Two Strategies For a Bus Queueing Model

by

Minjian Yuan

A Major Research Paper

Submitted to the Faculty of Graduate Studies

through the Department of Mathematics and Statistics

in Partial Fulfillment of the Requirements for

the Degree of Master of Science at the

University of Windsor

Windsor, Ontario, Canada

Two Strategies For a Bus Queueing Model

by

Minjian Yuan

APPROVED BY:

A. Hussein
Department of Mathematics and Statistics

M. Hlynka, Advisor
Department of Mathematics and Statistics

August 20, 2017

Author's Declaration of Originality

I hereby certify that I am the sole author of this major paper and that no part of this major paper has been published or submitted for publication.

I certify that to the best of my knowledge, my major paper does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my major paper, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted materials that surpass the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission of the copyright owner(s) to include such materials in my major paper and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my major paper, including any final revisions, as approved by committee and the Graduate Studies Office, and that this major paper has not been submitted for a higher degree to any other University or Institution.

Abstract

When waiting for a bus/elevator/subway with a long queue, which is the better strategy: keep waiting or go to the previous station/floor. We discuss a type of queue with random batch service ($M/M^Y/1$) based on a bus queueing model. We compare two strategies based on different conditions.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. Myron Hlynka for his continuous support of my study and research. The door to Dr. Hlynka's office was always open to me whenever I ran into any problem. He helped me with patience, motivated guidance and inspiration and steered me in the right direction.

Besides my supervisor, I would like to thank Dr. Abdulkadir Hussein for his encouragement and comments, which broadened my research perspectives.

Moreover, I would like to thank everyone in the Department of Mathematics and Statistics at the University of Windsor. From all of the generous help from you, I made it through three graduate terms happily and fully.

I also want to thank my best friend Yizhou Chen specifically, who not only provided encouragement to me, but also gave useful ideas and suggestions me to plot the graphs. I am lucky to have such a helpful and kind-hearted friend.

Last but not the least, I must express my very profound gratitude to my parents and my family for providing me with endless support and continuous encouragement throughout my years of study and life. This accomplishment would not have been possible without them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we will first introduce some theoretical results before analyzing two strategies. To analyze the waiting time in both strategies, we have to discuss the number of passengers, or the queue length, in both stations. Obviously, the queue length is influenced by how often a bus and a passenger arrive at the station. This is exactly a queueing system. Therefore we have to understand how a queueing system works in order to solve our problem.

To explain how a queueing system works, we have to introduce a very important process in queueing theory – the Poisson process. It has many nice properties. To simplify the model, we often assume passenger arrivals follow a Poisson process. The Laplace transform is also a very powerful tool that we use in our analysis.

Then we will focus on the queue model itself. We will first discuss one of the most basic models, the $M/M/1$ system, and also discuss some methods to analyze the system. Then we expand the model to a more complex $M/M^Y/1$ system.

1.1 Poisson Process and "PASTA"

As we mentioned before, the Poisson process is important in queue theory due to its nice properties. We begin from the definition. The following statement is retrieved from "Queueing Systems", written by I. Adan and J. Resing, March,2015.[1]

" Let $N(t)$ be the number of arrivals in $[0, t]$ for a *Poisson process* with rate λ, i.e. the time between successive arrivals is exponentially distributed with parameter λ and independent of the past. Then $N(t)$ has a *Poisson* distribution with parameter λt, so

$$P(N(t) = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad \text{for } k = 0, 1, 2... \tag{1.1}$$

" The mean, variance and coefficient of variation of $N(t)$ are

$$E(N(t)) = \lambda t, \quad Var(N(t)) = \lambda t, \quad c^2_{N(t)} = \frac{1}{\lambda t} \tag{1.2}$$

" By the memoryless property of Poisson distribution, we can verify that

$$P(arrival\ in\ (t, t + \Delta t]) = \lambda \Delta t + o(\Delta t) \tag{1.3}$$

" Hence, when Δt is small,

$$P(arrival\ in\ (t, t + \Delta t]) \approx \lambda \Delta t \tag{1.4}$$

“ So in each small time interval of length Δt the occurrence of an arrival is equally likely. In other words, Poisson arrivals occur completely randomly in time.

“ The Poisson Process is an extremely useful process for modeling purposes in many practical applications. An important property of the Poisson Process is called “PASTA”. (Poisson Arrivals See Time Averages).

“($PASTA$) For queueing systems with Poisson arrivals, ($M/./.$ systems), arriving customers find on average the same situation in the queueing system as an outside observer looking at the system at an arbitrary point in time. More precisely, the fraction of customers finding on arrival the system in some state A is exactly the same as the fraction of time the system is in state A.”

We will use this property to analyze an $M/M^Y/1$ system later.

1.2 Laplace Transform

The Laplace transform $L_X(s)$ of a nonnegative random variable X with distribution function f(x) is define as:

$$L_X(s) = E(e^{-sX}) = \int_{x=0}^{\infty} e^{-sx} f(x)dx \tag{1.5}$$

Notice that

$$L_X(0) = E(e^{-X \cdot 0}) = E(1) = 1 \tag{1.6}$$

and

$$
\begin{aligned}
L_X'(0) &= E((e^{-sX})')|_{s=0} \\
&= E(-Xe^{-sX})|_{s=0} \\
&= -E(X)
\end{aligned} \tag{1.7}
$$

Similarly,

$$L_X^{(k)}(0) = (-1)^k E(X^k) \tag{1.8}$$

There are many useful properties of Laplace Transform. These properties can make our calculations easier when dealing with probability.

Let X, Y, Z be three random variables with $Z = X + Y$ and X, Y are independent. Then the Laplace Transform of Z can be found:

$$L_Z(s) = L_X(s) \cdot L_Y(s) \tag{1.9}$$

Moreover, when Z with probability p equals X, with probability $1 - p$ equals Y, then

$$L_Z(s) = pL_X(s) + (1 - p)L_Y(s) \tag{1.10}$$

Now we will introduce Laplace Transforms of some useful distributions.

- Suppose X is a random variable which follows an exponential distribution with rate λ. The Laplace Transform of X is

$$L_X(s) = \frac{\lambda}{\lambda + s} \tag{1.11}$$

- Suppose X is a random variable which follows an Erlang $- r$ distribution with rate λ. Then X can be written as:

$$X = X_1 + X_2 + \cdots + X_r \tag{1.12}$$

where X_i are *i.i.d.* exponential with rate λ. Therefore, we have

$$\begin{aligned} L_X(s) &= L_{X_1}(s) \cdot L_{X_2}(s) \ldots L_{X_r}(s) \\ &= \left(\frac{\lambda}{\lambda + s}\right)^r \end{aligned} \tag{1.13}$$

- Suppose X is a constant real number c, then

$$\begin{aligned} L_X(s) &= E(e^{-sX}) \\ &= E(e^{-sc}) \\ &= e^{-sc} \end{aligned} \tag{1.14}$$

1.3 Basic Queueing Systems

We use Kendall's notation to describe a queueing system [3] and denote by:

$$A/B/m/K/n/D \tag{1.15}$$

where

- A: distribution of the interarrival times
- B: distribution of the service times
- m: number of servers
- K: capacity of the system, the maximum number of customers in the system including the one being serviced
- n: population size of sources of customers
- D: service discipline

We use G to denote general distribution, use M for exponential distribution (M stands for Memoryless), use D for deterministic times.[1]

We usually only use $A/B/m$ to describe a queueing system, where A stands for distribution of interarrival times, B stands for distribution of service times and m stands for number of servers. Hence $M/M/1$ denotes a system with Poisson arrivals, exponentially distributed service times and a single server. $M/G/m$ denotes an m-

server system with Poisson arrivals and generally distributed service times, and so on.

In this section, we will introduce one of the most basic queueing models, the $M/M/1$ system, which is a system with Poisson arrivals, exponentially distributed service times and a single server. The following part is retrieved from "Queueing Systems", written by Adan and Resing, March,2015.[1]

We first assume the interarrivals follow the exponential distribution with rate λ, and service time follows the exponential distribution with rate μ. Further, in the single service model, to avoid queue length instability, we assume that:

$$\rho = \frac{\lambda}{\mu} < 1 \tag{1.16}$$

Here ρ is the fraction of time the server is working (called the utility factor).

We first consider time-dependent behavior of this system, then the limiting behavior. Let $p_n(t)$ denote the probability that at time t there are n customers in the system. Then by (1.3), when $\Delta t \to 0$,

$$\begin{aligned} p_0(t+\Delta t) &= (1-\lambda\Delta t)p_0(t) + \mu\Delta t p_1(t) + o(\Delta t) && (1.17) \\ p_n(t+\Delta t) &= \lambda\Delta t p_{n-1}(t) + (1-(\lambda+\mu)\Delta t)p_n(t) + \mu\Delta t p_{n+1}(t) + o(\Delta t) && (1.18) \end{aligned}$$

where $n = 1, 2, ...$

Hence, by letting $\Delta t \to 0$, we obtain the following infinite set of differential equations for $p_n(t)$.

$$p_0^{'}(t) = -\lambda p_0(t) + \mu p_1(t) \tag{1.19}$$

$$p_n^{'}(t) = \lambda p_{n-1}(t) - (\lambda + \mu) p_n(t) + \mu p_{n+1}(t), \quad , n = 1, 2, ... \tag{1.20}$$

It is very difficult to solve these differential equations. However, when we focus on the limiting or equilibrium behavior of this system, it is much easier.

It can be shown[2] that when $t \to \infty, p_n^{'}(t) \to 0$ and $p_n(t) \to p_n$. It follows that the limiting probabilities p_n satisfy equations

$$0 = -\lambda p_0 + \mu p_1 \tag{1.21}$$

$$0 = \lambda p_{n-1} - (\lambda + \mu) p_n + \mu p_{n+1}, \quad , n = 1, 2, ... \tag{1.22}$$

Moreover, p_n also satisfy

$$\sum_{n=0}^{\infty} p_n = 1, \tag{1.23}$$

which is called the normalization equation. We can also use a *flow diagram* to derive the normalization equations directly. For an $M/M/1$ system, the flow diagram is shown in figure (1.1):

Figure 1.1: Process diagram for $M/M/1$ Queue, k=1,2,3,...

The rate matrix of the system is:

$$
Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 & \dots \\ \mu & -(\mu+\lambda) & \lambda & 0 & 0 & \dots \\ 0 & \mu & -(\mu+\lambda) & \lambda & 0 & \dots \\ 0 & 0 & \mu & -(\mu+\lambda) & \lambda & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{1.24}
$$

Notice that the sum of each row equals 0.

To determine the equations from the flow diagram, we need to use a global balance principle. Global balance principle states that for each set of states A under the equilibrium condition, *the flow out of set is equal to the flow into that set.*

Based on figure(1.1), we have

$$
\begin{array}{cccc}
\text{State} & \text{Rate In} & = & \text{Rate Out} \\
0 & \mu p_1 & = & \lambda p_0 \\
1 & \lambda p_0 + \mu p_2 & = & (\lambda+\mu)p_1 \\
2 & \lambda p_1 + \mu p_3 & = & (\lambda+\mu)p_2 \\
 & \vdots & &
\end{array}
$$

This is exactly the normalization equation. To solve the equation, we first assume $\rho = \frac{\lambda}{\mu}$, which is known as the *utilization factor*. From the equilibrium equation of state 0, we have:

$$
p_1 = \rho p_0 \tag{1.25}
$$

When we plug (1.25) into the equilibrium equation of state 1, we have:

$$\begin{aligned} \lambda p_0 + \mu p_2 &= (\lambda + \mu)\rho p_0 \\ &= (\frac{\lambda^2}{\mu} + \lambda)p_0 \end{aligned} \tag{1.26}$$

That is

$$\mu p_2 = \frac{\lambda^2}{\mu} p_0 \tag{1.27}$$

Therefore,

$$p_2 = \rho^2 p_0 \tag{1.28}$$

Generally, we have

$$p_k = \rho^k p_0 \tag{1.29}$$

Since

$$\sum_{n=0}^{\infty} p_n = 1 \tag{1.30}$$

Using (1.29), we can replace p_k by p_0. Then we have

$$\sum_{n=0}^{\infty} \rho^n p_0 = 1 \tag{1.31}$$

That is

$$\begin{aligned} \frac{1}{1-\rho} p_0 &= 1 \\ p_0 &= 1 - \rho \end{aligned} \tag{1.32}$$

Moreover, for any k, we have

$$p_k = \rho^k(1-\rho) \tag{1.33}$$

Finally, we find the limiting probability p_k in the $M/M/1$ system. The expected queue length L is given by

$$\begin{aligned} E(L) &= \sum_{i=0}^{\infty} i \cdot p_i \\ &= \sum_{i=0}^{\infty} i \cdot \rho^i(1-\rho) \\ &= (1-\rho)\sum_{i=0}^{\infty} i\rho^i \\ &= \rho(1-\rho)\sum_{i=1}^{\infty} i\rho^{i-1} \\ &= \rho(1-\rho)(\sum_{i=1}^{\infty} \rho^i)' \\ &= \rho(1-\rho)(\frac{\rho}{1-\rho})' \\ &= \frac{\rho}{1-\rho} \end{aligned} \tag{1.34}$$

In the next section, we will use a similar method to analyze a more complicated queueing system.

