

**DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES**

**STATISTICAL EVALUATION OF
PERFORMANCE MEASURES IN BATCH
QUEUEING SYSTEMS BY SIMULATION**

by
Şerife ÖZKAR

**December, 2011
İZMİR**

**STATISTICAL EVALUATION OF
PERFORMANCE MEASURES IN BATCH
QUEUEING SYSTEMS BY SIMULATION**

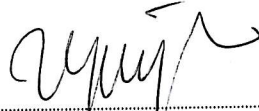
**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for
the Degree of Master of Science in Statistics**

**by
Şerife ÖZKAR**

**December, 2011
İZMİR**

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “STATISTICAL EVALUATION OF PERFORMANCE MEASURES IN BATCH QUEUEING SYSTEMS BY SIMULATION” completed by ŞERİFE ÖZKAR under supervision of YRD. DOÇ. DR. UMAY UZUNOĞLU KOÇER and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Yrd. Doç. Dr. Umay UZUNOĞLU KOÇER

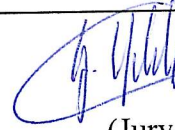
Supervisor



Prof. Dr. C. Cengiz ÇELİKOĞLU

(Jury Member)

Yrd. Doç. Dr. Gökalg YILDIZ



(Jury Member)



Prof. Dr. Mustafa SABUNCU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I am grateful to Yrd. Doç. Dr. Umay UZUNOĞLU KOÇER, my supervisor, for her encouragement and insight throughout my research and for guiding me through the entire thesis process from start to finish. Has it not been for her faith in me, I would not have been able to finish the thesis.

I'm also grateful for the insights and efforts put forth by the examining committee; Prof. Dr. Cengiz ÇELİKOĞLU and Yrd. Doç. Dr. Gökâl YILDIZ.

I want to take the opportunity to thank Yalçın ÇETİNKAYA, my friend, who shared his sweet home with me, had a great contribution on my thesis and gave me hope when I felt desperate.

I wish to give a heartfelt thanks to my mother, Havva ÖZKAR, and my father, İsmail ÖZKAR. They offered their endless support and constant prayers. This academic journey would not have been possible without their love, patience, and sacrifices along the way.

Şerife ÖZKAR

STATISTICAL EVALUATION OF PERFORMANCE MEASURES IN BATCH QUEUEING SYSTEMS BY SIMULATION

ABSTRACT

In this study two different queuing systems have been considered, such that; the batch arrival queues with fixed batch size and the batch service queues. The first aim of the study is to compare the numerical results to the simulation results and to make statistical precise statement on the accuracy of the performance measures. Therefore, the mathematical formulas related with the performance measures have been provided and the simulation programs have been written by using MATLAB 7.0 for the corresponding queuing systems. Repeating runs of the simulation, a finite random sample has been obtained, the performance measures have been attained by point estimate, and then, statistical precise statements on the accuracy of these estimates have been constructed by confidence interval estimate. The second aim is to show whether the statistical precision of these estimates is affected by the batch size. For this reason, the simulation programs have been repeated plenty of times for different batch size values, and also tables and graphs have been built by using the results of the simulation programs.

Keywords : batch arrival queues with fixed batch size, batch service queue, point estimate, confidence interval estimate.

YIĞIN KUYRUK SİSTEMLERİNDE PERFORMANS ÖLÇÜTLERİNİN BENZETİM YOLU İLE İSTATİSTİKSEL DEĞERLENDİRMESİ

ÖZ

Bu çalışmada iki farklı kuyruk sistemi incelenmiştir; yığın boyutu sabit olan yığın varışlı kuyruklar ve hizmetin yığın olarak alındığı kuyruklar. Çalışmanın birinci amacı sayısal sonuçlar ile benzetim sonuçlarını karşılaştırmak ve bulunan benzetim sonuçlarının doğruluğu hakkında istatistiksel olarak kesin bir ifade kullanmaktır. Bu amaç doğrultusunda performans ölçütlerinin matematiksel formülleri elde edildi ve MATLAB 7.0 kullanılarak simulasyon programları yazıldı. Programlar bir çok kez çalıştırılarak sonlu rastgele bir örneklem elde edildi, performans ölçümlerinin nokta tahminlerini yapıldı ve daha sonra güven aralığı tahminlerini kullanılarak bu ölçümlerin tamlığı istatistiksel olarak ifade edildi. Bu çalışmanın ikinci amacı yığın boyutunun tahminlerin istatistiksel tamlığı üzerinde etkisi olup olmadığını göstermektir. Bu amaç doğrultusunda farklı yığın boyutları için simulasyon programları birçok kez çalıştırıldı ve elde edilen sonuçlar kullanarak tablolar ve grafikler elde edildi.

Anahtar sözcükler : yığın boyutu sabit olan yığın varışlı kuyruklar, hizmetin yığın olarak alındığı kuyruklar, nokta tahmini, güven aralığı tahmini.

CONTENTS

	Page
M.Sc. THESIS EXAMINATION RESULT FORM.....	ii
ACKNOWLEDGMENTS	iii
ABSTRACT.....	iv
ÖZ	v
CHAPTER ONE - INTRODUCTION	1
1.1 Where do queue occur?	1
1.1.1 Commercial service systems.....	1
1.1.2 Transportation service systems.....	1
1.1.3 Internal service systems	2
1.1.4 Social service systems	2
1.2 Why do queues occur?.....	2
1.3 Why do management of queues is important?	5
1.4 Basic structure of queuing models	6
1.4.1 The input source or calling population	7
1.4.2 Queue discipline	8
1.4.3 System capacity or queue	8
1.4.4 Service mechanism	9
1.4.5 Notation	10
CHAPTER TWO - STOCHASTIC PROCESSES AND STEADY-STATE SOLUTION.....	12
2.1 Stochastic processes	12
2.1.1 Counting Process	14
2.1.2 Poisson Process.....	14

2.1.2.1	The number of arrivals to a system at time t	15
2.1.2.2	The time between successive arrivals (interarrival time).....	18
2.1.2.3	The arrival time of the n th event	19
2.1.3	Birth-and-death process	20
2.2	Steady-state solution.....	22
2.2.1	Methods of solving steady-state difference equations.....	23
2.2.1.1	Iterative method	23
2.2.1.2	Solution by generating functions	24
 CHAPTER THREE - QUEUING MODELS		27
3.1	The method of stages.....	27
3.2	Basic queueing model and Erlangian queueing models.....	30
3.2.1	M/M/1 model.....	30
3.2.2	M/E _r /1 model	32
3.2.3	E _r /M/1 model	33
3.3	The Little's Formula	34
3.3.1	Relationships among L , W , L_q , and W_q	37
3.4	Batch queue models.....	38
3.4.1	Batch arrival systems - M ^x /M/1 model.....	38
3.4.1.1	Generation function, $E(X)$, and $V(X)$	39
3.4.1.2	Waiting times	42
3.4.2	Batch arrival systems with fixed batch size- M ^f /M/1 model.....	43
3.4.2.1	Generating function, $E(X)$, and $V(X)$	44
3.4.2.2	Waiting times	46
3.4.2.3	The performance measures	47
3.4.3	Batch service systems- M/M ^x /1	47
3.4.3.1	Batch service systems- in policy (1)	48
3.4.3.2	The performance measures	52

CHAPTER FOUR - SIMULATION	53
4.1 Introduction to simulation	53
4.2 Discrete event simulation	55
4.2.1 The performance measures	57
4.2.2 Statistical analyses of simulation output.....	58
CHAPTER FIVE - APPLICATION	61
5.1 Simulation model of queuing systems with batch arrival	61
5.1.1 Determination of run length and number of replications.....	77
5.1.2 Comparison between the results of simulation and the analytic results	83
5.1.3 Impact of batch size over the performance measures	87
5.2 Simulation model of queuing systems with batch service	90
5.2.1 Determination of run length and number of replications.....	107
5.2.2 Comparison between the results of simulation and the analytic results ..	110
5.2.3 Impact of batch size over the performance measures	115
CHAPTER SIX - CONCLUSION	118
REFERENCES.....	121
APPENDIX 1 - STATISTICAL DISTRIBUTIONS, GENERATING FUNCTIONS, TRANSFORMS, AND STATISTICAL INFERENCE.....	124
A1.1 Statistical distributions	124
A1.1.1 Poisson Distribution.....	124
A1.1.2 Exponential Distribution.....	124
A1.1.3 Erlang Distribution.....	125
A1.1.4 Gamma Distribution.....	125
A1.2 Generating functions	126

A1.2.1 Generating function	126
A1.2.2 Moment generating function.....	126
A1.2.3 Probability generating function.....	127
A1.3 Transforms.....	127
A1.3.1 Laplace transform	127
A1.3.2 z-transform.....	128
A1.4 Statistical inference	129
A1.4.1 Point estimation	130
A1.4.2 Interval estimation	131
APPENDIX 2 - ANALITICAL SOLUTION FOR QUEUE MODELS	132
A2.1 Analytical solution for queue model with batch arrival	132
A2.2 Analytical solution for queue model with batch service	135
APPENDIX 3 - SIMULATION CODE FOR THE QUEUE MODELS	139
A3.1 Simulation code for the queue model with batch arrival.....	139
A3.2 Simulation code for the queue model with batch service.....	141

CHAPTER ONE

INTRODUCTION

All of us have spent a great deal of time waiting lines or queues. A queue is formed whenever the demand for service exceeds the capacity to provide service at that point in time. In this chapter, we begin by explaining some questions like "Where do queue occur?", "Why do queues occur?", and "Why do management of queues is important?" And then, in Section 1.4, we continue by discussing basic structure of queuing models.

1.1 Where do queue occur?

Queuing systems are common in a wide variety of contexts. We can give various examples of real queuing systems;

1. Commercial service systems,
2. Transportation service systems,
3. Internal service systems,
4. Social service systems.

1.1.1 Commercial service systems

The queuing systems that we encounter in our daily lives are commercial service systems (Hiller, & Lieberman, 2001). In this type of systems, outside customers receive service from commercial organizations. Many of these include person-to-person service at a fixed location. For example, a barber shop, bank teller service, and a cafeteria line.

1.1.2 Transportation service systems

For some of the transportation service systems, the vehicles are the customers, such as cars waiting at tollbooth or traffic light, and airplanes waiting to land or take off from a runway (Hiller, & Lieberman, 2001). Queues can be built up around parking areas with limited capacity such as parking garages for cars, piers in harbors for ships. In addition, queues are also built up at ticket-offices and check-in counters. Passengers usually have to wait for the departure of their train or bus (Blanc, 2011).

1.1.3 Internal service systems

Internal service systems are commonly encountered. In these systems, the customers receiving service are *internal* to the organization (Hiller, & Lieberman, 2001). Examples include *materials-handling systems*, where materials-handling units (the servers) move loads (the customers); *maintenance systems*, where maintenance crews (the servers) repair machines (the customers); and *inspection stations*, where quality control inspectors (the servers) inspect items (the customers).

1.1.4 Social service systems

Another system encountered in the queuing systems is social service systems. Examples include *judicial systems*, where the courts are service facilities, the judges are the servers, and the cases waiting to be tried are the customers; *health-care systems*, where the hospitals are service facilities, the doctors or the nurses are the servers, and the patients are the customers. On the other hand, we can also view ambulances, x-ray machines, and hospital beds as the servers (Hiller, & Lieberman, 2001).

1.2 Why do queues occur?

Figure 1.1, 1.2, and 1.3 concern deterministic service systems with constant interarrival times of customers, constant service times (the service times are 3 time units in all three figures), a single server and an unlimited waiting line. The figures show the number of customers present in the system represented by $N(t)$, as a function of time. Arrival instants are indicated by “A”, departure instants by “D”. The first customer arrives at time 1 and the service time of the customer is 3 time units. In Figure 1.1, the interarrival times are 4 time units. If the interarrival times are larger than the service times, then the preceding customer has already left the system when a new customer arrives, and no queuing ever occurs.

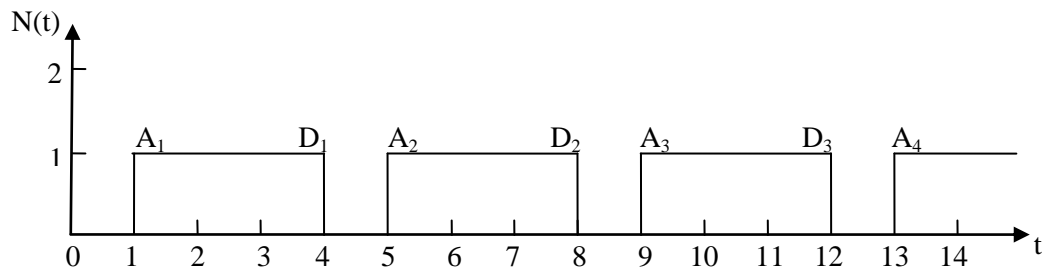


Figure 1.1 A deterministic, underloaded single-server queuing system.

Figure 1.2 concerns the case that the interarrival times are equal to the service times. In this system the preceding customer departs from the system precisely at the arrival instant of a new customer. The server is continuously busy, but no queuing occurs.

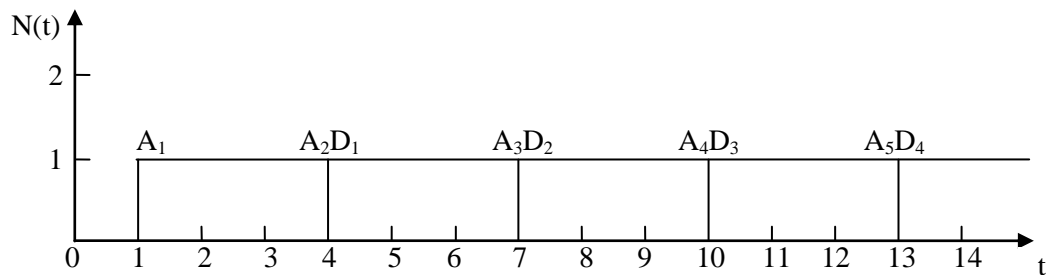


Figure 1.2 A deterministic, critically loaded single-server queuing system.

In Figure 1.3, the interarrival times are 2 time units. In this system, we see a queue gradually being built up. Because the service capacity is too small to handle all requests, this queue will grow without bound.

These simple examples illustrate the important concept of *stability*. A service system is said to be stable if it can handle all admitted service requests in the long run (Blanc, 2011). A deterministic service system with a single server is stable if the constant service time is smaller than or equal to the constant interarrival time (Blanc, 2011). Stability is the most relevant characteristics for systems with unlimited waiting line.

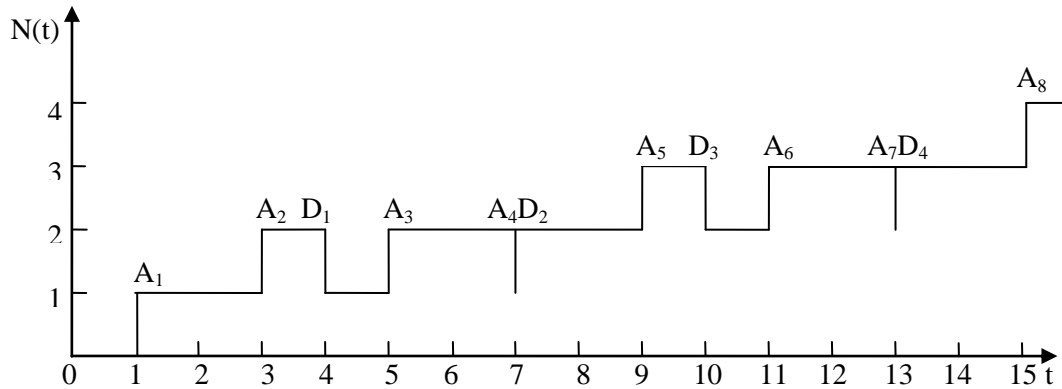


Figure 1.3 A deterministic, overloaded single-server queuing system.

The example illustrated in Figure 1.4. is the periodic deterministic single-server system. Three customers arrive in each cycle of 15 time units. The first customer arrives at time 1 and requires 10 time units of service. The second customer arrives at time 2 and requires 1 time unit of service. The second customer has to join to queue and wait 9 time units before the server becomes idle, because the service of the first customer is only completed at time 11. The service of the second customer is completed at time 12. The third customer arrives at time 13, does not have to wait and after a service of 1 time unit she leaves at time 14, before the first customer of the next cycle arrives. Clearly, this system is stable.

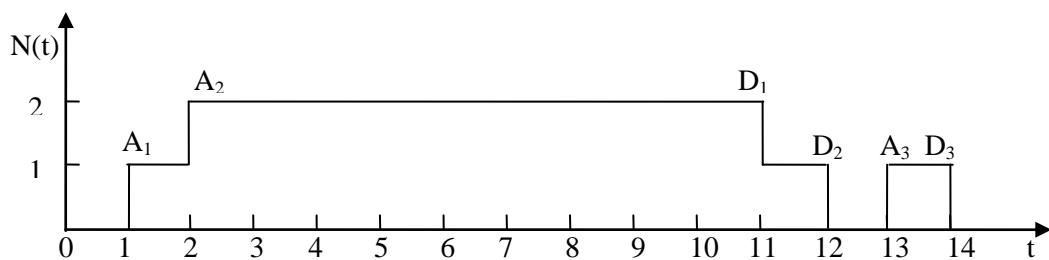


Figure 1.4 A periodic deterministic single-server queuing system.

Figure 1.5 shows the number of customers present in a single-server system with interarrival times of 6 time units for batches of four customers and service times of 1 time unit per individual customer. A queue is formed at an arrival instant and then gradually disappears when customers have been served in this stable system.

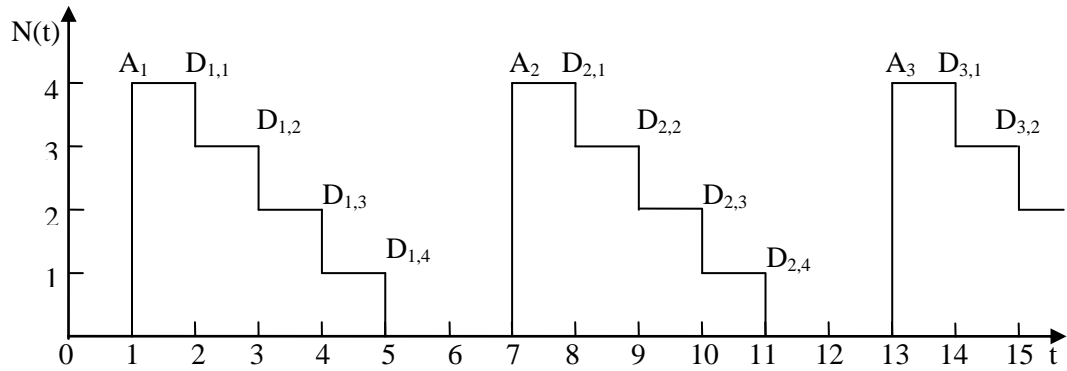


Figure 1.5 A deterministic, underloaded single-server queuing system with batch arrival.

Figure 1.6 shows the number of customers present in a single-server system with interarrival times of 1 time unit for individual customers and service times of 3 time units per batch of three customers. A queue of three customers has to form before a service can start. The queue continues to grow during a service until three customers simultaneously depart at a service completion instant. Observe that also this stable system never becomes empty again after the first customer has entered the system.

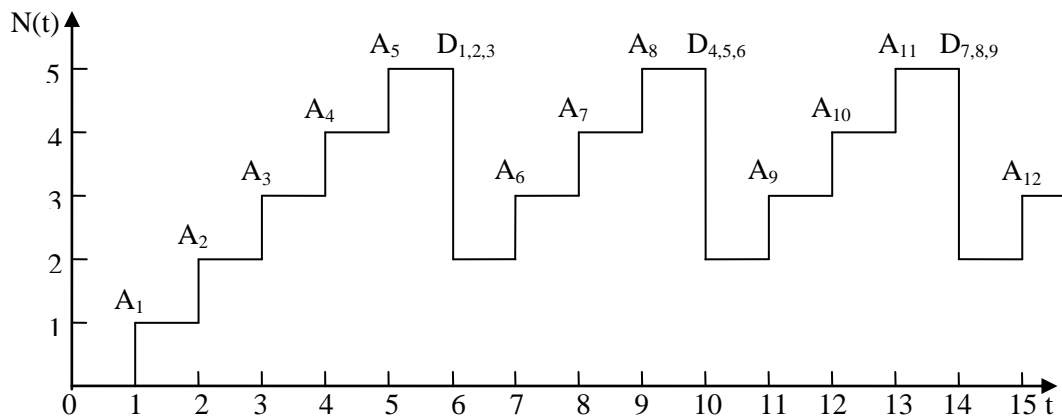


Figure 1.6 A deterministic, underloaded single-server queuing system with batch service.

1.3 Why do management of queues is important?

We describe some problem situations in which management of queues is important in Table 1.1 (Adan, & Resing, 2002).

Table 1.1 Queuing situations

Place	Problems
Supermarket.	<p>How long do customers have to wait at the checkouts?</p> <p>What happens with the waiting time during peak-hours?</p> <p>Are there enough checkouts?</p>
Production system, in which a machine produces different types of products.	<p>What is the production lead time of an order?</p> <p>What is the reduction in the lead time when we have an extra machine?</p> <p>Should we assign priorities to the orders?</p>
Post office, where there are counters specialized in e.g. stamps, packages.	<p>Are there enough counters?</p> <p>Separate queues or one common queue in front of counters with the same specialization?</p>
<p>Parking place.</p> <p>Managers are going to make a new parking place in front of a super market.</p>	<p>How large should it be?</p>
<p>Call centers of an insurance company.</p> <p>Questions by phone, regarding insurance conditions, are handled by a call center. This call center has a team structure, where each team helps customers from a specific region only</p>	<p>How long do customers have to wait before an operator becomes available?</p> <p>Is the number of incoming telephone lines enough?</p> <p>Are there enough operators?</p> <p>Pooling teams?</p>
Traffic lights.	<p>How do we have to regulate traffic lights such that the waiting times are acceptable?</p>

1.4 Basic structure of queuing models

A queuing model can be described as the following (Hiller, & Lieberman, 2001): *Customers* requiring service are generated by an *input source*. These customers enter the *queuing system* and join a *queue*. A member of the queue is selected for service by some rule known as the *queue discipline*. The service is performed for the customer by the *service mechanism*. Then the customer leaves the queuing system. This process is shown in Figure 1.7.

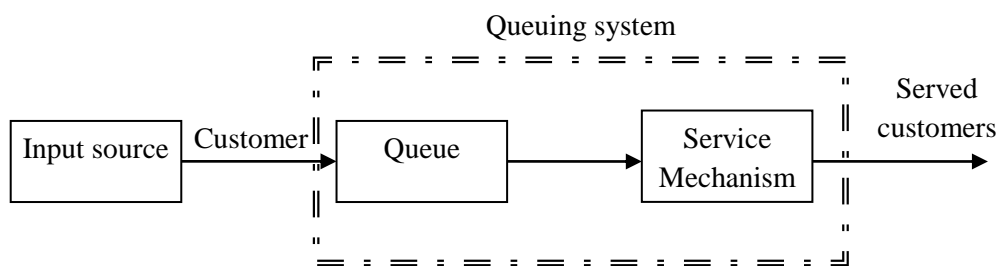


Figure 1.7 The basic queuing process.

The customers and servers are the key elements of a queuing system. The *customer* can refer to people, machines, truck, patients, airplanes, or e-mail, namely, anything that arrives at a facility and requires service. The server can refer to receptionists, repairpersons, runways at an airport, CPUs in a computer, namely, any resource which provides the requested service. The number of different systems is listed in Table 1.2 (Banks, Carson, Nelson, & Nicol, 2001).

Table 1.2 Examples of queuing systems

System	Customers	Server(s)
Airport	Airplanes	Runway
Road network	Cars	Traffic light
Computer	Jobs	CPU, disk
Telephone	Calls	Exchange
Repair facility	Machines	Repairperson
Hospital	Patients	Nurses, doctors
Laundry	Dirty linen	Washing machines/dryers
Garage	Trucks	Mechanic

1.4.1 The input source or calling population

The arrival pattern or input to a queuing system is often measured in terms of the average number of arrivals per some unit of time or by the average time between successive arrivals, and in the event that the stream of input is deterministic, then the arrival pattern is determined by either the *mean arrival rate* or the *mean interarrival time*. If there is uncertainty in the arrival pattern (referred to as random or stochastic), then these mean values provide only measures of central tendency for the input process (Gross, & Harris, 1974). Arrivals can occur in batches. In the event that more than one arrival can enter the system simultaneously, the input is said to occur in batch or batches (Gross, & Harris, 1974).

It is necessary to know the reaction of a customer upon entering the system. A customer may decide to wait no matter how long the queue becomes, or if the queue is too long to suit him, may decide not to enter it. If a customer decides not to enter the queue upon arrival, he is said to have *balked*. A customer may enter to queue, but after a time lose patience and decide to leave. In this case he is said to have *renege*d. In the event that there are two or more parallel waiting lines, customers may switch from one to another, that is, *jockey* for position (Gross, & Harris, 1974).

1.4.2 Queue discipline

Queue discipline refers to the manner by which customers are selected for service. The most common discipline is first come, first served (FCFS). Some others in common usage are last come, first served (LCFS), which is applicable to many inventory systems when there is no obsolescence of stored units as it is easier to reach the nearest items which are last in; selection for service in random order independent of the time of arrival to the queue (RSS); and a variety of priority schemes, where customers are given priorities upon entering the system, the ones with higher priorities to be selected for service ahead of those with lower priorities, regardless of their time of arrival to the system (Gross, & Harris, 1974).

There are two general situations in priority disciplines. In the first, which is called preemptive, the customer with the highest priority is allowed to enter service immediately even if a customer with lower priority is already in service when the higher priority customer enters the system; that is, the lower priority customer in service is preempted, his service stopped, to be resumed again after the higher priority customer is served. In the second general priority situation, called the nonpreemptive case, the highest priority customer goes to the head of the queue but cannot get into service until the customer presently in service is completed, even though this customer has a lower priority (Gross, & Harris, 1974).

1.4.3 System capacity or queue

A system may have an infinite capacity –that is, the queue in front of the server(s) may grow to any length. Otherwise, there may be limitation of space, so that when

the space is filled to capacity, an arrival will not be able to join the system and will be lost to system. The system is called a *delay system* or a *loss system*, according to whether the capacity is infinite or finite.

1.4.4 Service mechanism

Service patterns can be described by a rate (number of customers served per some unit of time) or as a time (time required to service a customer) (Gross, & Harris, 1974).

Service may be single or in batches. One generally thinks of one customer being served at a time by a given server, but there are many situations where customers may be served simultaneously by the server, such as sightseers on a guided tour, or people boarding a train.

The service rate may depend on the number of customers waiting for service. A server may work faster if he sees that the queue is building up or, conversely, he may get flustered and become less efficient. The situation in which service depends on the number of customers waiting is referred to as *state-dependent service* (Gross, & Harris, 1974).

Service systems are usually classified in terms of their number of channels, or numbers of servers. Some examples can be given as following:

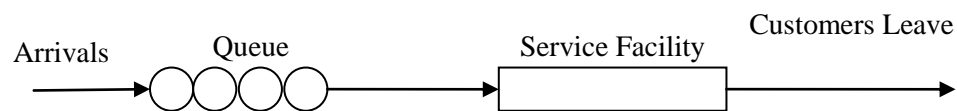


Figure 1.8 Single server- single queue model.

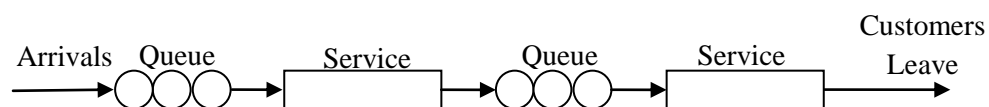


Figure 1.9 Multiple servers in a series.

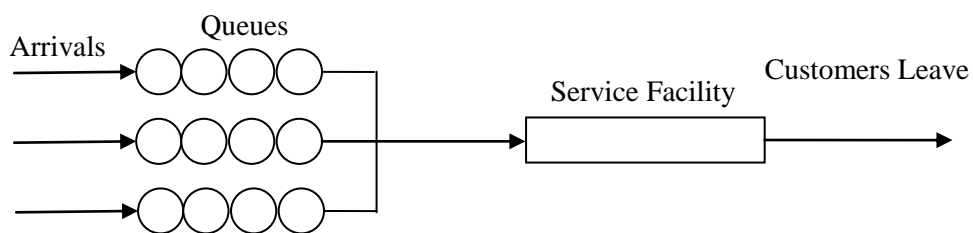


Figure 1.10 Single server- several queues model.

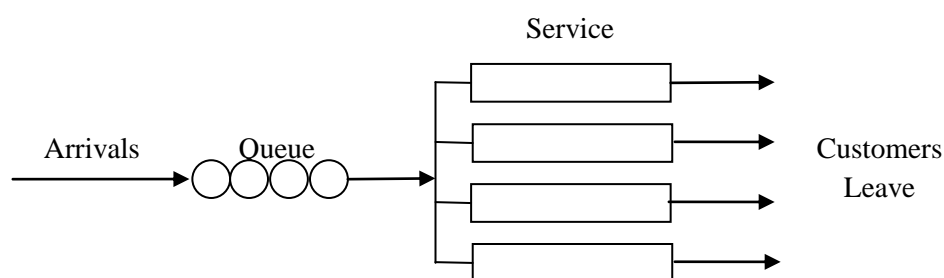


Figure 1.11 Several parallel servers- single queue model.

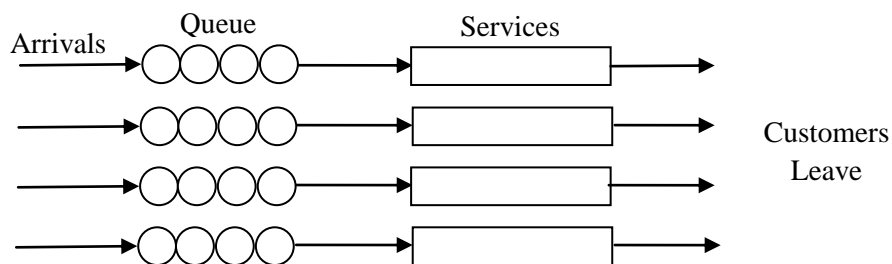


Figure 1.12 Several parallel servers- several queues model.

1.4.5 Notation

The notation introduced by Kendall is generally adopted to denote a queuing process. A queuing process is described by a series of symbols and slashes such as $A/B/X/Y/Z$ (Gross, & Harris, 1974).

A : the interarrival-time distribution,

B : the service time distribution,

X : the number of parallel service channels,

Y : the restriction on system capacity,

Z : the queue discipline.

In many situations only the first three symbols are used. Current practice is to omit the service-capacity symbol if no restriction is imposed ($Y = \infty$) and to omit the queue discipline if it is first come, first served ($Z = \text{FCFS}$) (Kleinrock, & Gail, 1996). For example, $M/D/2$ is used to represent a queuing system with exponential input, deterministic service, two servers, no limit on system capacity, and with discipline first-come, first-served. A and B take on values from the following symbols that refer to distributions :

M = exponential (**M**arkovian)

E_r = r -stage **E**rlangian

D = **D**eterministic

G = **G**eneral

The M and E_r symbols represent exponential distribution and type k Erlang distribution, respectively. The G symbol represents a general distribution, such that, it has no assumption. In these cases, results are applicable to any distribution, however, it is required that independent and identically distributed random variables. We can summarize all notation in Table 1.3 (Gross, & Harris, 1974).

Table 1.3 Queue notation

Characteristics	Symbol	Explanations
Interarrival-time distribution (A)	M	Exponential
	D	Deterministic
	E_k	Erlang type k ($k = 1, 2, \dots$)
	G	General
Service-time distribution (B)	M	Exponential
	D	Deterministic
	E_k	Erlang type k ($k = 1, 2, \dots$)
	G	General
Number of parallel servers (X)	$1, 2, \dots, \infty$	
Restriction on system capacity (Y)	$1, 2, \dots, \infty$	
Queue discipline (Z)	FCFS	First come, first served
	LCFS	Last come, first served
	RSS	Random selection for service
	PR	Priority

CHAPTER TWO

STOCHASTIC PROCESSES AND STEADY-STATE SOLUTION

A stochastic process is the mathematical concept of an empirical process whose development is governed by probabilistic laws. Some of these processes are the Poisson process, counting process and birth-and-death process. The counting process is introduced in Section 2.1.1. The Poisson process is discussed in Section 2.1.2. Relations among some specified distributions are also shown in detail in the subsections of Section 2.1.2. The birth-and-death process and the steady-state difference equations are discussed in Section 2.1.3. Finally, in Section 2.2, some methods of solving the steady-state difference equations are explained.

2.1 Stochastic processes

The stochastic process is described and classified by Medhi (2003) as following: Assume that t is a parameter values in a set T , and $X(t)$ is a random variable for all $t \in T$. At this point, the collection of random variables $\{X(t), t \in T\}$ is called a stochastic process. The parameter t , and the random variable $X(t)$ are interpreted as time and the state of the process at time t , respectively. The elements of T are time points. If T is countable, the stochastic process $\{X(t), t \in T\}$ is said to be a discrete-parameter (or discrete-time) process. If T is an interval of the real line, the stochastic process is said to be a continuous-parameter (or continuous-time) process. The set of the random variable $X(t)$ is called the state space of the process, and this set may be countable or uncountable. Stochastic processes are classified as following:

- (i) discrete-time and discrete state space,
- (ii) discrete-time and continuous state space,
- (iii) continuous-time and discrete state space,
- (iv) continuous-time and continuous state space.

Another classification is made by Gross, & Harris (1974) as following: A continuous-time stochastic process or a discrete-time stochastic process are said to

be a Markov process. On the other hand, process with a discrete state space is referred to as a chain (Medhi, 2003). This classification is summarized in Table 2.1.

Table 2.1 Classification of Markov processes

TIME PARAMETER	STATE SPACE	
	Discrete	Continuous
Discrete	Discrete-time Markov chain	Discrete-time Markov process
Continuous	Continuous-time Markov chain	Continuous-time Markov process

In mathematical language Markov chain is determined by Tijms (2003), such that, the sequence $\{X_n\}$ is a Markov chain if each random variable X_n is discrete and for any set of m points $n_1 < n_2 < \dots < n_m$, the conditional distribution of X_{n_m} , given values of $X_{n_1}, X_{n_2}, \dots, X_{n_{m-1}}$, depends only on $X_{n_{m-1}}$; that is

$$\Pr \left\{ X_{n_m} = x_{n_m} \mid X_{n_1} = x_{n_1}, \dots, X_{n_{m-1}} = x_{n_{m-1}} \right\} = \Pr \left\{ X_{n_m} = x_{n_m} \mid X_{n_{m-1}} = x_{n_{m-1}} \right\}.$$

And, in nonmathematical language it is said that, the future probabilistic behaviour of the process depends only on the present state of the process and is not influenced by its past history. This is called the *Markovian* property.

The class all continuous-time Markov chains has an important subclass formed by the *birth-and-death* processes that are characterized by the property that whenever a transition occurs from one state to another (Medhi, 2003). This transition can be to a neighboring state only, namely, we suppose that a transition can occur only from state i to a neighboring state $(i-1)$ or $(i+1)$. The birth-and-death process is discussed in detail in Section 2.1.3. Consequently, a continuous-time Markov chain is a birth-and-death process, and Poisson process is a birth-and-death process. The Poisson process is discussed in detail in Section 2.1.2.

2.1.1 Counting Process

A stochastic process $\{N(t), t \geq 0\}$ is said to be a *counting process* if $N(t)$ represents the total number of events that have occurred up to time t , and the counting process $N(t)$ satisfies that (Ross, 2003):

- i. $N(t) \geq 0$,
- ii. $N(t)$ is integer valued,
- iii. If $s < t$, then $N(s) \leq N(t)$,
- iv. For $s < t$, $N(t) - N(s)$ equals the number of events that have occurred in the interval (s, t) .

A counting process is said to possess *independent increments* if the numbers of events which occur in disjoint time intervals are independent (Ross, 2003). This means that the number of events which have occurred by time t must be independent of the number of events occurring between times t and $t + s$.

A counting process is said to possess *stationary increments* if the distribution of the number of events which occur in any interval of time depends only on the length of the time interval (Ross, 2003). This means that the number of events in the interval $(t_1 + s, t_2 + s)$ has the same distribution as the number of events in the interval (t_1, t_2) for all $t_1 < t_2$, and $s > 0$.

