LOCAL BALANCE MODELS OF
COMPUTER SYSTEMS

by

Donald Fred Towsley

December 1975 TR-60

This report constituted the author's Doctoral Dissertation in Computer Sciences at The University of Texas at Austin and was supported in part by National Science Foundation Grants GJ-1084 and DCR74-13302.

DEPARTMENT OF COMPUTER SCIENCES
THE UNIVERSITY OF TEXAS AT AUSTIN

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

The use of queueing networks as a basis for models of computer systems has led to the development of powerful tools for system analysis. Crucial to the utility of these tools is the development of rapid and simple solution techniques for queueing network models which can sufficiently represent critical factors in system performance. This dissertation extends the range of queueing network models which can be readily and accurately solved to include two factors which are becoming increasingly significant. These factors are <u>state dependent routing in a network</u> and the <u>simultaneous use of several devices or resources by a job</u>.

State dependent routing manifests itself in various areas of computing system design. An important problem in the design of ARPA type computer-communication networks is the selection of a routing strategy which decreases the average message delay between sites within the network. Within computer systems, as the number of storage devices (disk, drum, tape mounts) increase, various forms of load levelling techniques are being implemented to decrease response times. For the case of identical devices, the routing strategy is straightforward; route the job to the device with the lightest load. For the case of non-identical devices, the selection of a load levelling routing strategy can be more complicated. All these problems require simple and efficient tools for qualitatively selecting an optimal or near-optimal routing strategy and for producing accurate performance measure values. We develop analytic techniques which provide accurate and efficient solutions to a subset of these problems, namely those concerned with load levelling.

The manifestation of the simultaneous use of several devices or resources is more widespread. The ability to overlap central processor (CPU) activity and input/output (I/O) activity is common in many systems. Some systems allow the overlap of several I/O activities by a job. This capability of device sharing will become more important as multiprocessor systems (parallel-CPU's, CPU array processor combinations) become more widespread. The design of these systems requires efficient and accurate device sharing models, These are necessary to study the tradeoffs between increased performance, increased software complexity and increased buffer storage requirements. We develop numerical techniques which provide efficient and accurate solutions to a wide range of device sharing structures.

The basis for efficient solution capabilities of both factors is the efficient solution techniques which exist for queueing networks satisfying local balance [C1] or station balance [C4] conditions. We develop a thorough and extended characterization of networks satisfying these properties and determine functional forms of state dependent branching probabilities which will preserve these properties. These branching probabilities may depend on the queue lengths of the servers which comprise what we define to be parallel subnetworks. The extension of efficient solution capabilities to include device sharing is developed through an extended version of Norton's Theorem

[C2]. Exact numerical solutions are obtained for networks consisting of two queues in series with first come first served (FCFS) disciplines and exponential service time distributions [D1] which exhibit CPU activity and I/O activity (CPU-I/O) overlap. Accurate numerical approximations are obtained for central server networks (see Chapter II) with FCFS disciplines and exponential service times which exhibit CPU-I/O processing overlap or I/O-I/O processing overlap (two independent I/O activities by a single job).

Chapter II contains background material (including a survey of the literature) useful for the discussions encountered in the remainder of the dissertation. Chapter III presents a detailed theory of the network components, queues and servers, necessary for the development of the remainder of the dissertation. This discussion is taken mainly from Chandy, Howard, and Towsley [C4]. Chapter IV presents closed form solutions for networks with particular state dependent routing strategies. Chapter V presents techniques for the numerical analysis of CPU-I/O and I/O-I/O overlap processing models. Examples are presented throughout Chapters IV and V.

CHAPTER II

GENERAL CONCEPTS AND PREVIOUS WORK

The behavior of a computer system depends on the interaction between two elements, the workload and the system components. The workload is characterized by the jobs and their needs. The term system components covers hardware components such as memory, central processors, secondary storage (I/O) devices, channels, controllers, and the operating system, in particular the schedulers.

In this chapter we discuss how queueing models capture the essential characteristics of the workload and system components interactions. We discuss previous research in the area of queueing models of computer systems. Finally, we discuss briefly the problems in parameterizing queueing models.

