22) call the objective function to maximize sb_i, sf_{ij} and s_i by pulling it up to the minimum of their right hand sides, provided that enough incentives has been given to hold right hand side unchanged during the course of minimization or maximization.

Here is an example of such simple function:

$$min\{sf_i + sfl_i - sb_i - s_i - \sum_{\forall j} sf_{ij}\} \tag{4.23}$$

However, as we can see, the above simple LP is not optimal. It is so because, maximizing some of the variables might conflict with minimizing some of the variables. For example, minimizing sf_i in equation (4.18) will conflict with the goal of maximizing sf_{ij} in equation (4.17); because minimizing sf_i induces the inclination to have smaller $sfij$'s, which contradicts the goal of maximizing $sfij$ in (4.17).

Similarly, simply multiplying the variables with some coefficients in the objective function or Linear Programming (LP) does not work because, their coefficients would depend upon the data given in the particular problem, and determining their values will require as much effort as running the entire simulation.

So, here we will use a new technique to solve this problem. This technique is to transform the LP into a certain form so that, its optimal solution will be identical to the simulation results. There are two steps involved in this transformation. First step is to change all constraints into equalities by adding slack variables. Now, in the next step, which is second step, we define an objective function of minimizing all slack variables scaled by coefficients, c^{n-i}, where c is some constant and i is the index of the customer that the variable is associated with and n is the total number of simulated jobs. (Chan and Schruben, 2008)

Now, we perform transformation. After performing the transformation on the above constraints along with the suggested objective function, we obtain an LP with the optimal solution identical to the simulation trajectory of a SDQ.

SDQ-LP

$$min \sum_{\forall i} c^{(n-i)} * (y_{i1}^F + y_{i2}^F + \sum_{\forall k} y_{ik}^{sb} + \sum_{\forall j,k} y_{ijk}^{sfj} + \sum_{\forall j} y_{ij}^{sf} + y_{i1}^{sfl} + y_{i2}^{sfl} + y_{i1}^{s} + y_{i2}^{s}) \tag{4.24}$$

which is subjected to the following equalities:

$$F_i - A_{i-1} - y_{i1}^F = s_i, \qquad i = 1, 2, ..., n \tag{4.25}$$

$$F_i - F_{i-1} - y_{i2}^F = s_i, \qquad i = 2, ..., n \tag{4.26}$$

$$sb_i + y_{ik}^{sb} = sb_i^k, \qquad i = 1, ..., n; k = 0, ..., n - i \tag{4.27}$$

$$sf_{ij} + y_{ijk}^{sfi} = sf_{ij}^{k}, \qquad i = 2, ..., n-1; \tag{4.28}$$
$$j = i+1, ..., n; k = 0, ..., n-j+1$$

$$sf_i - y_{ij}^{sf} = sf_{ij}, \qquad i = 2, ..., n-1; j = i+1, ..., n \tag{4.29}$$

$$sfl_i - y_{i1}^{sfl} = sf_i, \qquad i = 2, ..., n-1 \tag{4.30}$$

$$sfl_i - y_{i2}^{sfl} = sl_i, \qquad i = 2, ..., n-1 \tag{4.31}$$

$$s_i + y_{i1}^{s} = sb_i, \qquad i = 1, ..., n \tag{4.32}$$

$$s_i + y_{i2}^{s} = sfl_i, \qquad i = 2, ..., n-1 \tag{4.33}$$

4.4 Simulations

In this sub-section, we will try to simulate a state dependent queue (SDQ) and compare it's results with uncorrelated queue. Let's assume that in a state dependent queue, service rate (μ_k) is linearly dependent on the queue-length (k), such that:

$$\mu_k = (1+k) \times \mu_0 \tag{4.34}$$

where, μ_0 is the service rate when there are no customers in the queue i.e., $k = 0$. Here, we will assume that once a customer n enters the server, service rate remains constant throughout it's service and is equal to $(1+k) \times \mu_0$, where k is the number of customers present in the queue at the time customer n entered the service.

Table 4.1: Here is the table comparing the *mean queue length* and the *Variance of queue length* of both types of queues for different combinations of *arrival rates* and *service rates*:

λ	μ	$mean_1$	$mean_2$	var_1	var_2
0.10	0.20	0.8568	0.43	2.23	1.92
0.20	0.40	0.8772	0.46	1.21	0.18
0.30	0.60	0.9269	0.46	1.01	0.09
0.40	0.80	1.04	0.48	1.19	0.05
0.50	1.00	1.00	0.49	1.16	0.05
0.60	1.20	1.00	0.50	1.02	0.04
0.50	1.20	0.71	0.41	0.63	0.03
0.60	1.00	1.48	0.59	2.48	0.05

From the results of the simulation we can see that if the service rate increases with queue length linearly, then the mean queue length is smaller. Also the variance of the

queue length is smaller. So, we can see that having SDQ help us in designing jitter-buffer of optimal size.