2.1.2 Poisson Process

The counting process $\{N(t), t \geq 0\}$ is said to be the *Poisson process* having rate λ , $\lambda > 0$. $N(t)$ represents the number of events that occur in the time interval $[0, t]$. The Poisson process satisfies that (Ross, 2006):

- i. $N(0) = 0$,
- ii. The numbers of events that occur in disjoint time intervals are independent, that is, the process has *independent increments*,

iii. The distribution of the number of events that occur in a given interval depends only on the length of that interval and not on its location, that is, the process has *stationary increments*,

$$\text{iv. } P\{N(t) = 1\} = \lambda t + o(\Delta t),$$

$$\text{v. } P\{N(t) \geq 2\} = o(\Delta t).$$

Δt is an incremental element, and the probability that more than one arrival between t and $t + \Delta t$ is $o(\Delta t)$ (Gross, & Harris, 1974). It becomes negligible when compared to Δt as $\Delta t \rightarrow 0$; that is,

$$\lim_{\Delta t \rightarrow 0} \frac{o(\Delta t)}{\Delta t} = 0.$$

2.1.2.1 The number of arrivals to a system at time t

We calculate the probability of n arrivals in a time interval of length t , $p_n(t)$, $n \geq 0$ as the following:

$$\begin{aligned} p_n(t + \Delta t) = & \Pr\{n \text{ arrivals in } t \text{ and zero in } \Delta t\} \\ & + \Pr\{n-1 \text{ arrivals in } t \text{ and one in } \Delta t\} \\ & + \Pr\{n-2 \text{ arrivals in } t \text{ and two in } \Delta t\} \\ & + \dots + \Pr\{0 \text{ arrivals in } t \text{ and } n \text{ in } \Delta t\} \quad (n \geq 1). \end{aligned} \quad (2.1)$$

The equation is rewritten as the following:

$$p_n(t + \Delta t) = p_n(t)[1 - \lambda\Delta t - o(\Delta t)] + p_{n-1}(t)[\lambda\Delta t + o(\Delta t)] + o(\Delta t). \quad (2.2)$$

For the case $n = 0$, we have

$$p_0(t + \Delta t) = p_0(t)[1 - \lambda\Delta t - o(\Delta t)]. \quad (2.3)$$

We can rewrite (2.2) and (2.3) as follows:

$$p_0(t + \Delta t) - p_0(t) = -\lambda\Delta t p_0(t) + o(\Delta t) \quad (2.4)$$

$$p_n(t + \Delta t) - p_n(t) = -\lambda \Delta t p_n(t) + \lambda \Delta t p_{n-1}(t) + o(\Delta t) \quad (n \geq 1). \quad (2.5)$$

In order to obtain the differential-difference equations, we divide (2.4) and (2.5) by Δt , take the limit as $\Delta t \rightarrow 0$:

$$\left\{ \begin{array}{l} \lim_{\Delta t \rightarrow 0} \left[\frac{p_0(t + \Delta t) - p_0(t)}{\Delta t} = -\lambda p_0(t) + \frac{o(\Delta t)}{\Delta t} \right] \\ \lim_{\Delta t \rightarrow 0} \left[\frac{p_n(t + \Delta t) - p_n(t)}{\Delta t} = -\lambda p_n(t) + \lambda p_{n-1}(t) + \frac{o(\Delta t)}{\Delta t} \right] \end{array} \right. \quad (n \geq 1),$$

and then, we have

$$\frac{dp_0(t)}{dt} = -\lambda p_0(t) \quad (2.6)$$

$$\frac{dp_n(t)}{dt} = -\lambda p_n(t) + \lambda p_{n-1}(t) \quad (n \geq 1). \quad (2.7)$$

We now have an infinite set of linear differential equations of the first order in (2.6) and (2.7). The linear differential equation of the first order is in the form of the following:

$$\frac{dy(x)}{dx} + \phi(x)y(x) = \psi(x), \quad (2.8)$$

and it's solution is given as (2.9).

$$y(x) = C e^{-\int \phi(x) dx} + e^{-\int \phi(x) dx} \int e^{\int \phi(x) dx} \psi(x) dx, \quad (2.9)$$

where C is a constant which is determined by boundary conditions $p_n(0) = 0$ for $n > 0$ and $p_0(0) = 1$. To solve the infinite set of equations given by (2.6) and (2.7), (2.9) can be used. To find $p_0(t)$, $\phi(x) = \lambda$, $y(x) = p_0(t)$ and $\psi(x) = 0$,

$$\frac{dp_0(t)}{dt} + \lambda p_0(t) = 0.$$

Then the solution can be written as follows:

$$p_0(t) = Ce^{-\int \lambda dt} + e^{-\int \lambda dt} \int e^{\int \lambda dt} (0) dt$$

$$p_0(t) = Ce^{-\lambda t}$$

Using boundary condition $p_0(0) = 1$, we obtain $C = 1$ and

$$p_0(t) = e^{-\lambda t}. \quad (2.10)$$

To find $p_1(t)$, $\phi(x) = \lambda$, $y(x) = p_1(t)$ and $\psi(x) = \lambda p_0(t)$,

$$\frac{dp_n(t)}{dt} + \lambda p_n(t) = \lambda p_{n-1}(t) \Rightarrow \frac{dp_1(t)}{dt} + \lambda p_1(t) = \lambda p_0(t)$$

Then the solution can be written as follows:

$$p_1(t) = Ce^{-\int \lambda dt} + e^{-\int \lambda dt} \int e^{\int \lambda dt} \lambda e^{-\lambda t} dt$$

$$p_1(t) = Ce^{-\lambda t} + e^{-\lambda t} \int e^{\lambda t} \lambda e^{-\lambda t} dt$$

$$p_1(t) = Ce^{-\lambda t} + e^{-\lambda t} \lambda t.$$

Using boundary condition, $p_1(0) = 0$, we obtain $C = 0$ and

$$p_1(t) = \lambda t e^{-\lambda t}. \quad (2.11)$$

To find $p_2(t)$, $\phi(x) = \lambda$, $y(x) = p_2(t)$ and $\psi(x) = \lambda p_1(t)$,

$$\frac{dp_n(t)}{dt} + \lambda p_n(t) = \lambda p_{n-1}(t) \Rightarrow \frac{dp_2(t)}{dt} + \lambda p_2(t) = \lambda p_1(t)$$

Then the solution can be written as follows:

$$p_2(t) = Ce^{-\int \lambda dt} + e^{-\int \lambda dt} \int e^{\int \lambda dt} \lambda \lambda t e^{-\lambda t} dt$$

$$p_2(t) = Ce^{-\lambda t} + e^{-\lambda t} \int e^{\lambda t} \lambda^2 e^{-\lambda t} dt$$

$$p_2(t) = Ce^{-\lambda t} + e^{-\lambda t} \lambda^2 \frac{t^2}{2}$$

Using boundary condition, $p_2(0) = 0$, we obtain $C = 0$ and

$$p_2(t) = \frac{(\lambda t)^2}{2} e^{-\lambda t}. \quad (2.12)$$

We employ the same way and obtain the equations as noted below:

$$p_3(t) = \frac{(\lambda t)^3}{3!} e^{-\lambda t}, \quad (2.13)$$

$$p_4(t) = \frac{(\lambda t)^4}{4!} e^{-\lambda t}. \quad (2.14)$$

From (2.11), (2.12), (2.13), and (2.14) we conjecture the general formula to be

$$p_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}, \quad n = 0, 1, 2, \dots \quad (2.15)$$

The equation (2.15) is the Poisson probability distribution with mean λt .

2.1.2.2 The time between successive arrivals (interarrival time)

If the arrival process follows the Poisson distribution, the random variable defined as the time between successive arrivals, interarrival time, follows the exponential distribution (İnal, 1998).

Let T represents the random variable “time between successive arrivals”, then $\Pr\{T \geq t\} = \Pr\{\text{zero arrivals in time } t\} = p_0(t) = e^{-\lambda t}$.

Let $F(t)$ represents the cumulative distribution function of T , then we have

$$F(t) = \Pr\{T \leq t\} = 1 - e^{-\lambda t}.$$

Let $f(t)$ is the density function of T , then it is given by

$$f(t) = \frac{dF(t)}{dt} = \lambda e^{-\lambda t}, \quad t > 0 \quad (2.16)$$

The equation (2.16) is the exponential probability distribution with mean $1/\lambda$.

2.1.2.3 The arrival time of the n th event

Let S_n represents the arrival time of the n^{th} event. It is also called the waiting time until the n^{th} event (Ross, 2003). It can be written as

$$S_n = \sum_{i=1}^n T_i \quad n \geq 1.$$

T_1, \dots, T_i are exponential random variables having mean $1/\lambda$, and each of them has the moment generating function is given by

$$M_T(t) = \left(1 - \frac{t}{\lambda}\right)^{-1} = \frac{\lambda}{\lambda - t}.$$

We can write

$$\begin{aligned} M_{S_n}(t) &= E\left[e^{t \sum_{i=1}^n T_i}\right] = E\left[e^{tT_1 + tT_2 + \dots + tT_n}\right] \\ &= E\left[e^{tT_1}\right] E\left[e^{tT_2}\right] \dots E\left[e^{tT_n}\right] \\ &= M_{T_1}(t) M_{T_2}(t) \dots M_{T_n}(t) \\ &= \underbrace{\left(\frac{\lambda}{\lambda - t}\right) \left(\frac{\lambda}{\lambda - t}\right) \dots \left(\frac{\lambda}{\lambda - t}\right)}_{n \text{ times}} \end{aligned}$$

and have

$$M_{S_n}(t) = \left(\frac{\lambda}{\lambda - t} \right)^n \quad (2.17)$$

The equation (2.17) is the moment generating function of the gamma probability distribution with mean n/λ . Then it is concluded that S_n has the gamma probability distribution with parameters n and λ .

2.1.3 Birth-and-death process

Birth-and-death process is a continuous parameter Markov chain. The process is Markovian and instantaneous changes in the system state can only amount to an increase (birth) or decrease (death) of one.

$p_n(t) = \Pr\{\text{population is at size } n \text{ at time } t\}$ is the state probabilities for an arbitrary birth-death process. The probability of a birth occurring in a small interval of length Δt which began with the system in state n is assumed to be $\lambda_n \Delta t + o(\Delta t)$, while that of a death is assumed to be $\mu_n \Delta t + o(\Delta t)$, independent of λ_n and t .

The system may get to state n at time $t + \Delta t$. To do so, the system might have been in state n at time t and had no net change during Δt , or the system might have found itself in state $n-1$ and had a birth, or in state $n+1$ and had a death. We can express this in mathematical form as follows:

For $n \geq 1$,

$$\begin{aligned} p_n(t + \Delta t) &= p_n(t)[1 - \lambda_n \Delta t][1 - \mu_n \Delta t] \\ &\quad + p_n(t)[\lambda_n \Delta t][\mu_n \Delta t] \\ &\quad + p_{n+1}(t)[1 - \lambda_{n+1} \Delta t][\mu_{n+1} \Delta t] \\ &\quad + p_{n-1}(t)[\lambda_{n-1} \Delta t][1 - \mu_{n-1} \Delta t] \\ &\quad + o(\Delta t). \end{aligned}$$

For $n = 0$,

$$p_0(t + \Delta t) = p_0(t)[1 - \lambda_0 \Delta t] + p_1(t)[\mu_1 \Delta t][1 - \lambda_1 \Delta t] + o(\Delta t).$$

The corresponding differential-difference equations are found by transposing $p_n(t)$ from the right-hand side to the left, dividing through by Δt , and taking the limit as $\Delta t \rightarrow \infty$. They are

$$\begin{cases} \frac{\partial p_n(t)}{\partial t} = -(\lambda_n + \mu_n) p_n(t) + (\mu_{n+1}) p_{n+1}(t) + (\lambda_{n-1}) p_{n-1}(t) & (n \geq 1) \\ \frac{\partial p_0(t)}{\partial t} = -\lambda_0 p_0(t) + \mu_1 p_1(t). \end{cases} \quad (2.18)$$

The stationary solution is found as follows. Since $p_n(t)$ is to be independent of time, $dp_n(t)/dt$ is zero and (2.18) becomes

$$\begin{cases} 0 = -(\lambda_n + \mu_n) p_n + (\mu_{n+1}) p_{n+1} + (\lambda_{n-1}) p_{n-1} & (n \geq 1) \\ 0 = -\lambda_0 p_0 + \mu_1 p_1 \end{cases} \quad (2.19a)$$

$$\begin{cases} p_{n+1} = \frac{\lambda_n + \mu_n}{\mu_{n+1}} p_n - \frac{\lambda_{n-1}}{\mu_{n+1}} p_{n-1} & (n \geq 1) \\ p_1 = \frac{\lambda_0}{\mu_1} p_0. \end{cases} \quad (2.19b)$$

To solve these equations for the birth-death steady-state probabilities, we consider the special case where $\lambda_n = \lambda$ and $\mu_n = \mu$ for all values of n . Equations (2.19a) and (2.19b) reduce to

$$\begin{cases} 0 = -(\lambda + \mu) p_n + \mu p_{n+1} + \lambda p_{n-1} & (n \geq 1) \\ 0 = -\lambda p_0 + \mu p_1 \end{cases} \quad (2.20a)$$

and

$$\begin{cases} p_{n+1} = \frac{\lambda + \mu}{\mu} p_n - \frac{\lambda}{\mu} p_{n-1} & (n \geq 1) \\ p_1 = \frac{\lambda}{\mu} p_0. \end{cases} \quad (2.20b)$$

These equations are the steady-state difference equations for the M/M/1 queue.

2.2 Steady-state solution

The density functions for the interarrival times and service times are given, respectively, as

$$a(t) = \lambda e^{-\lambda t}$$

$$b(t) = \mu e^{-\mu t}$$

where $1/\lambda$ then is the mean interarrival time and $1/\mu$ is the mean service time. Interarrival times are assumed to be statistically independent. We have

$$\Pr\{\text{an arrival occurs in an infinitesimal interval of length } \Delta t\} = \lambda \Delta t + o(\Delta t)$$

$$\Pr\{\text{more than one arrival occurs in } \Delta t\} = o(\Delta t)$$

$$\Pr\{\text{a service completion in } \Delta t \mid \text{system not empty}\} = \mu \Delta t + o(\Delta t)$$

$$\Pr\{\text{more than one service completion in } \Delta t \mid \text{more than one in system}\} = o(\Delta t).$$

We have, then, a birth-death process with $\lambda_n = \lambda$ and $\mu_n = \mu$, for all n . Arrivals can be considered as “births” to the system, since if the system is in state n and an arrival occurs, the state is changed to $n + 1$. Departures can be considered as “death” from the system, since if the system is in state n and a departure occurs, the state is changed to $n - 1$. Hence, the steady-state equations are given by (2.20a) or (2.20b).

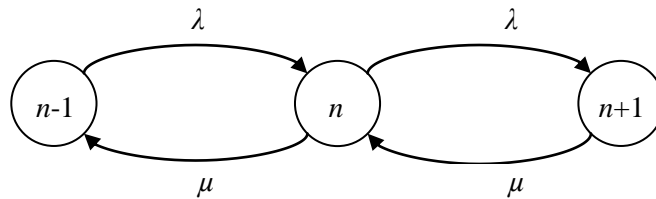


Figure 2.1 Rate transition diagram.

The analysis looks at a given state and requires that total flow into the state be equal to total flow out of the state if steady-state conditions exist. Then a rate transition diagram would appear as shown in Figure 2.1. From state n , the system

goes to $n-1$ if a service is completed or $n+1$ if an arrival occurs. The system can go to state n from $n-1$ if an arrival comes or to state n from $n+1$ if a service is completed.

2.2.1 Methods of solving steady-state difference equations

There are two methods for solving (2.20a) and (2.20b). The reason for presenting the two methods of solution is that one may be more successful than the others, depending on the particular model. For example, in this thesis we have used solution by generating function for the queuing model with batch arrival in the section 3.4.1 and solution by use of operators for the queuing model with batch service in the section 3.4.3.

2.2.1.1. Iterative method

The equation 2.20b can be used iteratively to obtain the following:

$$\begin{cases} p_1 = \frac{\lambda}{\mu} p_0 \\ p_2 = \left(\frac{\lambda}{\mu}\right)^2 p_0 \\ p_3 = \left(\frac{\lambda}{\mu}\right)^3 p_0 \end{cases} \quad (2.21)$$

at this point, we obtain

$$p_n = \left(\frac{\lambda}{\mu}\right)^n p_0. \quad (2.22)$$

$$p_{n+1} = \left(\frac{\lambda}{\mu}\right)^{n+1} p_0 \quad (2.23)$$

It remains only to obtain p_0 . This can be accomplished by utilizing the boundary condition that $\sum_{n=0}^{\infty} p_n = 1$, since p_n is a probability distribution. Using (2.23),

$$1 = \sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^n p_0.$$

We define ρ as λ/μ . The ratio ρ is often called the *utilization factor*. It is the expected number of arrivals per mean service time in the limit, and it is often also called the *traffic intensity*. We rewrite

$$p_0 = \frac{1}{\sum_{n=0}^{\infty} \rho^n}.$$

$\sum_{n=0}^{\infty} \rho^n$ is the geometric series and converges if and only if $|\rho| < 1$. Thus for the existence of a steady-state solution, $\rho = \lambda/\mu$ must be less than 1, or in other words, λ must be less than μ .

Making use of the well-known expression for the sum of the terms of geometric progression,

$$\sum_{n=0}^{\infty} \rho^n = \frac{1}{1-\rho} \quad (\rho < 1),$$

we have

$$p_0 = 1 - \rho \quad (\rho = \lambda/\mu < 1), \quad (2.24)$$

Thus, substituting (2.24) to (2.22) the steady-state solution is obtained by given

$$p_n = \rho^n (1 - \rho) \quad (\rho = \lambda/\mu < 1) \quad . \quad (2.25)$$

2.2.1.2 Solution by generating functions

The probability generating function $P(z) = \sum_{n=0}^{\infty} p_n z^n$ can be use to find p_n . We rewrite (2.20b) in terms of ρ and obtain

$$\begin{cases} p_{n+1} = (\rho + 1)p_n - \rho p_{n-1} & (n \geq 1) \\ p_1 = \rho p_0 \end{cases} \quad (2.26)$$

when both sides of the first line of (2.26) are multiplied by z^n we find

$$p_{n+1}z^n = (\rho + 1)p_n z^n - \rho p_{n-1}z^n$$

or

$$z^{-1}p_{n+1}z^{n+1} = (\rho + 1)p_n z^n - \rho z p_{n-1}z^{n-1}$$

And, when both sides of the foregoing equation are summed from $n = 1$ to ∞ , it is found that

$$z^{-1} \sum_{n=1}^{\infty} p_{n+1}z^{n+1} = (\rho + 1) \sum_{n=1}^{\infty} p_n z^n - \rho z \sum_{n=1}^{\infty} p_{n-1}z^{n-1}$$

or

$$z^{-1} \left[\sum_{n=1}^{\infty} p_{n+1}z^{n+1} - p_1 z - p_0 \right] = (\rho + 1) \left[\sum_{n=0}^{\infty} p_n z^n - p_0 \right] - \rho z \sum_{n=1}^{\infty} p_{n-1}z^{n-1}$$

Note that

$$\sum_{n=-1}^{\infty} p_{n+1}z^{n+1} = \sum_{n=0}^{\infty} p_n z^n = \sum_{n=1}^{\infty} p_{n-1}z^{n-1} = P(z),$$

we get

$$z^{-1} [P(z) - p_1 z - p_0] = (\rho + 1) [P(z) - p_0] - \rho z P(z). \quad (2.27)$$

From (2.26) we have that $p_1 = \rho p_0$; hence

$$z^{-1} [P(z) - (\rho z + 1)p_0] = (\rho + 1) [P(z) - p_0] - \rho z P(z).$$

Solving for $P(z)$ we have

$$P(z) = \frac{p_0}{1 - z\rho}. \quad (2.28)$$

To find p_0 we use the boundary condition that $\sum_{n=0}^{\infty} p_n = 1$.

$$P(z) = \sum_{n=0}^{\infty} p_n z^n \quad \Rightarrow \quad P(1) = \sum_{n=0}^{\infty} p_n 1^n = \sum_{n=0}^{\infty} p_n = 1.$$

From (2.28), we have

$$p_0 = 1 - \rho$$

and

$$P(z) = \frac{1 - \rho}{1 - z\rho} \quad (\rho < 1). \quad (2.29)$$

The sum of a geometric series is $\frac{1}{1 - z\rho} = 1 + z\rho + (z\rho)^2 + (z\rho)^3 + \dots$, and thus the probability generating function is

$$P(z) = \sum_{n=0}^{\infty} \underbrace{(1 - \rho)\rho^n}_{p_n} z^n. \quad (2.30)$$

Thus the steady-state solution is obtain by given

$$p_n = \rho^n (1 - \rho) \quad (\rho = \lambda/\mu < 1).$$

CHAPTER THREE

QUEUEING MODELS

In order to understand the special Erlangian distribution E_r which is applied to the queueing systems $M/E_r/1$ and $E_r/M/1$, we first begin by discussing the method of stages in Section 3.1. And then, we introduce basic queueing model and Erlangian queueing models in detail in Section 3.2. The aim here is that the batch queueing systems to be understood easily. Because the system $M/E_r/1$ has an interpretation as a batch arrival process, similarly, the system $E_r/M/1$ may be interpreted as a batch service system. In Section 3.3, we introduce the Little formula which is the most important equation in queueing theory. Finally, in Section 3.4, we discuss the batch queueing models.

3.1 The method of stages

We define a service facility with an exponentially distributed service time pdf given by

$$b(x) = \mu e^{-\mu x} \quad x \geq 0 \quad (3.1)$$

The exponential distribution has a mean and variance given by

$$E(x) = \frac{1}{\mu} \quad \text{and} \quad \sigma_x^2 = \frac{1}{\mu^2}.$$

In Figure 3.1 the oval represents the service facility. The oval is labeled with symbol μ . μ represents the service-rate parameter as in (3.1).

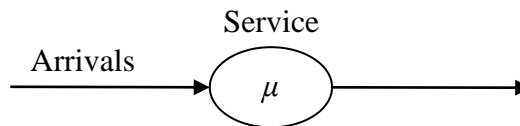


Figure 3.1 The single-stage exponential server.

In Figure 3.2 the large oval represents the service facility. The internal structure of this service facility is showed as connection of two smaller ovals. Each of these ovals represents a single exponential server such as that described in Figure 3.1. But the small ovals are labeled with the parameter 2μ , and they have a pdf given by

$$h(y) = 2\mu e^{-2\mu y} \quad y \geq 0 \quad (3.2)$$

The mean and variance for $h(y)$ are given by

$$E(y) = \frac{1}{2\mu} \quad \text{and} \quad \sigma_y^2 = \left(\frac{1}{2\mu}\right)^2.$$

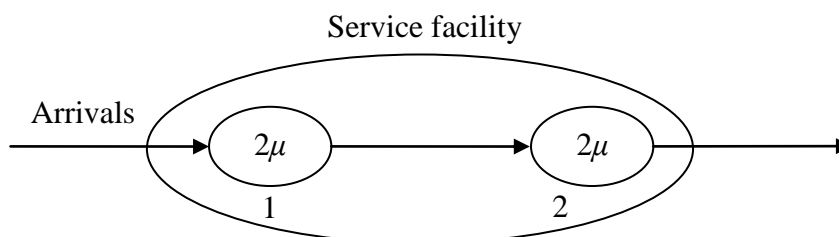


Figure 3.2 The two-stage Erlangian server E_2

A new customer is allowed to enter from the left when a customer departs from this service facility. This new customer enters stage 1 and remains there for an amount of time. Upon his departure from this first stage he then proceeds into the second stage and remains there for an amount of time. His departure from this second stage is called he departs from the service facility. At this point a new customer may enter the facility from the left. Namely, only one customer is allowed into the service facility at any time. This implies that at least one of the two service states must always be empty.

The Laplace transform for the exponential density function in (3.1) is given by

$$A^*(s) = \frac{\mu}{s + \mu}. \quad (3.3)$$

The Laplace transform for (3.2) is given by

$$H^*(s) = \frac{2\mu}{s + 2\mu}. \quad (3.4)$$

We request to know the specific distribution of total time spent in the service facility. This random variable is the sum of two independent and identically distributed random variables.

The characteristic function is denoted by $\phi_X(u)$ and is given by

$$\phi_X(u) \triangleq E[e^{jux}].$$

If we form the characteristic function for $Y=X_1+X_2$,

$$\begin{aligned}\phi_Y(u) &\triangleq E[e^{juy}] \\ \phi_Y(u) &= E[e^{jux_1} e^{jux_2}] \\ \phi_Y(u) &= E[e^{jux_1}] E[e^{jux_2}] \\ \phi_Y(u) &= \phi_{X_1}(u) \phi_{X_2}(u) = [\phi_X(u)]^2.\end{aligned}$$

This result also can be applied to the Laplace transform as follows:

$$\begin{aligned}B^*(s) &= [H^*(s)]^2 \\ B^*(s) &= \left[\frac{2\mu}{s+2\mu} \right]^2.\end{aligned}\tag{3.5}$$

We invert the (3.5) and obtain the specified distribution with mean $E(x) = \frac{1}{\mu}$ and

variance $\sigma_x^2 = \frac{1}{2\mu^2}$, given by

$$b(x) = 2\mu(2\mu x)e^{-2\mu x} \quad x \geq 0.\tag{3.6}$$

We generalize to r -stage exponential server in Figure 3.3 similar to the two-stage exponential server as given in Figure 3.2. The small ovals are labeled with the parameter $r\mu$, and they have a pdf given by

$$h(y) = r\mu e^{-r\mu y} \quad y \geq 0\tag{3.7}$$

The mean and variance for $h(y)$ are given by

$$E(y) = \frac{1}{r\mu} \quad \text{and} \quad \sigma_y^2 = \left(\frac{1}{r\mu} \right)^2.$$

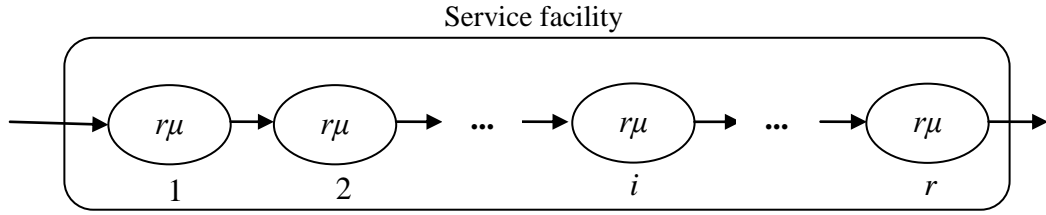


Figure 3.3 The r -stage Erlangian server E_r

The total time that a customer spends in this service facility is the sum of r independent identically distributed random variables. Thus, the Laplace transform for (3.7) is given by

$$B^*(s) = \left[\frac{r\mu}{s + r\mu} \right]^r \quad (3.8)$$

We invert the (3.8) and obtain the specified distribution with mean $E(x) = \frac{1}{\mu}$

and variance $\sigma_x^2 = \frac{1}{r\mu^2}$, given by

$$b(x) = \frac{r\mu(r\mu x)^{r-1} e^{-r\mu x}}{(r-1)!} \quad x \geq 0 \quad (3.9)$$

3.2 Basic queueing model and Erlangian queueing models

The aim here is to introduce basic queue model M/M/1 and to lead in the batch queue models. As it is mentioned before, the system M/E_r/1 has an interpretation as a batch arrival system, similarly, the system E_r/M/1 may be interpreted as a batch service system.

3.2.1 M/M/1 model

The M/M/1 queue has identically independent distributed interarrival times, which are exponentially distributed with parameter $1/\lambda$ and service times are distributed as exponential distribution with parameter $1/\mu$. The system has only a single server and uses FCFS service discipline. The waiting line is infinite size. The M/M/1 system is a

pure birth-and-death system where at any point in time at most one event occurs, with an event either being the arrival of a new customer or the completion of a customer's service.

The distribution of interarrival is exponential, hence the average interarrival time is $1/\lambda$. The distribution of service times is exponential, hence the average service time is $1/\mu$. So, the service facility or traffic intensity ρ is determined by $\rho = \lambda/\mu$. The general rule in queueing systems is that ρ must be less than 1. Since the queue can not last to infinity, μ must be greater than λ in order to make system stable. This is called steady-state condition.

The performance measures are as follows:

L : Expected number of customers in queuing system.

$$L = \frac{\rho}{1 - \rho} \quad (3.10)$$

or

$$L = \frac{\lambda}{\mu - \lambda} \quad (3.11)$$

L_q : Expected queue length (excludes customers being served).

$$L_q = \frac{\rho^2}{1 - \rho} \quad (3.12)$$

or

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} \quad (3.13)$$

\hat{W} : Waiting time in system (includes service time) for each individual customer.

$W = E(\hat{W})$.

$$W = \frac{1}{\mu - \lambda} \quad (3.14)$$

\hat{W}_q : Waiting time in queue (excludes service time) for each individual customer.

$$W_q = E(\hat{W}_q).$$

$$W_q = \frac{\lambda}{\mu(\mu - \lambda)} \quad (3.15)$$

3.2.2 M/E_r/1 model

The arrival is Poisson with parameter λ . The service time has an Erlang type- r distribution- r exponential stages. In each stage, the service time distribution is exponential with parameter $r\mu$ so that the total mean service time is $r(1/r\mu) = 1/\mu$. The density functions for the interarrival times and service times are given, respectively, as

$$a(t) = \lambda e^{-\lambda t} \quad t \geq 0$$

$$b(x) = \frac{r\mu(r\mu x)^{r-1} e^{-r\mu x}}{(r-1)!} \quad x \geq 0$$

A customer enters the first stage of the service, then progresses through the remaining stage and must complete the last stage before the next customer enters the first stage. We represent the state-transition-rate diagram for stages in system as shown in Figure 3.4 (Kleinrock, 1975).

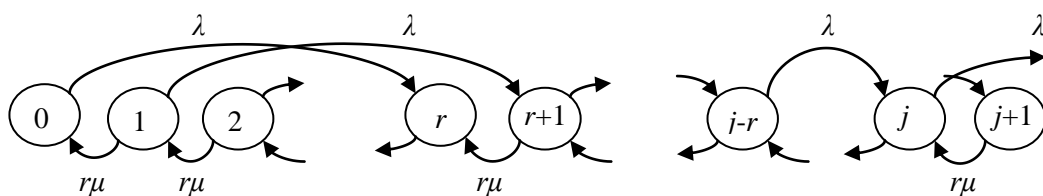


Figure 3.4 State-transition-rate diagram for number of stages: M/E_r/1.

Rate In = Rate Out Principle. For any state of the system n ($n = 0, 1, 2, \dots$), mean entering rate = mean leaving rate, and the equation expressing this principle is called the *balance equation* for state n (Hiller, & Lieberman, 2001). These balance equations are summarized in Table 3.1.

Table 3.1 Balance equation for M/E_r/1

State	Rate Out = Rate In
0	$\lambda p_0 = r\mu p_1$
1	$(\lambda + r\mu) p_1 = r\mu p_2$
:	:
j	$(\lambda + r\mu) p_j = \lambda p_{j-r} + r\mu p_{j+1}$

The forward equations in equilibrium is given by

$$\lambda p_0 = r\mu p_1 \quad (3.16)$$

$$(\lambda + r\mu) p_j = \lambda p_{j-r} + r\mu p_{j+1} \quad j = 1, 2, \dots \quad (3.17)$$

These equations is solved to find p_n . In this system, the states denote the number of stages in the system requiring service instead of number of customers.

3.2.3 E_r/M/1 model

The interarrival times are Erlang type- r distributed with a mean of $1/\lambda$. Each arrival is passing through r stages, with mean time of $1/r\lambda$ in each stages. We define the state variable as the number of arrival stages in the system. The density functions for the interarrival times and service times are given, respectively, as

$$a(t) = \frac{r\lambda(r\lambda t)^{r-1} e^{-r\lambda t}}{(r-1)!} \quad t \geq 0$$

$$b(x) = \mu e^{-\mu x} \quad x \geq 0$$

We can consider an arriving facility instead of service facility. When this arriving customer is inserted from the left side he must then pass through r exponential stages each with parameter $r\lambda$. When he exists from the right side of arriving facility he is then said to “arrive” to the queuing system E_r/M/1. Upon his arrival, new customer is inserted into the left side of arriving facility and the process is repeated. We represent

the state-transition-rate diagram for stages in system as shown in Figure 3.5 (Kleinrock, 1975). And, balance equations are summarized in Table 3.2.

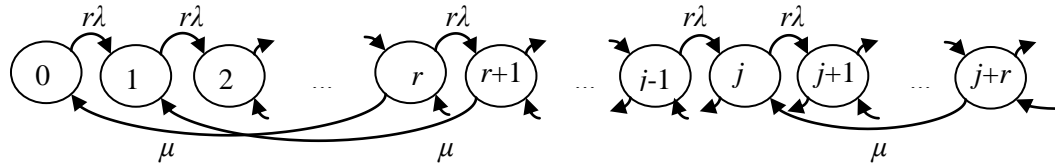


Figure 3.5 State-transition-rate diagram for number of stages: $E_r/M/1$.

Table 3.2 Balance equation for $E_r/M/1$

State	Rate Out = Rate In
0	$r\lambda p_0 = \mu p_r$
1	$r\lambda p_1 = r\lambda p_0 + \mu p_{r+1}$
2	$r\lambda p_2 = r\lambda p_1 + \mu p_{r+2}$
:	:
r	$(r\lambda + \mu) p_r = r\lambda p_{r-1} + \mu p_{2r}$

The forward equations in equilibrium is given by

$$r\lambda p_0 = \mu p_r \quad (3.18)$$

$$r\lambda p_j = r\lambda p_{j-1} + \mu p_{j+r} \quad 1 \leq j \leq r-1 \quad (3.19)$$

$$(r\lambda + \mu) p_j = r\lambda p_{j-1} + \mu p_{j+r} \quad r \leq j \quad (3.20)$$

3.3 The Little's Formula

The Little's formula is a 'law of nature' that applies to almost any type of queuing system. It relates the long-run averages such as the long-run average number of customers in system L (or L_q) and the long-run average amount of time spent per customer in the system W (or W_q). Tijms (2003) considered an example to illustrate the formula of Little $L = \lambda W$ as following. A hospital admits on 25 new patients per day. A patient stays on average 3 days in the hospital. What is the average number of occupied beds? Let $\lambda = 25$ denote the average number of new patients who are

admitted per day, $W = 3$ the average number of days a patient stays in the hospital and L the average number of occupied beds. Then $L = \lambda W = 25 \times 3 = 75$ beds.

A queuing system is described by the arrival process of customers, the service facility and the service discipline to name the most important elements. In formulating the law of Little, there is no need to specify these basic elements (Tijms, 2003). On the other hand, it needs to steady- stead condition (Medhi, 2003).

The formula of Little has some heuristic or rigorous proofs. Morse (1958) gave heuristic proof is simple enough for a long time. His student, Little (1961), gave rigorous proof of the formula, and so the formula is known as Little's formula. Jewel (1967) gave a proof based on renewal theory, and the proof does not require steady- stead conditions. Elion (1969) gave the following simple proof. This proof needs to steady-stead condition, and it does not depend on

- (i) the arrival or service time distributions,
- (ii) the number of servers in the system,
- (iii) the queue discipline.

We now consider a part of system in a time interval T in Figure 3.6 to give the proof.

$A(T)$ = total number of arrivals during T

$B(T)$ = total waiting time in the system of all the customers who arrive during T .

The area under the curve equals to $B(T)$, and it is shown as following (Veeraraghavan, 2004). We can also see the relation in Figure 3.6.

$$\int_0^T N_i dt = \sum_{i=1}^{A(T)} W_i = B(T)$$

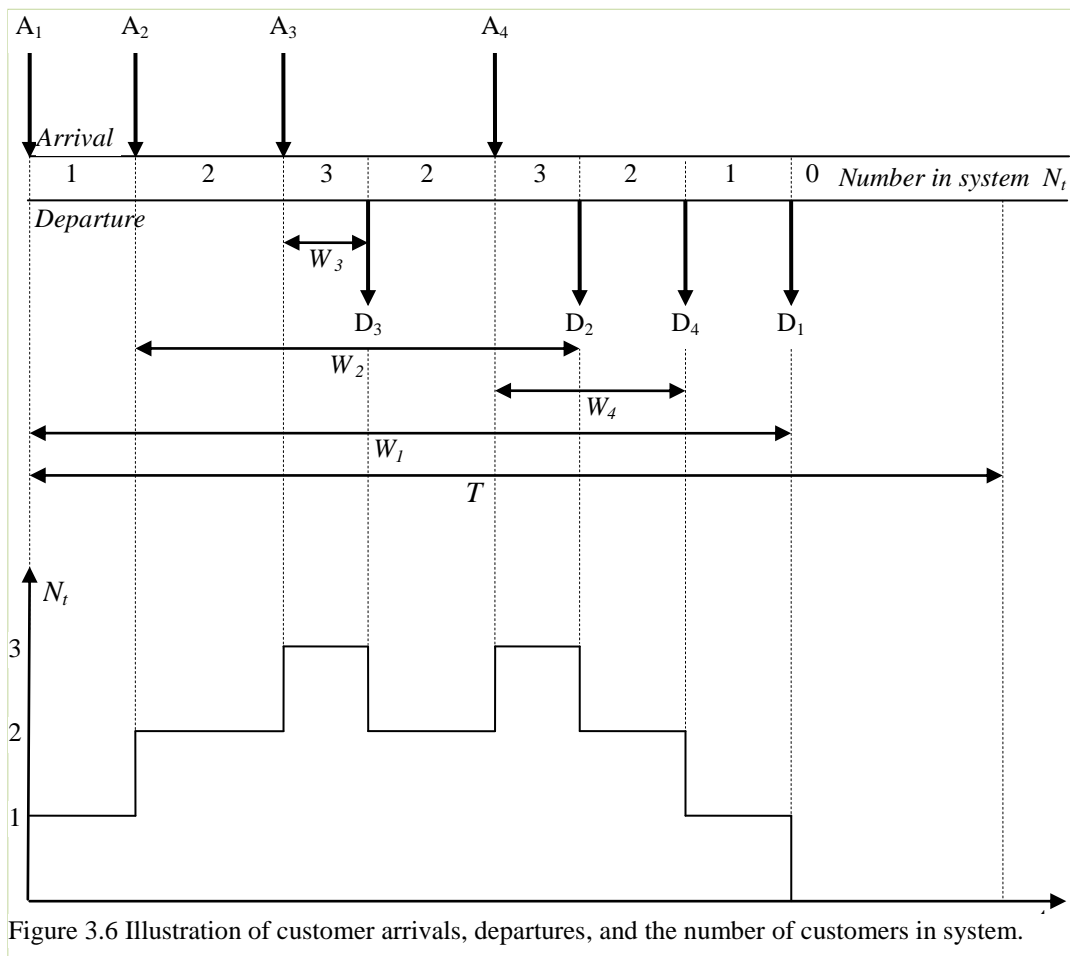


Figure 3.6 Illustration of customer arrivals, departures, and the number of customers in system.