2.1 Queueing Networks as Models of Computer Systems

The workload of a computer system is characterized by the arrival pattern of jobs into the system and their pattern of needs. Generally a job consists of a computation phase and an I/O phase. The computation phase consists of processing by the CPU of user data possibly found in input buffers and the placing of results into output buffers. When output buffers become full or input buffers become empty, the job may request transfer of data to or from secondary storage. The use of multiple buffering allows the possibility of overlapping computation and I/O activities.

In a multiprogrammed environment the situation arises where one job requires the same processor or device as another job, either the CPU or an I/O device. This situation results in the creation of a queue for that device. The presence of these queues suggests the use of a queueing network model where the hardware components, CPU's and I/O devices are represented by servers with associated queues.

The most commonly used queueing network model is the central server model. A fixed number of jobs (programs, customers) traverse a closed network consisting of a CPU and several I/O devices. Each job alternately receives service from the CPU and one of the I/O devices. When a server is busy, a job may have to wait in a queue. After completing service at the CPU, if there is no overlap, the job selects an I/O device according to probabilities associated with that device and the job. If CPU-I/O overlap is present, this selection of an I/O device may occur during the CPU phase. The service time of a job on a device may depend on the device, customer, and the queue lengths for that device. A central server network with three I/O devices appears in Figure 2.1.

The number of I/O devices and the scheduling disciplines associated with each queue depends primarily on the components of the system. The branching probabilities associated with each device depends on the job composition of the workload. The service time distributions depend both on the workload and system components. We assume that the distributions depend both on the workload and that substitution of a different processor changes the distribution by a multiplicative constant. Finally, the number of jobs within the network depends both on the memory needs of the jobs making up the workload, and on the size of main memory.

2.2 Previous Work

We review the literature of two areas: overlap processing models and queueing models of computing systems. There is a scarcity of literature in the first area. Hellerman and Smith [H1] considered the difference in the total time necessary to complete a single program for the cases of no overlap, of compute-input (C-I) overlapped processing, and of compute-input-output (C-I-O) overlapped processing. In all cases, all processing times were constant. Cotten and Abd-Alla [C8] also considered a uni-programmed environment to study the effects of overlapping output processing with compute processing. They assumed the processing times to be exponentially distributed. Both studies [H1, C8] are useless for studying the effects of overlapped processing in a multiprogrammed environment except for providing upper bounds in the improvement over a non-overlapped system.

In contrast the area of queueing models has produced a rich literature. These models usually consider computing systems as central server networks of queues and servers representing central processing units (CPU's) and Input/Output (I/O) devices. Baskett [B1] studied the effects of CPU round robin fixed quantum scheduling while varying the quantum size and the service time distributions with such a model. Buzen [B5] called these models central server models and used them with FCFS disciplines and exponential distributions in the analysis of

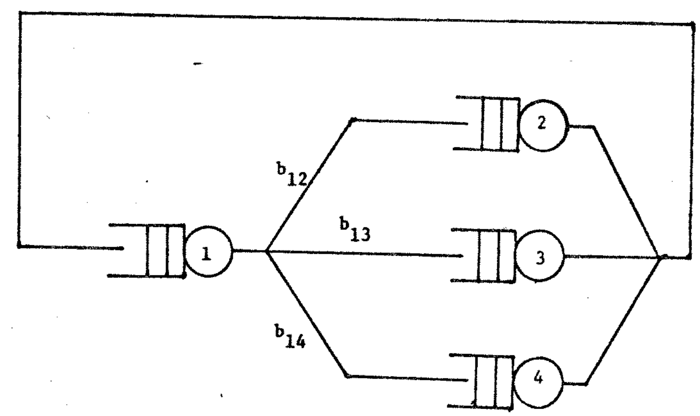


Figure 2.1

system bottlenecks. They have been imbedded into more general models which consider the effects of limited memory (Brown [B3]), and also the effects of the placement of files within memory hierarchies (Foster [F1]). Browne et al. [B4] have used approximation techniques in extending these models to evaluate and improve the performance of a complex computing system. Chen [C5] has done the same for an interactive computing system.