1.4 $M/M^Y/1$ Queue

We consider a more complicated queueing system with a single server, Customers are served in a batch. The batch size of customers may be fixed or random. We assume customer interarrivals and service follow exponential distributions with rate λ, μ respectively. Then we have an $M/M^Y/1$ queue. We further assume the batch size Y of customers being served follows a general distribution:

$$P(Y = k) = p_k, \text{ for } k = 1, 2, 3, ...c \tag{1.35}$$

where c is the maximum number of customers in a service and p_k satisfies:

$$\sum_{k=1}^{c} p_k = 1 \tag{1.36}$$

We now consider the queue length after each service when the system is in equilibrium. The probability that the system has m customers before a service and the system is empty after the service, is the probability that the server can serve at least m customers, which is $\sum_{i=m}^{c} p_i$. On the other hand, the probability that the system has m customers before the service and the system has n customers after the service, where n is a positive integer and less than m, is the probability that the server serves exactly $m - n$ customers, which is p_{m-n}.

The flow diagram for $M/M^Y/1$ is shown as below.

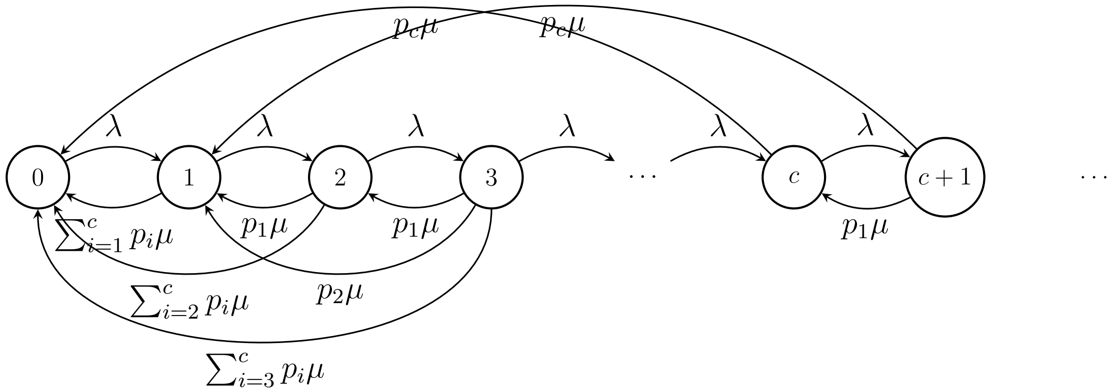

Figure 1.2: Process diagram for $M/M^Y/1$ Queue

The corresponding rate matrix is:

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 & \dots \\ \mu(\sum_{i=1}^{c} p_i) & -(\mu(\sum_{i=1}^{c} p_i) + \lambda) & \lambda & 0 & 0 & \dots \\ \mu(\sum_{i=2}^{c} p_i) & \mu p_1 & -(\mu(\sum_{i=1}^{c} p_i) + \lambda) & \lambda & 0 & \dots \\ \mu(\sum_{i=3}^{c} p_i) & \mu p_2 & \mu p_1 & -(\mu(\sum_{i=1}^{c} p_i) + \lambda) & \lambda & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{1.37}$$

Since

$$\sum_{k=1}^{c} p_k = 1 \tag{1.38}$$

the rate matrix can be simplified as:

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 & 0 & 0 & \dots \\ \mu(\sum_{i=1}^{c} p_i) & -(\mu+\lambda) & \lambda & 0 & 0 & 0 & 0 & \dots \\ \mu(\sum_{i=2}^{c} p_i) & \mu p_1 & -(\mu+\lambda) & \lambda & 0 & 0 & 0 & \dots \\ \mu(\sum_{i=3}^{c} p_i) & \mu p_2 & \mu p_1 & -(\mu+\lambda) & \lambda & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \dots \\ \mu p_c & \mu p_{c-1} & \mu p_{c-2} & \dots & \mu p_1 & -(\mu+\lambda) & \lambda & \dots \\ 0 & \mu p_c & \mu p_{c-1} & \mu p_{c-2} & \dots & \mu p_1 & -(\mu+\lambda) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{1.39}$$

It is a banded matrix.

To make this clear, we can rewrite the rate matrix as a block matrix:

$$Q = \begin{bmatrix} A & B & 0 & 0 & 0 & \dots \\ C & D & B & 0 & 0 & \dots \\ 0 & C & D & B & 0 & \dots \\ 0 & 0 & C & D & B & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{1.40}$$

A, B, C, D are both $(c+1) \times (c+1)$ matrices. Where:

$$A = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & \dots & 0 & 0 \\ \mu(\sum_{i=1}^{c} p_i) & -(\mu+\lambda) & \lambda & 0 & \dots & 0 & 0 \\ \mu(\sum_{i=2}^{c} p_i) & \mu p_1 & -(\mu+\lambda) & \lambda & \dots & 0 & 0 \\ \mu(\sum_{i=3}^{c} p_i) & \mu p_2 & \mu p_1 & -(\mu+\lambda) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu(\sum_{i=c-1}^{c} p_i) & \mu p_c & \mu p_{c-1} & \mu p_{c-2} & \dots & -(\mu+\lambda) & \lambda \\ \mu p_c & \mu p_{c-1} & \mu p_{c-2} & \mu p_{c-3} & \dots & \mu p_1 & -(\mu+\lambda) \end{bmatrix} \tag{1.41}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \lambda & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{1.42}$$

$$C = \begin{bmatrix} 0 & \mu p_c & \mu p_{c-1} & \mu p_{c-1} & \dots & \mu p_2 & \mu p_1 \\ 0 & 0 & \mu p_c & \mu p_{c-1} & \dots & \mu p_3 & \mu p_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mu p_c & \mu p_{c-1} \\ 0 & 0 & 0 & 0 & \dots & 0 & \mu p_c \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \tag{1.43}$$

$$D = \begin{bmatrix} -(\mu+\lambda) & \lambda & 0 & 0 & \dots & 0 & 0 \\ \mu p_1 & -(\mu+\lambda) & \lambda & 0 & \dots & 0 & 0 \\ \mu p_2 & \mu p_1 & -(\mu+\lambda) & \lambda & \dots & 0 & 0 \\ \mu p_3 & \mu p_2 & \mu p_1 & -(\mu+\lambda) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu p_{c-1} & \mu p_c & \mu p_{c-1} & \mu p_{c-2} & \dots & -(\mu+\lambda) & \lambda \\ \mu p_c & \mu p_{c-1} & \mu p_{c-2} & \mu p_{c-3} & \dots & \mu p_1 & -(\mu+\lambda) \end{bmatrix} \tag{1.44}$$

We are interesting in the performance when the queue is in equilibrium. We treat the number of customers in the queue as states and denote the limiting probability of being in state i by π_i, which represents that the queue length is i. Obviously the summation of all π_i is 1. Moreover, we assume when the system is in equilibrium, the flow rate into any state is always equal to the flow rate out of the state (See section (1.3)). So, we can determine a series of equations for π_i based on Figure (1.2), with respect to parameters λ, μ and p_i.

State	Rate In	=	Rate Out
0	$(\pi_1 \sum_{i=1}^{c} p_i + \pi_2 \sum_{i=2}^{c} p_i + \cdots + \pi_c p_c)\mu$	=	$\pi_0 \lambda$
1	$\pi_0 \lambda + (\pi_2 p_1 + \cdots + \pi_{c+1} p_c)\mu$	=	$\pi_1 \lambda + \pi_1 \mu$
	$\vdots$		
k	$\pi_{k-1} \lambda + (\pi_{k+1} p_1 + \cdots + \pi_{k+c} p_c)\mu$	=	$\pi_k \lambda + \pi_k \mu$
	$\vdots$		
c	$\pi_{c-1} \lambda + (\pi_{c+1} p_1 + \cdots + \pi_{2c} p_c)\mu$	=	$\pi_c \lambda + \pi_c \mu$
	$\vdots$		
c+k	$\pi_{c+k-1} \lambda + (\pi_{c+k+1} p_1 + \cdots + \pi_{2c+k} p_c)\mu$	=	$\pi_{c+k} \lambda + \pi_{c+k} \mu$
	$\vdots$		

Defining $\dfrac{\lambda}{\mu} = \rho_o$ (notice that this is NOT the utilization factor in $M/M^Y/1$ system), we can then simplify the equations as follows:

$$\pi_k = \rho_o^{-1}(\pi_{k+1} \sum_{i=1}^{c} p_i + \pi_{k+2} \sum_{i=2}^{c} p_i + \cdots + \pi_{c+k} p_c), \text{ for } k = 0, 1, 2, ... \tag{1.45}$$

or

$$\pi_k = \rho_o^{-1} \sum_{j=1}^{c} (\pi_{j+k} \sum_{i=j}^{c} p_i), \text{ for } k = 0, 1, 2, ... \tag{1.46}$$

With the fact

$$\sum_{i=1}^{\infty} \pi_i = 1 \tag{1.47}$$

we can solve for π_i in practice. Then, we can also find the expected number of waiting passengers $E(L)$ in the $M/M^Y/1$ queue by the following equation.

$$E(L) = \sum_{i=0}^{\infty} i\pi_i \tag{1.48}$$

Chapter 2

Comparison of two strategies

Now, consider the following situation. We have to wait for a bus in a rush hour at Station A. Since the queue is long, it might take some time to get on the bus. One strategy is to take a bus going the opposite direction and go to the previous station B and wait for a bus. It might take some time to get to the previous station, but it might take less time to get on a bus. Following graphs indicates two different strategies. Strategy A(stay at station A):

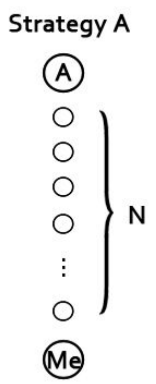

Figure 2.1: Strategy A

Strategy B(go to previous station B and return)

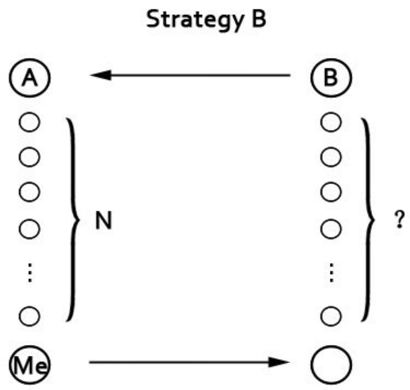

Figure 2.2: Strategy B

We are interested in which strategy is less time-consuming. We will use the knowledge we introduced in Chapter 1 to find the distribution of time to get on a bus. We will also compare the expected time to get on a bus for both strategies based on different conditions. Similar problems exist for subways, elevators.

To simplify the calculation, we first make several assumptions and add notations.

- When we arrive at station A, there are already N passengers in the queue. We only care about the time spent to get on a bus. We don't care about passengers arriving after us. Hence, we are not concerned with the passenger arrivals at station A.

- Unlike station A, the queue length at station B is unknown until we get there. In order to find the expected queue length at station B, we have to consider the passenger arrivals. We assume passenger arrivals at station B follow a Poisson Process with rate λ.

- Bus interarrivals at stations A and B are exponential with the same rate μ.
- The maximum capacity of each bus is c.
- We define Y_{A_i} and Y_{B_i} as the number of passengers who can get on the i^{th} bus at station A,B. Here Y_{A_i} and Y_{B_i} are two discrete random variables with values between 0 and c. Moreover, Y_{A_i} are *i.i.d.* and Y_{B_i} are *i.i.d.*.
- We assume there is no full bus at station B, i.e. $P(Y_{B_i} = 0) = 0$

Now we begin to analyze the simplest model.

2.1 No Waiting At Station B and constant service

We assume passengers at station B are so few that we can always get on the first coming bus at station B. In this case, we make two more assumptions to simplify the model.

- Each bus at station A can serve S passengers, where S is a constant positive integer, i.e. $Y_A = S$.
- A bus will spend time t from station B to A and from A to B, where t is a constant.