MAT Lab Code Snippet

```
    if newA < newD, (new arrival before new departure)
epoh= newA;


    present= present + 1;
newA= epoh + (-1/a)log(rand); (new arrival)


    if present == 1,


    newD= epoh + (1/d);
end


    else
epoh= newD; (new departure)
present= present - 1;
if present > 0,


    newD= epoh + (1/(d(present+1)));


    else
newD= inf;
end


    disp(transpose(s));


    m= mean(s); standard deviation = std(s);
```

CHAPTER 5

Jitter Control in Qos Networks

This chapter has been inspired from the paper (12. Mansour and Boaz , 2001). It talks about jitter control in networks and proposes on-line (arrival sequence is unknown or real time algorithm) jitter control algorithm and compare their results with the best possible jitter control algorithm (off-line algorithm) for a given arrival sequence. We have tried to understand and the simulated the on-line algorithm given here.

Jitter is measured in two terms. One measure, called *delay jitter*, bounds the maximum difference in the total delay of different packets. The second measure called the *rate jitter*, bounds the difference in packet delivery rate at various times. It measures the difference between the minimal and the maximal inter-arrival times. We will focus on *rate jitter* part.

For jitter control implementation, traffic incoming into the switch is input into a *jitter-regulator*, which reshapes the traffic by holding packets in an internal buffer. When a packet is released from a jitter-regulator, it is passed to a *link-scheduler*, which schedules packet transmission on the output link.

For our *rate jitter* algorithms, we assume that the average inter-arrival time of the input stream (X_a)is given ahead of the time. Apart from that, we parameters denoted I_{min} and I_{max} are also given which are lower and upper bound on the desired time between consecutive packets in the output stream. The on-line algorithm uses a buffer size of $2B + h$, where $h \geq 1$ is a parameter, B is such that an off-line algorithm using buffer-space B can release the packets with inter-departure times in the interval $[I_{min}, I_{max}]$.

The algorithm guarantees that the rate jitter of the released sequence is at most the best off-line jitter plus an additive term of $2(B + 2)(I_{max} - I_{min})/h$.

5.1 Rate Jitter Control

We consider the problem of minimizing the rate-jitter or how to keep the rate at which packets are released within the tightest possible bounds given as $[I_{min}, I_{max}]$. We will use the equivalent concept of minimizing the difference between inter-departure times. In this section, we will present an on-line algorithm for rate-jitter control using space $2B+h$ and compare it to an off-line algorithm using space B guaranteeing jitter J. Our algorithm guarantees rate jitter $J + cB/h$ at most , where c is a constant. The algorithm can work without knowledge of the exact average inter-arrival time. In this case, jitter guarantees will come into effect after an initial period in which packets may be released too slowly or we can say after a *transition-time*.

These are the parameters of the *online rate-jitter control algorithm*:

B = Buffer size of an off-line algorithm; $B_{off} = B$;

$h \geq 1$= space parameter for the on-line algorithm, such that $B_{on} = 2B + h$;

I_{min}, I_{max}= bounds on the minimum and maximum inter-departure time of an off-line algorithm respectively;

X_a = average inter-departure time of the input sequence and also the output sequence

The parameters I_{min}, I_{max} are the worst rate-jitter bounds the application would tolerate in order to reach optimal level of jitter. The goal of the rate jitter control algorithm is to minimize the rate jitter subject to the assumption that space B is sufficient (for an off-line algorithm) to bound the inter-departure times in the range I_{min}, I_{max}. The jitter guarantees will be expressed in terms of $I_{max}, I_{min}, X_a and J$, where J is best rate jitter for the given arrival sequence attainable by an offline algorithm using space B.

The key idea of the algorithm is to have next departure time as a monotonically function of the current number of packets in the buffer. More the packets in the buffer; lesser is the inter-departure time.

The algorithm uses $2B+h$ buffer space. With each packet in the buffer $0 \leq 2B+h$, we have a inter-departure time $IDT(j)$ defined as follows. Let $\delta = (I_{max} - I_{min})/h$.

$$\begin{array}{ll} IDT(j) = I_{max} & 0 \leq j \leq B \\ I_{max} - (j - B)\delta & B \leq j \leq B + h \\ I_{min} & B + h \leq j \leq 2B \end{array}$$

The algorithm starts with a buffer loading stage in which the packets are only accumulated and not released until for the first time $IDT(j)$ is less than X_a.