Taking advantage of the relation above we can write the equations as following:

The **average arrival rate** during T

$$\lambda(T) = \frac{A(T)}{T}$$

The **average waiting time of customers in the system** during T

$$W(T) = \frac{B(T)}{A(T)}$$

The **average number of customers in the system** during T

$$L(T) = \frac{1}{T} \int_0^T N_t dt = \frac{B(T)}{T}$$

We have

$$L(T) = \frac{B(T)}{T} = \frac{B(T)}{A(T)} \frac{A(T)}{T}$$

$$L(T) = W(T)\lambda(T).*$$

We suppose that limits, are given as below, exist as $T \rightarrow \infty$

$$\lim_{T \rightarrow \infty} \lambda(T) = \lambda$$

$$\lim_{T \rightarrow \infty} W(T) = W$$

$$\lim_{T \rightarrow \infty} L(T) = L.$$

This three limits satisfy the relation $L = \lambda W$. The last equation is referred to as Little's formula.

3.3.1 Relationships among L , W , L_q , and W_q

Certain relationships can be seen among of the performance measures. We can see from (3.13) and (3.15) that

$$L_q = \lambda W_q \tag{3.21}$$

and also from (3.11) and (3.14) that

$$L = \lambda W. \tag{3.22}$$

The other relations can be summarize in Table 3.3 (Gross, & Harris, 1974).

Table 3.3 Relations among the performance measures.

Relation	Comments on Validity
$W = W_q + \frac{1}{\mu}$	Holds in general.
$\left. \begin{array}{l} W = \frac{L}{\lambda} \\ W_q = \frac{L_q}{\lambda} \end{array} \right\}$	Holds for ergodic systems.
$L = L_q + (1 - p_0)$	Holds for all single channel one at a time service queues.

Table 3.3 Relations among the performance measures. (continued)

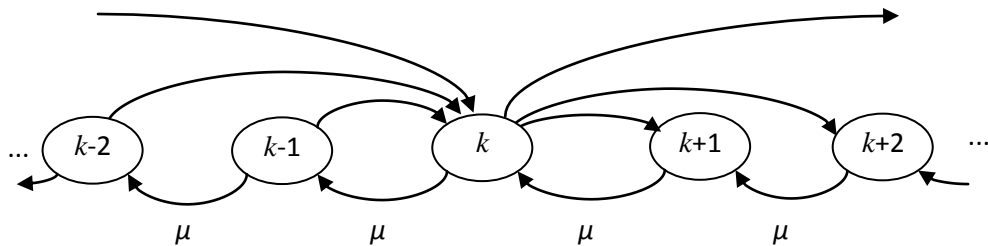
$L = L_q + \frac{\lambda}{\mu}$	Holds whenever Little's formulas are valid.
$p_0 = 1 - \frac{\lambda}{\mu}$	Holds for all single channel one at a time service queues in which Little's formulas are valid.
$W_q = L/\mu$	Limited use M/M/1 and like models.

3.4 Batch queue models

We know that Markovian queueing process can be studied as birth-and-death process. There the transitions occur to neighboring states. In this section, we consider Markovian models of **the non-birth-and-death type**. Transitions occur from a state to a state not necessarily neighboring (Medhi, 2003). The Chapman-Kolmogorov equations can be obtained in a similar way and memoryless property are still valid (Gross, & Harris, 1974).

3.4.1 Batch arrival systems - $M^x/M/1$ model

We take the number of customers in the system as state variable. We have the state-transition-rate diagram of Figure 3.7 (Ross, 2006). We can enter stage k from any stage below it (it is permitted batches of any size to arrive), and move from stage k to any state above it.

Figure 3.7 The batch arrival state-transition-rate diagram : $M^x/M/1$.

We define p_k to be the equilibrium probability for the number of customers in the system. Hence, the **equilibrium equations** are given by

$$(\lambda + \mu)p_k = \mu p_{k+1} + \sum_{i=0}^{k-1} p_i \lambda g_{k-i} \quad k \geq 1 \quad (3.23)$$

$$\lambda p_0 = \mu p_1 \quad (3.24)$$

3.4.1.1 Generation function, $E(X)$, and $V(X)$

We solve these equilibrium equations using the method of z -transforms. When both sides of (3.23) are multiplied by z^k we find

$$(\lambda + \mu)p_k z^k = \mu p_{k+1} z^k + \sum_{i=0}^{k-1} p_i \lambda g_{k-i} z^k \quad (3.25)$$

and, when both sides of the foregoing equation are summed from $n=1$ to ∞ , it is found that

$$(\lambda + \mu) \sum_{k=1}^{\infty} p_k z^k = \frac{\mu}{z} \sum_{k=1}^{\infty} p_{k+1} z^{k+1} + \sum_{k=1}^{\infty} \sum_{i=0}^{k-1} p_i \lambda g_{k-i} z^k \quad (3.26)$$

We interchange the order of summation for the double sum as following

$$\sum_{k=1}^{\infty} \sum_{i=0}^{k-1} = \sum_{i=0}^{\infty} \sum_{k=i+1}^{\infty} \quad (3.27)$$

The last term of (3.26) is found by using (3.27) as below

$$\begin{aligned} \sum_{k=1}^{\infty} \sum_{i=0}^{k-1} p_i \lambda g_{k-i} z^k &= \lambda \sum_{i=0}^{\infty} p_i z^i \sum_{k=i+1}^{\infty} g_{k-i} z^{k-i} \\ &= \lambda \sum_{i=0}^{\infty} p_i z^i \sum_{j=1}^{\infty} g_j z^j \end{aligned}$$

We firstly obtain to (3.26) as given by

$$(\lambda + \mu) \sum_{k=1}^{\infty} p_k z^k = \frac{\mu}{z} \sum_{k=1}^{\infty} p_{k+1} z^{k+1} + \lambda \sum_{i=0}^{\infty} p_i z^i \sum_{j=1}^{\infty} g_j z^j. \quad (3.28)$$

then

$$(\lambda + \mu)[P(z) - p_0] = \frac{\mu}{z}[P(z) - p_0 - p_1 z] + \lambda P(z)G(z).$$

Applying (3.24) we have

$$P(z) = \frac{\mu p_0 (1-z)}{\mu(1-z) - \lambda z[1-G(z)]}. \quad (3.29)$$

To eliminate p_0 we use $P(1)=1$. The application yields the indeterminate form $0/0$, so we must use L'Hospital's rule as follow:

$$\lim_{z \rightarrow 1} \left\{ \frac{[\mu p_0(1-z)]'}{[\mu(1-z) - \lambda z[1-G(z)]]'} \right\} = \lim_{z \rightarrow 1} \left\{ \frac{\mu p_0}{\mu + \lambda - \lambda G(z) - G'(z)\lambda z} \right\} = 1$$

We know $G(1)=1$ and $G'(1) = E(X)$, (see A1.2.3).

$$\begin{aligned} \Rightarrow \quad & \frac{\mu p_0}{\mu - E(X)\lambda} = 1 \\ \Rightarrow \quad & p_0 = 1 - \frac{\lambda E(X)}{\mu}. \end{aligned} \tag{3.30}$$

$G(z)$ is the distribution of batch size with $E(X)$. We know that ρ is the average arrival rate of customers times the average service time, $\rho = \lambda \left(\frac{1}{\mu} \right)$. In our case, the average arrival rate of customers is product of the average arrival rate of batch λ , and the average batch size $E(X)$. Namely, the average arrival rate of customer is $\lambda E(X)$.

Thus,

$$\rho = \lambda E(X) \left(\frac{1}{\mu} \right) \tag{3.31}$$

Substituting (3.31) to (3.30) we obtain

$$p_0 = 1 - \rho \tag{3.32}$$

Substituting (3.32) to (3.29) we obtain to the **generating function for the number of customers in the system.**

$$P(z) = \frac{\mu(1-\rho)(1-z)}{\mu(1-z) - \lambda z[1-G(z)]} \tag{3.33}$$

We now obtain to the expected number in the system $E(N)$, and the variance of the number in the system $V(N)$ from first derivative of the equation (3.33), and $V(N)$ from second derivative of the equation (3.33), respectively (Medhi, 2003).

Let the k th moment of the batch size be denoted by $a^{(k)} = E(X^k)$, $k \geq 2$. Note that $G'(1) = \bar{a}$, $G''(1) = a^{(2)} - \bar{a}$, and $G'''(1) = a^{(3)} - 3a^{(2)} + 2\bar{a}$, (see A1.2.3).

$$P(z) = \frac{\mu(1-\rho)(1-z)}{\mu(1-z) - \lambda z[1-G(z)]} = \frac{N(z)}{D(z)} \quad (3.34)$$

$$E(X) = \left. \frac{dP(z)}{dz} \right|_{z=1} = \left. \frac{N'(z)D(z) - D'(z)N(z)}{[D(z)]^2} \right|_{z=1}$$

Since $N(1)=D(1)=0$, the application yields the indeterminate form $0/0$, so we must use L'Hospital's rule twice as follow:

$$\begin{aligned} E(X) = P'(1) &= \left. \frac{N''(z)D(z) - D''(z)N(z)}{2D'(z)D(z)} \right|_{z=1} \\ &= \left. \frac{N'''(z)D(z) + N''(z)D'(z) - N'(z)D''(z) - N(z)D'''(z)}{2D''(z)D(z) + 2[D'(z)]^2} \right|_{z=1} \end{aligned} \quad (3.35)$$

$$\begin{cases} N(z) = \mu(1-\rho)(1-z) & \Rightarrow N(1) = 0 \\ N'(z) = -\mu(1-\rho) & \Rightarrow N'(1) = -\mu(1-\rho) \\ N''(z) = 0 & \Rightarrow N''(1) = 0 \\ N'''(z) = 0 & \Rightarrow N'''(1) = 0 \end{cases}$$

$$\begin{cases} D(z) = \mu(1-z) - \lambda[1-G(z)] & \Rightarrow D(1) = 0 \\ D'(z) = -\mu - \lambda + \lambda G(z) + \lambda z G'(z) & \Rightarrow D'(1) = -\mu + \lambda \bar{a} \quad (\text{or } D'(1) = -\mu(1-\rho)) \\ D''(z) = 2\lambda G'(z) + \lambda z G''(z) & \Rightarrow D''(1) = \lambda(a^{(2)} + \bar{a}) \\ D'''(z) = 3\lambda G''(z) + \lambda z G'''(z) & \Rightarrow D'''(1) = \lambda(a^{(3)} - \bar{a}) \end{cases}$$

Substituting these moments to (3.35) we obtain to the **expected number in the system**:

$$E(N) = \frac{\rho}{1-\rho} \left(\frac{a^{(2)} + \bar{a}}{2\bar{a}} \right). \quad (3.36)$$

For the variance, we use the below equation. Note that $V(N) = E(N^2) - [E(N)]^2$ and $P''(1) = E(N^2) - E(N)$.

$$V(N) = \frac{d^2 P(z)}{dz^2} \Big|_{z=1} + E(N) - [E(N)]^2. \quad (3.37)$$

$$P''(1) = \frac{D(z)[D(z)N''(z) - N(z)D''(z)] - [2D'(z)[D(z)N'(z) - N(z)D'(z)]]}{[D(z)]^3} \Big|_{z=1}$$

The application yields the indeterminate from 0/0, so we apply L'Hospital's rule three times and obtain

$$P''(1) = \frac{-N'(1)[2D'(1)D'''(1) - 3[D''(1)]^2]}{6[D'(1)]^3}. \quad (3.38)$$

Substituting above moments to (3.38) we obtain

$$P''(1) = \frac{\lambda(a^{(3)} - \bar{a})}{3\mu(1-\rho)} + \frac{\lambda^2(a^{(2)} + \bar{a})^2}{2\mu^2(1-\rho)^2}. \quad (3.39)$$

Substituting (3.36) and (3.39) to (3.37), we obtain to the **variance of the number in the system** $V(N)$ as follow:

$$V(N) = \frac{\rho}{1-\rho} \left(\frac{2a^{(3)} + 3a^{(2)} + \bar{a}}{6\bar{a}} \right) + [E(N)]^2. \quad (3.40)$$

3.4.1.2 Waiting times

The waiting time of a test unit consists of two component:

- 1) the time required to complete service of all the units in the system found by an arriving group, D_1

$$E(D_1) = \frac{\lambda}{2\mu^2(1-\rho)}(a^{(2)} + \bar{a}) \quad (3.41)$$

- 2) the time to serve all units of the group who are served prior to the start of service of the of the test unit, D_2

$$E(D_2) = \frac{1}{2\mu\bar{a}}(a^{(2)} + \bar{a}) - \frac{1}{\mu}. \quad (3.42)$$

The **expected waiting time in the system** of a test unit is given by

$$E(W) = E(D_1) + E(D_2) + \frac{1}{\mu}$$

$$E(W) = \frac{1}{2\mu(1-\rho)} \left(\frac{a^{(2)} + \bar{a}}{\bar{a}} \right). \quad (3.43)$$

Using Little's Law, we get $E(N) = (\lambda\bar{a})E(W)$. On the other hand, using $W = W_q + (1/\mu)$, we get **the expected waiting time in the queue** of a test unit is given by

$$E(W_q) = \frac{a^{(2)} - \bar{a}(1-2\rho)}{2\mu\bar{a}(1-\rho)}. \quad (3.44)$$

3.4.2 Batch arrival systems with fixed batch size- $M^r/M/1$ model

The system is special case of the system $M^x/M/1$ in previous section. Here, all batch sizes are the same, namely,

$$g_k = \begin{cases} 1 & k = r \\ 0 & k \neq r \end{cases}$$

In system $M/E_r/1$ each customer has to pass through r stages of service to complete his total service. The solution of this system is to count the number of service stages remaining in the system, each customer contributing r stages. Now we consider each "customer" arrival to be the arrival of r customers. Each of these r customers will require only a single stage of service.

The service consists of r stages, each exponential with mean $(1/r\mu)$, and the completion of the service requires total service in r stages with mean $r(1/r\mu) = 1/\mu$. The system is considered as one in which “each customer” arrival amounts to arrival of “ r customers”. Each of these require a single stage of exponential service with mean $(1/r\mu)$. Thus, the distribution of the number of customers in the system $M^r/M/1$ is the same as the distribution of the number of stages in the system $M/E_r/1$.

If we want to draw the state-transition-rate diagram for the number of customers in the system, then the batch arrival system lead to the diagram for the number of stages in the system $M/E_r/1$ (Kleinrock, 1975). We must make the minor modification that μ must now replaced by $r\mu$.

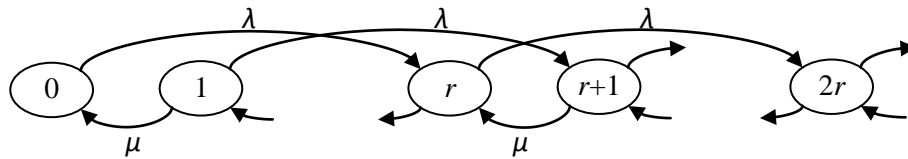


Figure 3.8 The batch arrival state-transition-rate diagram : $M^r/M/1$.

We define p_k to be the equilibrium probability for the number of customers in the system. For this the **equilibrium equations** are given by

$$\begin{aligned} \lambda p_0 &= \mu p_1 & k &= 0 \\ (\lambda + \mu) p_k &= \mu p_{k+1} & 1 \leq k &\leq r-1 \\ (\lambda + \mu) p_k &= \lambda p_{k-1} + \mu p_{k-1} & k &\geq r. \end{aligned}$$

3.4.2.1 Generating function, $E(X)$, and $V(X)$

We solve these equilibrium equations using the method of z -transforms. We obtain the generating function $P(z)$ for the system. In another way, for the batch arrival system with fixed batch size r , we have $G(z) = z^r$. Substituting this to equation (3.33) and the **generating function for this system** is obtained as follow:

$$P(z) = \frac{\mu(1-\rho)(1-z)}{\mu(1-z) - \lambda z [1-z^r]} \quad (3.45)$$

We obtain moments of the probability distribution by calculating $F^{(n)}(1)$. In here, $F(z)$ is a probability generating function, and f_n is a probability distribution of a discrete random variable X .

$$F(z) = \sum_{n=0}^{\infty} z^n f_n$$

The first factorial moment:

$$\begin{aligned} F'(z) &= \sum_{n=0}^{\infty} n z^{n-1} f_n & \Rightarrow F'(1) &= \sum_{n=0}^{\infty} n f_n \\ & & \Rightarrow F'(z) &= E(X) \end{aligned}$$

The second factorial moment:

$$\begin{aligned} F''(z) &= \sum_{n=0}^{\infty} n(n-1) z^{n-2} f_n & \Rightarrow F''(1) &= \sum_{n=0}^{\infty} n(n-1) f_n \\ & & \Rightarrow F''(1) &= \sum_{n=0}^{\infty} n^2 f_n - \sum_{n=0}^{\infty} n f_n \\ & & \Rightarrow F''(z) &= E(X^2) - E(X) \end{aligned}$$

The third factorial moment:

$$\begin{aligned} F'''(z) &= \sum_{n=0}^{\infty} n(n-1)(n-2) z^{n-3} f_n & \Rightarrow F'''(1) &= \sum_{n=0}^{\infty} n(n-1)(n-2) f_n \\ & & \Rightarrow F'''(1) &= \sum_{n=0}^{\infty} n^3 f_n - 3 \sum_{n=0}^{\infty} n^2 f_n + 2 \sum_{n=0}^{\infty} n f_n \\ & & \Rightarrow F'''(z) &= E(X^3) - 3E(X^2) + 2E(X) \end{aligned}$$

Using results above we obtain moments of distribution of batch size as follow:

$$\begin{aligned} G'(z) &= r z^{r-1} & \Rightarrow G'(1) &= r \\ & & \Rightarrow E(X) &= \bar{a} = r. \end{aligned} \quad (3.46)$$

$$\begin{aligned}
G''(z) = r(r-1)z^{r-2} & \Rightarrow G''(1) = r^2 - r \\
& \Rightarrow E(X^2) - E(X) = r^2 - r \\
& \Rightarrow E(X^2) = a^{(2)} = r^2.
\end{aligned} \tag{3.47}$$

$$\begin{aligned}
G'''(z) = r(r-1)(r-2)z^{r-3} & \Rightarrow G'''(1) = r^3 - 3r^2 + 2r \\
& \Rightarrow E(X^3) - 3E(X^2) - 2E(X) = r^3 - 3r^2 + 2r \\
& \Rightarrow E(X^3) = a^{(3)} = r^3.
\end{aligned} \tag{3.48}$$

Substituting results in (3.46) and (3.47) to equation (3.36) we obtain the **expected number in the system**, $E(N)$,

$$E(N) = \frac{\rho}{1-\rho} \left(\frac{r+1}{2} \right). \tag{3.49}$$

Substituting results in (3.46), (3.47), and (3.48) into equation (3.40) we obtain the **variance of the number in the system**, $V(N)$,

$$V(N) = \frac{\rho}{(1-\rho)^2} \left(\frac{(4r^2 + 6r + 2) + \rho(1-r^2)}{12} \right). \tag{3.50}$$

3.4.2.2 Waiting times

Substituting results in (3.46) and (3.47) into equation (3.43) we obtain the **expected waiting time in the system**, $E(W)$,

$$E(W) = \frac{(r+1)}{2\mu(1-\rho)} = \frac{(r+1)}{2(\mu-r\lambda)}. \tag{3.51}$$

Substituting results in (3.46) and (3.47) into equation (3.44) we obtain the **expected waiting time in the queue**, $E(W_q)$,

$$E(W_q) = \frac{(r+2\rho-1)}{2\mu(1-\rho)} = \frac{\mu(r-1) + 2r\lambda}{2\mu(\mu-r\lambda)}. \tag{3.52}$$

3.4.2.3 The performance measures

Using Little's Law, we get $E(N) = (r\lambda)E(W)$ and $E(N_q) = (r\lambda)E(W_q)$. We can summarize the performance measures in Table 3.4.

Table 3.4 Performance measures for batch arrival system with fixed batch size r

Batch size: r	ρ	W	L	W_q	L_q
2	$\frac{2\lambda}{\mu}$	$\frac{3}{2} \left(\frac{1}{\mu - 2\lambda} \right)$	$2\lambda.W$	$\frac{4\lambda + \mu}{2\mu(\mu - 2\lambda)}$	$2\lambda.W_q$
3	$\frac{3\lambda}{\mu}$	$2 \left(\frac{1}{\mu - 3\lambda} \right)$	$3\lambda.W$	$\frac{3\lambda + \mu}{\mu(\mu - 3\lambda)}$	$3\lambda.W_q$
4	$\frac{4\lambda}{\mu}$	$\frac{5}{2} \left(\frac{1}{\mu - 4\lambda} \right)$	$4\lambda.W$	$\frac{8\lambda + 3\mu}{2\mu(\mu - 4\lambda)}$	$4\lambda.W_q$
5	$\frac{5\lambda}{\mu}$	$3 \left(\frac{1}{\mu - 5\lambda} \right)$	$5\lambda.W$	$\frac{5\lambda + 2\mu}{\mu(\mu - 5\lambda)}$	$5\lambda.W_q$

3.4.3 Batch service systems- $M/M^x/1$

We consider systems where services are offered in batches at a time. The policies of batch service are considered in literature are given below (Medhi, 2003).

(1) The server serves in batch of size not more than the maximum capacity of the server, b units. If the server finds not more than b units waiting, then he takes all of them in a batch for service. If he finds more than b units waiting, then he takes for service a batch of a size b units. The others wait and join the queue. For example, an elevator is this type of server.

(2) A service batch may be of a fixed size, k units. The server waits until there are k units in the queue and starts service as soon as the queue reaches this size. If he finds more than k units waiting, then he takes a batch of size k units. The others wait in the queue.

(3) The server takes in a batch a minimum number, a units which is less than or equal to his capacity b units. If he finds q units waiting, then he adopts the following policy:

- i) $0 \leq q < a$, he waits until the queue size grows to a units,

- ii) $a \leq q \leq b$, he takes a batch of size q units for service,
- iii) $q > b$, he takes a batch of size b units for service. The others wait in the queue.

This rule is called the general batch service rule. We can give an example for this rule as following application to traffic flow (Neuts, 1967). We consider a main and a minor road. A traffic light on the main road interrupts its traffic flow after a fixed length of time if at least a cars have activated a tripplate on the minor road. Otherwise the light stays green until a cars have arrived. The red cycle is timed so that at most b cars can merge during it. We count as service times the time required for the platoon to merge, plus the fixed length of the green cycle on the main road. The model then studies the queue forming on the minor road.

3.4.3.1 Batch service systems- in policy (1)

The arrivals occur to a single-channel facility as Poisson process. They are served FCFS. There is no waiting capacity constraint. At the system, r customers are served at a time, otherwise, if there are customers less than r , all customers are served. We take the number of customers in the system as state variable. We have the state-transition-rate diagram of Figure 3.9 (Kleinrock, 1975).

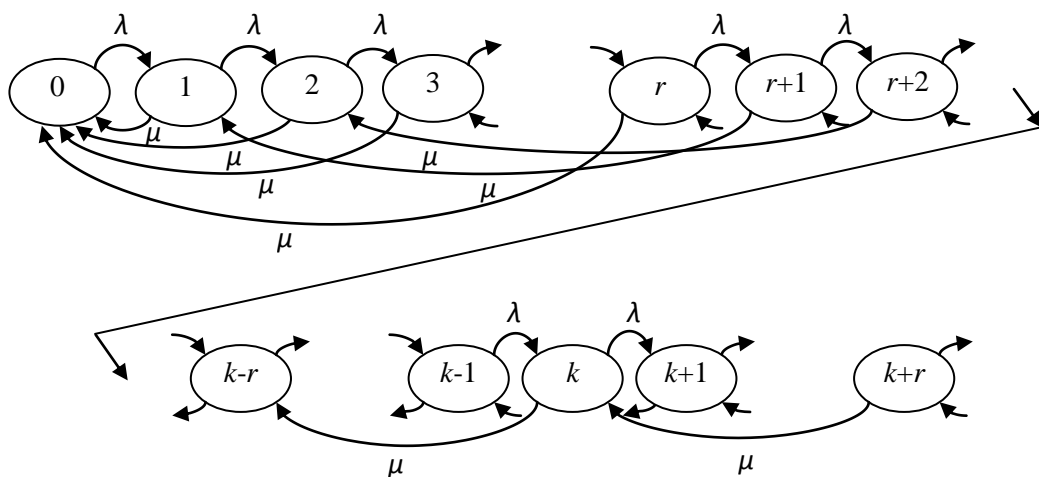


Figure 3.9 The batch service state-transition-rate diagram : $M/M^r/1$.

In this figure all stages (except for stage 0) behave in the same way. They are entered from their left-hand neighbor by an arrival, and from their neighbor r units to

the right by a group departure. They are exited by either an arrival or a group departure. On the other hand, stage 0 can be entered from any one of the r stages to its right and can be exited only by an arrival.

We define p_k to be the equilibrium probability for the number of customers in the system. For this the **equilibrium equations** are given by

$$\begin{aligned} (\lambda + \mu)p_k &= \mu p_{k+r} + \lambda p_{k-1} & k \geq 1 \\ \lambda p_0 &= \mu(p_1 + p_2 + \dots + p_r) \end{aligned} \quad (3.53)$$

We may solve the equilibrium equations by using operator notation or generating function.

Firstly, we solve the equilibrium equations by using operator notation. The first line of (3.53) may be rewritten as follows

$$\mu p_{k+1+r} - (\lambda + \mu)p_{k+1} + \lambda p_k = 0$$

and then, the last equation is written in operator notation as follows

$$\begin{aligned} \mu D^{r+1} p_k - (\lambda + \mu) D p_k + \lambda p_k &= 0 \\ [\mu D^{r+1} - (\lambda + \mu) D + \lambda] p_k &= 0 \quad (k \geq 0) \end{aligned} \quad (3.54)$$

The operator has roots (r_1, \dots, r_{r+1}) , then

$$p_k = \sum_{i=1}^{r+1} C_i r_i^k \quad (k \geq 0). \quad (3.55)$$

Each r_i must be less than one or $C_i = 0$ for all r_i not less than one, since $\sum_{k=0}^{\infty} p_k = 1$.

Thus, it is found only one root in $(0,1)$ and denoted by r_0 . Therefore, we can rewrite the equation (3.55) as follows

$$p_k = C r_0^k \quad (k \geq 0, 0 < r_0 < 1). \quad (3.56)$$

We now apply the boundary condition that $\sum_{k=0}^{\infty} p_k = 1$;

$$\left. \begin{array}{l} p_0 = C \\ p_1 = Cr_0 \\ p_2 = Cr_0^2 \\ p_3 = Cr_0^3 \\ \vdots \end{array} \right\} \quad \begin{array}{l} 1 = C[1 + r_0 + r_0^2 + \dots] \\ 1 = p_0 \left(\frac{1}{1-r_0} \right) \end{array}$$

and we have

$$C = p_0 = 1 - r_0. \quad (3.57)$$

Finally, we substitute (3.57) to (3.56), and obtain to the **distribution for the number of customers in the system:**

$$p_k = (1 - r_0)r_0^k \quad (k \geq 0, 0 < r_0 < 1). \quad (3.58)$$

We now solve the equilibrium equations by using generating function. On the other hand, we solve these equilibrium equations by using the method of z -transforms.

We first apply the z -transform to the first line of (3.53). When both sides of (3.53) are multiplied by z^k we find

$$(\lambda + \mu)p_k z^k = \mu p_{k+r} z^k + \lambda p_{k-1} z^k$$

When both sides of the foregoing equation are summed from $k=1$ to ∞ , it is found that

$$\begin{aligned} (\lambda + \mu) \sum_{k=1}^{\infty} p_k z^k &= \mu \sum_{k=1}^{\infty} p_{k+r} z^k + \lambda \sum_{k=1}^{\infty} p_{k-1} z^k \\ (\lambda + \mu)[P(z) - p_0] &= \frac{\mu}{z^r} \left[P(z) - \sum_{k=0}^r p_k z^k \right] + \lambda z P(z) \\ P(z) &= \frac{\mu \sum_{k=0}^r p_k z^k - (\lambda + \mu) p_0 z^r}{\lambda z^{r+1} - (\lambda + \mu) z^r + \mu} \end{aligned}$$

The negative term in the numerator of last equation is rewritten by using the second line of (3.53), and substituted to the last equation as follows:

$$-z^r (\lambda + \mu) p_0 = -\mu z^r \sum_{k=0}^r p_k$$

$$P(z) = \frac{\mu \sum_{k=0}^r p_k z^k - \mu z^r \sum_{k=0}^r p_k}{\lambda z^{r+1} - (\lambda + \mu) z^r + \mu} = \frac{\mu \left[(p_0 + p_1 z + \dots + p_r z^r) - (p_0 z^r + p_1 z^r + \dots + p_r z^r) \right]}{\mu r \left[\frac{\lambda}{r\mu} z^{r+1} - \frac{\lambda z^r}{r\mu} - \frac{\mu z^r}{r\mu} + \frac{1}{r} \right]}.$$

We know that $\rho = \lambda/r\mu$, because r customers are served at a time, and so have

$$P(z) = \frac{\sum_{k=0}^{r-1} p_k (z^k - z^r)}{r\rho z^{r+1} - (1+r\rho)z^r + 1}.$$

The generating function $P(z)$ must converge inside the unit cycle. Only one root is in interval $(0,1)$. We denote by z_0 .

The **generating function for the number of customers in the system** is given by

$$P(z) = \frac{1-1/z_0}{1-z/z_0}. \quad (3.59)$$

We invert the last equation to obtain the **distribution for the number of customers in the system** is given by

$$p_k = \left(1 - \frac{1}{z_0}\right) \left(\frac{1}{z_0}\right)^k \quad k = 0, 1, 2, \dots \quad (3.60)$$

We say that there is a relation between the root is found by the generating function z_0 and the root is found by using of operators r_0 (Gross, & Harris, 1974);

$$r_0 z_0 = 1. \quad (3.61)$$

We substitute the relation in (3.61) to (3.60), and we can obtain to the same **distribution for the number of customers in the system** in (3.58):

$$p_k = (1-r_0)r_0^k \quad k = 0, 1, 2, \dots \quad (3.62)$$

3.4.3.2 The performance measures

The performance measures for this model are obtained similar to M/M/1 system by substituting r_0 to ρ .

L : Expected number of customers in queuing system.

$$L = \frac{r_0}{1 - r_0} \quad (3.63)$$

L_q : Expected queue length (excludes customers being served).

$$L_q = \frac{r_0^2}{1 - r_0} \quad (3.64)$$

\hat{W} : Waiting time in system (includes service time) for each individual customer.

$W = E(\hat{W})$.

$$W = \frac{r_0}{\lambda(1 - r_0)} \quad (3.65)$$

\hat{W}_q : Waiting time in queue (excludes service time) for each individual customer.

$W_q = E(\hat{W}_q)$.

$$W_q = \frac{r_0^2}{\lambda(1 - r_0)} \quad (3.66)$$

Finally, we can write equations for finding root in Table 3.5.

Table 3.5 Equations for finding root

Batch size	Equations for finding root
1	$\mu r^2 - (\lambda + \mu)r + \lambda = 0$
2	$\mu r^3 - (\lambda + \mu)r + \lambda = 0$
3	$\mu r^4 - (\lambda + \mu)r + \lambda = 0$
4	$\mu r^5 - (\lambda + \mu)r + \lambda = 0$

CHAPTER FOUR

SIMULATION

We first give a brief information concerning simulation in Section 4.1. Afterwards, in Section 4.2, we discuss discrete event simulation, the performance measures, and statistical analysis of simulation output.

4.1 Introduction to simulation

Simulation can be defined as a technique, and it is the most popular technique of operation research. The technique of simulation usually involves using a computer to simulate the real-world process or system over time, and it is used to describe and analyze the behaviors of a system. We prefer to simulate the system rather than doing experiments on the real system, because the system may not exist yet, or the experiment of the system may be too expensive, time consuming and dangerous.

The **advantages of simulation modeling** can be summarized as follows (Banks et al., 2001):

1. Simulation enables the study of a complex system.
2. Informational, and environmental changes can be simulated, then the effect of these changes to the model's behavior can be observed.
3. The most important variables can be obtained by changing simulation inputs and observing the resulting outputs.
4. Simulation can be used as an educational device to reinforce analytic solution methodologies.
5. Simulation methods are easier to apply than analytical methods, for example, the analytical solution can consist of differential equations of high order.
6. Simulation enables better understanding by animation.

The **disadvantages of simulation modeling** can be summarized as follows (Banks et al., 2001):

1. Simulation results can be difficult to interpret
2. Simulation modeling and analysis can be time consuming and expensive

Some common types of application of simulation are given as follow (Hiller, & Lieberman, 2001):

1. Design and operation of queuing systems.
2. Managing inventory systems.
3. Design and operation of manufacturing systems.
4. Design and operation of distribution systems.
5. Financial risk analysis.
6. Health care applications.

There are two types of simulation; (i) *discrete event simulation* which uses a mathematical/logical model of a physical system that determines state changes at constant points in simulated time. The system state variables remain constant over intervals of time and change value only at certain points, namely, at event times. (ii) *Continuous simulation* which uses models of a physical system that do not determine constant time and state relationships, and the system state variables defined by differential or difference equations may change continuously over time (Albrecht, & Az, 2010). We consider the discrete-event simulation for the modeling of queuing systems.

In simulation the input data are random variables, whereas the output results may vary depending on the simulation length. When analyzing simulation output data, there are two types of simulation; *finite-horizon* (or *terminating*) ones and *steady-state* (or *non-terminating*) ones (Banks et al., 2001). The first type is related to performance measures evaluated over a sample path of finite length. The initial conditions of the system at time 0 and the stopping time must be specified. On the other hand, a steady-state simulation concerns the long-term behavior of the system of interest.

Initially the queueing system is empty and it takes some time for the system to reach the steady-state situation. The initial stage is called the *transient* or *warm-up* period (Kolahi, 2011). The warm-up period occurs in non-terminating simulations, but in some instances it can also occur in terminating simulations. Collecting data

during this period can affect the accuracy of the results. Therefore, a warm-up period must be searched in a simulation study.

4.2 Discrete event simulation

In every simulation study, the passage of time is the most important notion. So, every model contains a variable which is called the simulation clock. We advance simulated time by two approach such that (i) *fixed-increment time advance or time step* approach that the simulation clock is updated by the same time increment, and (ii) *next-event time advance* approach that time is advanced from the time of the current event to the time of the next scheduled event (Buss, & Rowaei, 2010). We consider to the next-event time advance approach for the modeling of queuing systems.

We consider a common framework for the modeling of queuing systems using discrete-event simulation, and next-event time advance. Therefore, we use simulation model for predicting the steady state behavior of the systems whose states change only at discrete points in time.

We define some components (used in the applications) for discrete event simulation models:

1. **Simulation clock:** It is used to track the current simulated time (Banks et al., 2001).
2. **System state:** A collection of variables, system variables, contains all the information to describe the system at any time (Banks et al., 2001). In our case, one of the system states correspond to the state of the server SS , which represents whether the server is idle or not. If the server is idle, the arrival customer enters to service. Otherwise, the arrival customer joins the queue. The other system state is the number of waiting customers in queue (WL) which represents whether there is any customer in the queue at the end of any service. If there is any customer in the queue, the server becomes busy for removing the first customer in the queue to service. Otherwise, the server becomes idle.

3. **Event** : An instantaneous occurrence that changes the state of a system (Banks et al., 2001). We have two events. One of them is *arrival event* which represents an arrival into the system. If the server is idle, it becomes busy, or the number of customers in the queue increases. The other one is *departure event* which represents a departure from the system. If the server is busy, it becomes idle, or the number of customer in the queue reduces.
4. **Event list**: A list of event contains information for future events that ordered by time of occurrence (Banks et al., 2001). We schedule corresponding events to describe the next event.
5. **Ending condition**: A time is specified by the simulation designer. It ends the simulation (Banks et al., 2001). Namely, if the simulation clock equal or exceed to the specified time, the simulation run ends.
6. **Random number generator**: It is used to generate random numbers from specified distributions (Banks et al., 2001). For example, we generate random numbers from exponential distribution for the service times of customers and the interarrival times of customers.
7. **Simulation table**: It provides a systematic method to track system state over time (Banks et al., 2001).

Firstly, we summarize the logical relationships among these components. The simulation starts with the initialization such that the simulation clock, system state, statistical counters and event list are initialized. Then, if an event i is the next to occur, the simulation clock is advanced to the time of the event i will occur. At this point, system state updating, occurrence of times of future events and gathering of information about system performance. Then the results is recorded in simulation table, and a check is made whether the simulation clock equal or exceed to specified time. This is the case, the simulation should be terminated. If simulation is to be terminated, performance metrics are computed and printed. Otherwise, the simulation continues to the previous cycle of events until the simulation clock catches to the specified time.

Secondly, we show to these components in a table as follow. Note that, the stopping of the simulation related to the **ending condition**, and the variables in the event list are obtained by the **random number generator**.