Under such conditions, we consider our first strategy A. We need to wait $\lfloor \frac{N}{S} \rfloor + 1$ buses to get on, where $\lfloor . \rfloor$ is the floor function. Let T_{A_i} and T_{B_i} denote the time between $(i-1)^{st}$ and i^{th} bus arrival at station A,B respectively. Then by our assumption, T_{A_i} and T_{B_i} are *i.i.d.* with exponential distributions with rate μ. Let T_A

denote the time we spend from arrival at station A till we get on a bus with strategy A. Then we have:

$$T_A = T_{A_1} + T_{A_2} + ... + T_{A_{\lfloor \frac{N}{S} \rfloor + 1}} \tag{2.1}$$

Let $\tilde{T}_{A_i}(s)$ denote the Laplace transform for T_{A_i}. Since T_{A_i} are *i.i.d.* exponential with rate μ, we have

$$\begin{aligned} \tilde{T}_A(s) &= \tilde{T}_{A_1}(s) \cdot \tilde{T}_{A_2}(s) \cdot ... \cdot \tilde{T}_{A_{\lfloor \frac{N}{S} \rfloor + 1}}(s) \\ &= \tilde{T}_{A_1}(s)^{\lfloor \frac{N}{S} \rfloor + 1} \\ &= \Big(\frac{\mu}{\mu + s}\Big)^{\lfloor \frac{N}{S} \rfloor + 1}. \end{aligned} \tag{2.2}$$

This is the Laplace transform for an $Erlang - (\lfloor \frac{N}{S} \rfloor + 1)$ distribution, which implies T_A follows an $Erlang - (\lfloor \frac{N}{S} \rfloor + 1)$ distribution.

Moreover,

$$E(T_A) = -\tilde{T}_A'(0) = (\lfloor \frac{N}{S} \rfloor + 1)\frac{1}{\mu} \tag{2.3}$$

On the other hand, consider another strategy B. It includes four periods.

1. Waiting for the first bus from the opposite direction at station A. Let this time be T_O, where T_O follows an exponential distribution with rate μ.

2. The bus spends time t from station A to B.

3. Waiting for the first bus at station B, we use T_{WB} to denote it. Since we assume we can always get on the first bus at station B, so we have $T_W B = T_{B_1}$ here.

4. The bus spends time t from station B to A again.

Let T_B denote the time we spend from arrival at station A, until we return to station A, going through station B. Then we have:

$$T_B = T_O + t + T_{B_1} + t = T_O + T_{B_1} + 2t \tag{2.4}$$

Since T_{B_1} and T_O are *i.i.d.* with exponential distributions with rate μ, we have

$$\begin{aligned} \tilde{T_B}(s) &= \tilde{T}_{B_1}(s) \cdot \tilde{T_O}(s) \cdot e^{-2st} \\ &= e^{-2st}\Big(\frac{\mu}{\mu + s}\Big)^2. \end{aligned} \tag{2.5}$$

Therefore,

$$E(T_B) = -\tilde{T}'_B(0) = \frac{2}{\mu} + 2t \tag{2.6}$$

So we can compare the expectations of T_A and T_B to decide which strategy is better. When

$$t < (\lfloor \frac{N}{S} \rfloor - 1)\frac{1}{2\mu}, \tag{2.7}$$

we should use strategy B. Otherwise, we should wait patiently at station A.

In the next section, we discuss a more complex model. Instead of each bus serving S passengers, we assume each bus can serve Y passengers, where Y is a random variable with specified distribution.

2.2 No Waiting At Station B and random service

We now assume Y_{A_i} is the number of passengers who can get on the i^{th} bus at station A, where Y_{A_i} are $i.i.d.$ with values between 0 and c, where c is the maximum capacity of an empty bus. Further we let

$$P(Y_{A_i} = j) = p_{A_j}, \text{ for } i = 1, 2, 3, ...; \ j = 0, 1, 2, 3, ..., c \tag{2.8}$$

where

$$\sum_{i=0}^{c} p_{A_i} = 1 \tag{2.9}$$

Since we assume we can always get on the first bus at station B, the previous analysis doesn't change for strategy B. We still have:

$$E(T_B) = \frac{2}{\mu} + 2t \tag{2.10}$$

On the other hand, if we assume

$$S_n = \sum_{i=1}^{n} Y_{A_i} \tag{2.11}$$

then S_{A_n} denotes the number of passengers being served by the first n^{th} buses at station A. Then the probability that we need to wait exactly k buses to get on at station A is found as follows:

When $N \leqslant c$,

$$\begin{aligned} P(\sum_{i=1}^{k} Y_{A_i} > N, \sum_{i=1}^{k-1} Y_{A_i} < N+1) &= P(S_{A_k} > N, S_{A_{k-1}} < N+1) \\ &= \sum_{i=0}^{N} (P(S_{A_{k-1}} = i) \cdot \sum_{k=N-i+1}^{c} p_{A_k}) \\ &= \sum_{i=0}^{N} (P(S_{A_{k-1}} = i)(1 - \sum_{k=0}^{N-i} p_{A_k})) \quad (2.12) \end{aligned}$$

If we define $q_j = \sum_{k=0}^{j} p_{A_k}$, then we have

$$P(\sum_{i=1}^{k} Y_{A_i} > N, \sum_{i=1}^{k-1} Y_{A_i} < N+1) = \sum_{i=0}^{N} P(S_{A_{k-1}} = i) - \sum_{i=0}^{N} P(S_{A_{k-1}} = i) \cdot q_{N-i} \quad (2.13)$$

Further, we can simplify the result, noticing that:

$$\begin{aligned} \sum_{i=0}^{N} P(S_{A_k} = i) &= \sum_{i=0}^{N} \sum_{j=0}^{i} [P(S_{A_{k-1}} = j) P(Y_{A_k} = i - j)] \\ &= \sum_{i=0}^{N} \sum_{j=0}^{i} P(S_{A_{k-1}} = j) \cdot p_{A_{i-j}} \\ &= \sum_{j=0}^{N} \sum_{i=j}^{N} P(S_{A_{k-1}} = j) \cdot p_{A_{i-j}} \\ &= \sum_{j=0}^{N} P(S_{A_{k-1}} = j) \cdot \sum_{i=j}^{N} p_{A_{i-j}} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=0}^{N} P(S_{A_{k-1}} = j) \cdot \sum_{k=0}^{N-j} p_{A_k} \\
&= \sum_{j=0}^{N} P(S_{A_{k-1}} = j) \cdot q_{N-j} \qquad (2.14)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
P(we\ get\ on\ exactly\ the\ k^{th}\ bus) &= P(\sum_{i=1}^{k} Y_{A_i} > N, \sum_{i=1}^{k-1} Y_{A_i} < N+1) \\
&= \sum_{i=0}^{N} P(S_{A_{k-1}} = i) - \sum_{i=0}^{N} P(S_{A_k} = i) \qquad (2.15)
\end{aligned}
$$

so we have

$$
\begin{aligned}
E(T_A) &= \sum_{k=1}^{\infty} P(\sum_{i=1}^{k} Y_{A_i} > N, \sum_{i=1}^{k-1} Y_{A_i} < N+1) \cdot E(\sum_{j=1}^{k} T_{A_j}) \\
&= \sum_{k=1}^{\infty} P(\sum_{i=1}^{k} Y_{A_i} > N, \sum_{i=1}^{k-1} Y_{A_i} < N+1) \cdot (\frac{k}{\mu}) \\
&= \frac{1}{\mu} \cdot (\sum_{k=1}^{\infty} k \cdot P(\sum_{i=1}^{k} Y_{A_i} > N, \sum_{i=1}^{k-1} Y_{A_i} < N+1)) \\
&= \frac{1}{\mu} \cdot (\sum_{k=1}^{\infty} k(\sum_{i=0}^{N} P(S_{A_{k-1}} = i) - \sum_{i=0}^{N} P(S_{A_k} = i))) \qquad (2.16)
\end{aligned}
$$

We can find $P(S_{A_k} = i)$ by using a pseudo transition matrix. Define P_A as the pseudo transition matrix of queue length at station A. Then:

$$P_A = \begin{bmatrix} p_{A_0} & 0 & 0 & 0 & 0 & \dots & 0 \\ p_{A_1} & p_{A_0} & 0 & 0 & 0 & \dots & 0 \\ p_{A_2} & p_{A_1} & p_{A_0} & 0 & 0 & \dots & 0 \\ p_{A_3} & p_{A_2} & p_{A_1} & p_{A_0} & 0 & \dots & 0 \\ p_{A_4} & p_{A_3} & p_{A_2} & p_{A_1} & p_{A_0} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ p_{A_{N+1}} & p_{A_N} & p_{A_{N-1}} & p_{A_{N-2}} & p_{A_{N-3}} & \dots & p_{A_0} \end{bmatrix} \tag{2.17}$$

In strategy A, we are not interested in the passengers arriving after us, so we can assume no other passengers arrive after us. Thus the queue length never gets longer. Therefore, P_A is a lower-triangular matrix. Then

$P_A[i,j]$=P(next bus serves exactly $i-j$ passengers)

Here $P_A[i,j]$ denotes the element at the i^{th} column and the j^{th} row of the matrix P_A.

So

$$\begin{aligned} P(S_{A_k} = i) &= P(Y_{A_1} + Y_{A_2} + \cdots + Y_{A_k} = i) \\ &= P_A^k[i+1, 1] \end{aligned} \tag{2.18}$$

Similarly, $P_A^k[i+1,1]$ denotes the element at the $(i+1)^{th}$ column and the 1^{st} row of the matrix P_A^k. We will use this notation in following part.

When $N > c$, if we define $\sum_{i=a}^{b} p_i = 0$ for $a > b$, and define the pseudo transition matrix P_A as:

$$P_A = \begin{bmatrix} p_{A_0} & 0 & 0 & \dots & 0 & 0 & 0 & \cdots \\ p_{A_1} & p_{A_0} & 0 & \cdots & 0 & 0 & 0 & \cdots \\ p_{A_2} & p_{A_1} & p_{A_0} & \cdots & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots \\ p_{A_c} & p_{A_{c-1}} & p_{A_{c-2}} & \cdots & p_{A_0} & 0 & 0 & \cdots \\ 0 & p_{A_c} & p_{A_{c-1}} & \cdots & p_{A_1} & p_{A_0} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2.19}$$

we can still get the same result as in the case $N \leqslant c$.

Moreover, we can also use recursion formula to solve for $E(T_A)$. Assume $E(T_A|i)$ denotes the expected time for strategy A to get on a bus when there are i passengers before us. Then, we can rewrite $E(T_A)$ to:

$$E(T_A) = E(T_A|N) \tag{2.20}$$

When there are 0 passengers before us, we consider the first bus. If the bus is not full, we can always get on since there are no passengers before us. The probability of a bus not being full is $1 - p_{A0}$, and we need the expected time $\frac{1}{\mu}$ to get on, since we assume the bus interarrivals follow an exponential distribution with rate μ. Otherwise, the first bus is full with probability p_{A0}, so we still have to wait $E(T_A|0)$ to get on. In this case, the expected time we need is $\frac{1}{\mu} + E(T_A|0)$. So, we have:

$$E(T_A|0) = \frac{1}{\mu}(1 - p_{A_0}) + p_{A_0}(\frac{1}{\mu} + E(T_A|0)) \tag{2.21}$$

Similarly, we get recursion formulas for $E(T_A|i)$:

$$\begin{aligned}
E(T_A|0) &= \frac{1}{\mu}(1 - q_0) + p_{A_0}(\frac{1}{\mu} + E(T_A|0)) && (2.22)\\
E(T_A|1) &= \frac{1}{\mu}(1 - q_1) + p_{A_0}(\frac{1}{\mu} + E(T_A|1)) + p_{A_1}(\frac{1}{\mu} + E(T_A|0)) && (2.23)\\
E(T_A|2) &= \frac{1}{\mu}(1 - q_2) + p_{A0}(\frac{1}{\mu} + E(T_A|2)) + \cdots + p_{A_2}(\frac{1}{\mu} + E(T_A|0)) && (2.24)
\end{aligned}$$

where

$$q_k = \sum_{i=0}^{k} p_{A_i} \tag{2.25}$$

We can simplify the recursion formulas as:

$$\begin{aligned}
(1 - p_{A_0}) \cdot E(T_A|0) &= \frac{1}{\mu} && (2.26)\\
(1 - p_{A_0}) \cdot E(T_A|1) &= \frac{1}{\mu} + p_{A_1} \cdot E(T_A|0) && (2.27)\\
(1 - p_{A_0}) \cdot E(T_A|2) &= \frac{1}{\mu} + p_{A_1} \cdot E(T_A|1) + p_{A_2} \cdot E(T_A|0) && (2.28)
\end{aligned}$$

Generally, if we define $p_{Ai} = 0$ when $i > c$, then for any $k = 0, 1, 2...$, we have the recursion formula:

$$E(T_A|k) = \frac{\frac{1}{\mu} + \sum_{i=1}^{k} p_{A_i} \cdot E(T_A|k - i)}{1 - p_{A_0}} \tag{2.29}$$

In Chapter 3, we build two different programs based on (2.16) and (2.29). They give exactly the same results. However, using recursion is much more efficient.

Now, we can compare $E(T_A)$ and $E(T_B)$ as we did in section 2.1 to find which strategy is better.

2.3 Waiting At Station B and random service

In this section, we will use previous results and the formulas given in section 1.4 to get both Laplace transforms and expected times of the two strategies. We assume both bus and passenger interarrivals follow exponential distributions with rates μ, λ, respectively. We also assume we might need to wait at station B using strategy B.