Theorem 1 (12. Mansour and Boaz , 2001) : Let Jbe the best rate-jitter attainable (for an off-line algorithm) using buffer space B for a given arrival sequence. Then the maximal rate-jitter in the release sequence generated by Algorithm is at most $J + (I_{max} - I_{min})(2B + 4)/h$ and never more than $I_{max} - I_{min}$.

We did simulations based on above algorithms and found results in accordance to the *Theorem 1*.

CHAPTER 6

RESULTS OF BRAVO EFFECT

This chapter is more of a literature survey of papers published by (Yoni Nazarathy, 2009) on BRAVO effect. It claims that previous results have shown that *Balancing Reduces Asymptotic Variances of Outputs*.

Here are the results of the paper for different kind of the queues.

1. For $M/M/1/K$ queue a factor of $1/3$ appears for large K when $\lambda = \mu$

2. For $M/M/1$ queue a factor of $2(1 - 2/\pi)$ appears when $\lambda = \mu$

3. For $G/G/1$ queue a factor of $1 - 2/\pi$ appears when $\lambda = \mu$

4. For $G/G/1/K$ queue a factor of $1/3$ appears wen = μ

However, most of the above results are observation based and have been based on simulations.

However, we could not generate the same effect; it can be useful for future work on the variances of the inter-departure processes.

CHAPTER 7

CONCLUSION

We saw that state dependent queues are difficult to analyze, but we can still draw some valuable results. In chapter 3, we tried to implement a linear policy for service-time of customer n, which depends upon waiting time of customer n. We found the variance of the difference of the successive service times, which was dependent on some variable coefficient ϵ. Using this relation, we can reduce this variance and hence, reduce the fluctuations in the service times. We also derived the variance of inter-departure times, which again depends on some variable coefficient ϵ. Using this relationship, we tried to reduce the variance of the inter-departure time, which will help us in minimizing the jitter in the outgoing stream.

In chapter 4, we tried to obtain a mathematical programming representation for the sample path dynamics of a state dependent queue. Here, the service rate was dependent linearly on the number of customers in the queue. We derived equations for service times and derived the constrains from a $G/G/1$ queue simulation model from the equations for the service times. We derived the MPR formulation of the SDQ using convexity property. Finally we close the discussion by SDQ-LP (State dependent Queue- Linear Programming).

We simulated some state dependent queues were service rate depends upon the number of the customers and increases as the queue-length increases. We calculated the mean queue length and as expected it was low and the variance of the queue length dropped dramatically. It can help us in designing systems with lower buffer-sizes.

REFERENCES

1. **Leonard Kleinrock** (1975). Queuing Systems, vol 1: Theory.

2. **Sheldon M. Ross**(2010). Introduction to probability models, vol 10.

3. **Ward WHITT** (1990). Queues with service times and inter-departure times depending linearly and randomly upon waiting times,*Queuing Systems* , **6**, 335-352.

4. **A. Brandt** (1986). The stochastic equation $Y_{n+1} = A_n Y_n + B_n$ with stationary coefficients. *Advanced Applied Probability*, **18**, 211-220.

5. **Wai Kin Chan** and **Lee W. Schruben** (2008). Mathematical Programming Representations for state-dependent Queues *Proceedings of the 2008 Winter Simulation Conference*, (2), 2-15.

6. **Kingman J.F.C** (1961). The single server queue in heavy traffic. *Mathematical Proceedings of the Cambridge Philosophical Society*, **57(4)**, 902.

7. **Yoni Nazarathy** (2009). The Variance of Departure Process: Puzzling Behavior and Open Problems.

8. **R.Manivasakan** and **Usha Rani** (2012). Correlated Queue Modeling of Jitter Buffer in TDMoIP. *AICT 2012: The Eighth Advanced International Conference on Telecommunications*.

9. **Keyur Parikh** and **Junius Kim** (2007). TDM Services over IP Networks.

10. **Yoni Nazarathy** (2011).The Variance of the Departure Process: Puzzling Behavior and Open Problems.

11. **Dimitris J. Bertsimas** and **Daisuke Nakazato** (1990).The Departure Process from a *GI/G/1* Queue and its applications to the analysis of tandem queues. **WP 3275-91-MSA**

12. **Yishay Mansour** and **Boaz Patt-Shamir** (2001).Jitter Control in QoS Networks. **IEEE,Vol 9, No 4, Aug 2001**