The complexity of current queueing models is a result of an increasingly rich set of solution techniques for queueing networks. One of the first results for queueing networks was the solution by Jackson [J1] of open networks consisting of queues with exponential service time distributions. A network is open if customers may enter and depart. Jackson showed that the solutions have a "product form". By a product form, we mean that the Markovian state probabilities are a product of terms for each queue in the network. Gordon and Newell [G3] showed that product form solutions exist for closed networks with exponential servers. Buzen [B5] developed efficient computational techniques for obtaining performance measures such as CPU or I/O utilizations for closed networks with product form solutions. Chandy [C1] introduced the concept of local balance and used it to obtain product form solutions for a large class of networks. This class includes some networks with non-exponential service time distributions and several queueing disciplines other than first-come-first-serve (FCFS). Baskett, Chandy, Muntz, and Palacios [B2] extended this analysis to networks with a finite number of customer classes.

Chandy, Herzog, and Woo [C2] formulated a method to simplify the parametric analysis of queueing networks satisfying local balance. This method is directly analogous to Norton's Theorem from electrical circuit theory. For a given queue, the method allows the replacement of the remainder of the network by a "composite queue" without changing the behavior of the queue. Hence, performance measures for a queue may be obtained from a simpler two queue network. Reiser and Kobayashi [R1] developed computationally efficient procedures for local balance networks with several classes of customers where some might be open and others closed. Chandy, Howard and Towsley [C4] introduced the concept of station balance, a sufficient condition for local balance. They showed that, for queues exhibiting station balance, performance measures such as utilization are independent of the form of the service time distribution, so long as the mean service time remains unchanged.

Closed form solutions have been difficult to obtain for queueing models not satisfying local balance. There are several other techniques which produce exact numerical solutions. Semi-Markov techniques have been used by Gaver [G1], Baskett [B1], Shedler [S2], and Courtois and Georges [C9]. Iterative Markovian numerical techniques have been used by Wallace and Rosenberg [W1]. Finally, recursive Markovian state solution techniques have been developed by Herzog et al. [H2]. This last solution technique is one of the basic elements of the models we develop.

Approximation methods are becoming increasingly widespread as efficient techniques for accurate solution of quite complex queueing models. Gaver and Shedler [G2], and Kobayashi [K2, K3] have applied diffusion approximations to queueing networks. Muntz and Wong [M4] have determined asymptotic values for mean response times of a job within a queueing network. These values can be used to approximate mean response times for low and high levels of multiprogramming within the network. Techniques based on local balance techniques and numerical techniques have been used by Chandy, Herzog, and Woo [C3], Sauer [S1], Brown [B3], Keller [K1], Browne et al. [B4], and Chen [C5]. Our work is based on techniques similar to these last ones.

The modeling of routing strategies within computing systems has received little attention. Fultz [F2] used simulation techniques to model message routing strategies within the ARPA net. Chen [C5] used an iterative approximation method based on local balance techniques to model an interactive computing system with a particular state dependent routing scheme.

2.3 Model Parameterization

The accuracy of the solutions our queueing models provide depends on two factors. First, it depends on the accuracy of the model itself. Second, the accuracy depends on the accuracy of the parameter values given to the models. The determination of reasonable parameter values is a non-trivial problem and so we present a short discussion of it.

Our models consist of central server networks. These consist of queues representing the CPU and various I/O devices. The I/O devices may be combinations of channels and controllers, whatever can

be used to represent the data transfer. Associated with each queue is a queueing discipline. These can usually be determined from hardware and operating system specifications. Additionally, each queue has a service time distribution, and each I/O queue a probability of access. Finally, since our networks are closed, we need to specify the level of multiprogramming, i.e., the number of programs in memory.