Table 4.1 Simulation table

	System Variables			Event List	
Type of Event	Simulation Clock	Status of Server, SS	Length of Waiting Line, WL	Time of the Next Arrival, AT	Time of the Next Departure, DT
	Specified Time				

4.2.1 The performance measures

Our purpose is to obtain the performance measures by simulation. For this purpose, we use the average number in the system L by a performance measure. If the run of the simulation is made *very long*, and the time-weighted average of the number in the system L_T is computed, we know that this converges to the true average number in the system L (Gross, & Harris, 1974). Namely, if we run the simulation for a length of time T and assume computation of average number in the system is L_T , then we can easily see that

$$\lim_{T \rightarrow \infty} L_T = L.$$

In order to calculate the performance measures in simulation, the equations are given as following (Taha, 2003):

The expected number of customers in system:

$$\left(\begin{array}{c} \text{Average number} \\ \text{in system} \end{array} \right) = \left(\sum_{i=1}^n \begin{array}{c} \text{Waiting time in system} \\ \text{of customer } i \end{array} \right) // \left(\begin{array}{c} \text{clock time} \\ \text{of simulation} \end{array} \right) \quad (4.1)$$

The expected waiting time in system for each individual customer:

$$\left(\begin{array}{c} \text{Average waiting} \\ \text{time in system} \end{array} \right) = \left(\sum_{i=1}^n \begin{array}{c} \text{Waiting time in system} \\ \text{of customer } i \end{array} \right) / n \quad (4.2)$$

The expected waiting time in queue for each individual customer:

$$\left(\begin{array}{c} \text{Average waiting} \\ \text{time in queue} \end{array} \right) = \left(\sum_{i=1}^n \begin{array}{c} \text{Waiting time in queue} \\ \text{of customer } i \end{array} \right) / n \quad (4.3)$$

The expected number of customers in queue:

$$\left(\begin{array}{c} \text{Average queue} \\ \text{length} \end{array} \right) = \left(\sum_{i=1}^n \begin{array}{c} \text{Waiting time in queue} \\ \text{of customer } i \end{array} \right) / \left(\begin{array}{c} \text{clock time} \\ \text{of simulation} \end{array} \right) \quad (4.4)$$

The expected service facility:

$$\left(\begin{array}{c} \text{Average service} \\ \text{facility} \end{array} \right) = \left(\sum_{i=1}^n \begin{array}{c} \text{Service time} \\ \text{of customer } i \end{array} \right) / \left(\begin{array}{c} \text{clock time} \\ \text{of simulation} \end{array} \right) \quad (4.5)$$

4.2.2 Statistical analyses of simulation output

In the previous section, we have shown to obtain the performance measures by simulation. However, L_T may be too large depending on volume of traffic in the system, so it does not allow us to make any precise statement on the accuracy of L_T as an estimate of L . In this section, our purpose is to obtain the confidence interval estimate and point estimate by repeating runs of the simulation. Namely, our purpose is to make a precise statement. For this purpose, we use again the average number in the system L by a performance measure.

"If we have ergodicity, L is also the expected value of the ensemble average. The ensemble process can be approached by repeating runs of the simulation, keeping all design parameters constant, but using a different random number stream each time, that is, changing the starting seed used in the random number generator" (Gross, & Harris, 1974, p.486).

The ensemble average number in the system:

$$\lim_{p \rightarrow \infty} \sum_{i=1}^p \frac{L_i}{p} = L.$$

It is not possible to take an average over a infinite set, but we can make it over a finite set. To do so, we want to obtain a finite random sample. Then, we can obtain the performance measures by point estimate, and make precise statement on the accuracy of this *estimate* by obtaining confidence interval estimate.

We can obtain point estimate of L by *the average of a finite random sample*. Namely, we need a point estimator which is unbiased, and know that the sample mean \bar{L} is an unbiased estimator of the population mean L . At this point, we have obtained the random sample with size p by multiple runs, so we can say that L is population mean and L_1, L_2, \dots, L_p are independent random variables which are obtained from this population. We assume that this population has variance σ_L^2 . So we obtain the point estimate of L as follows:

$$\hat{L} = \sum_{i=1}^p \frac{L_i}{p}. \quad (4.6)$$

We now require to obtain a confidence interval estimate of L . At this point, we need the estimate of the variance of \hat{L} , namely, \bar{L} . We know that

$$V(\bar{L}) = \frac{\sigma_L^2}{p}.$$

The sample variance S^2 is an unbiased estimator of the population variance σ_L^2 , so we can rewrite to above equation as following

$$V(\hat{L}) = \frac{S^2}{p}.$$

This equation gives us the estimate of the variance of \hat{L} as follows

$$S_L^2 = \frac{1}{p} \frac{\sum_{i=1}^p (L_i - \hat{L})^2}{p-1}. \quad (4.7)$$

According to the central limit theorem, if p is large, the L_i 's are going to be approximately normally distributed. So we obtain to the confidence interval estimate of L as follows

$$\hat{L} - z_{\alpha/2} S_{\hat{L}} \leq L \leq \hat{L} + z_{\alpha/2} S_{\hat{L}} \quad (4.8)$$

In here, $z_{\alpha/2}$ is obtained from the standard normal tables and α is confidence level.

CHAPTER FIVE APPLICATION

In this chapter, we deal with simulation model of queueing systems both with batch arrival and with batch service. We first describe the batch arrival system, and give details corresponding the simulation model of the system in Section 5.1. Similarly, the batch service system is described, and the details of the simulation model of the system is given in Section 5.2.

5.1 Simulation model of queueing systems with batch arrival

In simulation of queueing system, firstly we have to describe the system. For this single-server system, we assume that arrivals are drawn from an infinite calling population. There is a unlimited waiting room capacity. The customers are served in the order of their arrival, FCFS. We assume each “customer” arrival to be the arrival of $r = 3$ customers at a time, and the distribution of interarrival times to be the exponential distribution. Each of these $r = 3$ customers requires only a single stage of service with the distribution of service times as exponential distribution. After service, all customers return to the calling population. This queueing system is represented in Figure 5.1.

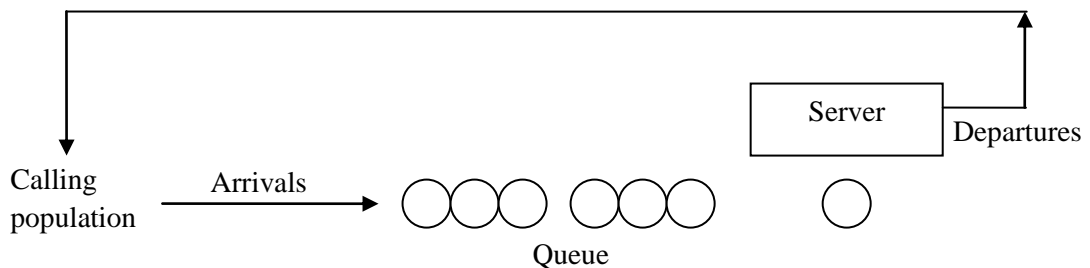


Figure 5.1 Single-server queueing system with batch arrival.

In order to explain the simulation model, we have provided a flowchart which consists of three steps: *the first step*, *the second step*, and *the third step*. And then, all the blocks in this flowchart have been numbered for easy reference (see Figure 5.2).

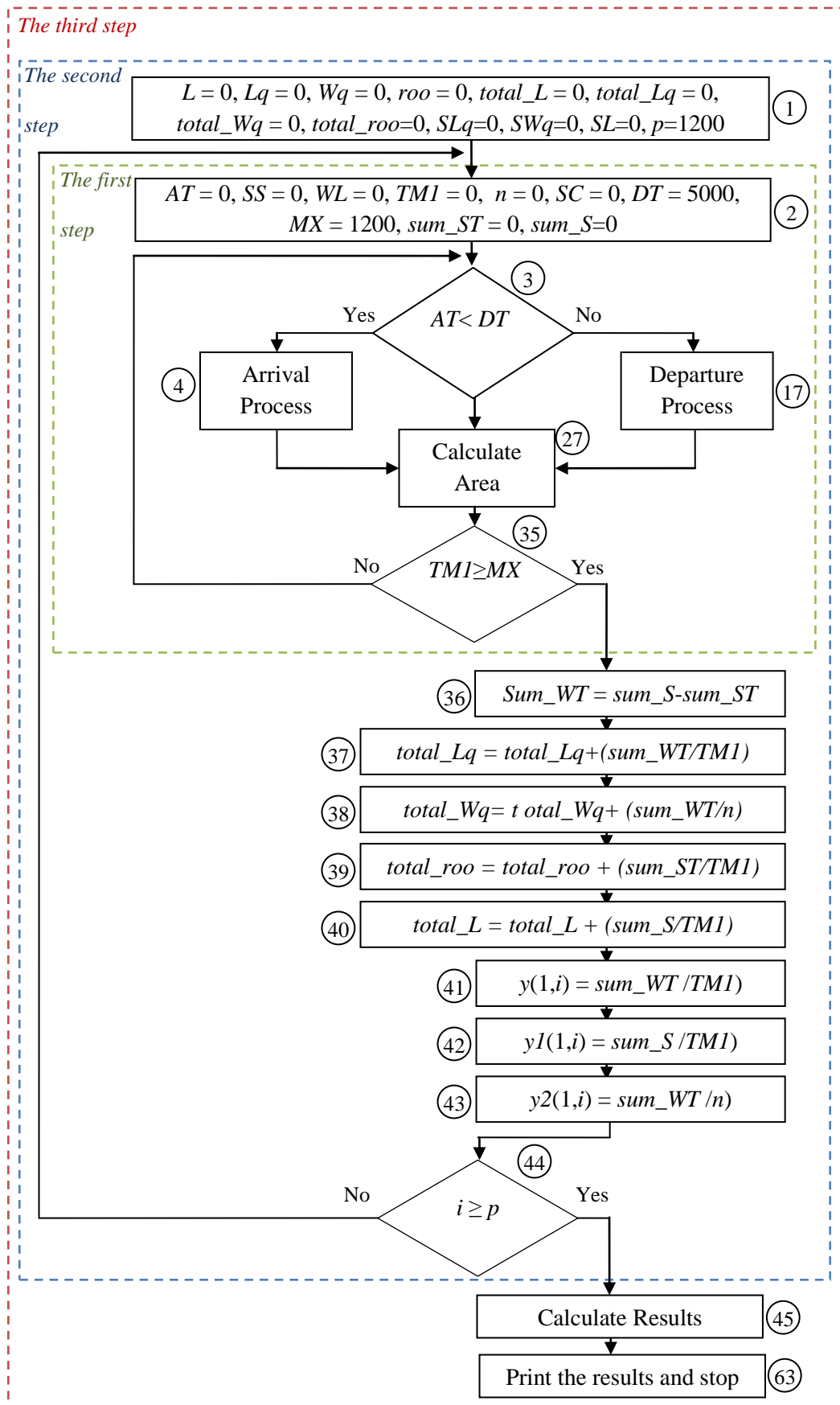


Figure 5.2 Flowchart for the queuing system with batch arrival.

The first step: This step consists of block 2, 3, 4, 17, 27, and 35 in the flowchart in Figure 5.2. All of the events occur in the first step. Namely, the arrival enters to the system at block 4, and the departure leaves the system at block 17. On the other hand, the total waiting time in the system of customers, Sum_S , and the total number of customers in the system, n , are calculated at block 27. The total service time, Sum_ST , is calculated at block 4 and 17. The simulation is run once at the first step. Therefore, at the end of the first step, we achieve values for Sum_S , Sum_ST , and n . On the other hand, only one single value for each of the following variables is obtained: the average queue length, L_q , the average number in the system, L , the service facility, roo , and the average waiting time in the queue, W_q .

The second step: This step consists of all blocks except for block 45 and 63. The total waiting time in the queue of customers, Sum_WT , is calculated by using the results achieved in the first step. When the simulation is run once, which means that the number of replication, p , is 1, the obtained results don't make it possible to get a variance and confidence interval in the third step. Therefore, we set $p = 1200$, namely, the simulation is run 1200 times. At the end of the second step, the values for the following variables are obtained: y_i ($i. L_q$), $y1_i$ ($i. L$), and $y2_i$ ($i. W_q$) where $i=1, \dots, 1200$, and also $total_L_q$ (sum of L_q), $total_L$ (sum of L), $total_W_q$ (sum of W_q), and $total_roo$ (sum of roo).

The third step: This step consists of all blocks. Using the results of the first and the second step, we are able to get the final results of the simulation. At the end of the third step, the values for the following variables are obtained: The point estimates, \hat{L} , \hat{L}_q , \hat{W}_q , and roo , the standard error estimates, $S_{\hat{L}}$, $S_{\hat{L}_q}$, and $S_{\hat{W}_q}$, and the confidence interval estimates for L , L_q , and W_q .

We start this simulation with an empty system and assume that our first event, a batch arrival (fixed batch size 3), takes place at clock time 0. This arrival finds the server idle and enters service immediately. Arrivals at other points in time may find the server either idle or busy. If the server is idle, then one of the customers in batch, the first customer, enters service, and the others join the queue. If the server is busy,

then all of the customers in batch join the waiting line, or the queue. These actions are summarized in the flowchart in Figure 5.3.

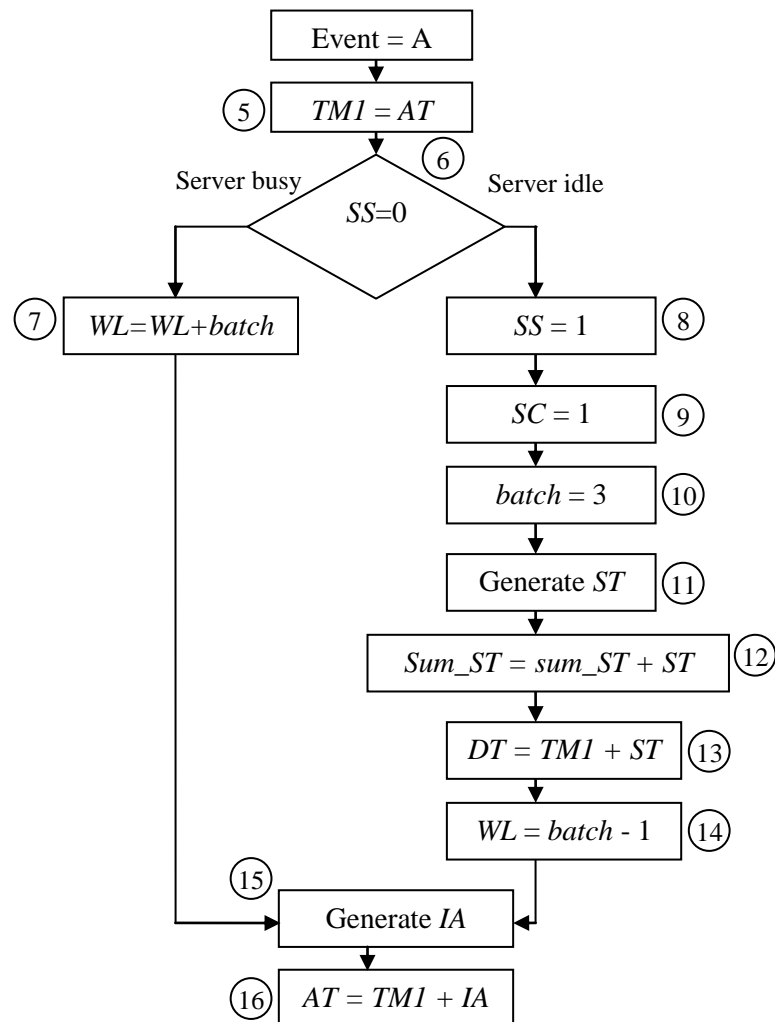


Figure 5.3 Flowchart for the arrival process in Figure 5.2.

We schedule the departure time of the first customer in the first batch by randomly generating a service time from service time distribution and setting the departure time as

$$\text{Departure time} = \text{clock time now} + \text{generated service time.} \quad (5.1)$$

We schedule the next batch arrival into the system by randomly generating an interarrival time from the interarrival time distribution and setting the arrival time as

$$\text{Arrival time} = \text{clock time now} + \text{generated interarrival time.} \quad (5.2)$$

Both these events and their scheduled times are maintained on the event list. If we complete all the actions for the first customer in the first arrival, we check the event list to determine the next scheduled event and its time. If the next event is an arrival, we move the clock time to the scheduled time of the arrival and the next process is an arrival. If the next event is a departure, we move the clock time to the time of the departure and the next process is a departure. For a departure, we control whether the length of the queue is greater than zero. If the length of the queue is greater than zero, then we remove the first customer from the queue and start service on this customer by setting a departure time using equation (5.1). If no one is waiting, then we set the status of the system to idle. This type of approach is called the *next-event time-advance mechanism* (Winston, 1994). In this approach, the clock time is updated. These actions can be summarized in the flowchart in Figure 5.4.

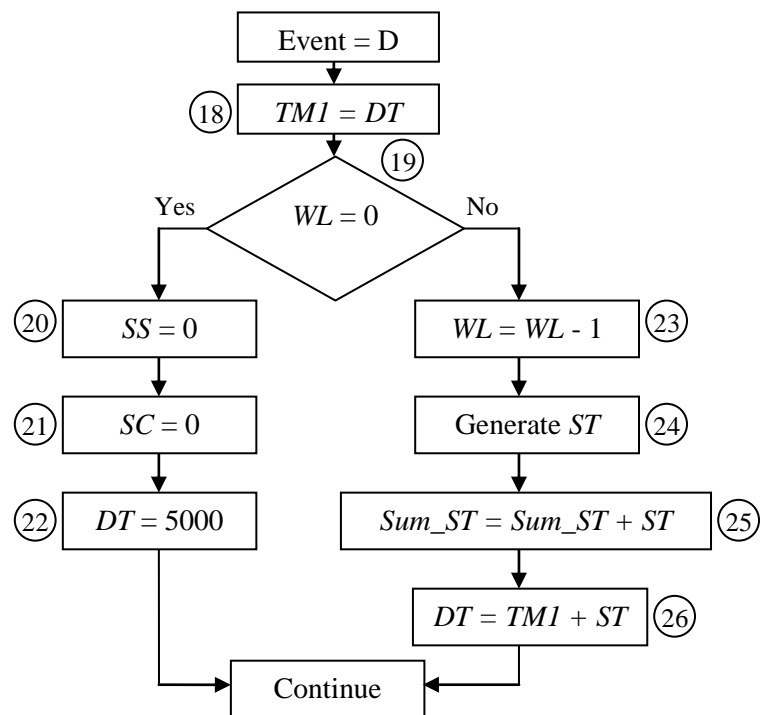


Figure 5.4 Flowchart for the departure process in Figure 5.2.

We act of passing over the periods of inactivity between the events by jumping from event to event, because the state variables change only an event time. We continue in this manner until prespecified stopping condition is satisfied. On the other hand, the procedure requires that we have either a batch arrival or a departure

scheduled for all of the next points in the simulation. When a new batch arrival into the system is concerned, the next batch arrival is always scheduled, but a departure time can only be scheduled when a customer is brought into service. Thus, if the system is idle, no departure can be scheduled. In such instances, we schedule a *dummy departure* by setting the departure time equal to a very large number such as 5000.

The general simulation process for the queuing model with batch arrival is explained above in detail. In this simulation, we assume that the interarrival and the service times are generated from exponential distribution for 18 customers, which are shown in Table 5.1 and 5.2. In Table 5.1, it can be realized that the time between the first arrival (customer 1, 2, 3) and the second arrival (customer 4, 5, 6) is **0.0235** time units, the time between the second arrival (customer 4, 5, 6) and the third arrival (customer 7, 8, 9) is **0.0644** time units, and so on. In Table 5.2, we can see that the service time for the customer 1 in the first arrival is **0.0067** time units, the service time for the customer 2 in the first arrival is **0.0047** time units, the service time for the customer 3 in the first arrival is **0.0001** time units, the service time for the customer 4 in the second arrival is **0.0102** time units, and so on.

Table 5.1 Interarrival times

Customer	IA
1,2,3	0.0235
4,5,6	0.0644
7,8,9	0.0273
10,11,12	0.0430
13,14,15	0.0046
16,17,18	0.0588

Table 5.2 Service times

Customer	ST
1	0.0067
2	0.0047
3	0.0001
4	0.0102
5	0.0039
6	0.0068
7	0.0128
8	0.0019
9	0.0006
10	0.0081
11	0.0000
12	0.0049
13	0.0006
14	0.0020
15	0.0037
16	0.0058
17	0.0045
18	0.0026

In order to demonstrate the first step of the simulation model, we define the following variables :

<i>TMI</i>	: clock time of the simulation
<i>AT</i>	: scheduled time of the next batch arrival
<i>DT</i>	: scheduled time of the next departure for a customer
<i>SS</i>	: status of the server (1=busy, 0=idle)
<i>SC</i>	: capacity of the server
<i>WL</i>	: length of the queue
<i>MX</i>	: length of a simulation run
<i>ST</i>	: service time randomly generated
<i>IA</i>	: interarrival time randomly generated
<i>n</i>	: number of customer in the system
<i>batch</i>	: number of customer in batch
L_q	: average queue length
W_q	: average waiting time in queue
<i>roo</i>	: service facility
<i>L</i>	: average number in system
<i>Sum_ST</i>	: sum of service times
<i>area_W</i>	: area under curve
<i>Sum_S</i>	: sum of the areas under curve

To start with, we set that *AT* is 0, because the first arrival is assumed to take place at time 0. We also assume that the system is empty at time 0, so we set *SS*=0, *WL*=0, and *DT*=5000. This means that our list of events now consists of two scheduled events; an arrival at time 0 and a dummy departure at time 5000. This completes the initialization process and gives us the computer representation of the simulation shown in Table 5.3.

Table 5.3 Computer representation of the first step of the simulation.

Event	Customer	Type of Event	TMI	SS	WL	Event List	
						AT	DT
0	-	Initialization	0	0	0	0	5000.00
1	1,2,3	Arrival	0.0000	1	2	0.0235	0.0067
2	1	Departure	0.0067	1	1	0.0235	0.0114
3	2	Departure	0.0114	1	0	0.0235	0.0115
4	3	Departure	0.0115	0	0	0.0235	5000.00
5	4,5,6	Arrival	0.0235	1	2	0.0879	0.0338
6	4	Departure	0.0338	1	1	0.0879	0.0376
7	5	Departure	0.0376	1	0	0.0879	0.0444
8	6	Departure	0.0444	0	0	0.0879	5000.00
9	7,8,9	Arrival	0.0879	1	2	0.1152	0.1007
10	7	Departure	0.1007	1	1	0.1152	0.1027
11	8	Departure	0.1027	1	0	0.1152	0.1033
12	9	Departure	0.1033	0	0	0.1152	5000.00
13	10,11,12	Arrival	0.1152	1	2	0.1582	0.1233
14	10	Departure	0.1233	1	1	0.1582	0.1234
15	11	Departure	0.1234	1	0	0.1582	0.1282
16	12	Departure	0.1282	0	0	0.1582	5000.00
17	13,14,15	Arrival	0.1582	1	2	0.1628	0.1588
18	13	Departure	0.1588	1	1	0.1628	0.1608
19	14	Departure	0.1608	1	0	0.1628	0.1646
20	16,17,18	Arrival	0.1628	1	3	0.2216	0.1646
21	15	Departure	0.1646	1	2	0.2216	0.1704
22	16	Departure	0.1704	1	1	0.2216	0.1749
23	17	Departure	0.1749	1	0	0.2216	0.1775
24	18	Departure	0.1775	0	0	0.2216	5000.00

We determine the first event by comparing AT and DT (block 3 in Figure 5.2). An arrival is indicated by $AT < DT$, and a departure is indicated by $AT > DT$. At this point, an arrival will take place, because $AT = 0$ and $DT = 5000$. We label this event 1 and update the clock time, TMI , to the time of event 1 (block 5 in Figure 5.3). Namely, we set $TMI = 0$. The arrival at time 0 finds the system empty, so a customer from the batch enters service immediately. We set $SS = 1$ to show that the server is now busy (block 8 in Figure 5.3), and set $SC = 1$ to show that there is one customer in service (block 9 in Figure 5.3). At this point, we require to say that three customer (customer 1, customer 2, and customer 3) arrive in the system, so we set $batch = 3$ (block 10 in Figure 5.2). Next, we generate a service time for customer 1 (block 11 in Figure 5.3). From Table 5.2, we can notice that the ST of customer 1 is 0.0067. At this point, we need to obtain sum of all service times. We update $Sum_ST = 0.0067$ by using the equation $Sum_ST = Sum_ST + ST$ (block 12 in Figure 5.3), because Sum_ST

is equal to 0. We set $DT = 0.0067$ by using the equation $DT = TMI + ST$ (block 13 in Figure 5.3), because TMI is 0. Namely, customer 1 will depart from the system at clock time 0.0067. We set $WL = 2$ (block 14 in Figure 5.3), because customer 2 and customer 3 join the queue while customer 1 enters service. We now schedule the next arrival into the system by generating an interarrival time (block 15 in Figure 5.3). From Table 5.1, we can notice that the IA is 0.0235. We set $AT = 0.0235$ by using the equation $AT = TMI + IA$ (block 16 in Figure 5.3), because TMI is 0. Namely, the second arrival (customer 4, customer 5, and customer 6) will take place at clock time 0.0235. We aim to calculate the waiting time in the queue of customer 2 and customer 3, and the service time of customer 1, namely, the waiting time in system of these customers until an event occurs (see Figure 5.6). Thus, we move on block 27 in Figure 5.2. We set $area_W = 0.0201$ by using the equation $area_W = (DT - TMI) * (WL + 1)$ (block 29 in Figure 5.5), because $DT = 0.0067$, $AT = 0.0235$, $TMI = 0$, and $WL = 2$ at this point. Then, we update $Sum_S = 0.0201$ by using the equation $Sum_S = Sum_S + area_W$ (block 34 in Figure 5.5), because $Sum_S = 0$. Table 5.3 shows the computer representation of the simulation at the end of event 1.

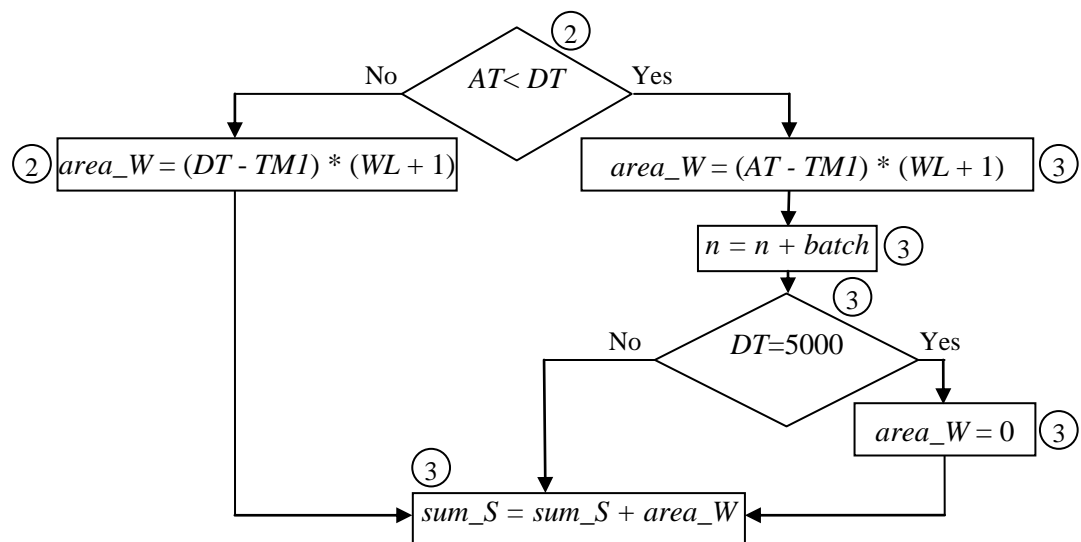


Figure 5.5 Flowchart for the calculate area in Figure 5.2.

We pass the block 35 in Figure 5.2 to determine whether the clock time, TMI , has exceeded the specified time length of simulation, MX . If it has, the first step of the simulation ends. If it has not, we go ahead with the first step of the simulation. At

this point, we loop back to block 3 in Figure 5.2 to determine the next event, event 2. Event 2 will be a departure at time 0.0067, because $AT = 0.0235$ and $DT = 0.0067$. We update the clock time, TMI , to the time of event 2 (block 18 in Figure 5.4), that is, we set $TMI = 0.0067$. At time 0.0067, customer 1 departs from system. We check the status of the waiting line to see if there are any customers waiting for service (block 19 in Figure 5.4). We have two customers waiting, since $WL = 2$. We remove customer 2 from the waiting line, set $WL = 1$ (block 23 in Figure 5.4). We bring customer 2 into service by generating a service time (block 24 in Figure 5.4). From Table 5.2, we see that the ST of customer 2 is 0.0047. At this point, we update $Sum_ST = 0.0114$ by using the equation $Sum_ST = Sum_ST + ST$ (block 25 in Figure 5.4), because $Sum_ST = 0.0067$. We set $DT = 0.0114$ by using the equation $DT = TMI + ST$ (block 26 in Figure 5.4), because $TMI = 0.0067$. Namely, customer 2 will depart from the system at clock time 0.0114. We require to calculate the waiting time in the queue of customer 3, and the service time of customer 2, namely, the waiting time in system of these customers until an event occur (see Figure 5.6), so we move on block 27 in Figure 5.2. We set $area_W = 0.0094$ by using the equation $area_W = (DT - TMI) * (WL + 1)$ (block 29 in Figure 5.5), because $DT = 0.0114$, $AT = 0.0235$, $TMI = 0.0067$, and $WL = 1$ at this point. Then, we update $Sum_S = 0.0295$ by using the equation $Sum_S = Sum_S + area_W$ (block 34 in Figure 5.5), because $Sum_S = 0.0201$. Table 5.3 shows the computer representation of the simulation at the end of the event 2.

We move on with block 35 in Figure 5.2 to state whether the clock time, TMI , has exceeded the specified time length of simulation, MX . At this point, we loop back to block 3 in Figure 5.2 to determine the next event, event 3. The event 3 will be a departure at time 0.0114, because $AT = 0.0235$ and $DT = 0.0114$. We update the clock time, TMI , to the time of event 3 (block 18 in Figure 5.4). Namely, we set $TMI = 0.0114$. At time 0.0114, customer 2 departs from system. We check the status of the waiting line to see whether there are any customers waiting for service (block 19 in Figure 5.4). We have one customer waiting, since $WL = 1$. We remove customer 3 from the waiting line, set $WL = 0$ (block 23 in Figure 5.4). We bring customer 3 into service by generating a service time (block 24 in Figure 5.4). From

Table 5.2, we see that the ST of customer 3 is 0.0001. At this point, we update $Sum_ST = 0.0115$ by using the equation $Sum_ST = Sum_ST + ST$ (block 25 in Figure 5.4), because $Sum_ST = 0.0114$. We set $DT = 0.0115$ by using the equation $DT = TMI + ST$ (block 26 in Figure 5.4), because $TMI = 0.0114$. It means that customer 3 will depart from the system at clock time 0.0115. We want to calculate that the service time of customer 3, that is, the waiting time in system of this customer until an event occurs (see Figure 5.6), therefore we move on block 27 in Figure 5.2. We set $area_W = 0.0001$ by using the equation $area_W = (DT - TMI) * (WL + 1)$ (block 29 in Figure 5.5), because $DT = 0.0115$, $AT = 0.0235$, $TMI = 0.0114$, and $WL = 0$ at this point. Then, we update $Sum_S = 0.0296$ by using the equation $Sum_S = Sum_S + area_W$ (block 34 in Figure 5.5), because $Sum_S = 0.0295$. Table 5.3 demonstrates the computer representation of the simulation at the end of event 3.

We move on with block 35 in Figure 5.2 to decide whether the clock time, TMI , has exceeded the specified time length of simulation, MX . At this point, we loop back to block 3 in Figure 5.2 to determine the next event, event 4. Event 4 will be a departure at time 0.0115, because $AT = 0.0235$ and $DT = 0.0115$. We update the clock time, TMI , to the time of event 4 (block 18 in Figure 5.4), that is, we set $TMI = 0.0115$. At time 0.0115, customer 3 departs from system. We control the status of the waiting line to see whether there are any customers waiting for service (block 19 in Figure 5.4). We have no customer waiting, since $WL = 0$. At this point, the system becomes idle. We set $SS = 0$ (block 20 in Figure 5.4), $SC = 0$ (block 21 in Figure 5.4), and $DT = 5000$ (block 22 in Figure 5.4). The system remains idle until an arrival takes place (See Figure 5.6). We pass the block 27 in Figure 5.2. Since $AT = 0.0235$ and $DT = 5000$, we set $area_W = 0.0120$ by using the equation $area_W = (AT - TMI) * (WL + 1)$ (block 30 in Figure 5.5). Then we set $n = 3$ (block 31 in Figure 5.5), because three customers pass over the system until this time. We check the dummy departure to see whether system is idle (block 32 in Figure 5.5), and see that there is the dummy departure because of $DT = 5000$, so we update $area_W = 0$. therefore, Sum_S does not change. Table 5.3 shows the computer representation of the simulation at the end of event 4.

The same procedure explained above for the first four events can be applied for the other events until the clock time, TMI , is either greater or equal than the specified time length of simulation, MX .

Figure 5.6 provides data for all the events. This figure also displays the changes in the system as a function of simulation clock time.

At the end of the first step, Sum_S , Sum_{ST} , and n are calculated. Additionally, only one single value is obtained for each of the following variables: the average queue length, L_q , the average number in system, L , the service facility, roo , and the average waiting time in queue, W_q .

We know that if the simulation is run only once, then it is not probable to calculate a variance and confidence interval in the third step, because we need to obtain a variety of L , L_q , W_q , and roo for these calculations. To sum up, our purpose is to acquire a plenty of L , L_q , W_q , and roo in this step. At this point, it involves that the simulation is run 1200 times, which means we set $p = 1200$ (block 1 in Figure 5.2). We now ready to show the second step. This step consists of all blocks except for block 45 and 63 in Figure 5.2.

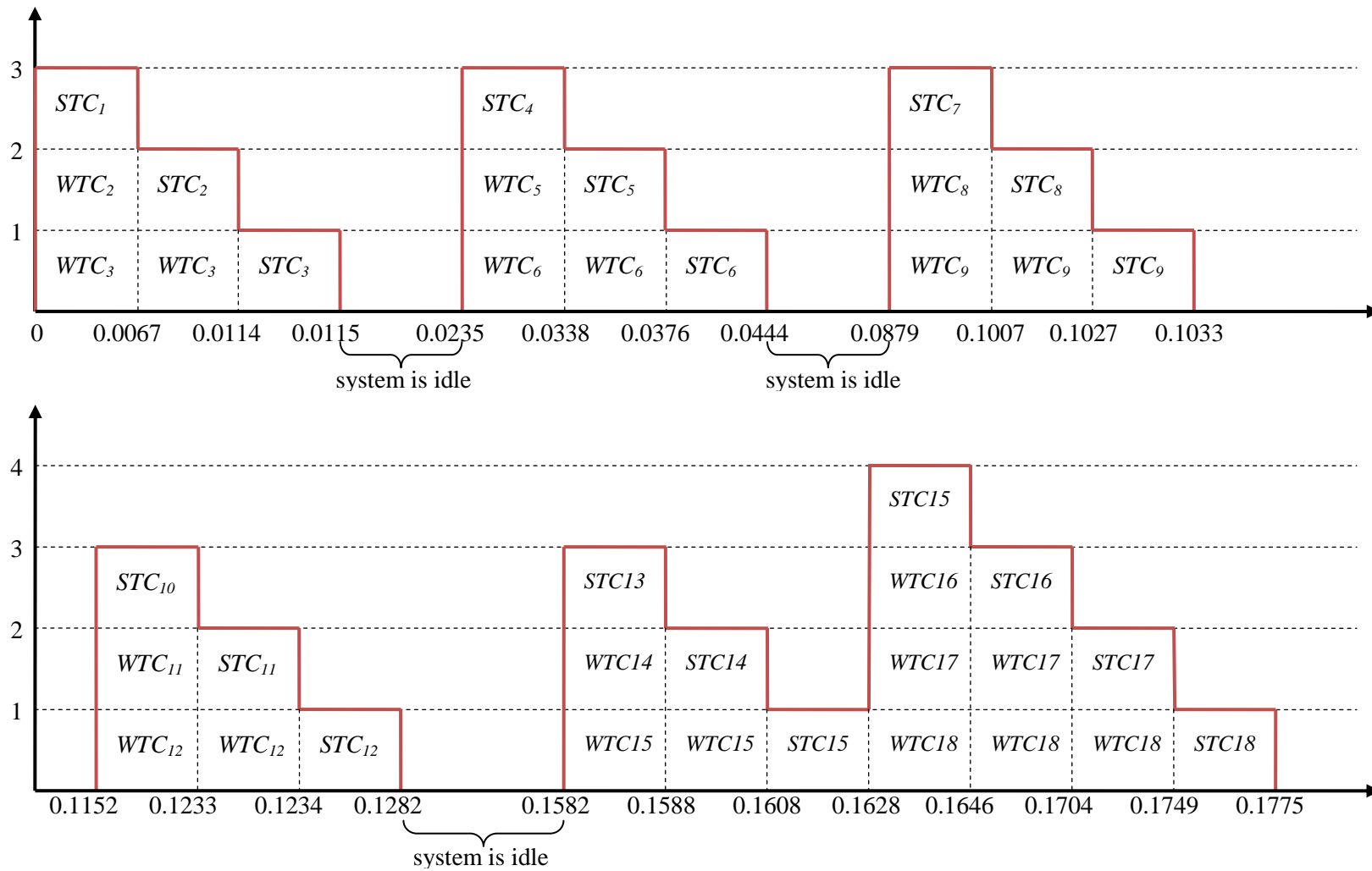


Figure 5.6 Changes in the system as a function of simulation clock time for queuing system with batch arrival.