We first obtain the Laplace transform for the time using strategy A with a given number N. We let $L_{A_N}(s)$ denote the Laplace transform with N passengers before us. Then:

$$L_{A_N}(s) = \begin{cases} \dfrac{\mu}{\mu+s}[p_{A_0}L_{A_N}(s) + \cdots + p_{A_S}L_{A_{N-c}}(s)] & \text{for } N > c; \\ \dfrac{\mu}{\mu+s}[p_{A_0}L_{A_N}(s) + \cdots + p_{A_N}L_{A_0}(s) + \sum_{i=N+1}^{c} p_{A_i} \cdot 1] & \text{for } N \leqslant c; \end{cases} \tag{2.30}$$

Specifically,

$$L_{A_0}(s) = \frac{\mu}{\mu+s}[p_{A_0}L_{A_0}(s) + (1-p_{A_0}) \cdot 1)] \tag{2.31}$$

That is:

$$L_{A_0}(s) = \frac{\mu(1-p_{A_0})}{\mu(1-p_{A_0})+s} \tag{2.32}$$

We can find numerical solutions for all $L_{A_i}(s)$.

Strategy A in this section is identical to strategy A in section (2.2). So we still have:

$$E(T_A) = \frac{1}{\mu} \cdot (\sum_{k=1}^{\infty} k(\sum_{i=0}^{N} P(S_{A_{k-1}} = i) - \sum_{i=0}^{N} P(S_{A_k} = i))) \tag{2.33}$$

where

$$S_{A_n} = \sum_{i=1}^{n} Y_{A_i} \tag{2.34}$$

and

$$P(S_{A_k} = i) \quad = \quad P_A^k[i+1, 1] \tag{2.35}$$

If $N_A \leqslant c$, P_A can be found in (2.17). Otherwise, when $N_A > c$, P_A is given by:

$$P_A = \begin{bmatrix} p_{A_0} & 0 & 0 & \dots & 0 & 0 & 0 & \cdots \\ p_{A_1} & p_{A_0} & 0 & \cdots & 0 & 0 & 0 & \cdots \\ p_{A_2} & p_{A_1} & p_{A_0} & \cdots & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots \\ p_{A_c} & p_{A_{c-1}} & p_{A_{c-2}} & \cdots & p_{A_0} & 0 & 0 & \cdots \\ 0 & p_{A_c} & p_{A_{c-1}} & \cdots & p_{A_1} & p_{A_0} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2.36}$$

Now, we focus on strategy B. Under this model, when we take a bus in the opposite direction from station A to station B, the queue length at station B is unknown. As in section (1.2), the system at station B is an $M/M^{Y_B}/1$ system.

Since we assume the passenger arrivals follow a Poisson Process, then, by $PASTA$, when we arrive at station B, the queue we see will match the equilibrium situation of the system. Let L_B be the queue length when we arrive at station B. Then:

$$P(L_B = k) = \pi_k, \text{ for } k = 0, 1, 2, ...; \tag{2.37}$$

where π_k can be solved by:

$$\pi_k = \rho_o^{-1} \sum_{j=1}^{c} (\pi_{j+k} \sum_{i=j}^{c} p_{B_i}), \text{ for } k = 1, 2, 3, ... \tag{2.38}$$

and

$$\sum_{k=1}^{\infty} \pi_k = 1 \tag{2.39}$$

We define p_{B_j} similar to p_{A_j} here. Notice that p_{A_j}, p_{B_j} are different in our model. Generally, when j is large, p_{B_j} is greater than p_{A_j}; when j is small, p_{B_j} is smaller than p_{A_j}. The idea is that buses at station B have fewer passengers and more space than buses at station A.

Now we obtain the Laplace transform and the expected time using strategy B. Let T_B be the time and let $L_B(s)$ be the Laplace transform.and T_B. As in section (2.1), we break strategy B into four different periods:

1. Waiting for the first bus from the opposite direction at station A. Let the time be T_O. Then T_O follows an exponential distribution with rate μ.

2. The bus spends time T_{AB} from station A to B.

3. Waiting for a bus at station B until we can get on. Let this time be T_{WB}.

4. The bus spends time T_{BA} from station B to A.

Then, we have

$$T_B = T_O + T_{AB} + T_{WB} + T_{BA} \tag{2.40}$$

T_B is the total time to get on a bus which is at station A by going backward to station B.

We assume T_{AB} and T_{BA} have same distribution (and may be constant). Using tilde for Laplace transforms, we have

$$\tilde{T}_B(s) \quad = \quad \tilde{T}_O(s) \cdot \tilde{T}_{AB}(s) \cdot \tilde{T}_{WB}(s) \cdot \tilde{T}_{BA}(s)$$

$$= \frac{\mu}{\mu+s} \cdot (\tilde{T}_{AB}(s))^2 \cdot \tilde{T}_{WB}(s) \tag{2.41}$$

If $T_{AB} = T_{BA} = t$, where t is a constant, then from section (1.2), we have:

$$\tilde{T}_{AB}(s) = e^{-st} \tag{2.42}$$

If T_{AB} and T_{BA} follow an exponential distribution with rate γ, then

$$\tilde{T}_{AB}(s) = \frac{\gamma}{\gamma+s} \tag{2.43}$$

Next we obtain $\tilde{T}_{WB}(s)$. Let L_{B_M} denote the Laplace transform of time at station B when there are M passengers in the queue at station B at the time we arrive. Then:

$$\begin{aligned} \tilde{T}_{WB}(s) &= \pi_0 L_{B_0} + \pi_1 L_{B_1} + \pi_2 L_{B_2} + ... \\ &= \sum_{i=0}^{\infty} \pi_i L_{B_i}(s) \end{aligned} \tag{2.44}$$

and $L_{B_M}(s)$ satisfies

$$L_{B_M}(s) = \begin{cases} \dfrac{\mu}{\mu+s}[p_{B_0} L_{B_M}(s) + \cdots + p_{B_c} L_{B_{M-c}}(s)] & \text{for } M > c; \\ \dfrac{\mu}{\mu+s}[p_{B_0} L_{B_M}(s) + \cdots + p_{B_M} L_{B_0}(s) + \sum_{i=M+1}^{c} p_{Bi} \cdot 1] & \text{for } M \leqslant c; \end{cases} \tag{2.45}$$

with

$$L_{B_0}(s) = \frac{\mu(1-p_{B_0})}{\mu(1-p_{B_0})+s} \tag{2.46}$$

This is the Laplace Transform for an exponential distribution with rate $\mu(1-p_{B_0})$. By our assumption at the beginning, there is no full bus at station B, which implies $p_{B_0} = 0$, so

$$L_{B_0}(s) = \frac{\mu}{\mu+s} \tag{2.47}$$

which is exactly the Laplace transform of an exponential distribution with rate μ.

Further,

$$L_{B_1}(s) = \Big(\frac{\mu}{\mu+s}\Big)^2 p_{B_1} + \frac{\mu}{\mu+s}(1-p_{B_1}) \tag{2.48}$$

which implies the time at station B given one passenger ahead of us follows an $Erlang-2$ distribution with rate μ with probability p_{B_1}, and follows an exponential distribution with rate μ with probability $1-p_{B_1}$. Next

$$L_{B_2}(s) = \Big(\frac{\mu}{\mu+s}\Big)^3 p_{B_1}^2 + \Big(\frac{\mu}{\mu+s}\Big)^2 (p_{B_1}(1-p_{B_1})+p_{B_2}) + \frac{\mu}{\mu+s}(1-p_{B_1}-p_{B_2}) \tag{2.49}$$

We know the time follows an $Erlang-3$ distribution with rate μ with probability $p_{B_1}^2$, and follows an $Erlang-2$ distribution with rate μ with probability $p_{B_1}(1-p_{B_1})+p_{B_2}$ and follows an exponential distribution with rate μ with probability $1-p_{B_1}-p_{B_2}$. We can solve L_{B_i} step by step. Finally, we can find $\tilde{T}_{WB}(s)$ by (2.44). With all results, we can find $\tilde{T}_B(s)$ from (2.41), the Laplace transform of time using strategy B.

Specifically, we can use induction to prove that $L_{B_k}(s)$ can be written as

$$L_{B_k}(s) = \sum_{i=0}^{k} \left(\frac{\mu}{\mu+s}\right)^{i+1} \cdot \gamma_i \tag{2.50}$$

where γ_i is a polynomial in $p_{B_1}, p_{B_2}, \ldots, p_{B_k}$. This gives us the idea that the distribution of time to get on a bus when there are k passengers when we arrive at station B follows:

- $Erlang-(k+1)$ distribution with rate μ with probability γ_{k+1}.
- $Erlang-k$ distribution with rate μ with probability γ_k.

  $\ldots$

- $Erlang-1$ distribution (also known as exponential distribution) with rate μ with probability γ_1.

To find the expected time spent using strategy B, we can use the property of Laplace Transforms:

$$\tilde{T}_B'(0) = -E(T_B) \tag{2.51}$$

We can also directly verify the expected time using strategy B from (2.40),

$$\begin{aligned} E(T_B) &= E(T_O + T_{AB} + T_{WB} + T_{BA}) \\ &= \frac{1}{\mu} + 2E(T_{AB}) + E(T_{WB}) \end{aligned} \tag{2.52}$$

We use the same assumption for T_{AB} and T_{BA}, namely

$$T_{AB} = T_{BA} = t \tag{2.53}$$

Then we have

$$E(T_B) = \frac{1}{\mu} + 2t + E(T_{WB}) \tag{2.54}$$

Recall that T_{WB} denotes the waiting time at station B. When we arrive at station B, there is probability π_k of having k passengers in the queue. Recall T_{B_k} denotes our waiting time when there are exactly k passengers in the queue at station B. Then,

$$E(T_{WB}) = \sum_{i=0}^{\infty} \pi_i E(T_{B_i}) \tag{2.55}$$

When considering T_{B_i}, there are exactly i passengers before us, and each bus will serve a random number of passengers. This is a similar system to what we had at station A. We already discussed this system in Section 2.2 and 2.3. Recall that Y_{Bi} denotes the number of passengers served by the i^{th} bus at station B, p_{B_i} denotes the probability that a bus at station B can serve exactly i passengers. By using the result (2.33), we have

$$E(T_{B_k}) = \frac{1}{\mu} \cdot (\sum_{i=1}^{\infty} i(\sum_{j=0}^{k} P(S_{B_{i-1}} = j) - \sum_{j=0}^{k} P(S_{B_i} = j))) \tag{2.56}$$

where

$$S_{B_n} = \sum_{i=1}^{n} Y_{B_i} \tag{2.57}$$

From the previous section, we have:

$$P(S_{B_k} = i) \quad = \quad P_B^k[i+1, 1] \tag{2.58}$$

where

$$P_B = \begin{bmatrix} p_{B_0} & 0 & 0 & 0 & 0 & \dots & 0 \\ p_{B_1} & p_{B_0} & 0 & 0 & 0 & \dots & 0 \\ p_{B_2} & p_{B_1} & p_{B_0} & 0 & 0 & \dots & 0 \\ p_{B_3} & p_{B_2} & p_{B_1} & p_{B_0} & 0 & \dots & 0 \\ p_{B_4} & p_{B_3} & p_{B_2} & p_{B_1} & p_{B_0} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ p_{B_{W+1}} & p_{B_W} & p_{B_{W-1}} & p_{B_{W-2}} & p_{B_{W-3}} & \dots & p_{B_0} \end{bmatrix} \tag{2.59}$$

Here W is an arbitrary positive integer that is large enough. When we say large enough, we mean W should be greater than the queue length at station B with high probability.

Hence, we finally get $E(T_B)$ by using (2.54), (2.55) and(2.56), such that:

$$E(T_B) = \frac{1}{\mu} + 2t + \frac{1}{\mu}\sum_{k=0}^{\infty}\pi_i \cdot (\sum_{i=1}^{\infty} i(\sum_{j=0}^{k} P(S_{B_{i-1}} = j) - \sum_{j=0}^{k} P(S_{B_i} = j))) \tag{2.60}$$

Chapter 3

Further Analysis

In Chapter 1 and Chapter 2, we discussed the properties of an $M/M^Y/1$ system, and analyzed the expected time for two strategies based on three different assumptions. We now use R to simulate the $M/M^Y/1$ system, and compare the simulation result to our theoretical result to make sure the theory is correct. We also compare two strategies with different parameters, in order to have a deeper understanding.

3.1 Limiting behavior of $M/M^Y/1$ System

We are going to discuss the limiting behavior in this section. First we want to show that the number of customers in the system at the beginning will not impact the probability of the number of customers in the system when it is in equilibrium.

To simulate this, we build up an $M/M^Y/1$ system with:

- $\lambda = \mu = \dfrac{1}{2}$;

- $c = 6, p_0 = 0.1, p_1 = 0.4, p_2 = 0.3, p_3 = p_4 = p_5 = p_6 = 0.05$;
- The system will run for 300,000 steps.