We are not concerned with a program's memory needs in our models. It is possible to use other techniques [B3] to include the effects of finite memory. This has been done to model a CDC CYBER-70 computer system with a SCOPE 3.4 operating system [I1]. Without using the above techniques, two cruder methods may be used. First, the average number of jobs expected in the system may be used. Determination of this number consists of measurement from the actual system or an analysis of typical program memory requirements. The second method is a parametric analysis over the range of levels of multiprogramming.

For each I/O queue, our models assume exponential service time distributions. Only the mean service times need to be determined. For an existing system, this can be determined by properly designed and implemented hardware and or software probes. The proper design and implementation of these probes may be difficult. For a proposed system, accurate determination can be even more difficult. Some I/O subsystems may be quite complex. For example, some disk subsystems allow the overlap of seeks and data transfer. These may require preliminary analysis to determine the parameter values for the central server network [B4]. The probabilities for each queue can be determined by measuring the frequency of accesses of files within the device represented by the queue. Again, for a future system this can be quite difficult.

Our models require accurate knowledge of the overlap behavior of the typical job. Simple determination can be made if the measurement probe and analyzer possess the capability to note each occurrence of an I/O access overlapped with computing within one program. Less accurate and more time consuming is analysis of the structure of typical programs. Another method would be to run the computer system in a uniprogramming mode for typical programs and take measurements. Under these conditions, with good probes, an accurate estimate can still be made. The difference between the measurement period length and the sum of the total CPU and I/O times is the total time that CPU and I/O activities are overlapped. This overlap time can be used to obtain an approximate typical program structure. Finally, CPU burst times can also be determined from good probes.

When measurement facilities are not adequate, some parameters may be obtained from partial information. For example, for a given I/O device, if the utilization, the length of the measurement period, and the number of accesses are known, then the mean service time is the utilization multiplied by the measurement period and divided by the number of accesses.

CHAPTER III

SINGLE QUEUE SYSTEMS

3.1 Introduction

Crucial to the efficiency of our solution techniques is the assumption at some step that the model satisifes local balance. Several studies [B2, C1, C2] show that local balance is closely related to product form solutions. Additionally, other works [B5, C2] provide efficient computational techniques for solutions with product form solutions. In Chapters III and IV we characterize the relationships between local balance and product form solutions.

To develop a useful theory of queueing networks satisfying the properties of local balance and station balance, it is desirable to study their components, single queue (SQ) systems. A SQ system consists of a queue and server and the associated queueing discipline. It is the aim of this chapter to present properties and solutions for the subset of SQ systems satisfying local balance and/or station balance. Most of the notation and results presented in this chapter are taken from [C4].

In studying a SQ system, the behavior is characterized by the load on the system and its associated queueing discipline. The load consists of the customers in the system with their service requirements. We allow customers of several classes, each class with its corresponding arrival process and service time needs. The load on the system is formalized in section 3.2. The queueing discipline imposes rules on the order of service and amount of service for customers in the queue. These rules can be represented by two families of parameters. These are presented in section 3.3. Once the load and queueing discipline are specified a system of differential equations describing the behavior of the system can be derived. These equations are presented in section 3.4. Finally we limit our study to SQ systems exhibiting the properties of local or station balance. Equilibrium solutions for systems of practical and theoretical interest are presented in section 3.5. Table 3.1 can be used for quick reference to the notation presented in this chapter.

3.2 Queue Load

3.2.1 Customer Classes

We allow K customer classes labelled $k = 1,\ldots,K$. We assume that the load of the queue can be specified by the inter-arrival distribution function H_k and service time density function F_k. However, this work concerns itself with single queue (SQ) systems for which the arrival process of customer class k is Poisson with a mean interarrival time of $1/\lambda_k$. The reader is invited to glance at Appendix A for the generalization of the following discussion to arbitrary differentiable interarrival time distributions.