The second step of the simulation consists of the following variables, in addition to the ones mentioned in the first step:

p	: number of replication
y_i	: i . average queue length ($i. L_q$)
$y1_i$: i . average number in system ($i. L$)
$y2_i$: i . average waiting time in queue ($i. W_q$)
Sum_WT	: sum of the waiting times in queue (sum of W_q)
$total_L_q$: sum of the average queue lengths
$total_W_q$: sum of the average waiting times in queue
$total_roo$: sum of the service facilities
$total_L$: sum of the average number in system

In case of $p = 1$, Sum_WT is obtained by using results achieved in the first step (block 36 in Figure 5.4). Now, it can be said that we have Sum_WT (sum of W_q), sum_ST (sum of ST), and sum_S (sum of W_s) for the customers in the first replication. We firstly calculate L_q , W_q , roo , and L by means of the equations (4.4), (4.3), (4.5), and (4.1), respectively. Then, $total_L_q$, $total_W_q$, $total_roo$, and $total_L$ are updated by using these results (block 37, 38, 39, and 40 in Figure 5.4, respectively). Finally, we set y_1 (L_q), $y1_1$ (L), and $y2_1$ (W_q) by using the values of L_q , L , and W_q , respectively (block 41, 42, and 43 in Figure 5.4). At the end of the first replication, we have the data for y_1 (L_q), $y1_1$ (L), and $y2_1$ (W_q), $total_L_q$, $total_L$, $total_W_q$, and $total_roo$.

In case of $p = 2$, we loop back to block 2 in Figure 5.4 to initialize all the variables in the first step (block 2 in Figure 5.4). At this point, the first step of the simulation is run again for new customers. Sum_WT is obtained by using data obtained in the first step (block 36 in Figure 5.4). Now, we have Sum_WT (sum of W_q), sum_ST (sum of ST), and sum_S (sum of W_s) for the customers in the second replication. We firstly calculate L_q , W_q , roo , and L by using the equations (4.4), (4.3), (4.5), and (4.1), respectively. Then, $total_L_q$, $total_W_q$, $total_roo$, and $total_L$ are updated by using these results (block 37, 38, 39, and 40 in Figure 5.4, respectively). Finally, we set y_2

(2. L_q), $y1_2$ (2. L), and $y2_2$ (2. W_q) by using the values of L_q , L , and W_q , respectively (block 41, 42, and 43 in Figure 5.4). At the end of the second replication, we have the data for y_1 (1. L_q), $y1_1$ (1. L), $y2_1$ (1. W_q), y_2 (2. L_q), $y1_2$ (2. L), and $y2_2$ (2. W_q). Furthermore, $total_L_q$, $total_L$, $total_W_q$, and $total_roo$ are updated.

The same procedure explained above can be applied for the rest of the replications until p is 1200. At the end of the second step of the simulation, we get the values of y_i (i . L_q), $y1_i$ (i . L), and $y2_i$ (i . W_q) where $i=1,\dots,1200$, and also of $total_L_q$ (sum of L_q), $total_L$ (sum of L), $total_W_q$ (sum of W_q), and $total_roo$ (sum of roo).

Now, we can move to analyze the third step. As explained before, this step consists of all blocks demonstrated in Figure 5.2. In the light of the results obtained in the first and second step, now we calculate the final outcomes for the simulation since all the data we need to make calculations in the third step has already been obtained. Also, at the end of this step, we will be able to complete the simulation.

The third step of the simulation consists of the following variables, in addition to the ones mentioned in the first and second step:

SL_q	: sum of squared of errors for L_q
SL	: sum of squared of errors for L
SW_q	: sum of squared of errors for W_q
SL_q1	: standard error for L_q
$SL1$: standard error for L
SW_q1	: standard error for W_q
$cdown$: down limit for confidence interval of L_q
cup	: up limit for confidence interval of L_q
$cdown1$: down limit for confidence interval of W_q
$cup1$: up limit for confidence interval of W_q
$cdown2$: down limit for confidence interval of L
$cup2$: up limit for confidence interval of L .

We calculate the point estimates of L_q , W_q , roo , and L by using the equation (4.6). (block 46, 47, 48, and 49 in Figure 5.7). After that, we calculate the sum of squared of errors for \hat{L}_q , \hat{L} , and \hat{W}_q (block 50, 51, and 52 in Figure 5.7), and also the standard error estimates of \hat{L}_q , \hat{L} , and \hat{W}_q by using the equation (4.7), SL_qI , SLI , and SW_qI (block 54, 55, and 56 in Figure 5.7). Lastly, the confidence interval estimates by using the equation (4.8); $cdown$, cup for L_q (block 57 and 58 in Figure 5.7), $cdown1$, $cup1$ for W_q (block 59 and 60 in Figure 5.7), $cdown2$, $cup2$ for L (block 61 and 62 in Figure 5.7) are calculated.

At the end of the third step, in other words, at the end of the simulation, the data for the point estimates, \hat{L} , \hat{L}_q , \hat{W}_q , and roo , the standard error estimates, $S_{\hat{L}}$, $S_{\hat{L}_q}$, and $S_{\hat{W}_q}$, and the confidence interval estimates for L , L_q , and W_q are achieved.

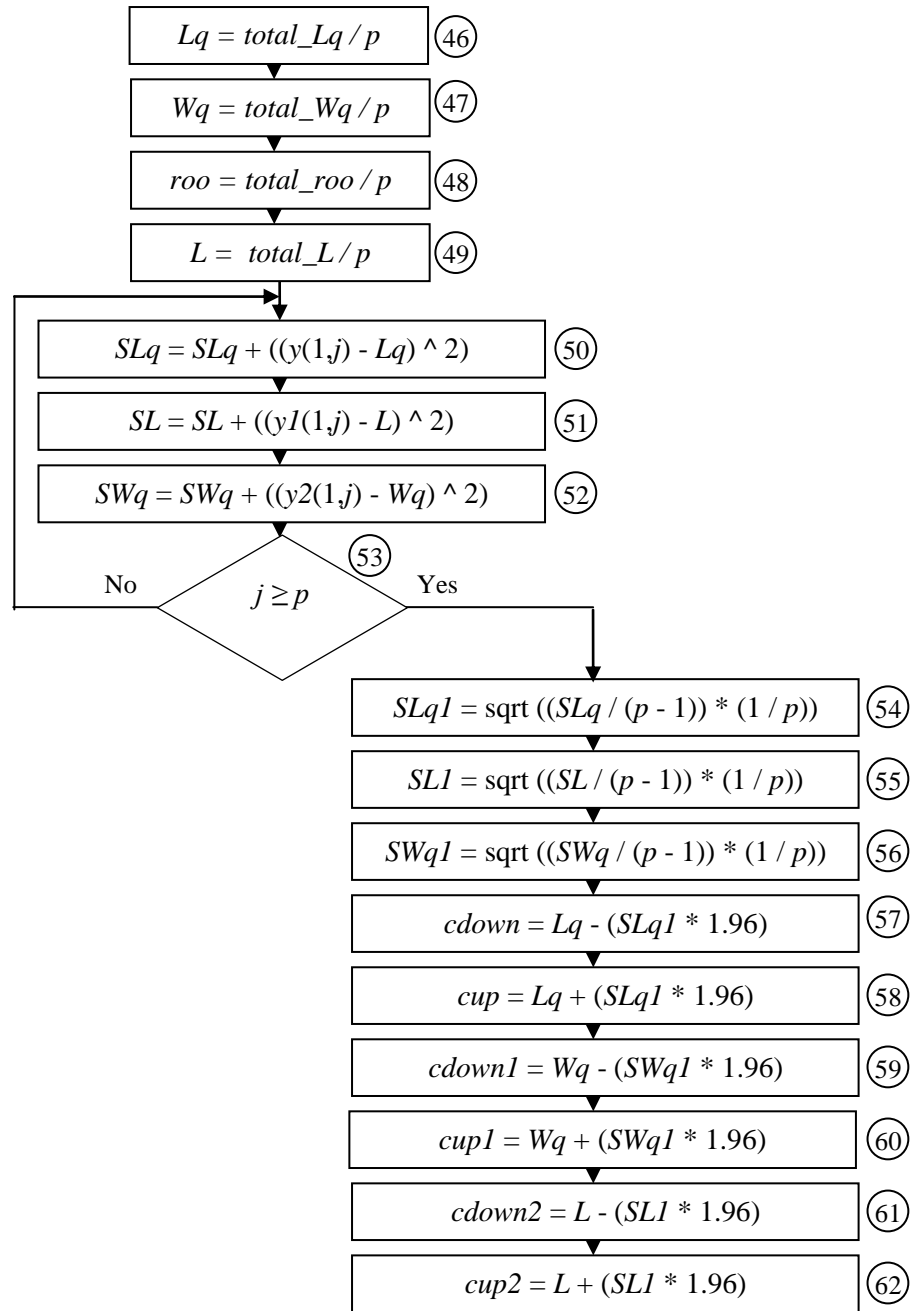


Figure 5.7 Flowchart for the calculate results in Figure 5.2.

5.1.1 Determination of run length and number of replications

In this section, we consider how to determine the run length, MX , and the number of replications, p . Firstly, we have needed to analyze whether there is a warm-up period in this queueing system. At this point, the three simulation studies have been chosen as following:

- (1) batch size 2, batch arrival rate 30, and service rate 100,
- (2) batch size 3, batch arrival rate 30, and service rate 100,
- (3) batch size 4, batch arrival rate 20, and service rate 100.

Note that, for batch size 2, 3, and 4 the systems have highest traffic intensity in (1), (2), and (3) respectively (see Table 5.8). Therefore, we analyze the three systems to find whether there is a warm-up period. If warm-up period is not occurred when the traffic intensity is high, we can stop the simulation run at specified time.

If the queuing system that is working with higher traffic intensity does not involve a warm-up period, the systems with lower traffic intensity never occupy a warm-up period. That's why, we have chosen the above mentioned systems with the highest traffic intensity for each batch size. To find a warm-up period, we have obtained the graphs which illustrate waiting time in the system versus the simulation clock time by the run length 2000. These graphs are represented in Figure 5.8, 5.9, and 5.10.

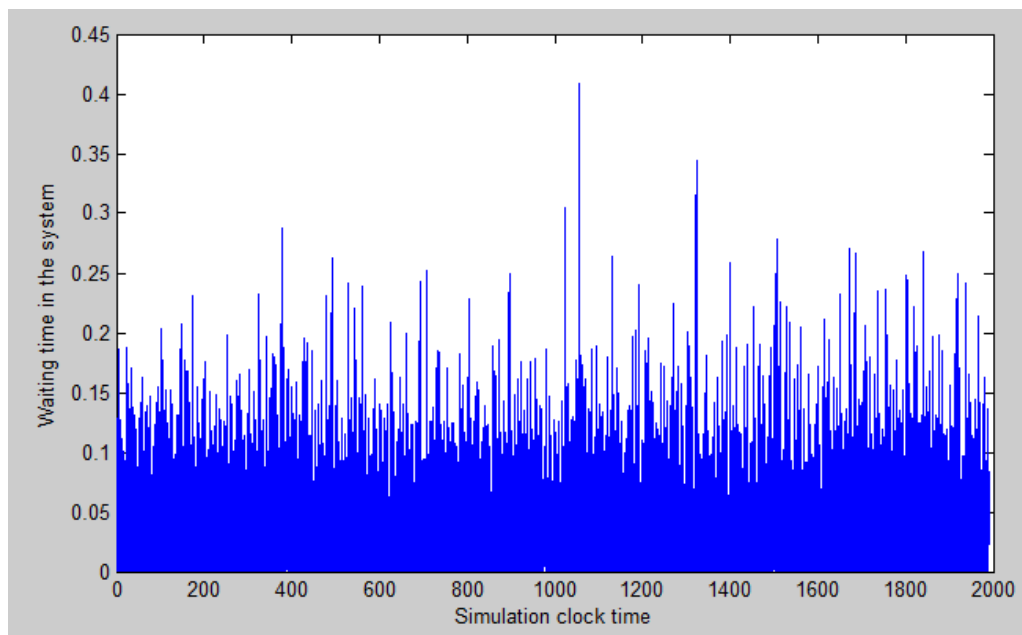


Figure 5.8 Batch size 2, batch arrival rate 30, and service rate 100, traffic intensity 0.6.

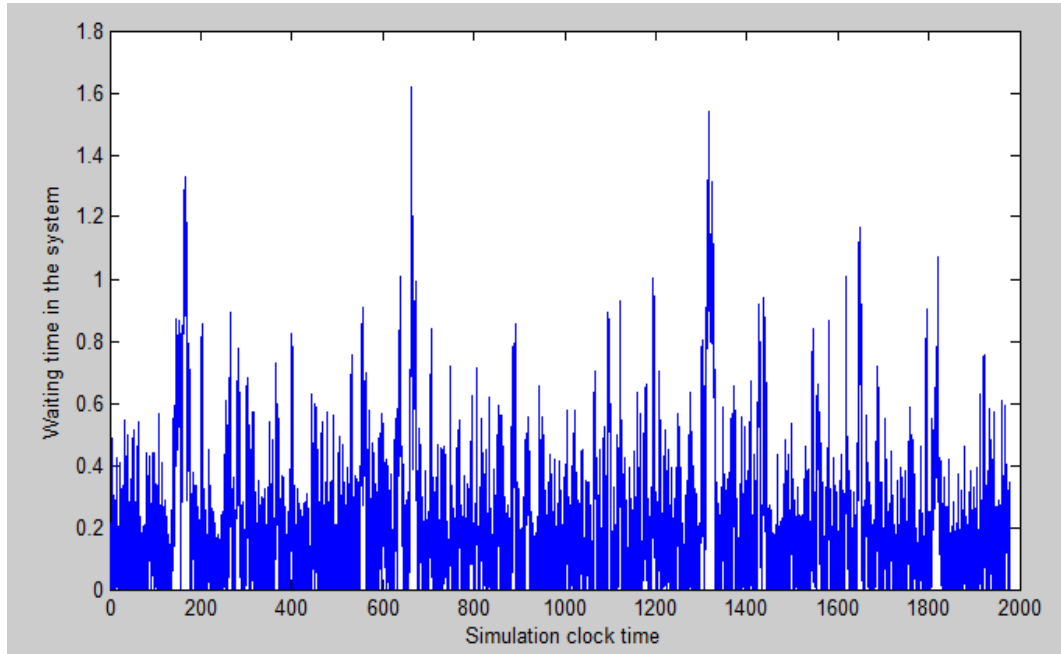


Figure 5.9 Batch size 3, batch arrival rate 30, and service rate 100, traffic intensity 0.9.

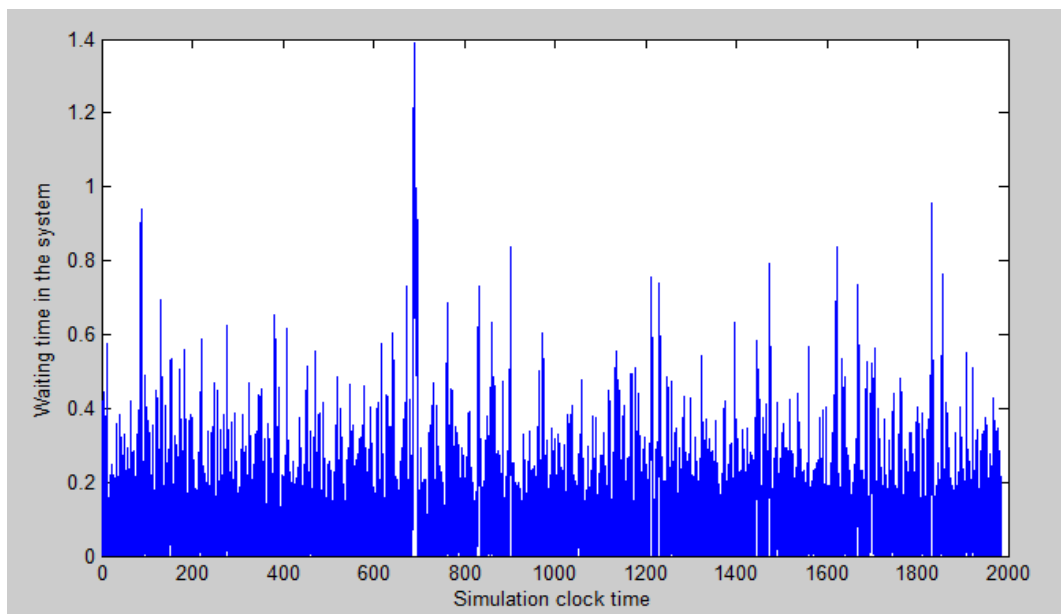


Figure 5.10 Batch size 4, batch arrival rate 20, and service rate 100, traffic intensity 0.8.

The figures illustrate that the effect of initial conditions on later observations is fairly less. Besides observations appear around a constant mean in long-time for each graph. Namely, we have no warm-up period in each study. Therefore, we have collected all data in the run length.

We know that the large run length reduces $S_{\hat{L}}$ and the large number of replications also reduces $S_{\hat{L}}$. Since we require the smallest value for $S_{\hat{L}}$ in order to obtain the narrow confidence intervals, we need to get the optimal values for MX and p . To accomplish this aim, some pilot-runs need to be made with different run-lengths and numbers of replication.

For the first pilot-runs, we have decided to keep the total simulation run length constant to be 50.000. We have run the simulation with the values of 50 and 1.000 for MX and p , respectively, but realized that the results were not satisfactory. And then, we have decided to make new calculations by increasing MX so decreasing p since the total simulation run-length was kept constant. Since it is not required for p to be less than 30, at the last try, we have run the simulation with the values of 1.000 for MX and of 50 for p . However, we have observed again that these values were far away from being satisfactory. As seen in Table 5.4, when we increase MX with the constant value for the total simulation run length, $S_{\hat{L}}$ doesn't change significantly (for instance; see column 6). Therefore, for these pilot-runs, it may be stated that it is not enough to change the values of MX and p when the total simulation run length is kept constant.

Table 5.4 Traffic intensity versus the total simulation run length for 50.000

Batch Size:		2	3	4	5	6	7	8	9
ρ		0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000
MX	p								
50	1000	0.00092	0.00235	0.00528	0.01191	0.02515	0.06337	0.17521	0.70223
100	500	0.00093	0.00240	0.00498	0.01078	0.02521	0.05959	0.17143	0.76826
200	250	0.00088	0.00225	0.00514	0.01208	0.02518	0.05787	0.16956	0.95169
500	100	0.00087	0.00234	0.00521	0.01145	0.02534	0.05903	0.18203	0.91955
1000	50	0.00082	0.00253	0.00608	0.01022	0.02120	0.05892	0.15118	0.87952

Whitt (1989) described the heavy-traffic limits as following: "The heavy-traffic limits describe how the queuing process behave as the traffic intensity ρ approaches its critical value for stability,... always taken to be 1" (p.1342). In these pilot runs, batch arrival rate, λ , and service rate, μ , are kept constant as 10 and 100, respectively, therefore, when batch size is 10, the heavy-traffic limit is exceeded

because of $\rho = \lambda r / \mu = (10)(10) / 100 = 1$. Thus, the pilot-runs are made with the values of batch size less than 9. The results for $S_{\hat{L}}$ can be seen in Table 5.4.

"The heavy-traffic limits allow us to relate how time t or customer index n should grow (the simulation run length) as ρ approaches 1;.... " (Whitt, 1989, p.1342). As far as our pilot runs are concerned, then, we can say that when the batch size is equal to or less than 6, or the traffic intensity is equal to or less than 0.6, the total simulation run length can be 50.000. The reason behind this is that in this study, we assume that the standard error less than 0.05 is adequate and as seen in Table 5.4, the values of $S_{\hat{L}}$ for the batch sizes less than 7 are less than 0.05. However, we want the length to be acceptable for all batch sizes. That's why, we have decided to increase the total simulation run length by taking the simulation run length equal to the number of replications. Therefore, we made new pilot runs, the results of which can be seen in Table 5.5.

Table 5.5 Traffic intensity versus the total simulation run length for more than 50.000

Batch Size:		6	7	8	9
ρ		0.6000	0.7000	0.8000	0.9000
MX	p				
400	400	0.01452	0.03459	0.09881	-
500	500	-	-	0.08224	-
600	600	-	-	0.06857	-
700	700	-	-	0.05867	-
800	800	-	-	0.05157	-
900	900	-	-	0.04380	0.22526
1000	1000	-	-	-	0.20700
1100	1100	-	-	-	0.18480
1200	1200	-	-	-	0.17257
1800	1800	-	-	-	0.11195
2300	2300	-	-	-	0.08762
2700	2700	-	-	-	0.07565
3400	3400	-	-	-	0.06048

Firstly, the total simulation run length 50.000 has been increased to 160.000, 250.000, 360.000, 490.000, and 640.000, respectively, but it has not been adequate for the traffic intensity 0.8 and 0.9 while it has been adequate for the rest. When the length has been increased to 810.000, it has become adequate for the traffic intensity

0.8, but this time it has been still insufficient for 0.9. Finally, we have continued to the pilot runs until the total simulation run length has been 11.560.000, but seen that it has not been adequate for 0.9. The last pilot run has lasted too much time, that is, 11 hours. At this point, we can say that if the traffic intensity approaches to the heavy-traffic limit, it is unavoidable that the total simulation run is longer. And, we can say that if batch size increases, then the traffic intensity increases too. For example, while batch size increases from 2 to 9, the traffic intensity increases from 0.2 to 0.9, simultaneously. This case can be seen in Table 5.4. Consequently, we may say that small batch size equals to the less waste of time. Therefore, we have chosen batch size 2, 3, and 4, and determined the total simulation run length for the most intensity case.

The most intensity case has been obtained for the batch size 3, the batch arrival rate 30, and the service rate 100 (see Table 5.8). So, the plot runs have been made in order to determine the total simulation run length. The results can be seen in Table 5.6.

Table 5.6 - Pilot runs for traffic intensity 0.9

MX	p	$S_{\hat{L}_q}$	$S_{\hat{W}_q}$	$S_{\hat{L}}$
300	300	0.164681	0.001755	0.165131
400	400	0.140822	0.001513	0.141144
500	500	0.098995	0.001059	0.099253
600	600	0.090084	0.000964	0.090318
700	700	0.076201	0.000816	0.076403
800	800	0.067933	0.000725	0.068114
900	900	0.058872	0.000629	0.059027
1000	1000	0.052429	0.000560	0.052572
1100	1100	0.047646	0.000511	0.047761
1200	1200	0.045378	0.000486	0.045491
1400	1400	0.035795	0.000383	0.035893

Firstly, we have run the simulation by taking $MX = 300$ and $p = 300$, seen that the total simulation run length 90.000 was not adequate, decided to increase these values because of the same reasons mentioned before. We have accomplished our aim by taking the total simulation run length 1.440.000. Consequently, we have decided to run the simulation program by taking $MX = 1200$ and $p = 1200$.

5.1.2 Comparison between the results of simulation and the analytic results

The simulation program for the queuing model with batch arrival has been run by taking $MX = 1200$ and $p = 1200$, which means that one simulation run length MX is 1200, and the simulation run is repeated 1200 times because of $p = 1200$.

In this study, we have two purposes to achieve. Firstly, we point estimate for the performance measures by using simulation program and analyze the statistical precision of these estimates by using the confidence interval estimate. Secondly, we require showing whether the statistical precision of estimates is affected by the batch size.

Throughout the first purpose, we have run the simulation program (see A3.1) 14 times for different batch size values and batch arrival rates, and obtained the estimates of the performance measures. The simulation results are shown in Table 5.7.

Table 5.7 The simulation results or point estimates for the performance measures.

Batch Size: r	Batch Arrival Rate: λ	Customer Arrival Rate: $r\lambda$	Service Rate: μ	Simulation Results			
				$\hat{\rho}$	\hat{L}	\hat{L}_q	\hat{W}_q
2	10	20	100	0.2000	0.3749	0.1749	0.0087
	15	30	100	0.2999	0.6424	0.3425	0.0114
	20	40	100	0.3999	0.9992	0.5993	0.0150
	25	50	100	0.5000	1.5001	1.0001	0.0200
	30	60	100	0.5998	2.2486	1.6487	0.0275
3	10	30	100	0.3000	0.8573	0.5573	0.0186
	15	45	100	0.4500	1.6360	1.1861	0.0264
	20	60	100	0.5999	2.9987	2.3988	0.0400
	25	75	100	0.7497	5.9878	5.2381	0.0698
	30	90	100	0.9000	18.0065	17.1065	0.1900
4	5	20	100	0.2000	0.6248	0.4248	0.0212
	10	40	100	0.4001	1.6673	1.2671	0.0317
	15	60	100	0.5997	3.7444	3.1448	0.0524
	20	80	100	0.8000	9.9913	9.1913	0.1148

We have obtained some numerical results for the batch size values and batch arrival rates in simulation (see A2.1). The numerical results are shown in Table 5.8.

Table 5.8 The numerical results for performance measures.

Batch Size: r	Batch Arrival Rate: λ	Customer Arrival Rate: $r\lambda$	Service Rate: μ	Numerical Results			
				ρ	L	L_q	W_q
2	10	20	100	0.2000	0.3760	0.1760	0.0088
	15	30	100	0.3000	0.6420	0.3420	0.0114
	20	40	100	0.4000	1.0000	0.6000	0.0150
	25	50	100	0.5000	1.5000	1.0000	0.0200
	30	60	100	0.6000	2.2500	1.6500	0.0275
3	10	30	100	0.3000	0.8580	0.5580	0.0186
	15	45	100	0.4500	1.6380	1.1880	0.0264
	20	60	100	0.6000	3.0000	2.4000	0.0400
	25	75	100	0.7500	6.0000	5.2500	0.0700
	30	90	100	0.9000	18.000	17.100	0.1900
4	5	20	100	0.2000	0.6260	0.4260	0.0213
	10	40	100	0.4000	1.6680	1.2680	0.0317
	15	60	100	0.6000	3.7500	3.1500	0.0525
	20	80	100	0.8000	10.000	9.2000	0.1150

When we compare the simulation results in Table 5.7 to the numerical results in Table 5.8, it can be derived that the simulation results are fairly close to the numerical results. However, with these findings, anything about the statistical precision of these estimates cannot be claimed. Therefore, we have calculated the confidence interval estimates for the performance measures and standard error estimates necessary for these confidence intervals in the simulation program. In the calculations, we have used confidence level $\alpha = 0.05$, namely, confidence coefficient 0.95. The each interval including the true performance measure is confided in 95%, for example, in case of the batch size is 2, the batch arrival rate is 10, and the service rate is 100, the interval $0.374572 < L < 0.375243$ includes the true $L = 0.3760$ confided in 95%. The results can be seen in Table 5.9.

Table 5.9 The standard error estimates of estimators and confidence interval estimates of performance measures.

Batch Size: r	Batch Arrival Rate: λ	Customer Arrival Rate: $r\lambda$	Service Rate: μ	Standard Error Estimate			Confidence Interval Estimate (95%)		
				$S_{\hat{L}}$	$S_{\hat{L}_q}$	$S_{\hat{W}_q}$	L	L_q	W_q
2	10	20	100	0.000171	0.000114	0.000005	0.374572;0.375243	0.174694;0.175142	0.008737;0.008755
	15	30	100	0.000281	0.000218	0.000006	0.641842;0.642942	0.342092;0.342945	0.011406;0.011430
	20	40	100	0.000451	0.000377	0.000008	0.998287;1.000053	0.598552;0.600031	0.014967;0.014999
	25	50	100	0.000771	0.000691	0.000012	1.498637;1.501658	0.998770;1.001479	0.019973;0.020020
	30	60	100	0.001365	0.001275	0.000019	2.245899;2.251249	1.646240;1.651239	0.027445;0.027518
3	10	30	100	0.000427	0.000351	0.000009	0.856477;0.858144	0.556612;0.557989	0.018557;0.018593
	15	45	100	0.000926	0.000837	0.000015	1.634206;1.637835	1.184413;1.187696	0.026321;0.026381
	20	60	100	0.002022	0.001929	0.000028	2.994769;3.002696	2.395001;2.402564	0.039917;0.040028
	25	75	100	0.006309	0.006195	0.000075	5.975414;6.000143	5.225960;5.250244	0.069700;0.069993
	30	90	100	0.044487	0.044365	0.000474	17.919327;18.093714	17.019531;17.193444	0.189060;0.190919
4	5	20	100	0.000362	0.000293	0.000010	0.624091;0.625512	0.424258;0.425404	0.021224;0.021264
	10	40	100	0.000982	0.000888	0.000018	1.665327;1.669176	1.265366;1.268846	0.031626;0.031695
	15	60	100	0.002963	0.002844	0.000040	3.738636;3.750249	3.139204;3.150351	0.052344;0.052501
	20	80	100	0.014118	0.013990	0.000161	9.963616;10.018958	9.163838;9.218679	0.114524;0.115154

We always want to obtain the narrow confidence intervals so that we rearrange Table 5.9 to Table 5.10 in order to show the width of each confidence interval clearly.

Table 5.10 The width of the confidence intervals in Table 5.9.

Batch Size: r	Batch Arrival Rate: λ	Customer Arrival Rate: $r\lambda$	Service Rate: μ	ρ	Width of Confidence Interval		
					L	L_q	W_q
2	10	20	100	0.2	0.000671	0.000448	0.000018
	15	30	100	0.3	0.001100	0.000853	0.000024
	20	40	100	0.4	0.001766	0.001479	0.000032
	25	50	100	0.5	0.003021	0.002709	0.000047
	30	60	100	0.6	0.005350	0.004999	0.000073
3	10	30	100	0.3	0.001667	0.001377	0.000036
	15	45	100	0.45	0.003629	0.003283	0.000060
	20	60	100	0.6	0.007927	0.007563	0.000111
	25	75	100	0.75	0.024729	0.024284	0.000293
	30	90	100	0.9	0.174387	0.173913	0.001859
4	5	20	100	0.2	0.001421	0.001146	0.000040
	10	40	100	0.4	0.003849	0.003480	0.000069
	15	60	100	0.6	0.011613	0.011147	0.000157
	20	80	100	0.8	0.055342	0.054841	0.000063

We have obtained the narrowest confidence intervals when the traffic intensity has the smallest value; 0.2, 0.3 and 0.2, for batch sizes 2, 3, and 4, respectively. This can be seen in Table 5.10.

Whitt (1989) states that "For more complex queuing models than M/M/1, the statistical precision of estimates for a given simulation run length is not only affected by the traffic intensity, but also by the variability of the basic arrival and service processes." (p.1342). With the help of this explanation, we may think that the batch size can affect the statistical precision of estimates. That's why our second purpose in the study is to examine whether the statistical precision of estimates is affected by the batch size.

In order to understand the batch size effect, we have formed new tables from Table 5.10 (see Table 5.11 and 5.12). In Table 5.11, we have kept ρ constant while in Table 5.12 λ and μ have been kept constant.

We know that the traffic intensity $\rho = \lambda r / \mu$ affects the statistical precision of estimates. When the traffic intensity is kept constant as 0.6 as shown in Table 5.11, we can see that when the batch size changes, unsurprisingly the batch arrival rate changes. And, at last, as we can see, the statistical precision of estimates also changes although the traffic intensity does not change. Besides, we can observe that the minimum batch size value gives us the most confidential estimate.

Table 5.11 Relation between statistical precision of estimate and batch size for the same ρ .

Batch Size: r	Batch Arrival Rate: λ	Customer Arrival Rate: $r\lambda$	Service Rate: μ	Width of Confidence Interval		
				L	L_q	W_q
2	30	60	100	0.005350	0.004999	0.000073
3	20	60	100	0.007927	0.007563	0.000111
4	15	60	100	0.011613	0.011147	0.000157

Similarly, when the batch arrival rate and the service rate are kept constant as 20 and 100 respectively (see Table 5.12), once more, we can see that when the batch size changes, the statistical precision of estimates changes. Moreover, we see that the minimum batch size value gives us the most confidential estimate.

Table 5.12 Relation between statistical precision of estimate and batch size for different ρ .

Batch Size: r	Batch Arrival Rate: λ	Customer Arrival Rate: $r\lambda$	Service Rate: μ	Width of Confidence Interval		
				L	L_q	W_q
2	20	40	100	0.001766	0.001479	0.000032
3	20	60	100	0.007927	0.007563	0.000111
4	20	80	100	0.055342	0.054841	0.000063

5.1.3 Impact of batch size over the performance measures

In this section, we will examine how batch size affects the performance measures. Through this purpose, for different batch sizes, we have obtained the graphs of the batch arrival rates versus the performance measures (see Figure 5.11, 5.12, and 5.13).

When we have analyzed Figure 5.11 for the batch size 2, we see that the waiting time in queue increases as the batch arrival rate increases. Similarly, when we

examine the batch size 3 and 4, we realize again that the waiting time in queue increases as the batch arrival rate increases. Furthermore, it is also clear that the batch size increases, the increment rate of the waiting time becomes higher.

Similarly, it can be derived from the graph that the increment rates for different traffic intensities have been changing. For instance; as far as the interval (10, 15) is concerned, we can see that the waiting time in queue increases as batch size increases. On the other hand, when we analyze the interval (15, 20), it is obvious that the increment rate is higher compared to the interval (10, 15). This is because of the traffic intensity. It is closer to the heavy traffic limit in the interval (15, 20) than in the interval (10, 15). To sum up, the more intense the system is, the more waiting time it requires.

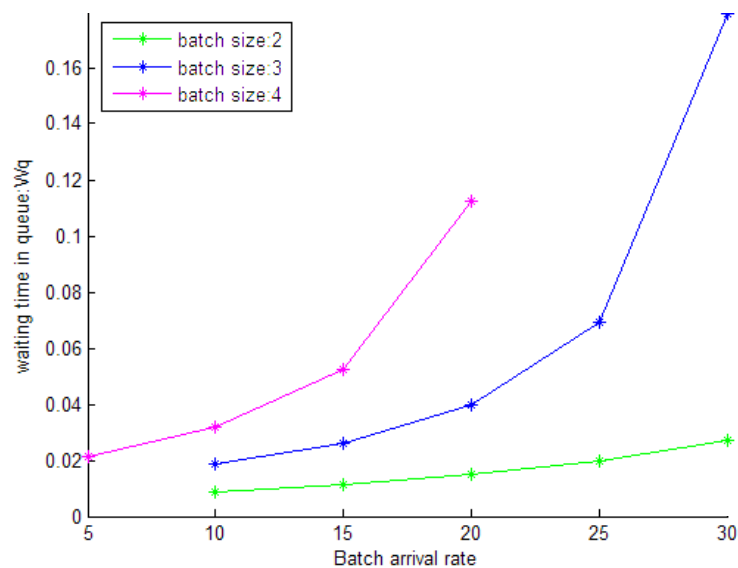


Figure 5.11 Batch arrival rate versus W_q for different batch sizes.

Similar explanations can be made for the number of customers in queue which means the queue length. When we analyze Figure 5.12 for the batch size 2, we see that the number of customers in queue increases as the batch arrival rate increases. Similarly, when we examine the batch size 3 and 4, we realize again that the number of customers in queue increases as the batch arrival rate increases. Furthermore, it is also clear that the batch size increases, the increase rate of the number of customers becomes higher.

In the same way, it can be derived from the graph that the increment rates for different traffic intensities have been changing. For instance; as far as the interval (10, 15) is concerned, we can see that the number of customers in queue increases as batch size increases. On the other hand, when we analyze the interval (15, 20), it is obvious that the increment rate is higher compared to the interval (10, 15). It can be explained by the traffic intensity. It is closer to the heavy traffic limit in the interval (15, 20) than in the interval (10, 15). To sum up, the more intense the system is, the more number of customers in queue it requires.

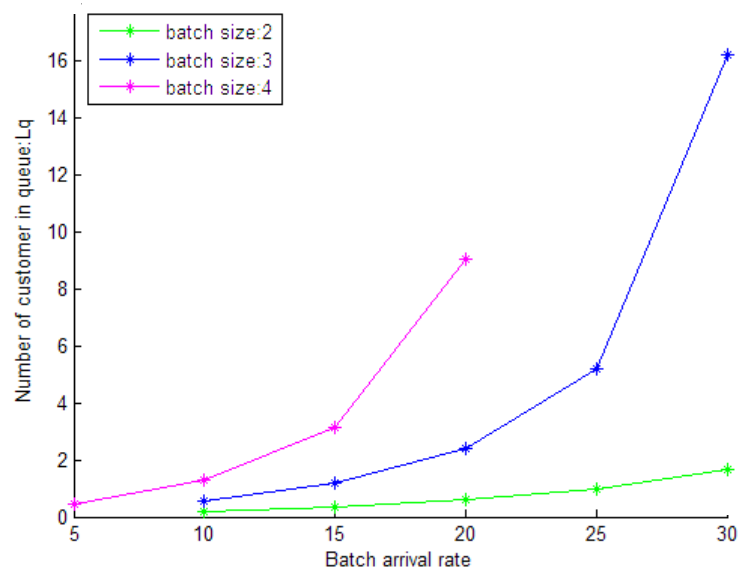


Figure 5.12 Batch arrival rate versus L_q for different batch sizes.

When we analyze Figure 5.13 for the batch sizes, we can understand that the service facility which refers to the traffic intensity increases as the batch arrival rate increases. It is also obvious that the batch size increases, the increment rate of the service facility becomes slightly higher. What's more, according to the graph, we can say that the large batch size causes the traffic intensity to approach heavy-traffic limit more quickly.