The system typical behavior is as follows (for 3 random examples):

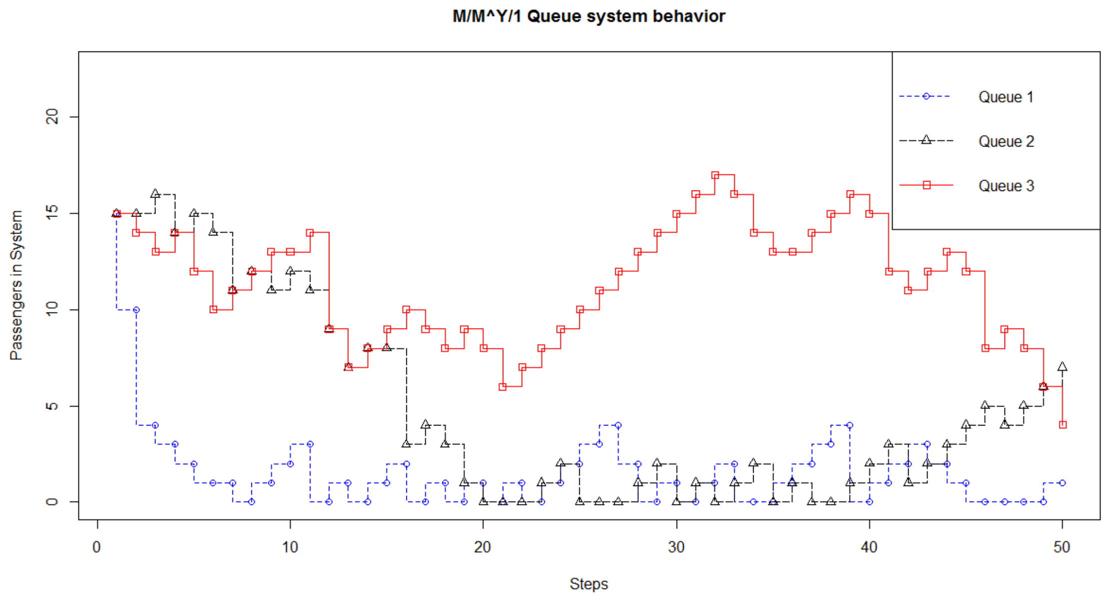

Figure 3.1: The number of passengers vs step number

The graph gives three different $M/M^Y/1$ system behaviors with exactly the same parameters – they perform differently. Therefore it is necessary to find the limiting behavior. To check the limiting behavior, we first simulate the system with three different initiate values, $N = 0, 30, 50$. If we treat the number of customers in the system as states, then the limiting probability of system in states 0,1,2,..., and the expected queue length with three different initiate values is:

N \ π_k	π_0	π_1	π_2	π_3	π_4	π_5	...	$E(Length)$
N=50	0.3019	0.2109	0.1474	0.1034	0.0718	0.0496	...	2.3127
N=30	0.3014	0.2098	0.1464	0.1032	0.0728	0.0511	...	2.3091
N=0	0.3013	0.2106	0.1467	0.1012	0.0710	0.0498	...	2.3336

Table 3.1: Comparison of probability for each state in $M/M^Y/1$ system with different N

As can be seen in the table, the probability of each state and expected queue length are almost the same. This confirms that the initiate value will not change the system behavior in the long run.

Next, we want to compare the simulation results to theoretical results based on our analysis in Chapter 1 by using R.

From Chapter 1, we know the limiting probability of each state, say π_i, can be solved from the equations:

$$\pi_k = \rho_o^{-1} \sum_{j=1}^{c} (\pi_{j+k} \sum_{i=j}^{c} p_i), \quad \text{for } k = 0, 1, 2, ... \tag{3.1}$$

and

$$\sum_{i=1}^{\infty} \pi_i = 1 \tag{3.2}$$

There are infinitely many equations with infinitely many unknowns. The method we use is to truncate the upper bound of states, since π_i will be close to 0 when i is large.

If we assume the maximum number of states in system is W. Then the equations will have $W+1$ unknowns. We rewrite them as:

$$\pi_k - \rho_o^{-1} \sum_{j=1}^{c} (\pi_{j+k} \sum_{i=j}^{c} p_i) = 0, \text{ for } k = 0, 1, 2, ...W \tag{3.3}$$

$$\sum_{i=0}^{W} \pi_i = 1 \tag{3.4}$$

We use q_k to denote $\sum_{i=0}^{k} p_i$. Then the previous equations can be written as:

$$\pi_k - \sum_{j=1}^{c} \rho_o^{-1} (1 - q_{j-1}) \cdot \pi_{k+j} = 0, \text{ for } k = 0, 1, 2, ...W \tag{3.5}$$

$$\sum_{i=0}^{W} \pi_i = 1 \tag{3.6}$$

Here we define $\pi_i = 0$ if $i > W$.

Now, define the $(W+1)\times(W+1)$ matrix A as:

$$A = \begin{bmatrix} 1 & -\rho_o^{-1}(1-q_0) & -\rho_o^{-1}(1-q_1) & -\rho_o^{-1}(1-q_2) & \dots & 0 & 0 \\ 0 & 1 & -\rho_o^{-1}(1-q_0) & -\rho_o^{-1}(1-q_1) & \dots & 0 & 0 \\ 0 & 0 & 1 & -\rho_o^{-1}(1-q_0) & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & -\rho_o^{-1}(1-q_0) \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix} \tag{3.7}$$

and define the $(W+1)\times 1$ vector b as:

$$b = (0, 0, 0, \dots, 0, 1)^T \tag{3.8}$$

Then $X = (\pi_0, \pi_1, \pi_2, \dots, \pi_W)^T$ is the solution of the equation

$$AX = b \tag{3.9}$$

We use R to solve the equation to find the limiting probabilities. Then, we compare this result with the simulation result with 2,500,000 steps, based on the same parameters as before. Here is the table:

π_k / N	π_0	π_1	π_2	π_3	π_4	π_5	...	$E(Length)$
Simulation	0.3007	0.2103	0.1473	0.1030	0.0719	0.0502	...	2.3227
Theoretical	0.3010	0.2104	0.1471	0.1028	0.0719	0.0502	...	2.3221

Table 3.2: Comparison of Simulation with Theoretical Results

The differences are extremely small. Our theory fits the simulation results. Hence in the next section, we will use the truncation method to calculate the expected times of two strategies.

We observe the change of the limiting state probabilities as λ changes. If i is small, then π_i increases as λ decreases.

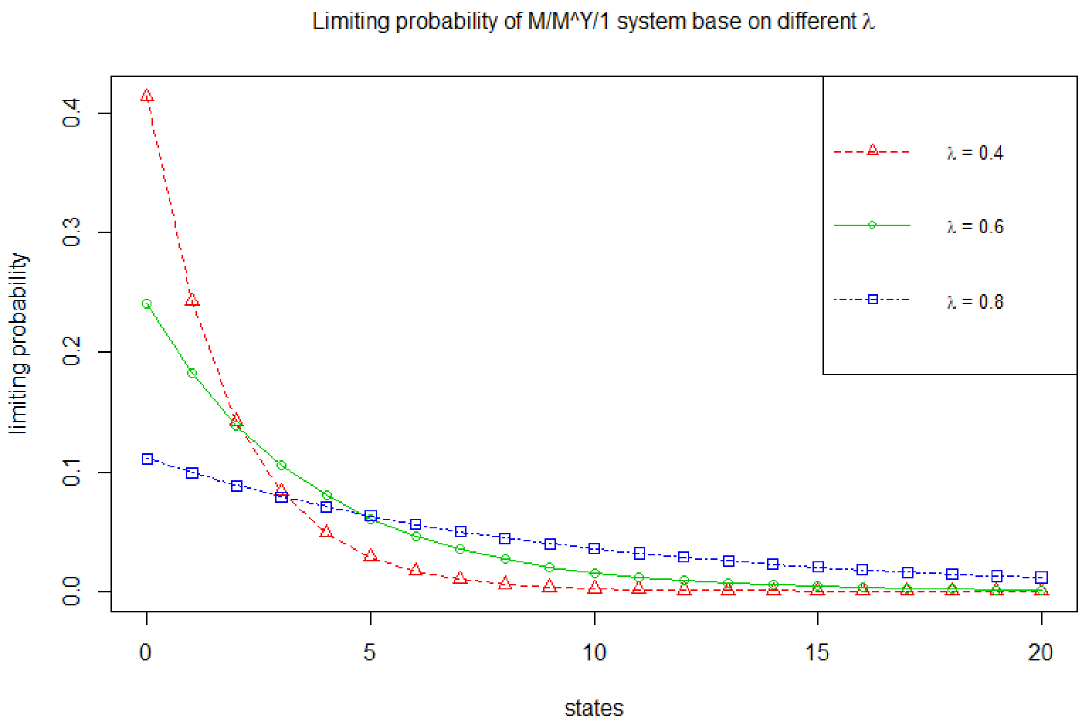

Figure 3.2: Limiting probability of $M/M^Y/1$ system with λ

3.2 Numerical results for Strategy A

From Section 2.2 and 2.3, we know the expected time to get on a bus for strategy A is:

$$E(T_A) = \frac{1}{\mu} \cdot (\sum_{k=1}^{\infty} k(\sum_{i=0}^{N} P(S_{A_{k-1}} = i) - \sum_{i=0}^{N} P(S_{A_k} = i))) \tag{3.10}$$

where

$$P(S_{A_k} = i) \quad = \quad P_A^k[i+1, 1] \tag{3.11}$$

and

$$P_A = \begin{bmatrix} p_{A_0} & 0 & 0 & \dots & 0 & 0 & 0 & \cdots \\ p_{A_1} & p_{A_0} & 0 & \cdots & 0 & 0 & 0 & \cdots \\ p_{A_2} & p_{A_1} & p_{A_0} & \cdots & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots \\ p_{A_c} & p_{A_{c-1}} & p_{A_{c-2}} & \cdots & p_{A_0} & 0 & 0 & \cdots \\ 0 & p_{A_c} & p_{A_{c-1}} & \cdots & p_{A_1} & p_{A_0} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{3.12}$$

Recall that:

$$E(T_A|k) = \frac{\frac{1}{\mu} + \sum_{i=1}^{k} p_{A_i} \cdot E(T_A|k-i)}{1 - p_{A_0}} \tag{3.13}$$

for $k = 0, 1, 2, ...N$.

Since in strategy A, we don't need to consider passenger arrival rate, $E(T_A)$ doesn't involve λ. From (3.10), we can justify that the expected time $E(T_A)$ to get on a bus for strategy A, is directly proportional to $\frac{1}{\mu}$.

We now consider the relationship between the expected time $E(T_A)$ and the number of passengers N in the queue at the beginning.

To study the relationship between $E(T_A)$ and N better, we selected three different groups of p_i:

- $S = 6, p_0 = p_1 = p_2 = p_3 = p_4 = 0, p_5 = 0.8, p_6 = 0.2$;
- $S = 6, p_0 = p_1 = p_2 = p_3 = p_4 = p_5 = 0, p_6 = 1$, an extreme situation, every bus is empty;
- $S = 6, p_0 = p_1 = p_2 = p_3 = 0, p_4 = p_5 = p_6 = \frac{1}{3}$, ;

We have the following graph:

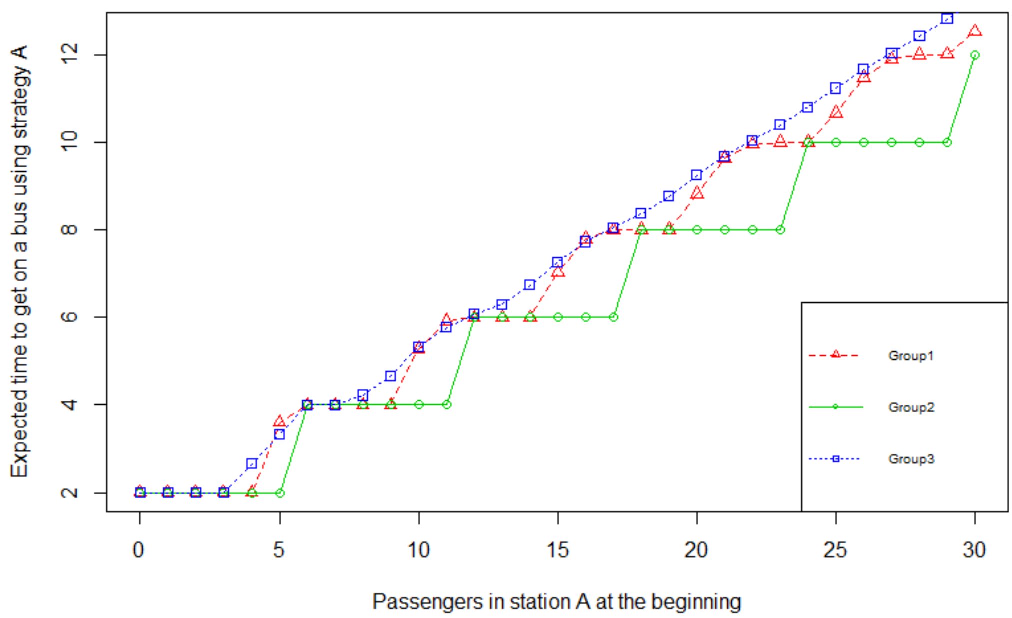

Figure 3.3: expected time for strategy A with different p_i

The graph indicates some interesting properties of how the probability of passengers being served each time will influence the expected time of strategy A as N grows.