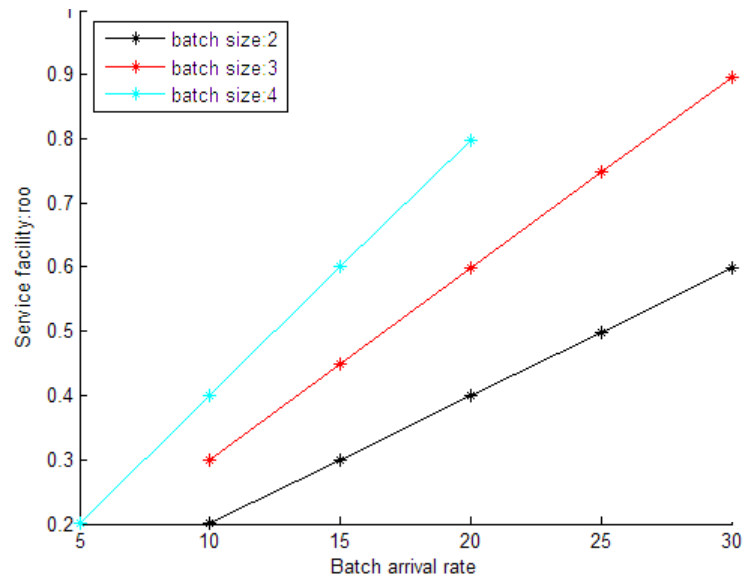


Figure 5.13 Batch arrival rate versus ρ for different batch sizes.

5.2 Simulation model of queuing systems with batch service

In the simulation of queuing system, we first have to describe the system. For this single-server system, we assume that arrivals are drawn from an infinite calling population. Furthermore, we assume that arrivals occur in such a way that one customer enters the system at a time, and also the type of interarrival times distribution is exponential distribution. The customers are served with the queue discipline FCFS and there is unlimited waiting room capacity. If the number of waiting customers in the queue is more than or equal to 3, then they are served as $r = 3$ at a time with the distribution of service times as exponential distribution. After service, all customers return to the calling population. This queuing system is represented in Figure 5.14.

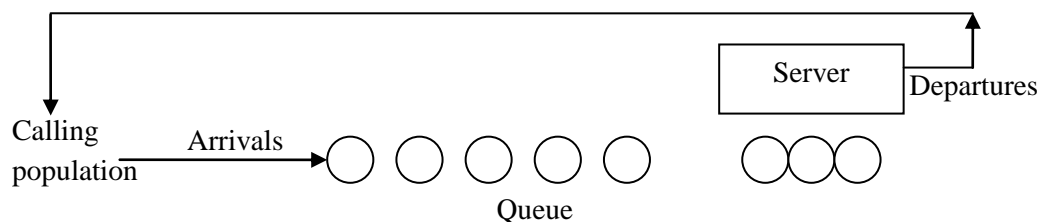


Figure 5.14 Single-server queuing system with batch service ($r = 3$).

However, if the number of waiting customers in the queue is less than 3, then, all the customers are served at a time according to the distribution of service times

which is exponential distribution. After service, all customers return to the calling population. This queuing system is shown in Figure 5.15.

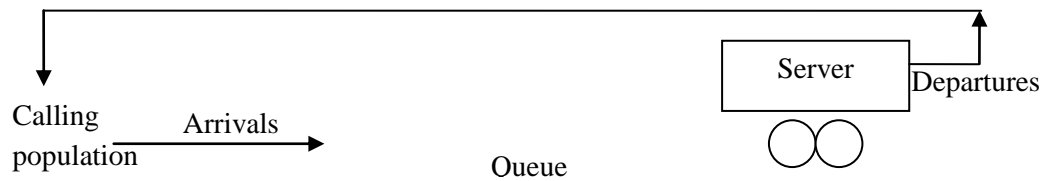


Figure 5.15 Single-server queuing system with batch service ($r < 3$).

We provide a flowchart which consists of three steps: *the first step*, *the second step*, and *the third step*. And, all the blocks have been numbered (see Figure 5.16).

The first step: This step consists of block 2, 3, 4, 24, 37, and 49 in the flowchart in Figure 5.16. All of the events occur in the first step. Namely, the arrival enters to the system at block 4 in Figure 5.16, and the departure leaves the system at block 24 in Figure 5.16. On the other hand, the total waiting time in the system of customers, Sum_S , the total waiting time in the queue of customers, Sum_WT , and the total number of customers in system, n , are calculated at block 37. The total service time, Sum_ST , is calculated at block 4 and block 24. The simulation is run once at the first step. Therefore, at the end of the first step, we achieve values for Sum_S , Sum_ST , Sum_WT , and n . On the other hand, only one single value for each of the following variables is obtained: the average queue length, L_q , the average number in system, L , the root, roo , and the average waiting time in queue, W_q .

The second step: This step consists of all blocks except for block 58 and 76. When the simulation is run once, which means that the number of replication, p , is 1, the obtained results don't make it possible to get a variance and confidence interval in the third step. Therefore, we set $p = 300$. At the end of the second step, the values for the following variables are obtained: y_i ($i. L$), $y1i$ ($i. L_q$), and $y2i$ ($i. W_q$) where $i=1, \dots, 300$, and also $total_L_q$, $total_L$, $total_W_q$, and $total_roo$.

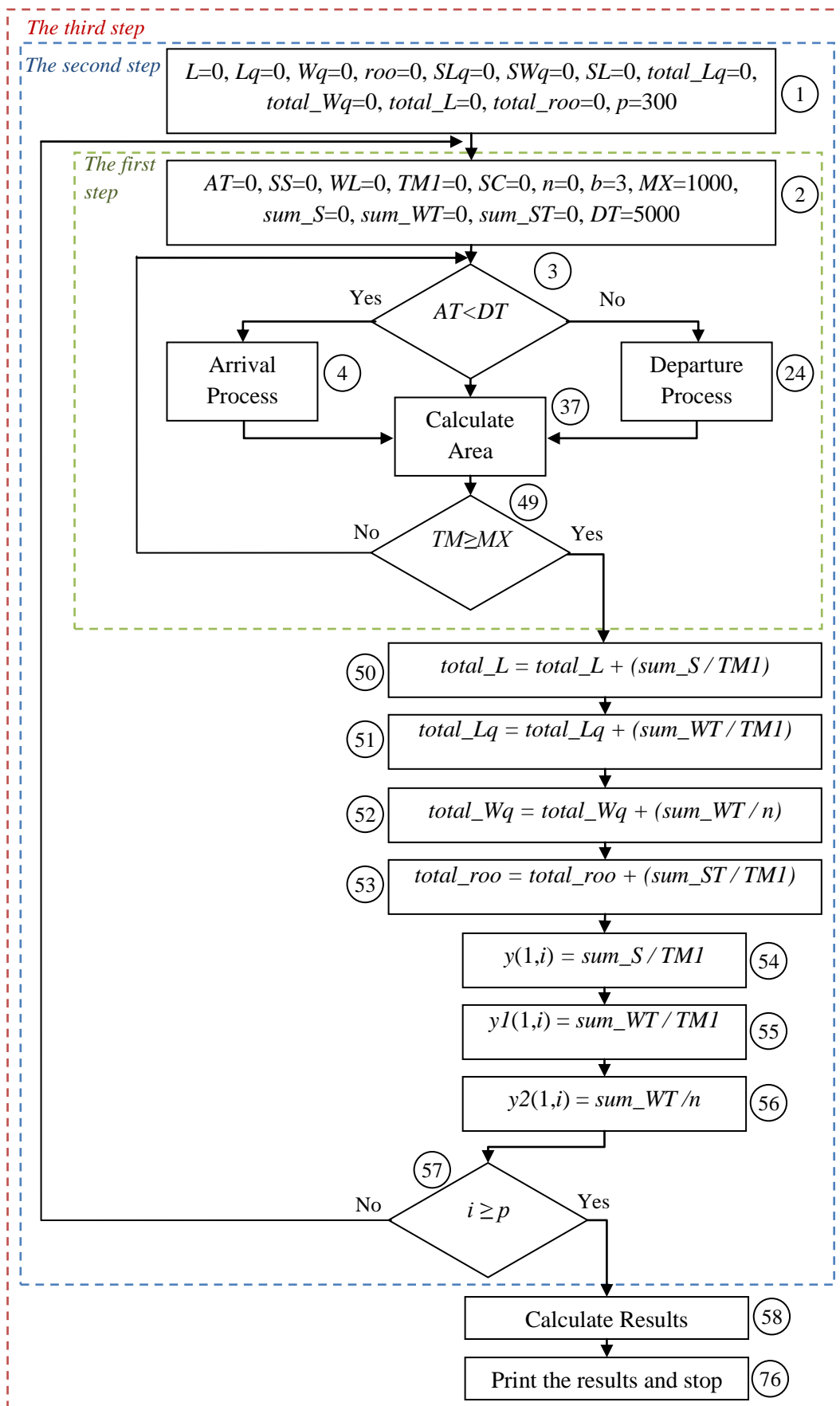
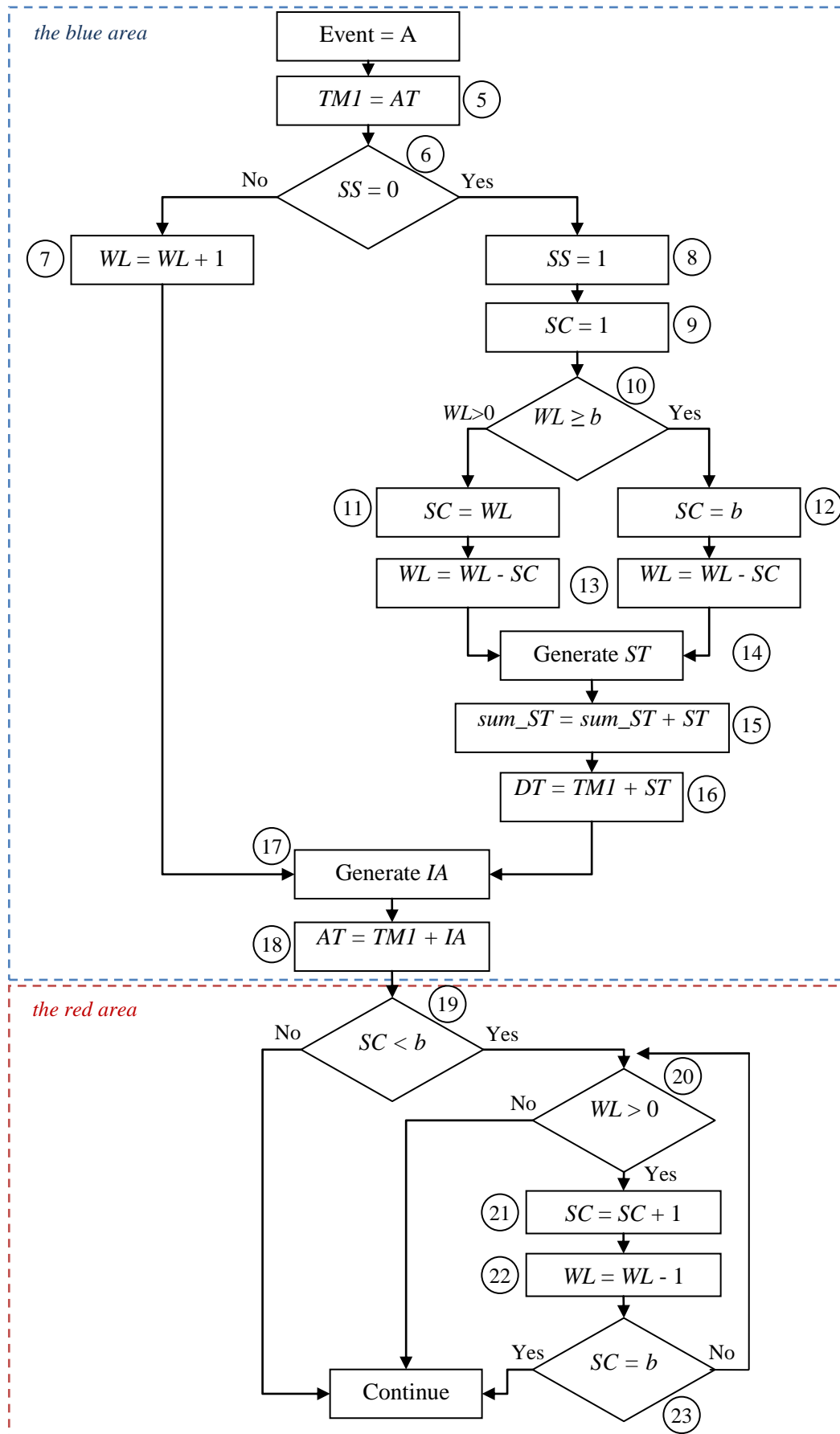


Figure 5.16 Flowchart for the queuing system with batch service.

The third step: This step consists of all blocks. Using the results of the first and the second step, we are able to get the final results of the simulation. At the end of the third step, the values for the following variables are obtained: the point estimates, \hat{L} , \hat{L}_q , \hat{W}_q , and roo , the standard error estimates, $S_{\hat{L}}$, $S_{\hat{L}_q}$, and $S_{\hat{W}_q}$, and the confidence interval estimates for L , L_q , and W_q .

In order to explain the arrival process of the simulation model, we provide a flowchart which consists of two areas: *the blue area* and *the red area* (see Figure 5.17).

We start this simulation with an empty system and assume that our first event takes place at clock time 0. This arrival finds the server idle and enters service immediately. We schedule the departure time of the first customer by randomly generating a service time from service time distribution, and then setting the departure time by means of the equation (5.1). We schedule the next arrival into the system by randomly generating an interarrival time from the interarrival time distribution, and then setting the arrival time by using the equation (5.2). The first event occurs in the blue area. At this point, it is controlled whether the capacity of server has exceeded the maximum level. If it has not, we check whether there is any customer waiting. If there is, the customer is removed from the queue, and the service is started immediately for this customer. This process occurs in the red area. We do not schedule the departure time because this customer departs from the system at the departure time of previous customer who is in service at that time. If the capacity of server has exceeded the maximum level, the next arrival joins the waiting line. Alike the simulation with batch arrival, it is important in this simulation model whether the capacity of server exceeds 3 at a time. Although in the simulation with batch arrival, the customer enters the service if the server is idle, in this simulation, any arrival enters service if the capacity of server has not exceeded the maximum level.



Both all these events and their scheduled times are maintained on the event list. If we complete all the actions for the first customer, we check the event list to determine the next scheduled event and its time. If the next event is an arrival, we move the clock time to the scheduled time of the arrival and the next process is an arrival. If the next event is a departure, we move the clock time to the time of the departure and the next process is a departure. For a departure, we control whether there are any customers in the waiting line. If there are not, we set the status of the system to idle, and the capacity of server becomes zero. If there are, we control whether the length of waiting line has exceeded the maximum capacity of the server. If it has, we remove three customers from the queue, if it has not, we remove all customers from the queue, and start the service on these customers by setting a departure time using equation (5.1). All the actions mentioned above can be summarized in the flowchart in Figure 5.18.

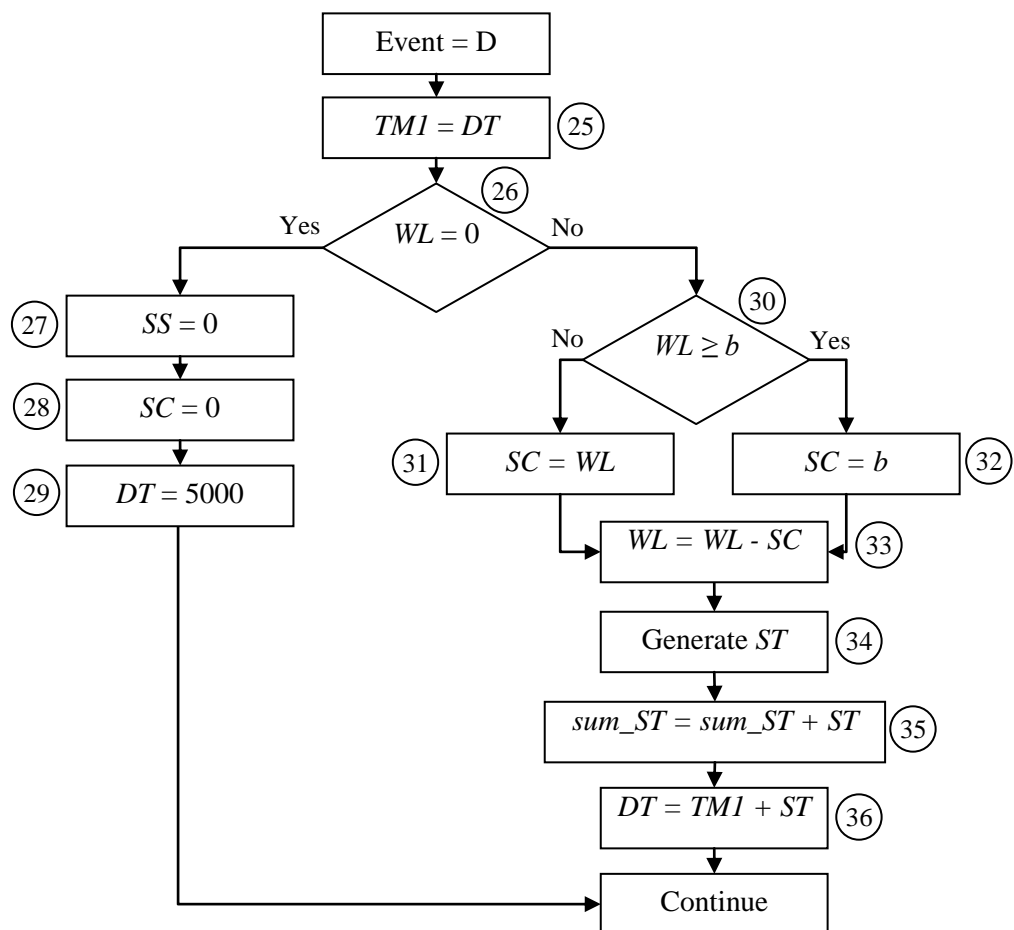


Figure 5.18 Flowchart for the departure process in Figure 5.16.

The general simulation process for the queuing model with batch service is explained above in detail. We assume that the interarrival times and the service times are generated from exponential distribution for 18 customers, which are shown in Table 5.13 and Table 5.14. In Table 5.13, we can see that the time between the first arrival (customer 1) and the second arrival (customer 2) is **0.0598** time units, the time between the second arrival (customer 2) and the third arrival (customer 3) is **0.0544** time units, and so on. In Table 5.14, we can see that the service time for the customer 1 and customer 2 is **0.0612** time units, the service time for the customer 3 and customer 4 is **0.0911** time units, and so on.

Table 5.13 Interarrival times

Customer	IA
1	0.0598
2	0.0544
3	0.0480
4	0.2192
5	0.0098
6	0.0216
7	0.2266
8	0.0335
9	0.0016
10	0.0046
11	0.0092
12	0.0001
13	0.0174
14	0.0009
15	0.1950
16	0.0055
17	0.0296
18	0.1352

Table 5.14 Service times

Customer	ST
1,2	0.0612
3,4	0.0911
5,6	0.0304
7	0.0315
8,9,10	0.0726
11,12,13	0.0114
14,15	0.1368
16,17,18	0.0890

In order to demonstrate the first step of the simulation model, we define variables used as follows:

- TM* : clock time of the simulation
AT : scheduled time of the next batch arrival
DT : scheduled time of the next departure for a customer
SS : status of the server (1=busy, 0=idle)
SC : capacity of the server

WL	: length of the queue
MX	: length of a simulation run
ST	: service time randomly generated
IA	: interarrival time randomly generated
n	: number of customer in the system
b	: number of customer in batch
L_q	: average queue length
W_q	: average waiting time in queue
$root$: root
L	: average number in system
Sum_ST	: sum of service times
Sum_WT	: sum of the waiting time in queue
Sum_S	: sum of the waiting time in system
$area_W$: the waiting time in queue
$area_S$: the waiting time in system

To start with, we first initialize all the variables (block 2 in Figure 5.16). We set $AT=0$, because the first arrival is assumed to take place at time 0. We also assume that the system is empty at time 0, so we set $SS=0$, $SC=0$, $WL=0$, and $DT=5000$, which means that our list of events now consists of two scheduled events; an arrival at time 0 and a dummy departure at time 5000. The initialization process is now completed and provides us the computer representation of the simulation shown in Table 5.15.

We again determine the first event by comparing AT and DT (block 3 in Figure 5.16). An arrival is indicated by $AT < DT$, and a departure is indicated by $AT > DT$. At this point, an arrival will take place, because $AT = 0$ and $DT = 5000$. We label this event 1 and update the clock time, TMI , to the time of event 1 (block 5 in Figure 5.17). Namely, we set $TMI = 0$. The arrival at time 0 finds the system empty, so that the customer enters the service immediately. We set $SS = 1$ in order to show that the server becomes busy (block 8 in Figure 5.17), and set $SC = 1$ in order to show that there is one customer having service (block 9 in Figure 5.17). We control whether

the length of waiting line has exceeded the maximum capacity of the server because the server can accept maximum three customers at a time. Therefore, we need to update the capacity of the server and the capacity of waiting line. At this point, we jump to block 14 in Figure 5.17, because $WL=0$.

Table 5.15 Computer representation of the first step of the simulation.

Event	Customer	Type of Event	TMI	SS	WL	SC	Event List	
							AT	DT
0	-	Initialization	0	0	0	0	0	5000.00
1	1	Arrival	0.0000	1	0	1	0.0598	0.0612
2	2	Arrival	0.0598	1	0	2	0.1142	0.0612
3	1,2	Departure	0.0612	0	0	0	0.1142	5000.00
4	3	Arrival	0.1142	1	0	1	0.1622	0.2053
5	4	Arrival	0.1622	1	0	2	0.3814	0.2053
6	3,4	Departure	0.2053	0	0	0	0.3814	5000.00
7	5	Arrival	0.3814	1	0	1	0.3912	0.4118
8	6	Arrival	0.3912	1	0	2	0.4128	0.4118
9	5,6	Departure	0.4118	0	0	0	0.4128	5000.00
10	7	Arrival	0.4128	1	0	1	0.6394	0.4443
11	7	Departure	0.4443	0	0	0	0.6394	5000.00
12	8	Arrival	0.6394	1	0	1	0.6729	0.7121
13	9	Arrival	0.6729	1	0	2	0.6745	0.7121
14	10	Arrival	0.6745	1	0	3	0.6791	0.7121
15	11	Arrival	0.6791	1	1	3	0.6883	0.7121
16	12	Arrival	0.6883	1	2	3	0.6884	0.7121
17	13	Arrival	0.6884	1	3	3	0.7057	0.7121
18	14	Arrival	0.7057	1	4	3	0.7067	0.7121
19	15	Arrival	0.7067	1	5	3	0.9017	0.7121
20	8,9,10	Departure	0.7121	1	2	3	0.9017	0.7235
21	11,12,13	Departure	0.7235	1	0	2	0.9017	0.8602
22	14,15	Departure	0.8602	0	0	0	0.9017	5000.00
23	16	Arrival	0.9017	1	0	1	0.9072	0.9907
24	17	Arrival	0.9072	1	0	2	0.9368	0.9907
25	18	Arrival	0.9368	1	0	3	1.0720	0.9907
26	16,17,18	Departure	0.9907	0	0	0	1.0720	5000.00

There is only one customer, customer 1 in the system. We next generate a service time for customer 1 (block 14 in Figure 5.17). From Table 5.14, we see that ST of customer 1 is 0.0612. At this point, we want to obtain sum of all service times. We update Sum_ST to 0.0612 by using the equation $Sum_ST = Sum_ST + ST$ (block 15 in Figure 5.17) because $Sum_ST = 0$. We set $DT = 0.0612$ by using the equation $DT = TMI + ST$ (block 16 in Figure 5.17), because $TMI = 0$. Namely, customer 1 will depart from the system at clock time 0.0612. We now schedule the next arrival into

the system by generating an interarrival time (block 17 in Figure 5.17). From Table 5.13, we see that the IA is 0.0598. We set $AT = 0.0598$ by using the equation $AT=TMl+IA$ (block 18 in Figure 5.17) because $TMl = 0$. Namely, the second arrival (customer 2) will take place at clock time 0.0598. At the end of these actions, the blue area of the arrival process ends, and we continue with the red area of the arrival process. We control whether the capacity of the server has exceeded the maximum capacity of the server (block 19 in Figure 5.17). At this point, we see that the server can accept one customer, because $SC=1$, and check whether there are any customers waiting (block 20 in Figure 5.17). At the end of these actions, the red area of the arrival process ends, because $WL=0$.

We require to calculate the waiting time in the system, $area_S$ for customer 1, and the waiting time in the queue, $area_W$ for the customers in the queue, so go ahead with block 37 (see Figure 5.16). We set $area_W=0$ (block 41 in Figure 5.19), because $WL=0$, which means that there is no customer in the queue, then, $area_S = 0.0598$ (block 42 in Figure 5.19), because $AT=0.0598$, $TMl=0$, $WL=0$ and $SC=1$. We set $n=1$ (block 43 in Figure 5.19), because it is the first arrival. We check whether there is a dummy departure (block 44 in Figure 5.19), and realize that there is not. We update $Sum_WT = 0$ by using the equation $Sum_WT = Sum_WT + area_W$ (block 47 in Figure 5.19), because $Sum_WT = 0$, and $Sum_S = 0.0598$ by using the equation $Sum_S = Sum_S + area_S$ (block 48 in Figure 5.19), because $Sum_S = 0$ (see Figure 5.20). Table 5.15 shows the computer representation of the simulation at the end of the event 1.

We move on with block 49 in Figure 5.16 in order to determine whether the clock time, TMl , has exceeded the specified time length of simulation, MX . At this point, we loop back to block 3 in Figure 5.16 in order to determine the next event, event 2. Event 2 will be an arrival at time 0.0598, because $AT = 0.0598$ and $DT = 0.0612$. We update the clock time, TMl , to the time of event 2 (block 5 in Figure 5.17). Namely, we set $TMl = 0.0598$. At time 0.0598, customer 2 enters the system. We check the status of the server in order to see whether the server is idle (block 6 in Figure 5.17). The customer 2 immediately joins the waiting line (block 7 in Figure 5.17), because

there is customer 1 in the service, so we set $WL=1$. We now schedule the next arrival into the system by generating an interarrival time (block 17 in Figure 5.17). From Table 5.13, we see that the IA is 0.0544. We set $AT = 0.1142$ by using the equation $AT = TMI + IA$ (block 18 in Figure 5.17), because $TMI = 0.0598$. Namely, the third arrival (customer 3) will take place at clock time 0.1142. At the end of these actions, the blue area of the arrival process is ended, and we continue with the red area of the arrival process. We control if the server accepts any customer (block 19 in Figure 5.17), and realize that it does, because $SC=1$. At this point, we control whether there are any customers waiting (block 20 in Figure 5.17). We set $SC=2$ (block 21 in Figure 5.17), $WL=0$ (block 22 in Figure 5.17) by removing customer 2 from the waiting line, because $WL=1$ and $SC=1$. Then, we control again whether the capacity of the server equals to the maximum capacity of the server (block 23 in Figure 5.17). At this point, we see that it does not, because $SC=2$. Namely, the server may accept another customer. Therefore, we control again whether there are any customers waiting by looping back to block 20 in Figure 5.17. At the end of these actions, the red area of the arrival process is ended, because WL is 0.

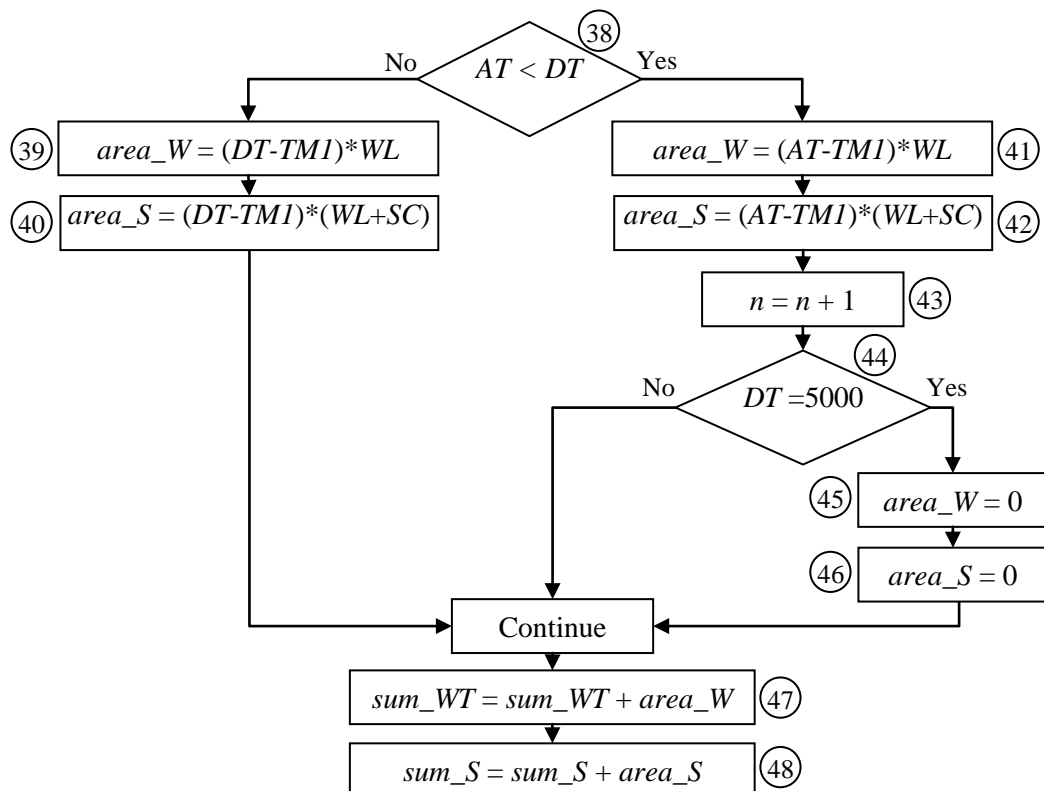


Figure 5.19 Flowchart for the calculate area in Figure 5.16.

We want to calculate the waiting time in the system, $area_S$ for customer 1 and customer 2, and the waiting time in queue, $area_W$ for the customers in the queue, so proceed to block 37 in Figure 5.16. We set $area_W=0$ (block 39 in Figure 5.19), because $WL=0$, then, $area_S = 0.0028$ (block 40 in Figure 5.19), because $DT=0.0612$, $TMI=0.0598$, $WL=0$ and $SC=2$. We update $Sum_WT = 0$ by using the equation $Sum_WT = Sum_WT + area_W$ (block 47 in Figure 5.19), because $Sum_WT = 0$, and $Sum_S = 0.0626$ by using the equation $Sum_S = Sum_S + area_S$ (block 48 in Figure 5.19), because $Sum_S = 0.0598$ (see Figure 5.20). Table 5.15 shows the computer representation of the simulation at the end of the event 2.

We move on with block 49 in Figure 5.16 in order to determine whether the clock time, TMI , has exceeded the specified time length of simulation, MX . At this point, we loop back to block 3 in Figure 5.16 in order to determine the next event, event 3. Event 3 will be a departure at time 0.0612, because $AT = 0.1142$ and $DT = 0.0612$. We update the clock time, TMI , to the time of event 3 (block 25 in Figure 5.18). Namely, we set $TMI = 0.0612$. At time 0.0612, customer 1 and customer 2 departure from the system. We check the waiting line in order to see whether there are any customers waiting (block 26 in Figure 5.18). At this point, since $WL=0$, we see that there is no customer, and set $SS=0$, $SC=0$, and $DT=5000$. We require to calculate the waiting times in the service or waiting line, so go on with block 37 (see Figure 5.16). We set $n=2$ (block 43 in Figure 5.19), because $n=1$. We set $area_W=0$ and $area_S=0$ (block 45 and 46 in Figure 5.19), because there is a dummy departure, $DT=5000$. Therefore, sum_WT and sum_S do not change. Table 5.15 shows the computer representation of the simulation at the end of the event 3.

The same procedure explained above for the first three events can be applied for the other events until the clock time, TMI , is either greater or equal than the specified time length of simulation, MX .

Figure 5.20 provides data for all the events. This figure also displays the changes in the system as a function of simulation clock time.

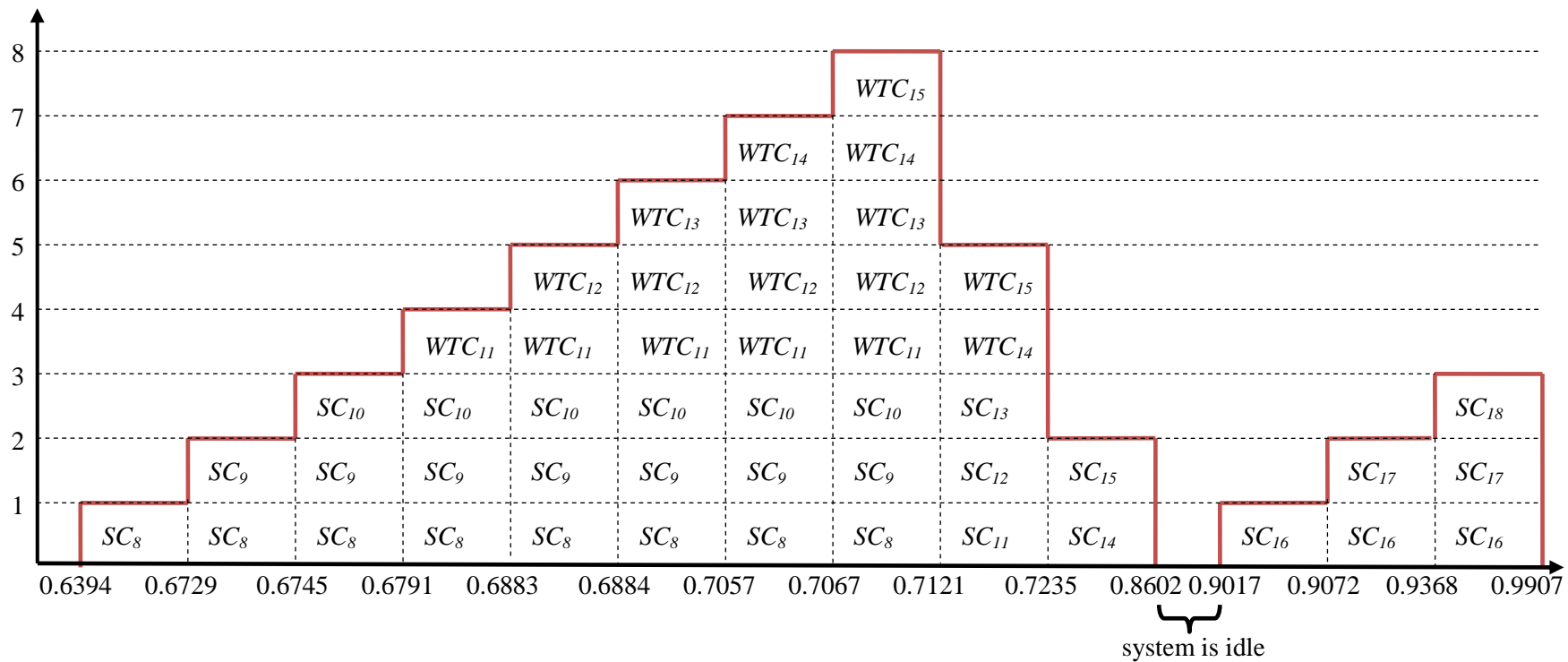
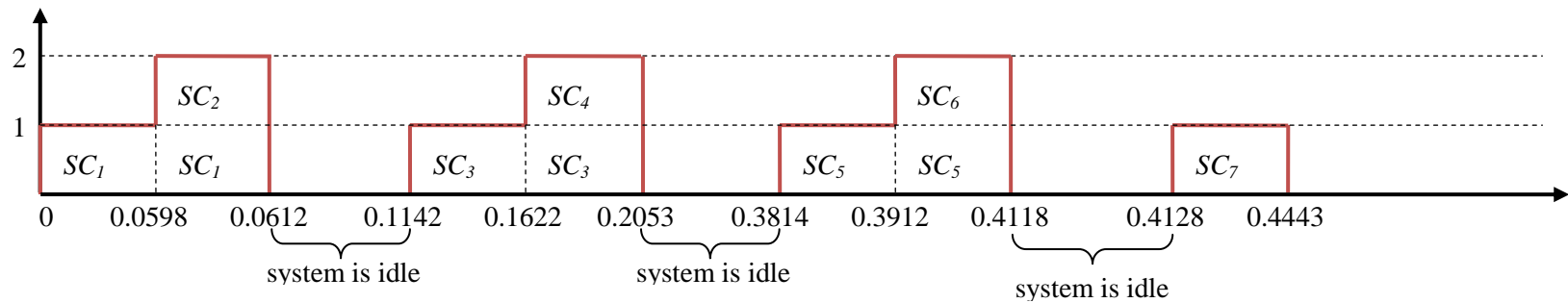


Figure 5.20 Changes in the system as a function of simulation clock time for the queuing system with batch service.

At the end of the first step, Sum_S , Sum_ST , Sum_WT , and n are calculated. Additionally, only one single value is obtained for each of the following variables: the average queue length, L_q , the average number in system, L , the root, roo , and the average waiting time in queue, W_q .

We know that if the simulation is run only once, then it is not probable to calculate a variance and confidence interval in the third step, because we need to obtain a variety of L , L_q , W_q , and roo for these calculations. To sum up, we purpose to acquire a plenty of L , L_q , W_q , and roo in this step. At this point, it involves that the simulation is run 300 times, which means we set $p = 300$ (block 1 in Figure 5.16). We now ready to show the second step. This step consists of all blocks except for block 58 and 76 in Figure 5.16.