- As N increases, the expected time increases. The length of each stage K equals the maximum passengers a bus can serve;
  (i.e. $K = \max_{0 \leqslant l \leqslant S, p_l \neq 0} l$)
- The graph will become smooth gradually and eventually becomes a line when N gets large except when each bus serves a constant number of passengers;
- When p_k is large for k close to K, which means buses tend to be empty or nearly empty, the stages are clear in the graph. Conversely, if p_k is large where k is close to 0, which means the buses tend to be full or nearly full, the stages become unrecognizable, and the expected time $E(T_A)$ will have almost a linear relationship with N.

Under our assumptions, buses arriving at station A will always be full or nearly full, so the expected time $E(T_A)$ to get on a bus for strategy A will most likely have a linear relationship with the number of passengers N in the queue at the beginning.

3.3 Numerical results for Strategy B

From Section 2.3, we calculate the expected time $E(T_B)$ to get on a bus for strategy B theoretically. Recall:

$$E(T_B) = \frac{1}{\mu} + 2t + E(T_{WB}) \tag{3.14}$$

where

$$E(T_{WB}) = \sum_{i=0}^{\infty} \pi_i E(T_{B_i}) \tag{3.15}$$

Then π_i can be solved from

$$\pi_k - \sum_{j=1}^{c} \rho_o^{-1}(1 - q_{j-1}) \cdot \pi_{k+j} \quad = \quad 0, \text{ for } k = 0, 1, 2, ...W \tag{3.16}$$

$$\sum_{i=0}^{W} \pi_i \quad = \quad 1 \tag{3.17}$$

As we mentioned in Section(3.1), let

$$q_k = \sum_{i=0}^{k} p_{B_i} \tag{3.18}$$

$E(T_{B_k})$ can be obtained by

$$E(T_B|k) = \frac{\frac{1}{\mu} + \sum_{i=1}^{k} p_{B_i} \cdot E(T_B|k-i)}{1 - p_{B_0}} \tag{3.19}$$

for $k = 0, 1, 2, ...W$.

Alternatively

$$E(T_{B_k}) = \frac{1}{\mu} \cdot (\sum_{j=1}^{\infty} j(\sum_{i=0}^{k} P(S_{B_{j-1}} = i) - \sum_{i=0}^{k} P(S_{B_j} = i))) \tag{3.20}$$

where

$$P(S_{B_k} = i) \quad = \quad P_B^k[i+1,1] \tag{3.21}$$

and

$$P_B = \begin{bmatrix} p_{B_0} & 0 & 0 & 0 & 0 & \dots & 0 \\ p_{B_1} & p_{B_0} & 0 & 0 & 0 & \dots & 0 \\ p_{B_2} & p_{B_1} & p_{B_0} & 0 & 0 & \dots & 0 \\ p_{B_3} & p_{B_2} & p_{B_1} & p_{B_0} & 0 & \dots & 0 \\ p_{B_4} & p_{B_3} & p_{B_2} & p_{B_1} & p_{B_0} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ p_{B_{W+1}} & p_{B_W} & p_{B_{W-1}} & p_{B_{W-2}} & p_{B_{W-3}} & \dots & p_{B_0} \end{bmatrix} \tag{3.22}$$

We use various parameters to study how the expected time changes when passenger and bus arrival rates change. Here are some parameter choices.

- The distribution of the number of passengers each bus can serve is: $p_0 = 0.1, p_1 = 0.4, p_2 = 0.3, p_3 = p_4 = p_5 = p_6 = 0.05$.
- The time t that a bus will spend from station A to B, or B to A is 3.
- Based on p_i the expected number of passengers a bus can serve is 1.9. Hence

the passenger service rate is 1.9μ. To keep the system stable, we need $\lambda < 1.9\mu$. When we construct the graph, the range for μ is $(0.5, 1.5)$, the range of λ is $(0.2, 0.9)$.

In order to find the expected time to get on a bus for strategy B, we need to use the conclusion in Section 3.1. Obviously, $E(T_B)$ does not depend on N, and the relationship between λ, μ and $E(T_B)$ can be found in following graphs:

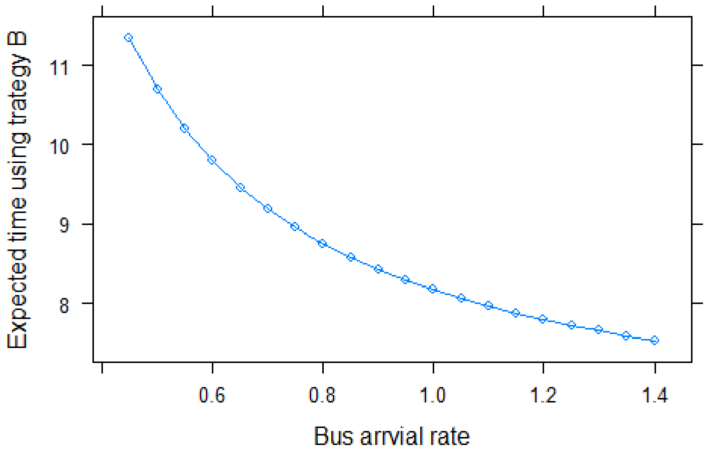

Figure 3.4: Expected time of Strategy B using μ

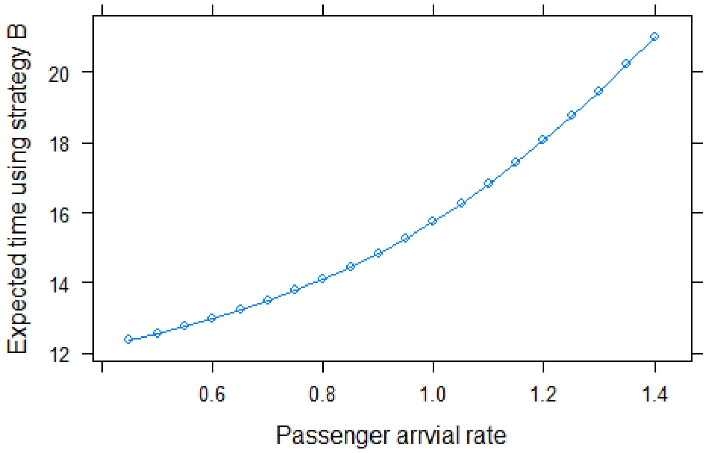

Figure 3.5: Expected time of Strategy B using λ

As can be seen, when μ decreases, $E(T_B)$ falls steadily. When λ increases, $E(T_B)$ increases smoothly. We show the trend more directly from the 3D graph

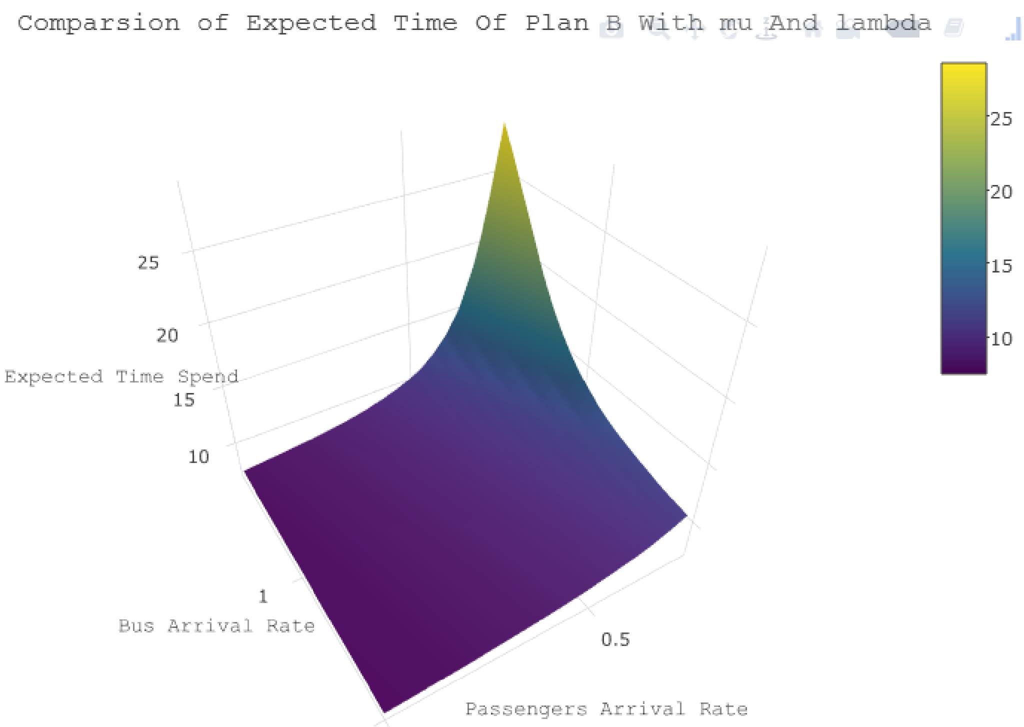

Figure 3.6: Expected time of Strategy B with λ and μ

From graph (3.6) we find that $E(T_B)$ increases dramatically in the neighbourhood of point $(0.5, 0.9)$, at which point the utilization factor ρ is very close to 1. This gives us the idea that the expected time of strategy B is strongly related to the utilization factor ρ.

3.4 Comparison of two strategies

From our previous analysis, the expected time to get on a bus in both strategies depends on many parameters. In practice, we want to set up a criterion to decide our selection of the two strategies. We should consider all parameters.

However, in our model, some of the parameters maybe unknown and not be immediately observable. i.e. the passenger and bus arrivals, the probabilities of the number of passengers who can be served by a bus. We can only use N, where N is the number of passengers at station A at the beginning, as our decision parameter. If we know the other parameters approximately, we can easily use N as a criterion to make our decision. So in this section, we want to fix all other parameters, and compare the expected time of both strategies with different values of N.

Here are our parameter choices.

- The distribution of the number of passengers each bus can serve at the two station is: $p_{A_0} = 0.1, p_{A_1} = 0.4, p_{A_2} = 0.3, p_{A_3} = p_{A_4} = p_{A_5} = p_{A_6} = 0.05$; $p_{B_0} = 0.0, p_{B_1} = 0.4, p_{B_2} = 0.4, p_{B_3} = p_{B_4} = p_{B_5} = p_{B_6} = 0.05$. Most of the time, when we consider going to station B, we assume that the bus at station

B will serve more people than the bus at station A.

- The time t that a bus will spend from station A to B, or B to A is 3.

We make a graph to compare the expected times of the two strategies with N:

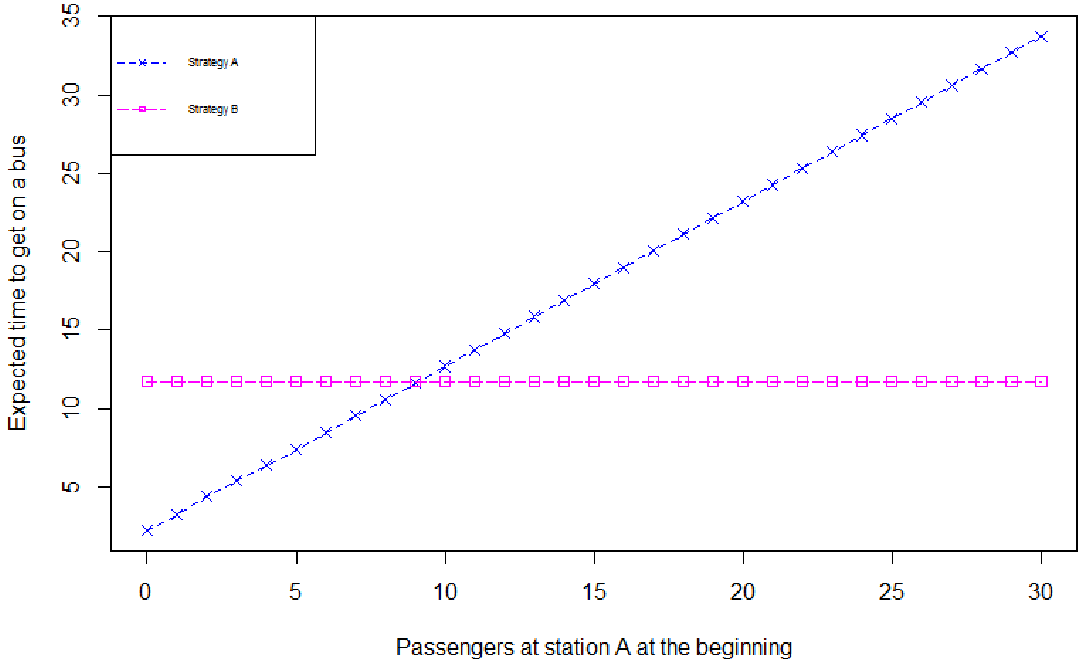

Figure 3.7: Comparison of time of Strategy A and Strategy B

Notice that $E(T_B)$, the expected time for strategy B, does not depend on N. Therefore, the graph of $E(T_B)$ with N will always be a horizontal line. Also, as we mentioned in section 3.2, since the buses arriving at station A are always nearly full, the graph of the expected time $E(T_A)$, will behave like a straight line with N. In this case, if there are more than 10 passengers before us at the beginning, then strategy B is better than strategy A.