The second step of the simulation consists of the following variables, in addition to the ones mentioned in the first step:

p	: number of replication
y_i	: i . average queue length ($i. L$)
$y1_i$: i . average number in system ($i. L_q$)
$y2_i$: i . average waiting time in queue ($i. W_q$)
$total_L_q$: sum of the average queue lengths
$total_W_q$: sum of the average waiting times in queue
$total_roo$: sum of the roots
$total_L$: sum of the average number in system

In case $p = 1$, we have Sum_WT (sum of W_q), sum_ST (sum of ST), and sum_S (sum of W_s) for the customers in the first replication. We firstly calculate L_q , W_q , roo , and L by using relations in the equations (4.4), (4.3), (4.5), and (4.1), respectively. Then, $total_L_q$, $total_W_q$, $total_roo$, and $total_L$ are updated by using these results, respectively (block 51, 52, 53, and 50 in Figure 5.16). Finally, we set y_1 ($i. L$), $y1_1$ ($i. L_q$), and $y2_1$ ($i. W_q$) by using values of L , L_q , and W_q , respectively (block 54, 55, and 56 in Figure 5.16). At the end of the first replication, we have the

data for y_1 ($1. L$), $y1_1$ ($1. L_q$), and $y2_1$ ($1. W_q$), $total_L_q$, $total_L$, $total_W_q$, and $total_roo$.

In case $p = 2$, we loop back to block 2 in Figure 5.16 to initialize all the variables in the first step (block 2 in Figure 5.16). At this point, the first step of the simulation is run again for new customers. At this point, we have Sum_WT (sum of W_q), sum_ST (sum of ST), and sum_S (sum of W_s) for the customers in the second replication. We firstly calculate L_q , W_q , roo , and L by using relations in the equations (4.4), (4.3), (4.5), and (4.1), respectively. Then $total_L_q$, $total_W_q$, $total_roo$, and $total_L$ are updated by using these results, respectively (block 51, 52, 53, and 50 in Figure 5.16). Finally, we set y_2 ($2. L$), $y1_2$ ($2. L_q$), and $y2_2$ ($2. W_q$) by using values of L , L_q , and W_q , respectively (block 54, 55, and 56 in Figure 5.16). At the end of the second replication we have the data for y_1 ($1. L$), $y1_1$ ($1. L_q$), $y2_1$ ($1. W_q$), y_2 ($2. L$), $y1_2$ ($2. L_q$), and $y2_2$ ($2. W_q$). In addition, $total_L_q$, $total_L$, $total_W_q$, and $total_roo$ are updated.

The same procedure explained above can be applied for the rest of the replications until p is 300. At the end of the second step of the simulation, we get the values of y_i ($i. L$), $y1_i$ ($i. L_q$), and $y2_i$ ($i. W_q$) where $i=1, \dots, 300$, and also of $total_L_q$ (sum of L_q), $total_L$ (sum of L), $total_W_q$ (sum of W_q), and $total_roo$ (sum of roo).

Now, we can move to analyze the third step. As explained before, this step consists of all blocks demonstrated in Figure 5.16. In the light of the results obtained in the first and second step, now we calculate the final outcomes for the simulation since all the data we need to make calculations in the third step has already been obtained. Also, at the end of this step, we will be able to complete the simulation.

The third step of the simulation consists of the following variables, in addition to the ones mentioned in the first and second step:

- SL_q : sum of squared of errors for L_q
- SL : sum of squared of errors for L

SW_q	: sum of squared of errors for W_q
SL_{q1}	: standard error for L_q
$SL1$: standard error for L
SW_{q1}	: standard error for W_q
$cdown$: down limit for confidence interval of L_q
cup	: up limit for confidence interval of L_q
$cdown1$: down limit for confidence interval of W_q
$cup1$: up limit for confidence interval of W_q
$cdown2$: down limit for confidence interval of L
$cup2$: up limit for confidence interval of L .

We calculate the point estimates of L , L_q , W_q , and roo by using the equation (4.6). (block 59, 60, 61, and 62 in Figure 5.21). We calculate the sum of squared of errors for \hat{L}_q , \hat{L} , and \hat{W}_q (block 63, 64, and 65 in Figure 5.21), and the standard error estimates of \hat{L}_q , \hat{L} , and \hat{W}_q by using the equation (4.7), SL_{q1} , $SL1$, and SW_{q1} (block 67, 68, and 69 in Figure 5.21). We finally calculate the confidence interval estimates by using the equation (4.8); $cdown$, cup for L_q (block 70 and 71 in Figure 5.21), $cdown1$, $cup1$ for W_q (block 72 and 73 in Figure 5.21), $cdown2$, $cup2$ for L (block 74 and 75 in Figure 5.21).

At the end of the third step, in other words, at the end of the simulation, the data for the point estimates, \hat{L} , \hat{L}_q , \hat{W}_q , and roo , the standard error estimates, $S_{\hat{L}}$, $S_{\hat{L}_q}$, and $S_{\hat{W}_q}$, and the confidence interval estimates of L , L_q , and W_q are achieved.

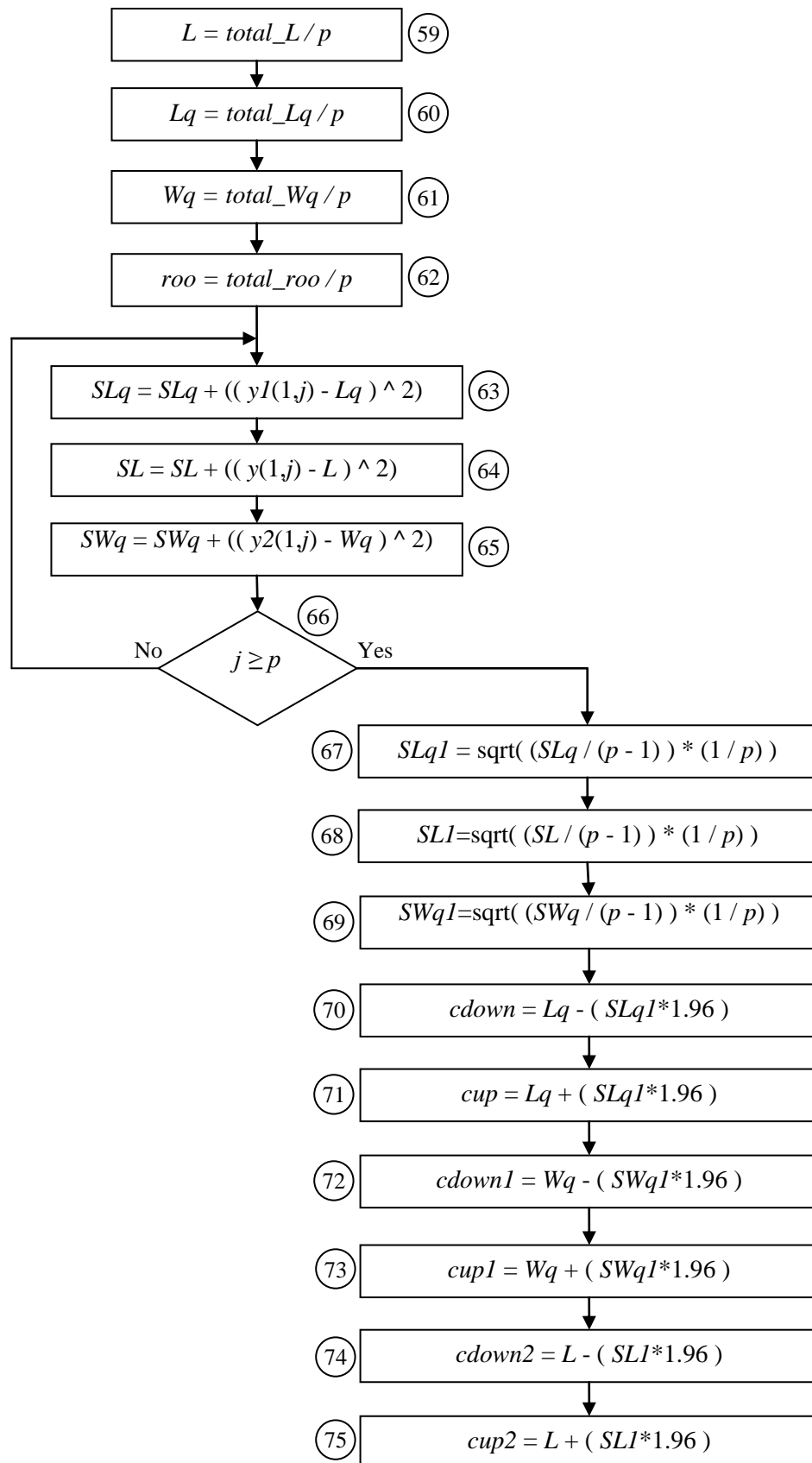


Figure 5.21 Flowchart for the calculate results in Figure 5.16.

5.2.1 Determination of run length and number of replications

We have analyzed whether there is a warm-up period in this queueing system. At this point, the three simulation studies have been again chosen as following:

- (1) batch size 2, arrival rate 20, and batch service rate 12,
- (2) batch size 3, arrival rate 20, and batch service rate 10,
- (3) batch size 4, arrival rate 20, and batch service rate 8.

Note that, for batch size 2, 3, and 4 the systems have highest traffic intensity in (1), (2), and (3) respectively (see Table 5.18). Therefore, we analyze the three systems to find whether there is a warm-up period. If warm-up period is not occurred when the traffic intensity is high, we can stop the simulation run at specified time.

If the queueing system that is working with higher traffic intensity does not involve a warm-up period, the systems with lower traffic intensity never occupy a warm-up period. That's why, we have chosen the above mentioned systems with the highest traffic intensity for each batch size. To find a warm-up period, we have obtained the graphs which illustrate waiting time in the system versus the simulation clock time by the run length 2000. These graphs are represented in Figure 5.22, 5.23, and 5.24.

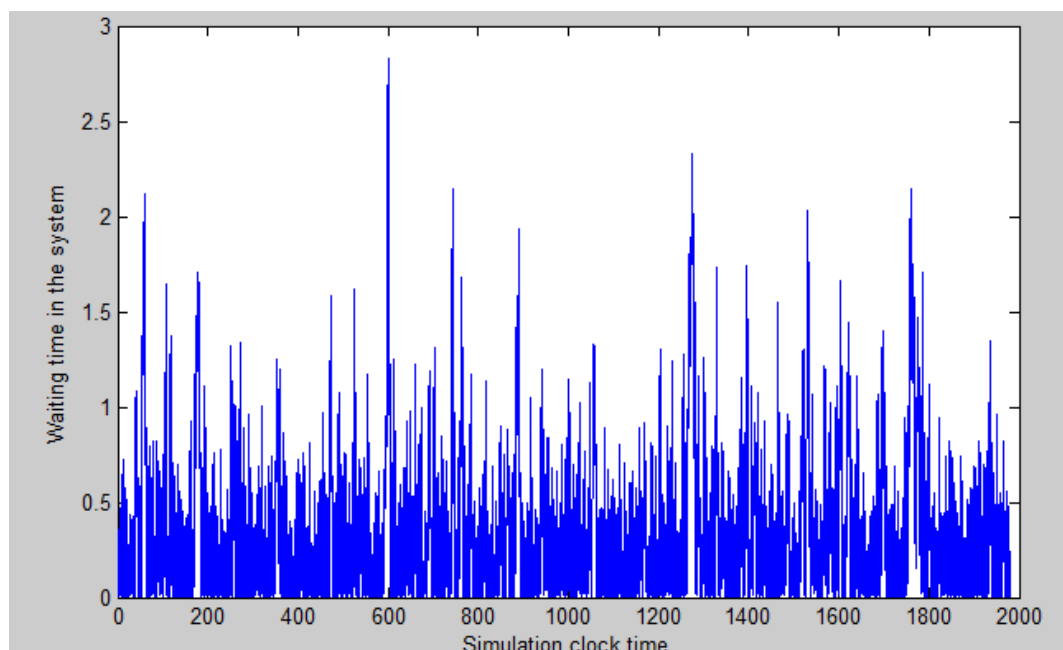


Figure 5.22 Batch size 2, arrival rate 20, batch service rate 12 and traffic intensity 0.8850

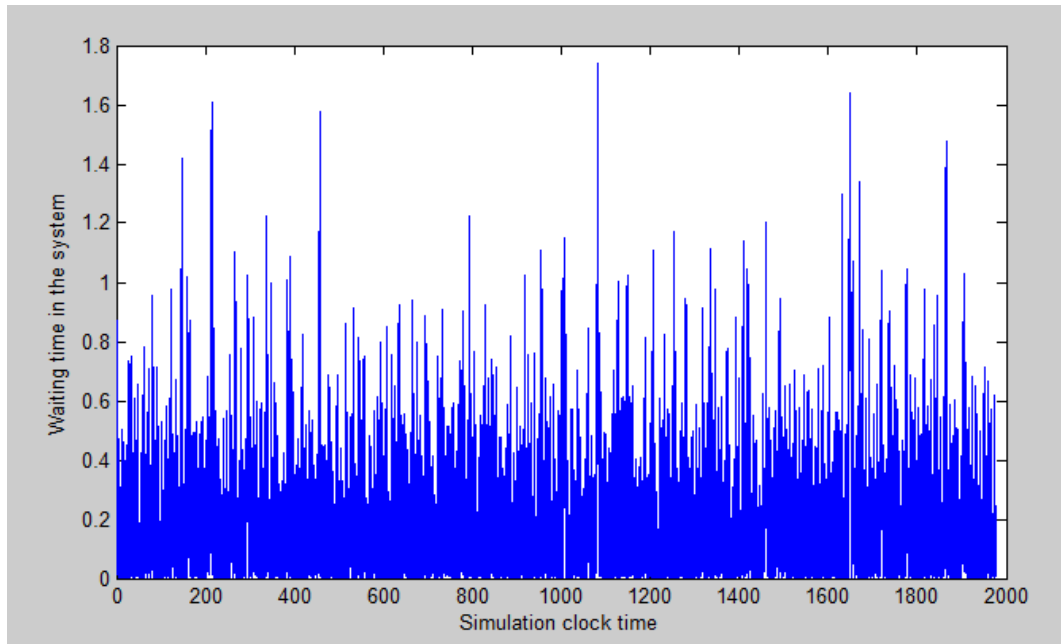


Figure 5.23 Batch size 3, arrival rate 20, batch service rate 10 and traffic intensity 0.8106

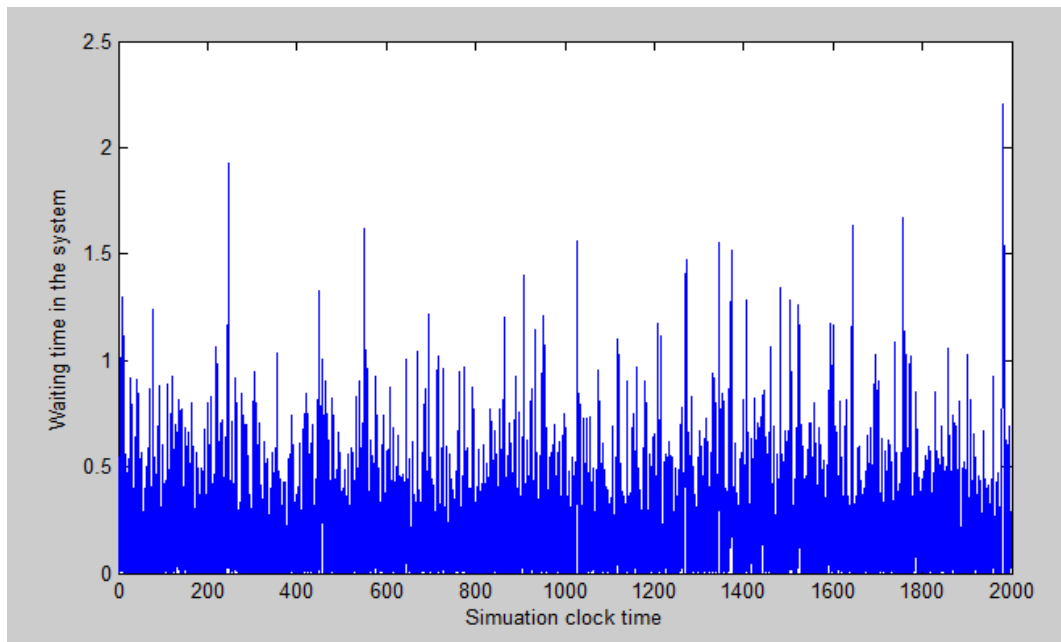


Figure 5.24 Batch size 4, arrival rate 20, batch service rate 8 and traffic intensity 0.8203

The figures illustrate that the effect of initial conditions on later observations is fairly less. Besides observations appear around a constant mean in long-time for each graph. Namely, we have no warm-up period in each study. Therefore, we have collected all data in the run length.

In this section, we will consider how to determine run length MX , and number of replications, p , because of the same reasons mentioned before in the section 5.1.1. Some pilot runs were made for different run length and numbers of replication for the queuing systems with batch service.

In these pilot runs, arrival rate λ and service rate μ are kept constant as 20 and 12, respectively. If the batch size is 10, the traffic intensity is $\rho = \lambda/r\mu = 20/(10)(12) = 0.1667$, and if the batch size is 11, the traffic intensity is $\rho = \lambda/r\mu = 20/(11)(12) = 0.1515$. We can easily see that the batch size decreases the traffic intensity. At this point, we can say that in batch service system, the traffic intensity decreases while the batch size increases opposite to the batch arrival system. We investigate when the system is intensive, so we continue to make the pilot runs until batch size is 10. The values of S_i are shown in Table 5.16.

For the first pilot-runs, we have run the simulation by taking $MX = 50$ and $p = 1000$, and seen that these values were not adequate. And then, we have decided to to make new calculations by increasing MX so decreasing p since the total simulation run-length was kept constant. Since it is not required for p to be less than 30, at the last try, we have run the simulation with the values of 1.000 for MX and of 50 for p .

Table 5.16 Traffic intensity versus the total simulation run length.

Batch Size:r		2	3	4	5	6	7	8	9
ρ		0.8333	0.5555	0.4167	0.3333	0.2778	0.2381	0.2083	0.1852
MX	p								
50	1000	0.09763	0.01359	0.00826	0.00577	0.00512	0.00472	0.00443	0.00431
100	500	0.09370	0.01386	0.00816	0.00597	0.00498	0.00455	0.00427	0.00423
200	250	0.10878	0.01373	0.00745	0.00615	0.00504	0.00482	0.00438	0.00419
500	100	0.09825	0.01310	0.00683	0.00680	0.00555	0.00477	0.00508	0.00372
1000	50	0.09828	0.01183	0.00784	0.00589	0.00459	0.00441	0.00488	0.00436
1000	200	0.04886	0.00631	0.00397	0.00321	0.00252	0.00233	0.00199	0.00221
1000	300	0.04194	0.00579	0.00319	0.00238	0.00209	0.00194	0.00181	0.00173

The total simulation run length is 50.000 as seen in the first five rows in Table 5.16. And this value for the total simulation run length can be accepted for all batch sizes except for 2 because we require the standard error to be less than 0.05. That's why, we have accepted that 50.000 is not adequate and decided to increase the total simulation run length and made new pilot runs. The results of the pilot runs can be seen in the last two rows of Table 5.16. Firstly, the total simulation run length 50.000 has been increased to 200.000, and it has been found adequate for all batch sizes. We have continued to make pilot runs, because 0.04886 is fairly close to 0.05. Finally, we have decided to run the simulation program by taking $MX = 1000$ and $p = 300$.

5.2.2 Comparison between the results of simulation and the analytic results

The simulation program for the queuing model with batch service has been run by taking $MX = 1000$ and $p = 300$. With the same purposes mentioned in section 5.1.2, we have firstly obtained point estimate for the performance measures by using simulation program and analyze the statistical precision of these estimates by using the confidence interval estimate. Secondly, we require to show whether the statistical precision of estimates is affected by the batch size.

Throughout the first purpose, we have run the simulation program (see A3.2) 15 times for different batch size values and batch service rates, and obtained the estimates for the performance measures. The simulation results are shown in Table 5.17.

Table 5.17 The simulation results or point estimates for the performance measures.

Batch Size: r	Arrival Rate: λ	Batch Service Rate: μ	Simulation Results			
			\hat{r}_0	\hat{L}	\hat{L}_q	\hat{W}_q
2	20	12	0.8850	7.7072	6.0390	0.3016
	20	14	0.7957	3.8811	2.4525	0.1226
	20	16	0.7246	2.6340	1.3844	0.0692
	20	18	0.6670	2.0047	0.8927	0.0446
3	20	10	0.8106	4.2724	2.2725	0.1136
	20	12	0.7335	2.7536	1.0875	0.0544
	20	14	0.6719	2.0489	0.6226	0.0312
	20	16	0.6220	1.6431	0.3938	0.0197
	20	18	0.5801	1.3826	0.2706	0.0135
4	20	8	0.8203	4.5665	2.0675	0.1034
	20	10	0.7408	2.8600	0.8635	0.0432
	20	12	0.6793	2.1207	0.4530	0.0227
	20	14	0.6290	1.6970	0.2666	0.0133
	20	16	0.5861	1.4159	0.1673	0.0084
	20	18	0.5500	1.2216	0.1115	0.0056

We have obtained some numerical results for the batch size values and batch service rates in simulation (see A2.2). The numerical results are shown in Table 5.18.

Table 5.18 The numerical results for the performance measures.

Batch Size: r	Arrival Rate: λ	Batch Service Rate: μ	Numerical Results			
			r_0	L	L_q	W_q
2	20	12	0.8844	7.6505	5.9838	0.2992
	20	14	0.7955	3.8899	2.4614	0.1231
	20	16	0.7247	2.6324	1.3824	0.0691
	20	18	0.6667	2.0003	0.8892	0.0444
3	20	10	0.8105	4.2770	2.2770	0.1138
	20	12	0.7336	2.7537	1.0870	0.0543
	20	14	0.6724	2.0525	0.6239	0.0311
	20	16	0.6221	1.6462	0.3962	0.0198
	20	18	0.5799	1.3804	0.2693	0.0135
4	20	8	0.8206	4.5741	2.0741	0.1037
	20	10	0.7413	2.8655	0.8655	0.0433
	20	12	0.6792	2.1172	0.4505	0.0225
	20	14	0.6287	1.6932	0.2646	0.0132
	20	16	0.5864	1.4178	0.1678	0.0084
	20	18	0.5502	1.2232	0.1121	0.0056

When we compare the simulation results in Table 5.17 to the numerical results in Table 5.18, it can be derived that the simulation results are fairly close to the numerical results. However, with these findings, anything about the statistical precision of these estimates cannot be claimed. Therefore, we have calculated the confidence interval estimates of performance measures and standard error estimates necessary for these confidence intervals in the simulation program. In the calculations, we have used confidence level $\alpha = 0.05$, namely, confidence coefficient 0.95. The each interval including the true performance measure is confided in 95%, for example, in case of the batch size is 2, the arrival rate is 20, and batch service rate is 12, the interval $7.624229 < L < 7.790220$ includes the true $L = 7.6505$ confided in 95%. These results are provided in Table 5.19.

Table 5.19 The standard error estimates of estimators and confidence interval estimates of performance measures.

Batch Size: r	Arrival Rate: λ	Batch Service Rate: μ	Standard Error Estimate			Confidence Interval Estimate (95%)		
			$S_{\hat{L}}$	$S_{\hat{L}_q}$	$S_{\hat{W}_q}$	L	L_q	W_q
2	20	12	0.042345	0.041374	0.002005	7.624229;7.790220	5.957900;6.120088	0.297668;0.305528
	20	14	0.012075	0.011367	0.000549	3.857443;3.904779	2.430222;2.474781	0.121494;0.123644
	20	16	0.006145	0.005407	0.000258	2.621938;2.646026	1.373761;1.394956	0.068734;0.069747
	20	18	0.003906	0.003266	0.000155	1.997040;2.012352	0.886337;0.899138	0.044318;0.044924
3	20	10	0.012298	0.011153	0.000545	4.248300;4.296510	2.250639;2.294359	0.112489;0.114627
	20	12	0.005546	0.004521	0.000219	2.742746;2.764485	1.078644;1.096366	0.053944;0.054802
	20	14	0.003500	0.002561	0.000124	2.042035;2.055753	0.617623;0.627660	0.030917;0.031402
	20	16	0.002354	0.001562	0.000075	1.638497;1.647723	0.390769;0.396892	0.019537;0.019832
	20	18	0.001913	0.001157	0.000056	1.378839;1.386336	0.268313;0.272849	0.013416;0.013635
4	20	8	0.011792	0.010160	0.000494	4.543423;4.589647	2.047608;2.087437	0.102396;0.104333
	20	10	0.005227	0.003941	0.000194	2.849758;2.870246	0.855749;0.871199	0.042795;0.043555
	20	12	0.003185	0.002064	0.000101	2.114483;2.126970	0.448927;0.457018	0.022459;0.022855
	20	14	0.002105	0.001211	0.000059	1.692919;1.701172	0.264193;0.268938	0.013196;0.013427
	20	16	0.001702	0.000812	0.000039	1.412558;1.419229	0.165677;0.168859	0.008287;0.008442
	20	18	0.001384	0.000613	0.000030	1.218914;1.224339	0.110343;0.112747	0.005518;0.005637

We always require obtaining the narrow confidence intervals so that we rearrange Table 5.19 to Table 5.20 in order to show the width of each confidence interval clearly.

Table 5.20 The width of the confidence interval in Table 5.19.

Batch Size: r	Arrival Rate: λ	Batch Service Rate: μ	ρ	Width of Confidence Interval		
				L	L_q	W_q
2	20	12	0.8333	0.1660	0.1622	0.0079
	20	14	0.7143	0.0473	0.0446	0.0022
	20	16	0.6250	0.0241	0.0212	0.0010
	20	18	0.5555	0.0153	0.0128	0.0006
3	20	10	0.6667	0.0482	0.0437	0.0021
	20	12	0.5555	0.0217	0.0177	0.0009
	20	14	0.4762	0.0137	0.0100	0.0005
	20	16	0.4167	0.0092	0.0061	0.0003
	20	18	0.3704	0.0075	0.0045	0.0002
4	20	8	0.6250	0.0462	0.0398	0.0019
	20	10	0.5000	0.0205	0.0154	0.0008
	20	12	0.4167	0.0125	0.0081	0.0004
	20	14	0.3571	0.0083	0.0047	0.0002
	20	16	0.3125	0.0067	0.0032	0.0002
	20	18	0.2778	0.0054	0.0024	0.0001

We obtain the most narrow confidence intervals when the traffic intensity has the smallest value; 0.5555, 0.3704, and 0.2778 for batch sizes 2, 3, and 4, respectively. This can be seen in Table 5.20.

For the second purpose, we plan to show whether the statistical precision of estimates is affected by the batch size. We have formed a new table from Table 5.20 (see Table 5.21).

We know that the traffic intensity $\rho = \lambda/r\mu$ affect to the statistical precision of estimates. When the arrival rate and the batch service rate are kept constant as 20 and 12, respectively, we can see that the minimum batch size value provides the maximum traffic intensity. Consequently, the maximum batch size value gives us the most confidential estimate (see Table 5.21).

Table 5.21 Relation between statistical precision of estimate and batch size.

Batch Size: r	Arrival Rate: λ	Batch Service Rate: μ	ρ	L	L_q	W_q
2	20	12	0.8333	0.1660	0.1622	0.0079
3	20	12	0.5555	0.0217	0.0177	0.0009
4	20	12	0.4167	0.0125	0.0081	0.0004

5.2.3 Impact of batch size over the performance measures

In this section, we have analyzed how batch size affects the performance measures. Through this purpose, for different batch sizes, we have obtained the graphs of the batch service rates versus the performance measures. (see in Figure 5.25, 5.26, and 5.27).

When we examine Figure 5.25 for the batch size 2, we see that the waiting time in queue decreases while the batch service rate increases. Similarly, when we look at the batch size 3 and 4, we realize again that the waiting time in queue decreases while the batch service rate increases. Furthermore, it is also clear that the batch size increases, the decrease rate of the waiting time becomes lower.

Similarly, it can be derived from the graph that the decrement rates for different traffic intensities change. For instance; as far as the interval (14, 16) is concerned, we can see that the waiting time in queue decreases as batch size increases. On the other hand, when we analyze the interval (12, 14), it is obvious that the decrease rate is higher compared to the interval (14, 16). It can be explained by the traffic intensity. It is closer to the heavy traffic limit in the interval (12, 14) than in the interval (14, 16). To sum up, the more intense the system is, the more waiting time it requires.

Similar explanations can be made for the number of customer in queue which means the queue length. When we look at Figure 5.26 for the batch size 2, 3, and 4, we can see that the number of waiting customers in queue decreases while the batch service rate increases.

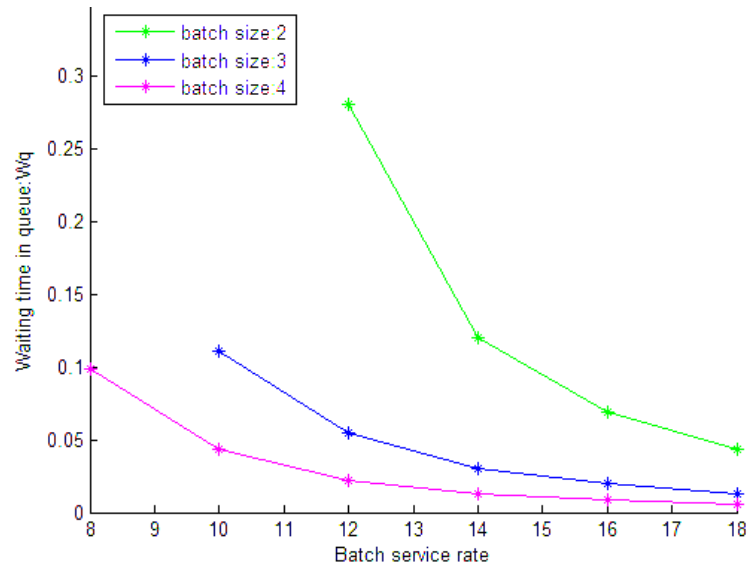


Figure 5.25 Batch service rate versus W_q for different batch sizes.

Similarly, it can be derived from the graph that the decrement rates for different traffic intensities change. For instance; as far as the interval (14, 16) is concerned, we can see that the number of waiting customers in queue decreases as batch size increases. On the other hand, when we analyze the interval (12, 14), it is obvious that the decrement rate is higher compared to the interval (14, 16). It can be explained by the traffic intensity.

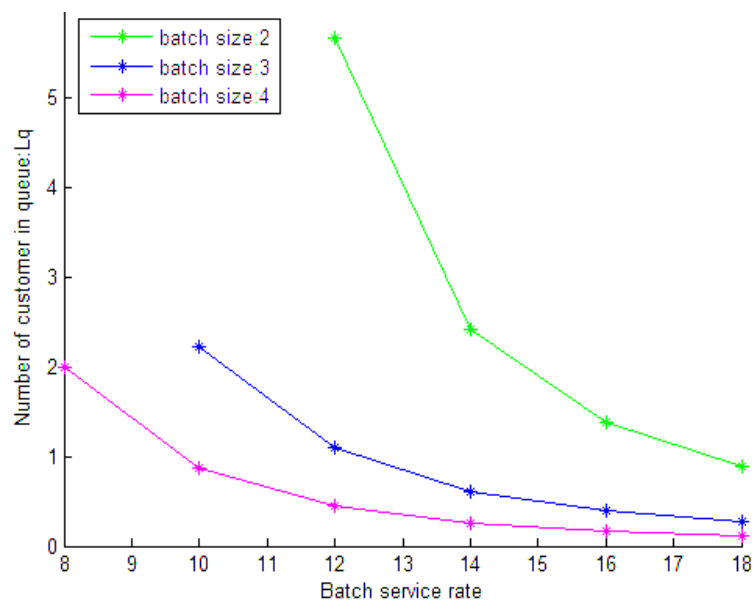


Figure 5.26 Batch service rate versus L_q for different batch sizes.

When we analyze Figure 5.27 for the batch sizes, we can understand that the service facility which refers to the traffic intensity decreases as the batch service rate increases. It is also obvious that the batch size increases, the decrement rate of the service facility becomes slightly higher. What's more, according to the graph, we can say that the large batch size causes the traffic intensity to approach heavy-traffic limit more slowly.

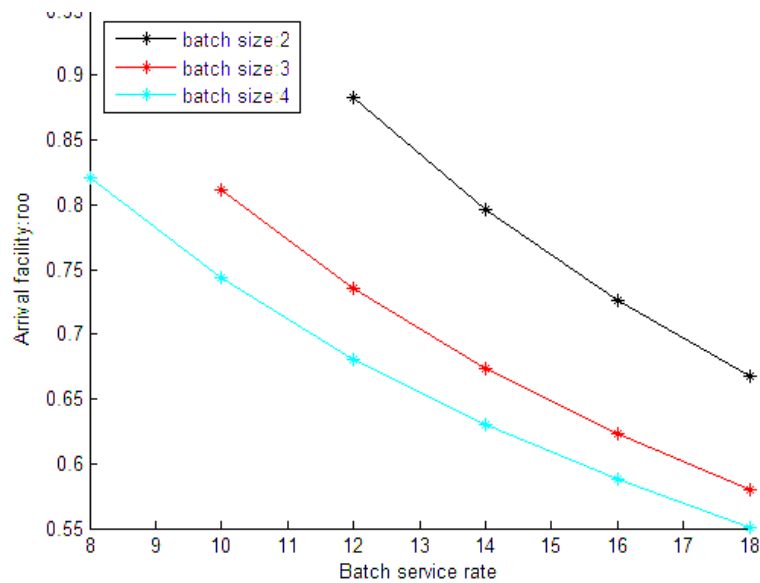


Figure 5.27 Batch service rate versus ρ for different batch sizes.

CHAPTER SIX

CONCLUSION

In this study, we have two purposes to achieve: (1) we obtain point estimates for the performance measures by using simulation program and analyze the statistical precision of these estimates by using the confidence interval estimate; (2) we require to show whether the statistical precision of estimates is affected by the batch size.

First of all, we have determined the run length and the number of replications for the batch arrival queueing system with fixed batch size. We can say that if the traffic intensity (ρ) approaches to the heavy-traffic limit, it is unavoidable that the total simulation run is longer. Therefore, we can say that if batch size increases, then the traffic intensity increases too. We have chosen batch size 2, 3, and 4, and determined the total simulation run length for the most intensity case. The most intensity case has been obtained for the batch size 3, the batch arrival rate 30, and the service rate 100. So, the plot runs have been made in order to determine the total simulation run length. Consequently, we have decided to run the simulation program by taking $MX = 1200$ and $p = 1200$.

Throughout the first purpose, we have run the simulation program (see A3.1) 14 times for different batch size values and batch arrival rates, and obtained the point estimates for the performance measures. We have also obtained some numerical results for the batch size values and batch arrival rates in simulation (see A2.1). We have compared the simulation results to the numerical results, and realized that the simulation results were fairly close to the numerical results. However, with these findings, anything about the statistical precision of these estimates cannot be claimed. Therefore, we have calculated the confidence interval estimates for the performance measures in the simulation program. In the calculations, we have used confidence level $\alpha = 0.05$, namely, confidence coefficient 0.95. We always want to obtain the narrow confidence intervals. As a result, in the calculations, we have seen

that the most narrow confidence intervals have been obtained when the traffic intensity has the smallest value.

With the help of Whitt's explanation (1989), we have examined whether the statistical precision of estimates is affected by the batch size. We know that the traffic intensity $\rho = \lambda r / \mu$ affects the statistical precision of estimates. When the traffic intensity is kept constant, we have seen that when the batch size changes, unsurprisingly the batch arrival rate changes. And, at last, as we have seen, the statistical precision of estimates also changes although the traffic intensity does not change. Besides, we have observed that the minimum batch size value gives us the most confidential estimate. Consequently, we have decided to batch size affects the statistical precision of estimates.

To examine how batch size affects the performance measures, for different batch sizes, we have obtained graphs of the batch arrival rates versus the performance measures. We have analyzed corresponding figures, and seen that the value for the performance measures increases as the batch arrival rate increases. Furthermore, we have also realized that the increment rates of the performance measures occur higher as the batch size increases. Similarly, it can be derived from the graph that the increment rates for different traffic intensities change. To sum up, the more intense the system is, the more waiting time or the number of waiting customers in queue it requires.

For the batch service queueing systems, first of all, we have determined run length and number of replications. Some pilot runs have been made for different run length and numbers of replication. We have easily seen that the batch size decreases the traffic intensity. At this point, we can say that in batch service system, the traffic intensity decreases while the batch size increases opposite to the batch arrival system. Consequently, we have decided to run the simulation program by taking $MX = 1000$ and $p = 300$.

Throughout the first purpose, we have run the simulation program (see A3.2) 15 times for different batch size values and batch service rates, and obtained the point estimates for the performance measures. We have also obtained some numerical results for the batch size values and batch service rates in simulation (see A2.2). We have compared the simulation results to the numerical results, and realized that the simulation results are fairly close to the numerical results. However, with these findings, anything about the statistical precision of these estimates cannot be claimed. Therefore, we have calculated the confidence interval estimates for the performance measures in the simulation program. In the calculations, we have again used confidence coefficient 0.95. Consequently, we have obtained the most narrow confidence intervals when the traffic intensity has the smallest value.