Chapter 4

Discussion

We can make further improvements based on our results.

- In our model, we only discuss the $M/M^Y/1$ system, in which we assume the bus interarrivals follows an exponential distribution. In practice, bus interarrivals are more likely to follow a deterministic distribution, which means the bus will come to the station every t time units. We don't need to change much in our results in such an $M/D^Y/1$ system, since we solved for the expected time to get on a bus for both strategies by using the expected number of buses we need, times the expected time to wait for a bus. If we change the bus interarrival times from an exponential distribution to a constant, we just need to change all $\frac{1}{\mu}$ to t, This won't change the result much.

- The passenger arrivals is another interesting topic. According to "Study on Distribution Law of Passenger Arrival of Passenger Service Facilities in Urban Rail Transit Station", by J. Sun, 2014 [4], there are many different predicted distributions for passenger interarrivals: exponential [5], uniform [7], normal [6], deterministic [8] and PH (phase-type) [9]. Each distribution has its own advantages and disadvantages, based on different modes of transportation, such as metro, railway or coach. These distributions might not all fit the passenger interarrival times at a regular bus station, but still it is possible and interesting to discuss a queueing system which has random service with a different passenger interarrival times.

- In section 3.2, we noticed that under some conditions, the expected time to get on a bus for strategy A will have an approximate linear relationship with the passengers number N at station A at the beginning, but we pointed out several special conditions and we did not study them deeply. It should be possible to verify the predicted linear models more generally.

- We chose the distribution of the number of passengers that each bus can serve, because we cannot find any realistic data to tell us what distribution should be used.

- The bus queueing model can be extend to different settings, such as subway or elevator. Some assumptions could change when the settings change. For example, unlike buses, elevators are expected to have different interarrival times at different floors. The interarrival times will be influenced by the algorithm that the elevator uses.

- Another interesting topic was mentioned by Dr. Qi-Ming He at CanQueue 2017.

If we desire that we get a seat on the bus/subway, instead of being concerned with the waiting time, what should the model be? In this case, we have to divide the passengers into two different types. Type A passengers just want to get on as soon as possible, type B passengers will not get on until they can have a seat. We will also assume there are two different maximum capacities, say c_1, c_2, where c_1 denotes the maximum capacity of the bus/subway, c_2 denotes the number of seats in the bus/subway. This will be a much more complex model, but we expect it can be solved.

Appendix A

R Code

Note: The code showing below is calculating one expected number point for specific λ's, β's.

```
#Program I:Limiting probability of M/M^Y/1 system by
   simulation:

library(lattice)

N=15; l=0.5; m=.5 ;t=2500000; s=1; c=7;# l stands for
   passengers interarrivals rate; m stands for bus
   interarrival rate; #c stands for maximum capacity of bus;
   s stands for repeat times.
p1=c(0.1,0.4,0.3,0.05,0.05,0.05,0.05) # p1[i] stands for p(
   a bus can serve i passengers).
p=p1[1]                    # p[i] stands for p(a bus can
   serve less than i passengers)
p[1]=p1[1]
for(i in 2:c)
+ {p=c(p,sum(p1[1:i]))}
ez=matrix(0:0,s);

for(kk in 1:s)
{
```

```
x=c(N)
for (i in 1:t)
{d=length(x); e=runif(1); f=runif(1);

for (j in 1:c)
{
if (f<p[j])
{k=j-1;break}       #k stands for number of passengers
   served in each steps
}
a=l/(l+m);n=(e<a)*(x[d]+1)+(e>a)*max(0,x[d]-k);x=c(x,n)
}
g=max(x[1:t]) #g stands for maximum of queue length during
   simulation
yy=table(factor(x, levels = 0:g))
y<-as.matrix(yy,header=FALSE)
z1=matrix(0:0,g+1)
for(i in 1:(g+1))
{z1[i]=y[i]/sum(y[1:(g+1)])}



zz=matrix(0:0,g) #zz=i*P(L=i)
for(i in 1:g)
{zz[i]=(i-1)*z1[i]}

ez[kk]=sum(zz[1:g]) #ez=sum(i*P(L=i))
}

z1 #z1 stands for the limiting probability at each states
ez #ez stands for expected queue length in system

#Program II:Limiting probability of M/M^Y/1 system by
   truncation (theoretical way)

library('expm')
l=0.5; m=.5 ;t=2; c=7;r=30;# l stands for passengers arrival
    rate; m stands for bus arrival rate;
#c stands for maximum capacity of bus +1;
#b stands for the queue length at the beginning;r stands for
    the upper bounded of state
p1=c(0:0,100)
q=c(0:0,100)
co=c(0:0,100)
```

```
p1=c(0.1,0.4,0.3,0.05,0.05,0.05,0.05) # p1[i] stands for P(L
   =i-1),IF CHANGE P1, NEED TO CHANGE c
p1[(c+1):100]=c(0:0)
rho=m/l

ct=c(0:0,r) #ct stands for b as in Ax=b

for(i in 1:r)
{if (i!=r)
{ ct[i]=0}else
{ct[i]=1}
}

ep=sum((0:(c-1))*p1[1:c]) #ep stands for expected numbers of
    passengers being served each time

for(i in 1:(c+1))

{q[i]=sum(p1[1:i])}

q[(c+2):100]=1           #q[i] stands for p1+p2+p3+...p(i-1)

for(i in 1:(c+2))

{if(i==1)
{co[i]=1}else

{ co[i]=-rho*(1-q[i-1])}


}

co[(c+3):100]=0


Pi=matrix(c(0:0),r,r);

for (i in 1:(r))
{for (j in 1:(r))
{
if(i==r)
{Pi[i,j]=1}else
if(j<i)
{Pi[i,j]=0}else if (j-i<(c+2))
```

```
{Pi[i,j]=co[j-i+1]}else
{Pi[i,j]=0}

}
}


pi=solve(Pi,ct)
pi  #pi stands for the limiting probability of the system


eq=0
for (i in 1:r)
{eq=eq+(i-1)*pi[i]}
eq  #eq stands for expected queue length in system

#Program III: Comparison of two strategies based on the same
    parameters

library('expm')
library('lattice')
l=0.5; m=.5 ;t=3; c=7;r=30;# l stands for passengers arrival
    rate; m stands for bus arrival rate; t stands for the
   time a bus takes from A to B
#c stands for maximum capacity of bus +1;b stands for the
   queue length at the beginning;r stands for the rank of
   the matrix
pb=c(0:0,100)
qb=c(0:0,100)
co=c(0:0,100)
pb=c(0,0.4,0.4,0.05,0.05,0.05,0.05) # p1[i] stands for P(L=i
   -1),IF CHANGE P1, NEED TO CHANGE c
pb[(c+1):100]=c(0:0)
rho=m/l

ct=c(0:0,r) #ct stands for b as in Ax=b

for(i in 1:r)
{if (i!=r)
{ ct[i]=0}else
{ct[i]=1}
}
```

```
ep=sum((0:(c-1))*pb[1:c]) #ep stands for expected numbers of
    passengers being served each time at B

for(i in 1:(c+1))

{qb[i]=sum(pb[1:i])}

qb[(c+2):100]=1           #q[i] stands for p1+p2+p3+...p(i
   -1)

for(i in 1:(c+2))

{if(i==1)
{co[i]=1}else

{ co[i]=-rho*(1-qb[i-1])}


}

co[(c+3):100]=0


Pi=matrix(c(0:0),r,r);

for (i in 1:(r))
{for (j in 1:(r))
{
if(i==r)
{Pi[i,j]=1}else
if(j<i)
{Pi[i,j]=0}else if (j-i<(c+2))
{Pi[i,j]=co[j-i+1]}else
{Pi[i,j]=0}

}
}


pi=solve(Pi,ct)
pi         #pi stands for limiting probability
```

```
ep=sum((0:(c-1))*pa[1:c]) #ep stands for expected numbers of
    passengers being served each time at A




etm=matrix(0:0,r)
etb=0
et=0

for(k in 0:(r-1))
{b=k; x=c(b+1)
;et=0;

PS=matrix(c(0:0),b+2,b+2);
# P stands for the transition matrix
for (i in 1:(b+2))
{for (j in 1:(b+2))
{if(j>i)
{PS[i,j]=0}else if (j<i&&i-j>c-1)
{PS[i,j]=0}else
{PS[i,j]=pb[i-j+1]}

}
}


for(i in 1:(c+1))

{qb[i]=sum(pb[1:i])}

qb[(c+2):100]=1

for (i in 1:100)
{
aa=0
ab=0
ac=0
for(j in 0:b)
{

ab=ab+((PS%^%(i-1))[(j+1),1])*qb[b-j+1]
ac=ac+((PS%^%(i))[(j+1),1])
```

```
aa=aa+((PS%^%(i-1))[(j+1),1])
}

et=et+i*(aa-ab)

}
et=et*1/m

etm[k]=et
etb=etb+et*pi[k+1]

}

etb #etb stands for expected waiting time at station B

ETB=etb+2*t+1/m

ETB #ETB stands for time spend in Plan B

b=10; # l stands for passengers arrival rate; m stands for
   bus arrival rate;
#c stands for maximum capacity of bus +1;b stands for the
   queue length at the beginning
pa=c(0:0,100)
qa=c(0:0,100)
pa=c(0.1,0.4,0.3,0.05,0.05,0.05,0.05) # p1[i] stands for P(L
   =i-1),IF CHANGE P1, NEED TO CHANGE c
pa[(c+1):100]=c(0:0)


n=x[1]
ep=sum((0:(c-1))*pa[1:c]) #ep stands for expected numbers of
    passengers being served each time

eta=matrix(0:0,30)

for(k in 0:30)
{b=k;
P=matrix(c(0:0),b+2,b+2);
PS=matrix(c(0:0),b+2,b+2);
for (i in 1:(b+2))
{for (j in 1:(b+2))
{if(j>i)
{P[i,j]=0}else if (j<i&&i-j>c-1)
```

```
{P[i,j]=0}else if (j==1&&i-j<c)
{P[i,j]=sum(pa[i:c])}else
{P[i,j]=pa[i-j+1]}

}
}
# P stands for the transition matrix
for (i in 1:(b+2))
{for (j in 1:(b+2))
{if(j>i)
{PS[i,j]=0}else if (j<i&&i-j>c-1)
{PS[i,j]=0}else
{PS[i,j]=pa[i-j+1]}

}
}


ea=0


for(i in 1:(c+1))

{qa[i]=sum(pa[1:i])}

qa[(c+2):100]=1

for (i in 1:100)
{
aa=0
ab=0
ac=0
for(j in 0:b)
{

ab=ab+((PS%^%(i-1))[(j+1),1])*qa[b-j+1]
ac=ac+((PS%^%(i))[(j+1),1])
aa=aa+((PS%^%(i-1))[(j+1),1])
}

ea=ea+i*(aa-ab)


}
```

```
ea=1/m*ea
eta[k+1]=ea #ea stands for E(T_A)
}


ETA=ea

X=matrix(0:0,31)
for (i in 1:31)
{X[i]=i-1;}
Y=matrix(0:0,31)
for(i in 1:31)
{Y[i]=eta[i]}
Z=matrix(0:0,31)
for(i in 1:31)
{Z[i]=ETB}
plot(Y~X,xlab="Passengers␣at␣station␣A␣at␣the␣beginning",
   ylab="Expected␣time␣to␣get␣on␣a␣bus",
type="o",lty=2,pch=4,col=4)
lines(Z~X,type="o",lty=5,pch=0,col=6)
legend("topleft",legend=expression("Strategy␣A","Strategy␣B"
   ),col=c(4,6),lty=c(2,5),pch=c(4,0),cex=0.5)



#Program IV: 3D Graph of Expected time spend using Strategy
   with both mu and lambda

library('expm')
library('lattice')
library('plot3D')
library("emdbook")
library('shape')
library('rgl')
library('plotly')
lambda=0.5; mu=.5 ;t=3; c=7;r=31;# lambda stands for
   passengers arrival rate; mu stands for bus arrival rate;
   t stands for the time a bus takes from A to B
#c stands for maximum capacity of bus +1;b stands for the
   queue length at the beginning;r stands for the rank of
   the matrix
pb=c(0:0,100)
qb=c(0:0,100)
co=c(0:0,100)
```

```
pb=c(0.1,0.4,0.3,0.05,0.05,0.05,0.05) # p1[i] stands for P(L
   =i-1),IF CHANGE P1, NEED TO CHANGE c
pb[(c+1):100]=c(0:0)
Ep=sum(c(0:(c-1))*pb[1:c])
Etbk=matrix(0:0,20,20)
etb=matrix(0:0,100)


for(k in 1:20)
{
mu=0.45+0.05*k;
for(l in 1:20)
{lambda=0.2+0.04*l

rho=mu/lambda


ct=c(0:0,r) #ct stands for b as in Ax=b

for(i in 1:r)
{if (i!=r)
{ ct[i]=0}else
{ct[i]=1}
}


for(i in 1:(c+1))

{qb[i]=sum(pb[1:i])}

qb[(c+2):100]=1          #q[i] stands for p1+p2+p3+...p(i
   -1)

for(i in 1:(c+2))

{if(i==1)
{co[i]=1}else

{ co[i]=-rho*(1-qb[i-1])}


}

co[(c+3):100]=0
```

```
Pi=matrix(c(0:0),r,r);

for (i in 1:(r))
{for (j in 1:(r))
{
if(i==r)
{Pi[i,j]=1}else
if(j<i)
{Pi[i,j]=0}else if (j-i<(c+2))
{Pi[i,j]=co[j-i+1]}else
{Pi[i,j]=0}

}
}


pi=solve(Pi,ct)

etb=matrix((1/mu):(1/mu),31)

for(i in 0:30)
{if(i==0)
{etb[i+1]=1/(mu*(1-pb[1]))}else
{etb[i+1]=(1/mu+sum(pb[2:(i+1)]*etb[i:1]))/(1-pb[1])

}

}

Etbk[k,l]=1/mu+2*t+sum(pi[1:31]*etb[1:31])


}  #lambda


} #mu




i=1:20
```

```
j=1:20
x<-0.45+i*0.05
y<-0.2+j*0.04
z<-Etbk[i,j] #z stands for E(T_B|mu,lambda)

f <- list(
family = "Courier␣New,␣monospace",
size = 14
)
g <- list(
family = "Courier␣New,␣monospace",
size = 18

)


require(latex2exp)

p<-plot_ly(x = x, y = y, z = z, type = "surface")%>%
layout(
title = "Comparison␣of␣Expected␣Time␣Of␣Plan␣B␣With␣mu␣And␣
   lambda",titlefont=g,
scene = list(
xaxis = list(title = "Bus␣Arrival␣Rate",titlefont=f),
yaxis = list(title = "Passengers␣Arrival␣Rate",titlefont=f),
zaxis = list(title = "Expected␣Time␣Spend",titlefont=f)
))
p


#Program V: Graph of Expected time spent using strategy A
   with different groups of p_i

library('expm')
library('lattice')
library('ggplot2')
l=0.5; m=.5 ;t=3; c=7;r=30;# l stands for passengers arrival
    rate; m stands for bus arrival rate; t stands for the
   time a bus takes from A to B
#c stands for maximum capacity of bus +1;b stands for the
   queue length at the beginning;r stands for the rank of
   the matrix
```

```
b=10; # l stands for passengers arrival rate; m stands for
   bus arrvial rate;
#c stands for maximum capacity of bus +1;b stands for the
   queue length at the begining

pa=c(0:0,100)
qa1=c(0:0,100)
pa1=c(0,0,0,0,0,0.8,0.2) # Group 1 p_i
pa1[(c+1):100]=c(0:0)


n=x[1]
ep1=sum((0:(c-1))*pa1[1:c]) #ep stands for expected numbers
   of passengers being served each time


eta1=matrix((1/m):(1/m),31)

for(i in 0:30)
{if(i==0)
{eta1[i+1]=1/(m*(1-pa1[1]))}else
{eta1[i+1]=(1/m+sum(pa1[2:(i+1)]*eta1[i:1]))/(1-pa1[1])

}

}

#
   ----------------------------------------------------------------


pa=c(0:0,100)
qa2=c(0:0,100)
pa2=c(0,0,0,0,0,0,1.0) # Group 2 p_i
pa2[(c+1):100]=c(0:0)


n=x[1]
ep2=sum((0:(c-1))*pa2[1:c]) #ep stands for expected numbers
   of passengers being served each time


eta2=matrix((1/m):(1/m),31)
```

```
for(i in 0:30)
{if(i==0)
{eta2[i+1]=1/(m*(1-pa2[1]))}else
{eta2[i+1]=(1/m+sum(pa2[2:(i+1)]*eta2[i:1]))/(1-pa2[1])

}

}

#
   ------------------------------------------------------------------------


pa=c(0:0,100)
qa3=c(0:0,100)
pa3=c(0,0,0,0,1/3,1/3,1/3) # Group 3 p_i
pa3[(c+1):100]=c(0:0)


n=x[1]
ep3=sum((0:(c-1))*pa3[1:c]) #ep stands for expected numbers
   of passengers being served each time


eta3=matrix((1/m):(1/m),31)

for(i in 0:30)
{if(i==0)
{eta3[i+1]=1/(m*(1-pa3[1]))}else
{eta3[i+1]=(1/m+sum(pa3[2:(i+1)]*eta3[i:1]))/(1-pa3[1])

}

}


X=matrix(0:0,31)
for (i in 1:31)
{X[i]=i-1;}
Y=matrix(0:0,31)
for(i in 1:31)
{Y[i]=eta1[i]}
Z=matrix(0:0,31)
```

```
for(i in 1:31)
{Z[i]=eta2[i]}
U=matrix(0:0,31)
for(i in 1:31)
{U[i]=eta3[i]}

plot(Y~X,xlab="Passengers␣in␣station␣A␣at␣the␣beginning",
   ylab="Expected␣time␣to␣get␣on␣a␣bus␣using␣strategy␣A",
   type="o",lwd=1,lty=2,col=2,pch=2)
lines(Z~X,lwd=1,type="o",lty=1,col=3,pch=1)
lines(U~X,lwd=1,type="o",lty=3,col=4,pch=0)
legend("bottomright",legend = c(expression("Group1"),
expression(paste("Group2")),
expression(paste("Group3"))),col=c(2,3,4),lty=c(2,1,3),pch=c
   (2,1,0),lwd=1,cex=0.6)


#Program VI: Graph of Limiting probability with different
   lambda values

library('expm')
library('lattice')
l=0.8; m=.5 ;t=3; c=7;r=30;# l stands for passengers arrival
    rate; m stands for bus arrival rate; t stands for the
   time a bus takes from A to B
#c stands for maximum capacity of bus +1;b stands for the
   queue length at the beginning;r stands for the rank of
   the matrix
pb=c(0:0,100)
qb=c(0:0,100)
co=c(0:0,100)
pb=c(0.1,0.3,0.4,0.05,0.05,0.05,0.05) # p1[i] stands for P(L
   =i-1),IF CHANGE P1, NEED TO CHANGE c
pb[(c+1):100]=c(0:0)
rho=m/l

ct=c(0:0,r) #ct stands for b as in Ax=b

for(i in 1:r)
{if (i!=r)
{ ct[i]=0}else
{ct[i]=1}
}
```

```
ep=sum((0:(c-1))*pb[1:c]) #ep stands for expected numbers of
    passengers being served each time at B

for(i in 1:(c+1))

{qb[i]=sum(pb[1:i])}

qb[(c+2):100]=1           #q[i] stands for p1+p2+p3+...p(i
   -1)

for(i in 1:(c+2))

{if(i==1)
{co[i]=1}else

{ co[i]=-rho*(1-qb[i-1])}


}

co[(c+3):100]=0


Pi=matrix(c(0:0),r,r);

for (i in 1:(r))
{for (j in 1:(r))
{
if(i==r)
{Pi[i,j]=1}else
if(j<i)
{Pi[i,j]=0}else if (j-i<(c+2))
{Pi[i,j]=co[j-i+1]}else
{Pi[i,j]=0}

}
}


pi=solve(Pi,ct)
pi         #pi stands for limiting probability

#Second curve
```

```
l=0.6
pb=c(0:0,100)
qb=c(0:0,100)
co=c(0:0,100)
pb=c(0.1,0.3,0.4,0.05,0.05,0.05,0.05) # p1[i] stands for P(L
   =i-1),IF CHANGE P1, NEED TO CHANGE c
pb[(c+1):100]=c(0:0)
rho=m/l

ct=c(0:0,r) #ct stands for b as in Ax=b

for(i in 1:r)
{if (i!=r)
{ ct[i]=0}else
{ct[i]=1}
}

ep=sum((0:(c-1))*pb[1:c]) #ep stands for expected numbers of
    passengers being served each time at B

for(i in 1:(c+1))

{qb[i]=sum(pb[1:i])}

qb[(c+2):100]=1          #q[i] stands for p1+p2+p3+...p(i
   -1)

for(i in 1:(c+2))

{if(i==1)
{co[i]=1}else

{ co[i]=-rho*(1-qb[i-1])}


}

co[(c+3):100]=0


Pj=matrix(c(0:0),r,r);

for (i in 1:(r))
{for (j in 1:(r))
```

```
{
if(i==r)
{Pj[i,j]=1}else
if(j<i)
{Pj[i,j]=0}else if (j-i<(c+2))
{Pj[i,j]=co[j-i+1]}else
{Pj[i,j]=0}

}
}


pj=solve(Pj,ct)
pj


##Third Curve
l=0.4
pb=c(0:0,100)
qb=c(0:0,100)
co=c(0:0,100)
pb=c(0.1,0.3,0.4,0.05,0.05,0.05,0.05) # p1[i] stands for P(L
   =i-1),IF CHANGE P1, NEED TO CHANGE c
pb[(c+1):100]=c(0:0)
rho=m/l

ct=c(0:0,r) #ct stands for b as in Ax=b

for(i in 1:r)
{if (i!=r)
{ ct[i]=0}else
{ct[i]=1}
}

ep=sum((0:(c-1))*pb[1:c]) #ep stands for expected numbers of
    passengers being served each time at B

for(i in 1:(c+1))

{qb[i]=sum(pb[1:i])}

qb[(c+2):100]=1          #q[i] stands for p1+p2+p3+...p(i
   -1)
```

```
for(i in 1:(c+2))

{if(i==1)
{co[i]=1}else

{ co[i]=-rho*(1-qb[i-1])}


}

co[(c+3):100]=0

pk=matrix(0:0,31)
Pk=matrix(0:0,r,r);

for (i in 1:(r))
{for (j in 1:(r))
{
if(i==r)
{Pk[i,j]=1}else
if(j<i)
{Pk[i,j]=0}else if (j-i<(c+2))
{Pk[i,j]=co[j-i+1]}else
{Pk[i,j]=0}

}
}


pk=solve(Pk,ct)
pk


X=matrix(0:0,21)
for (i in 1:21)
{X[i]=i-1;}



plot(pk[1:21]~X,xlab="states",ylab="limiting␣probability",
   type="o",lty=2,col=2,pch=2,lwd=1.5)
lines(pj[1:21]~X,lty=1,type="o",col=3,pch=1,lwd=1.5)
lines(pi[1:21]~X,lty=4,type="o",col=4,pch=0,lwd=1.5)
```

```
title(expression(paste("Limiting␣probability␣of␣M/M^Y/1␣
   system␣base␣on␣different␣",lambda)),cex.main=1.0)
legend("topright",legend = c(expression(paste(lambda, "␣=␣",
    0.4)),
expression(paste(lambda, "␣=␣", 0.6)),
expression(paste(lambda, "␣=␣", 0.8))),col=c(2,3,4),lty=c
   (2,1,4),pch=c(2,1,0),lwd=1.5,cex=0.8)
```

Bibliography

[1] I. Adan, J. Resing. *Queueing Systems*, 2016
retrieved from:
http://www.win.tue.nl/ iadan/queueing.pdf

[2] J.W. Cohen. *The Single Server Queue*, North Holland, 1982

[3] J. Sztrik. *Basic Queueing Theory:Foundations of System Performance Modeling*, GlobeEdit, 2016

[4] J. Sun. *Study on Distribution Law of Passenger Arrival of Passenger Service Facilities in Urban Rail Transit Station*, 2014

[5] L. Gunnar. *Modeling and Simulation of Pdestrian Traffic Flow[J].* Transportation Reasearch Part B: Methodological, 2014, 28(6), 429-443

[6] X. Hu. *Modeling and System of Pedestrian Flow in Rail Hubs*, 2010

[7] D. Sims. *See how they run:modeling evacuations in VR[J].* Computer Graphics and Application, 1995, 15(2), 11-13

[8] M. Zhong, C. Shi. *Study of the human evacuation simulation of metro fire safety analysis in China[J].* Journal of Loss Prevention in the Process Industries, 2008, 21(3), 287-298

[9] L. Hu, Y. Pu. *Fitting for the distribution of interval of passengers arriving at urban rail transit station* Systems Engineering- Theory & Practice, 2014, 34(7), 1835-1846

Vita Auctoris

Minjian Yuan was born in China in 1989. He received his Bachelor of Science (Honours) in Statistics from University of Windsor in 2016. He is currently a candidate for the Master of Science degree in Statistics at the University of Windsor and expects to graduate at the end of 2017.