For the second purpose, we have planned to show whether the statistical precision of estimates is affected by the batch size. We have seen that the minimum batch size value provides the maximum traffic intensity, and realized the maximum batch size value gives us the most confidential estimate opposite to the batch arrival system. Consequently, we have decided to batch size affects the statistical precision of estimates.

Moreover, we analyze how batch size affects the performance measures in graphical way. For different batch sizes, we have obtained graphs of the batch service rates versus the performance measures. We have examined corresponding figures, and seen that the values for the performance measures decrease while the batch service rate increases. Furthermore, we have realized that the decrement rates of the performance measures becomes lower as the batch size increases. Similarly, it can be derived from the graph that the decrement rates for different traffic intensities change.

REFERENCES

- Adan, I., & Resing, J. (2002). Queuing theory. Technical report in Eindhoven University of Technology. Eindhoven, The Netherlands.
- Albrecht, M. C., & Az, P. E. (2010). *Introduction to discrete event simulation*. Retrieved April 16, 2011 from [http://www.albrechts.com/mike/DES/Introduction to DES.pdf](http://www.albrechts.com/mike/DES/Introduction%20to%20DES.pdf).
- Bain, L. J., & Engelhardt, M. (1992). *Introduction to probability and mathematical statistics* (2nd.ed.). Boston: PWS-KENT.
- Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. M. (2001). *Discrete-event system simulation* (3rd ed.). New Jersey: Prentice-Hall.
- Blanc, J. P. C. (2011). *Queuing Models- Analytical and numerical methods*. Lecture note. Retrieved March 10, 2011 from <http://lyrawww.uvt.nl/~blanc/qm-blanc.pdf>.
- Buss, A., & Rowaei, A. A. (2010). *A comparison of the accuracy of discrete event and discrete time*. Proceedings of the 2010 Winter Simulation Conference. B. Johansson, S. Jain, J. Montoya-Torres, J. Hukan, and E. Yücesan, eds., Baltimore, Maryland, December 5 - 8, 2010.
- Eilon, S. (1969). A simple proof of $L=\lambda W$. *Operation Research*, 17, 915-916.
- Gross D., & Harris C. M. (1974). *Fundamentals of queueing theory* (2nd.ed.). New York: John Willey & Sons.
- Hiller, F. S., & Lieberman, G. J. (2001). *Introduction to operations research* (7th ed.). USA: McGraw-Hill.

- İnal, C. (1988). *Olasılıksal süreçlere giriş (Markov zincirleri)*. Ankara: ÖZTEK Matbaacılık.
- Jewell, W. S. (1967). A simple proof of $L=\lambda W$. *Operation research*, 17, 1109-1116.
- Kleinrock, L. (1975). *Queuing systems (vol.1)*. New York: Wiley.
- Kleinrock, L., & Gail, R. (1996). *Queuing systems-Problems and solutions*. New York: Wiley.
- Kolahi, S. S. (2011). Simulation Model, Warm-up Period, and Simulation Length of Cellular Systems. *Second International Conference on Intelligent Systems, Modelling and Simulation*.
- Little, J. D. C. (1961). A proof of the queueing formula $L=\lambda W$. *Operation Research* 9, 383-387.
- Medhi, J. (2003). *Stochastic models in queuing theory* (2nd ed.). USA: Academic Press.
- Morse, P. M. (1958). *Queues, Inventories and Maintenance*, Wiley, New York.
- Neuts, M. F. (1967). A general class of bulk queues with Poisson input. *Annals of Mathematical Statistics*, 38 (3), 759-770.
- Ross, S. (2003). *Introduction to probability models* (8th ed.). USA: Academic Press.
- Ross, S. (2006). *A first course in probability* (7th ed.). USA: Pearson Prentice Hall.
- Taha, H. A. (2003). *Operation research an introduction* (7th ed.). Arkansas: Pearson Prentice Hall.

Tijms, H. C. (2003). *A first course in stochastic models*. England: Wiley.

Veeraraghavan, M. (2004). *Derivation of Little's Law*. Lecture note. Retrieved March 10, 2011 from <http://www.ece.virginia.edu/mv/edu/715/lectures/littles-law/littles-law.pdf>.

Winston, W. L. (1994). *Operation research- Applications and algorithms*. USA: Duxbury Press.

Whitt, W. (1989). Planning queuing simulation. *Manager Science*, 35 (11), 1341-1366.

APPENDIX 1.
STATISTICAL DISTRIBUTIONS, GENERATING FUNCTIONS,
TRANSFORMS, AND STATISTICAL INFERENCE

A1.1 Statistical distributions

A1.1.1 Poisson Distribution

A discrete random variable X is said to have the Poisson distribution with parameter λ . This random variable has probability distribution that

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad x = 0, 1, 2, \dots$$

The Laplace-Stieltjes transform $\tilde{X}(s)$, the mean $E(X)$, the variance $V(X)$, and the moment generating function $M_X(t)$ are given as

$$\tilde{X}(s) = e^{\lambda(e^{-s}-1)}, \quad E(X) = \lambda, \quad V(X) = \lambda, \quad M_X(t) = e^{\lambda(e^t-1)}.$$

A1.1.2 Exponential Distribution

The density of exponential distribution with parameter μ is given by

$$f(x) = \mu e^{-\mu x} \quad x \geq 0$$

The Laplace-Stieltjes transform $\tilde{X}(s)$, the mean $E(X)$, the variance $V(X)$, and the moment generating function $M_X(t)$ are given as

$$\tilde{X}(s) = \frac{\mu}{\mu + s}, \quad E(X) = \frac{1}{\mu}, \quad V(X) = \frac{1}{\mu^2}, \quad M_X(t) = \left(1 - \frac{t}{\mu}\right)^{-1}.$$

An important property of an exponential random variable X with parameter μ is the *memoryless property*. If X the lifetime of a component, then this property insist on that the probability that the component will last more than $x+t$ time given that it has already lasted more than t is the same as that of a new component lasting more than x . Namely, an old component that still works is just as reliable as a new component. This property states that for all $x \geq 0$ and $t \geq 0$,

$$P(X > t + x | X > t) = P(X > x) = e^{-\mu x}.$$

The exponential probability distribution with mean $1/\mu$ is special class of the Erlang probability distribution with mean $1/\mu$, namely, type 1 where $k = 1$.

If X_1, \dots, X_n are exponential random variables having mean μ , then $X_1 + \dots + X_n$ has a gamma distribution with parameters α and μ .

A1.1.3 Erlang Distribution

A random variable X is said to have the Erlang family of probability distributions with parameters k and μ . This random variable has probability distribution that

$$f(x) = \frac{(\mu k)^k}{(k-1)!} x^{k-1} e^{-k\mu x} \quad x > 0.$$

The Laplace-Stieltjes transform $\tilde{X}(s)$, the mean $E(X)$, the variance $V(X)$, and the moment generating function $M_X(t)$ are given as

$$\tilde{X}(s) = \left(\frac{\mu}{\mu + s} \right)^k, \quad E(X) = \frac{1}{\mu}, \quad V(X) = \frac{1}{k\mu^2}, \quad M_X(t) = \left(1 - \frac{t}{\mu} \right)^{-k}.$$

The Erlang probability distribution with mean $1/\mu$ is a special class of the gamma probability distribution where $\alpha = k$ and $\beta = 1/k\mu$.

A1.1.4 Gamma Distribution

A continuous random variable X is said to have the gamma distribution with parameters $\alpha > 0$ and $\beta > 0$. This random variable has probability distribution that

$$f(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta} \quad x > 0$$

The Laplace-Stieltjes transform $\tilde{X}(s)$, the mean $E(X)$, the variance $V(X)$, and the moment generating function $M_X(t)$, are given as

$$\tilde{X}(s) = (1 + \beta s)^{-\alpha}, \quad E(X) = \alpha\beta, \quad V(X) = \alpha\beta^2, \quad M_X(t) = (1 - \beta t)^{-\alpha}$$

The gamma function is denoted by $\Gamma(\alpha)$ for all $\alpha > 0$, is given by

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t} dt.$$

The gamma function satisfies the following properties:

- i) $\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1) \quad \alpha > 1$
- ii) $\Gamma(n) = (n - 1)! \quad n = 1, 2, \dots$

A1.2 Generating functions

A1.2.1 Generating function

$G(z)$ is a function which has a power series expansion as given by

$$G(z) = \sum_{n=0}^{\infty} g_n z^n = g_0 + g_1 z + g_2 z^2 + g_3 z^3 + \dots$$

If the series converges for some range of z , $G(z)$ is called the *generating function* of the sequence $g_0, g_1, g_2, g_3, \dots$. The generating function is related to z -transform. Generating functions are helpful in solving difference equations. We easily solve a corresponding equation by using a generating function. We determine its series expansion, and then "pick off" the coefficients $\{g_n\}$ rather than to solve the original equation.

A1.2.2 Moment generating function

Let X be a random variable. The *moment generating function* (MGF) of X is given by

$$M_X(t) = E(e^{tX}),$$

and if the MGF of X exists, then

$$\left. \begin{aligned} M_X'(0) &= E(X) \\ M_X''(0) &= E(X^2) \\ M_X'''(0) &= E(X^3) \end{aligned} \right\} M_X^{(r)}(0) = E(X^r) \quad r = 1, 2, \dots$$

A1.2.3 Probability generating function- Factorial moment generating function

Let X be a random variable . The r th *factorial moment* of X is given by

$$E[X(X-1)\dots(X-r+1)],$$

and the *factorial moment generating function* (FMGF) of X is given by

$$G_X(t) = E(t^X).$$

If the FMGF of X exists, then

$$G'_X(1) = E(X)$$

$$G''_X(1) = E[X(X-1)] = E(X^2) - E(X)$$

$$G'''_X(1) = E[X(X-1)(X-2)] = E(X^3) - 3E(X^2) + 2E(X)$$

$$G_X^{(r)}(1) = E[X(X-1)\dots(X-r+1)] \quad r = 1, 2, \dots$$

A1.3 Transforms

A1.3.1 Laplace transform

A transform is a mapping of a function from one space to another. It may be very difficult to solve certain equations directly for a particular function of interest, so we solve corresponding equation in terms of a transform of the function, and then invert the transform to obtain the function. In here, we use the *Laplace transform* (LT) for solving differential equations.

If t is a random variable, and $a(t)$ is the probability density function (pdf) of t , then $A^*(s)$ is the Laplace transform of $a(t)$. The integral representation of this transform is given by

$$A^*(s) = \int_0^{\infty} a(t)e^{-st} dt.$$

The inverse of a Laplace transform is the function that we used to create the transform. Namely, the inverse of $A^*(s)$ is $a(t)$. A list of important Laplace transforms for $f(t)$ is given in Table

Table-Some Laplace transform pairs

Function	Transform
1. $f(t) \quad t \geq 0 \quad \Leftrightarrow$	$F^*(s) = \int_0^{\infty} f(t)e^{-st} dt$
2. Ae^{-at}	$\frac{A}{s+a}$
3. te^{-at}	$\frac{1}{(s+a)^2}$
4. $\frac{t^n}{n!} e^{-at}$	$\frac{1}{(s+a)^{n+1}}$

An important use of this transform is its moment-generating property, it means, the moments t^k may be generated from $A^*(s)$ by relationship

$$\left. \frac{d^k A^*(s)}{ds^k} \right|_{s=0} = (-1)^k \overline{t^k},$$

namely,

$$A^*(s) = M_X(-s).$$

In addition, Laplace transforms have properties very similar to MGFs, for example, the convolution property. We have the useful result that the LT of a random variable which is the sum of two other random variables is the product of the respective LTs as given by

$$Z = X + Y \quad \Rightarrow \quad Z^*(s) = X^*(s) Y^*(s).$$

A1.3.2 z-transform

The z-transform of the discrete non-negative random variable, X , is defined as given by

$$P(z) = \sum_{k=0}^{\infty} p_k z^k.$$

The inverse of a z-transform is the sequence of values that we used to create the transform. Namely, the inverse of $P(z)$ is the sequence p_k . A list of important z-transforms for sequences f_n is given in Table

Table-Some z -transform pairs

Sequence	z -Transform
1. $f_n \quad n = 0, 1, 2, \dots \quad \Leftrightarrow$	$F(z) = \sum_{n=0}^{\infty} f_n z^n$
2. $A\alpha^n$	$\frac{A}{1 - \alpha z}$
3. $n\alpha^n$	$\frac{\alpha z}{(1 - \alpha z)^2}$

The z -transform also has the moment-generating property, namely,

$$\left. \frac{dE[z^X]}{dz} \right|_{z=1} = E(X),$$

and,

$$\left. \frac{d^2 E[z^X]}{dz^2} \right|_{z=1} = E(X^2) - E(X).$$

A1.4 Statistical inference

We firstly give some useful definitions as following;

Definition1:

If X_1, X_2, \dots, X_n be a random sample with size n , then the sample mean, \bar{X} , and the sample variance, S^2 , are given by

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n},$$

$$S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}.$$

Definition2:

If X_1, X_2, \dots, X_n is a random variable from an infinite population with the mean μ , and the variance σ^2 , then

$$\mu_{\bar{X}} = E(\bar{X}) = \mu,$$

$$\sigma_{\bar{X}}^2 = V(\bar{X}) = \frac{\sigma^2}{n}.$$

Definition3: (Central Limit Theorem, CLT)

If X_1, X_2, \dots, X_n is a random sample from a distribution with the mean μ , and the variance σ^2 , then the limiting distribution of

$$Z_n = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$

is the standard normal distribution, $Z_n \xrightarrow{d} Z \sim N(0,1)$ as $n \rightarrow \infty$.

A1.4.1 Point estimation

A statistic, $\hat{\theta}$, that is used to estimate the value of θ is called an *estimator* of θ , and an observed value of the $\hat{\theta}$ is called an *estimate* of θ . It is requested that point estimators have some properties, such as unbiasedness and minimum variance. An estimator $\hat{\theta}$ is said to be an *unbiased estimator* of θ if

$$E(\hat{\theta}) = \theta.$$

Otherwise, we say that $\hat{\theta}$ is *biased estimator* of θ , and the bias is given by

$$\tau(\hat{\theta}) = E(\hat{\theta}) - \theta.$$

The *mean squared error* (MSE) of $\hat{\theta}$ is given by

$$MSE(\hat{\theta}) = E\left[(\hat{\theta} - \theta)^2\right] = V(\hat{\theta}) + [\tau(\hat{\theta})]^2.$$

We easily see that if $E(\hat{\theta}) = \theta$, then MSE is called variance. On the other hand, we can say that if the estimator is bias, then we use to a estimator that has minimum MSE.

A1.4.2 Interval estimation

The two-sided confidence intervals (CI) can be shown a general formula as following

$$(\text{Estimate}) \pm (\text{Reliability factor}) (\text{Standard error}).$$

The estimate (point estimate) is the value of a statistic computed from the data of a sample. The reliability factor depends on the amount of confidence that we want and the sampling distribution of the estimator. Standard error is the standard error of the sample statistic. The confidence coefficient specifies the degree of confidence. It indicates the proportion of confidence interval that we would expect to contain the population parameter being estimated.

The two-sided *confidence interval estimate* is given by

$$P\left(\hat{\theta} - z_{\alpha/2}\sigma_{\hat{\theta}} < \theta < \hat{\theta} + z_{\alpha/2}\sigma_{\hat{\theta}}\right) = 1 - \alpha.$$

APPENDIX 2
ANALITICAL SOLUTION FOR QUEUE MODELS

A2.1 Analytical solution for queue model with batch arrival

1. $r = 2, \lambda = 10, \mu = 100$

$$\rho = \frac{2\lambda}{\mu} = \frac{20}{100} = 0.2000$$

$$W = \frac{3}{2} \left(\frac{1}{\mu - 2\lambda} \right) = \frac{3}{2} \left(\frac{1}{100 - 20} \right) = 0.0188, \quad L = 2\lambda.W = 20(0.0188) = 0.3760$$

$$W_q = \frac{4\lambda + \mu}{2\mu(\mu - 2\lambda)} = \frac{40 + 100}{200(100 - 20)} = 0.0088, \quad L_q = 2\lambda.W_q = 20(0.0088) = 0.1760$$

2. $r = 2, \lambda = 15, \mu = 100$

$$\rho = \frac{2\lambda}{\mu} = \frac{30}{100} = 0.3000$$

$$W = \frac{3}{2} \left(\frac{1}{\mu - 2\lambda} \right) = \frac{3}{2} \left(\frac{1}{100 - 30} \right) = 0.0214, \quad L = 2\lambda.W = 30(0.0214) = 0.6420$$

$$W_q = \frac{4\lambda + \mu}{2\mu(\mu - 2\lambda)} = \frac{60 + 100}{200(100 - 30)} = 0.0114, \quad L_q = 2\lambda.W_q = 30(0.0114) = 0.3420$$

3. $r = 2, \lambda = 20, \mu = 100$

$$\rho = \frac{2\lambda}{\mu} = \frac{40}{100} = 0.4000$$

$$W = \frac{3}{2} \left(\frac{1}{\mu - 2\lambda} \right) = \frac{3}{2} \left(\frac{1}{100 - 40} \right) = 0.0250, \quad L = 2\lambda.W = 40(0.0250) = 1.0000$$

$$W_q = \frac{4\lambda + \mu}{2\mu(\mu - 2\lambda)} = \frac{80 + 100}{200(100 - 40)} = 0.0150, \quad L_q = 2\lambda.W_q = 40(0.0150) = 0.6000$$

4. $r = 2, \lambda = 25, \mu = 100$

$$\rho = \frac{2\lambda}{\mu} = \frac{50}{100} = 0.5000$$

$$W = \frac{3}{2} \left(\frac{1}{\mu - 2\lambda} \right) = \frac{3}{2} \left(\frac{1}{100 - 50} \right) = 0.0300, \quad L = 2\lambda.W = 50(0.0300) = 1.5000$$

$$W_q = \frac{4\lambda + \mu}{2\mu(\mu - 2\lambda)} = \frac{100 + 100}{200(100 - 50)} = 0.0200, \quad L_q = 2\lambda.W_q = 50(0.0200) = 1.0000$$

5. $r = 2, \lambda = 30, \mu = 100$

$$\rho = \frac{2\lambda}{\mu} = \frac{60}{100} = 0.6000$$

$$W = \frac{3}{2} \left(\frac{1}{\mu - 2\lambda} \right) = \frac{3}{2} \left(\frac{1}{100 - 60} \right) = 0.0375, \quad L = 2\lambda.W = 60(0.0375) = 2.2500$$

$$W_q = \frac{4\lambda + \mu}{2\mu(\mu - 2\lambda)} = \frac{120 + 100}{200(100 - 60)} = 0.0275, \quad L_q = 2\lambda.W_q = 60(0.0275) = 1.6500$$

6. $r = 3, \lambda = 10, \mu = 100$

$$\rho = \frac{3\lambda}{\mu} = \frac{30}{100} = 0.3000$$

$$W = 2 \left(\frac{1}{\mu - 3\lambda} \right) = 2 \left(\frac{1}{100 - 30} \right) = 0.0286, \quad L = 3\lambda.W = 30(0.0286) = 0.8580$$

$$W_q = \frac{3\lambda + \mu}{\mu(\mu - 3\lambda)} = \frac{30 + 100}{100(100 - 30)} = 0.0186, \quad L_q = 3\lambda.W_q = 30(0.0186) = 0.5580$$

7. $r = 3, \lambda = 15, \mu = 100$

$$\rho = \frac{3\lambda}{\mu} = \frac{45}{100} = 0.4500$$

$$W = 2 \left(\frac{1}{\mu - 3\lambda} \right) = 2 \left(\frac{1}{100 - 45} \right) = 0.0364, \quad L = 3\lambda.W = 45(0.0364) = 1.6380$$

$$W_q = \frac{3\lambda + \mu}{\mu(\mu - 3\lambda)} = \frac{45 + 100}{100(100 - 45)} = 0.0264, \quad L_q = 3\lambda.W_q = 45(0.0264) = 1.1880$$

8. $r = 3, \lambda = 10, \mu = 100$

$$\rho = \frac{3\lambda}{\mu} = \frac{60}{100} = 0.6000$$

$$W = 2 \left(\frac{1}{\mu - 3\lambda} \right) = 2 \left(\frac{1}{100 - 60} \right) = 0.0500, \quad L = 3\lambda.W = 60(0.0500) = 3.0000$$

$$W_q = \frac{3\lambda + \mu}{\mu(\mu - 3\lambda)} = \frac{60 + 100}{100(100 - 60)} = 0.0400, \quad L_q = 3\lambda.W_q = 60(0.0400) = 2.4000$$

9. $r = 3, \lambda = 25, \mu = 100$

$$\rho = \frac{3\lambda}{\mu} = \frac{75}{100} = 0.7500$$

$$W = 2\left(\frac{1}{\mu - 3\lambda}\right) = 2\left(\frac{1}{100 - 75}\right) = 0.0800, \quad L = 3\lambda.W = 75(0.0800) = 6.0000$$

$$W_q = \frac{3\lambda + \mu}{\mu(\mu - 3\lambda)} = \frac{75 + 100}{100(100 - 75)} = 0.0700, \quad L_q = 3\lambda.W_q = 75(0.0700) = 5.2500$$

10. $r = 3, \lambda = 30, \mu = 100$

$$\rho = \frac{3\lambda}{\mu} = \frac{90}{100} = 0.9000$$

$$W = 2\left(\frac{1}{\mu - 3\lambda}\right) = 2\left(\frac{1}{100 - 90}\right) = 0.2000, \quad L = 3\lambda.W = 90(0.2000) = 18.000$$

$$W_q = \frac{3\lambda + \mu}{\mu(\mu - 3\lambda)} = \frac{90 + 100}{100(100 - 90)} = 0.1900, \quad L_q = 3\lambda.W_q = 90(0.1900) = 17.100$$

11. $r = 4, \lambda = 5, \mu = 100$

$$\rho = \frac{4\lambda}{\mu} = \frac{20}{100} = 0.2000$$

$$W = \frac{5}{2}\left(\frac{1}{\mu - 4\lambda}\right) = \frac{5}{2}\left(\frac{1}{100 - 20}\right) = 0.0313, \quad L = 4\lambda.W = 20(0.0313) = 0.6260$$

$$W_q = \frac{8\lambda + 3\mu}{2\mu(\mu - 4\lambda)} = \frac{40 + 300}{200(100 - 20)} = 0.0213, \quad L_q = 4\lambda.W_q = 20(0.0213) = 0.4260$$

12. $r = 4, \lambda = 10, \mu = 100$

$$\rho = \frac{4\lambda}{\mu} = \frac{40}{100} = 0.4000$$

$$W = \frac{5}{2}\left(\frac{1}{\mu - 4\lambda}\right) = \frac{5}{2}\left(\frac{1}{100 - 40}\right) = 0.0417, \quad L = 4\lambda.W = 20(0.0417) = 1.6680$$

$$W_q = \frac{8\lambda + 3\mu}{2\mu(\mu - 4\lambda)} = \frac{80 + 300}{200(100 - 40)} = 0.0317, \quad L_q = 4\lambda.W_q = 40(0.0317) = 1.2680$$

13. $r = 4, \lambda = 15, \mu = 100$

$$\rho = \frac{4\lambda}{\mu} = \frac{60}{100} = 0.6000$$

$$W = \frac{5}{2} \left(\frac{1}{\mu - 4\lambda} \right) = \frac{5}{2} \left(\frac{1}{100 - 60} \right) = 0.0625, \quad L = 4\lambda.W = 60(0.0625) = 3.7500$$

$$W_q = \frac{8\lambda + 3\mu}{2\mu(\mu - 4\lambda)} = \frac{120 + 300}{200(100 - 60)} = 0.0525, \quad L_q = 4\lambda.W_q = 60(0.0525) = 3.1500$$

14. $r = 4, \lambda = 20, \mu = 100$

$$\rho = \frac{4\lambda}{\mu} = \frac{80}{100} = 0.8000$$

$$W = \frac{5}{2} \left(\frac{1}{\mu - 4\lambda} \right) = \frac{5}{2} \left(\frac{1}{100 - 80} \right) = 0.1250, \quad L = 4\lambda.W = 80(0.1250) = 10.000$$

$$W_q = \frac{8\lambda + 3\mu}{2\mu(\mu - 4\lambda)} = \frac{160 + 300}{200(100 - 80)} = 0.1150, \quad L_q = 4\lambda.W_q = 80(0.1150) = 9.2000$$

A2.2 Analytical solution for queue model with batch service

$$b = 2 \Rightarrow \mu r^3 - (\lambda + \mu)r + \lambda = 0$$

1. $\lambda = 20, \mu = 12$

$$12r^3 - 32r + 20 = 0 \Rightarrow r_0 = 0.8844$$

$$L = \frac{r_0}{1 - r_0} = \frac{0.8844}{0.1156} = 7.6505$$

$$L_q = L - \frac{\lambda}{\mu} = 7.6505 - \frac{20}{12} = 5.9838$$

$$W_q = \frac{L_q}{\lambda} = \frac{5.9838}{20} = 0.2992$$

2. $\lambda = 20, \mu = 14$

$$14r^3 - 34r + 20 = 0 \Rightarrow r_0 = 0.7955$$

$$L = \frac{r_0}{1-r_0} = \frac{0.7955}{0.2045} = 3.8899$$

$$L_q = L - \frac{\lambda}{\mu} = 3.8899 - \frac{20}{14} = 2.4614$$

$$W_q = \frac{L_q}{\lambda} = \frac{2.4614}{20} = 0.1231$$

3. $\lambda = 20, \mu = 16$

$$16r^3 - 36r + 20 = 0 \Rightarrow r_0 = 0.7247$$

$$L = \frac{r_0}{1-r_0} = \frac{0.7247}{0.2753} = 2.6324$$

$$L_q = L - \frac{\lambda}{\mu} = 2.6324 - \frac{20}{16} = 1.3824$$

$$W_q = \frac{L_q}{\lambda} = \frac{1.3824}{20} = 0.0691$$

4. $\lambda = 20, \mu = 18$

$$18r^3 - 38r + 20 = 0 \Rightarrow r_0 = 0.6667$$

$$L = \frac{r_0}{1-r_0} = \frac{0.6667}{0.3333} = 2.0003$$

$$L_q = L - \frac{\lambda}{\mu} = 2.0003 - \frac{20}{18} = 0.8892$$

$$W_q = \frac{L_q}{\lambda} = \frac{0.8892}{20} = 0.0444$$

$$b=3 \Rightarrow \mu r^4 - (\lambda + \mu)r + \lambda = 0$$

1. $\lambda = 20, \mu = 12$

$$12r^4 - 32r + 20 = 0 \Rightarrow r_0 = 0.7336$$

$$L = \frac{r_0}{1-r_0} = \frac{0.7336}{0.2664} = 2.7537$$

$$L_q = L - \frac{\lambda}{\mu} = 2.7537 - \frac{20}{12} = 1.0870$$

$$W_q = \frac{L_q}{\lambda} = \frac{1.0870}{20} = 0.0543$$

2. $\lambda = 20, \mu = 14$

$$14r^4 - 34r + 20 = 0 \Rightarrow r_0 = 0.6724$$

$$L = \frac{r_0}{1 - r_0} = \frac{0.6724}{0.3276} = 2.0525$$

$$L_q = L - \frac{\lambda}{\mu} = 2.0525 - \frac{20}{14} = 0.6239$$

$$W_q = \frac{L_q}{\lambda} = \frac{0.6239}{20} = 0.0311$$

3. $\lambda = 20, \mu = 16$

$$16r^4 - 36r + 20 = 0 \Rightarrow r_0 = 0.6221$$

$$L = \frac{r_0}{1 - r_0} = \frac{0.6221}{0.3779} = 1.6462$$

$$L_q = L - \frac{\lambda}{\mu} = 1.6462 - \frac{20}{16} = 0.3962$$

$$W_q = \frac{L_q}{\lambda} = \frac{0.3962}{20} = 0.0198$$

4. $\lambda = 20, \mu = 18$

$$18r^4 - 38r + 20 = 0 \Rightarrow r_0 = 0.5799$$

$$L = \frac{r_0}{1 - r_0} = \frac{0.5799}{0.4201} = 1.3804$$

$$L_q = L - \frac{\lambda}{\mu} = 1.3804 - \frac{20}{18} = 0.2693$$

$$W_q = \frac{L_q}{\lambda} = \frac{0.2693}{20} = 0.0135$$

$$b = 4 \Rightarrow \mu r^5 - (\lambda + \mu)r + \lambda = 0$$

1 $\lambda = 20, \mu = 12$

$$12r^5 - 32r + 20 = 0 \Rightarrow r_0 = 0.6792$$

$$L = \frac{r_0}{1-r_0} = \frac{0.6792}{0.3208} = 2.1172$$

$$L_q = L - \frac{\lambda}{\mu} = 2.1172 - \frac{20}{12} = 0.4505$$

$$W_q = \frac{L_q}{\lambda} = \frac{0.4505}{20} = 0.0225$$

2. $\lambda = 20, \mu = 14$

$$14r^5 - 34r + 20 = 0 \Rightarrow r_0 = 0.6287$$

$$L = \frac{r_0}{1-r_0} = \frac{0.6287}{0.3713} = 1.6932$$

$$L_q = L - \frac{\lambda}{\mu} = 1.6932 - \frac{20}{14} = 0.2646$$

$$W_q = \frac{L_q}{\lambda} = \frac{0.2646}{20} = 0.0132$$

3. $\lambda = 20, \mu = 16$

$$16r^5 - 36r + 20 = 0 \Rightarrow r_0 = 0.5864$$

$$L = \frac{r_0}{1-r_0} = \frac{0.5864}{0.4136} = 1.4178$$

$$L_q = L - \frac{\lambda}{\mu} = 1.4178 - \frac{20}{16} = 0.1678$$

$$W_q = \frac{L_q}{\lambda} = \frac{0.1678}{20} = 0.0084$$

4. $\lambda = 20, \mu = 18$

$$18r^5 - 38r + 20 = 0 \Rightarrow r_0 = 0.5502$$

$$L = \frac{r_0}{1-r_0} = \frac{0.5502}{0.4498} = 1.2232$$

$$L_q = L - \frac{\lambda}{\mu} = 1.2232 - \frac{20}{18} = 0.1121$$

$$W_q = \frac{L_q}{\lambda} = \frac{0.1121}{20} = 0.0056$$


```

        if (DT == 5000)
            area_S = 0;
        end
    else
        area_S = (DT - TMI) * (WL + 1);
    end
    sum_S = sum_S + area_S;           % sum of system times of all arrivals
end
sum_WT = (sum_S - sum_ST);         % sum of waiting times of all arrivals
total_Lq = total_Lq + (sum_WT / TMI);
total_Wq = total_Wq + (sum_WT / n);
total_roo = total_roo + (sum_ST / TMI);
total_L = total_L + (sum_S / TMI);
y(1,i) = (sum_WT / TMI);
y1(1,i) = (sum_S / TMI);
y2(1,i) = (sum_WT / n);
end
% Calculate Result
Lq = total_Lq / p;                 % average queue length
Wq = total_Wq / p;                 % average waiting time in queue
roo = total_roo / p;               % service facility
L = total_L / p;                   % average number in system
for j = 1 : p
    SLq = SLq + ((y(1,j) - Lq) ^ 2);
    SL = SL + ((y1(1,j) - L) ^ 2);
    SWq = SWq + ((y2(1,j) - Wq)^2);
end
SLqI = sqrt((SLq / (p-1)) * (1/p)); % standard error for Lq
SLI = sqrt((SL / (p-1)) * (1/p));  % standard error for L
SWqI = sqrt((SWq / (p-1)) * (1/p)); % standard error for Wq
cdown = Lq - (SLqI * 1.96);        % confidence interval lower limit for Lq
cup = Lq + (SLqI * 1.96);         % confidence interval upper limit for Lq
cdown1 = Wq - (SWqI * 1.96);      % confidence interval lower limit for Wq
cup1 = Wq + (SWqI * 1.96);        % confidence interval upper limit for Wq
cdown2 = L - (SLI * 1.96);        % confidence interval lower limit for L
cup2 = L + (SLI * 1.96);          % confidence interval upper limit for L
% Print the results
fprintf('....L.....=%4.4f\n', L);
fprintf('....Lq.....=%4.4f\n', Lq);
fprintf('....Wq.....=%4.4f\n', Wq);
fprintf('....roo.....=%4.4f\n', roo);
fprintf('.... Standard error for Lq.....=%4.6f\n', SLqI);
fprintf('.... Standard error for L.....=%4.6f\n', SLI);
fprintf('.... Standard error for Wq.....=%4.6f\n', SWqI);
fprintf('.... Confidence Interval for Lq.....=%4.6f %4.6f\n', cdown, cup);
fprintf('.... Confidence Interval for Wq.....=%4.6f %4.6f\n', cdown1, cup1);
fprintf('.... Confidence Interval for L.....=%4.6f %4.6f\n', cdown2, cup2);

```

A3.2 Simulation code for the queue model with batch service

```

L = 0; Lq = 0; Wq = 0; roo = 0; SLq = 0; SWq = 0; SL = 0; p = 1000; total_L = 0;
total_roo = 0; total_Lq = 0; total_Wq = 0;
for i = 1 : p
AT = 0; SS = 0; SC = 0; WL = 0; TMI = 0; DT = 5000; MX = 300; sum_ST = 0;
sum_WT = 0; n = 0; sum_S = 0;
    while (TMI <= MX)
        if (AT < DT)                                % Arrival Process
            event = 'A';
            b = 3;
            TMI = AT;
            if (SS == 0)
                SS = 1;
                SC = 1;
                if (WL >= b)
                    SC = b;
                    WL = WL - SC;
                elseif (WL >= b & WL > 0)
                    SC = WL;
                    WL = WL - SC;
                end
                ST = exprnd(1 / 12);
                sum_ST = sum_ST + ST;                % sum of service time
                DT = TMI + ST;
            else
                WL = WL + 1;
            end
            IA = exprnd(1 / 20);
            AT = TMI + IA;
            if (SC < b)
                while (WL > 0)
                    SC = SC + 1;
                    WL = WL - 1;
                    if (SC == b)
                        break;
                    end
                end
            end
        else                                % Departure Process
            event = 'D';
            TMI = DT;
            if (WL == 0)
                SS = 0;
                SC = 0;
                DT = 5000;
            else
                if (WL >= b)

```

```

                SC = b;
            else
                SC = WL;
            end
            WL = WL - SC;
            ST = exprnd(1 / 12);
            sum_ST = sum_ST + ST;           % sum of service time
            DT = TMI + ST;
        end
    end
    fprintf(' %4.4f %6.0f %8.0f %8.0f %10.4f %10.4f
    .....%s\n',TMI,SS,WL,SC,AT,DT,event);
    % Calculate Area
    if (AT < DT)
        area_W = (AT - TMI) * (WL);
        area_S = (AT - TMI) * (WL + SC);
        n = n + 1;                       % number of arrivals in system
        if (DT == 5000)
            area_W = 0;
            area_S = 0;
        end
    else
        area_W = (DT - TMI) * (WL);
        area_S = (DT - TMI) * (WL + SC);
    end
    sum_WT = sum_WT + area_W;           % sum of waiting times of all arrivals
    sum_S = sum_S + area_S;           % sum of system times of all arrivals
    end
    total_L = total_L + (sum_S / TMI);
    total_Lq = total_Lq + (sum_WT / TMI);
    total_Wq = total_Wq + (sum_WT / n);
    total_roo = total_roo + (sum_ST / TMI);
    y(1,i) = (sum_S / TMI);
    y1(1,i) = (sum_WT / TMI);
    y2(1,i) = (sum_WT / n);
    end
    L = total_L / p;                   % average number in system
    Lq = total_Lq / p;                 % average queue length
    Wq = total_Wq / p;                 % average waiting time in queue
    roo = total_roo / p;               % service facility
    for j = 1 : p
        SLq = SLq + ((y1(1,j) - Lq)^2);
        SL = SL + ((y(1,j) - L)^2);
        SWq = SWq + ((y2(1,j) - Wq)^2);
    end
    SLqI = sqrt((SLq / (p-1)) * (1/p)); % standard error for Lq
    SLI = sqrt((SL / (p-1)) * (1/p)); % standard error for L
    SWqI = sqrt((SWq / (p-1)) * (1/p)); % standard error for Wq

```



```

 $c_{down} = L_q - (SL_{q1} * 1.96);$            % confidence interval lower limit for  $L_q$ 
 $c_{up} = L_q + (SL_{q1} * 1.96);$            % confidence interval upper limit for  $L_q$ 
 $c_{down1} = W_q - (SW_{q1} * 1.96);$        % confidence interval lower limit for  $W_q$ 
 $c_{up1} = W_q + (SW_{q1} * 1.96);$        % confidence interval upper limit for  $W_q$ 
 $c_{down2} = L - (SL1 * 1.96);$          % confidence interval lower limit for  $L$ 
 $c_{up2} = L + (SL1 * 1.96);$          % confidence interval upper limit for  $L$ 
% Print the results
fprintf('....L.....=%4.4f\n', L);
fprintf('....Lq.....=%4.4f\n', Lq);
fprintf('....Wq.....=%4.4f\n', Wq);
fprintf('....roo.....=%4.4f\n', roo);
fprintf('....SLq.....=%4.6f\n', SLq1);
fprintf('....SL.....=%4.6f\n', SL1);
fprintf('....SWq.....=%4.6f\n', SWq1);
fprintf('.... Confidence Interval for Lq.....=%4.6f %4.6f\n', cdown, cup);
fprintf('.... Confidence Interval for Wq.....=%4.6f %4.6f\n', cdown1, cup1);
fprintf('.... Confidence Interval for L.....=%4.6f %4.6f\n', cdown2, cup2);

```