Let Y_k be a positive real random variable denoting the initial service time for a customer of class k. We assume Y_k is independent of the state of the system upon arrival and has the probability distribution function

$$F_k(t) = \Pr(0<Y_k<t)$$

Table 3.1 Notation in Chapter III

Formal symbolism	Meaning	Section reference
K	number of customer classes	3.2.1
k	generic index for customer class	3.2.1
F_k	service time distribution for class k	3.2.1
f_k	service time density function for class k	3.2.1
$1/\mu_k$	mean service time for class k	3.2.1
$1/\lambda_k$	mean interarrival time for class k	3.2.1
n	number of customers in the queue	3.2.2
i	generic index for queue station	3.2.2
$k(i)$	class of customer in station i	3.2.2
S	$(k(1),\ldots,k(n))$, occupancy of queue	3.2.2
X_i	remaining service time of customer in station i	3.2.2
X	$(S,X_1,\ldots,X_n)$, complete state description	3.2.2
$S-i$	$(k(1),\ldots,k(i-1),k(i+1),\ldots,k(n))$	3.2.2
$S+(i,k)$	$(k(1),\ldots,k(i-1),k,k(i+1),\ldots,k(n))$	3.2.2
$X-i$	$(S-i,X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_n)$	3.2.2
$X+(i,k,Y)$	$(S+(i,k),X_1,\ldots,X_{i-1},Y,X_{i+1},\ldots,X_n)$	3.2.2
n_k	number of class k customers in queue	3.2.2
N	$(n_1,\ldots,n_K)$	3.2.2
$N-k$	$(n_1,\ldots,n_k-1,\ldots,n_K)$	3.2.2
$N+k$	$(n_1,\ldots,n_k+1,\ldots,n_K)$	3.2.2
$\delta_{\ell,m}$	value 1 if $\ell=m$, value 0 if $\ell\neq m$	3.2.2

Table 3.1 Notation in Chapter III

Formal symbolism	Meaning	Section reference
$a(i\|k,S)$	arrival probability of class k customer into station i when the occupancy is S	3.3.1
$r(i\|S)$	service rate of customer in station i when the occupancy is S	3.3.1
$R(S)$	$\sum_{i=1}^{n} r(i\|S)$	3.3.1
$p(t,X)$	state probability density function (spdf) for state X at time t	3.4.1
$p(t,S)$	spdf for occupancy S at time t	3.4.1
$p(t,N)$	spdf for N at time t	3.4.1
$p(X),p(S),p(N)$	equilibrium values for $p(t,X)$, $p(t,S)$, $p(t,N)$	3.4.2
$q(S)$	see section 3.5.2	3.5.2
$R(k,N)$	service rate for all customers of class k when the system is in state N	3.5.4

with derivative (probability density function)

$$f_k(t) = dF_k(t)/dt$$

and mean

$$1/\mu_k = E[Y_k] = \int_0^\infty t\, f_k(t)\, dt \ .$$

The only constraint on F_k is that it be twice differentiable and have a finite nonzero mean.

3.2.2 States

Before considering the queueing discipline, let us consider the concept of a queue. A queue may be viewed as an ordered sequence of stations $i = 1,2,\ldots$, each occupied by only one customer. If there are n customers in the queue, then the first n stations $i = 1,\ldots,n$ are occupied. Let $k(i)$ be the class of the customer in the i^{th} station. Then $S = (k(1),\ldots,k(n))$ represents the occupancy state of the queue. Note that this description is complete only when F_k is exponential. When F_k is non-exponential it becomes necessary to specify the remaining service time of each customer. For completeness, let $X = (S, X_1,\ldots,X_n)$ be the complete state description where X_i is the remaining service time for the customer in station i. Note that the X_i's are continuous variables. As the customer in station i receives service X_i will decrease.

The behavior of a SQ system can be specified in terms of the occupancy S, the state X and the occupancies and states created either by the arrival or departure of an arbitrary customer from S and X. As a result, the following notation will become necessary.

$$S-i = (k(1),\dots,k(i-1),k(i+1),\dots,k(n))$$
$$S+(i,k) = (k(1),\dots,k(i-1),k,i(i),\dots,k(n))$$
$$X-i = (S-i,X_1,\dots,X_{i-1},X_{i+1},\dots,X_n)$$
$$X+(i,k,Y) = (S+(i,k),X_1,\dots,X_{i-1},Y,X_i,\dots,X_n)$$

S-i and X-i are the occupancy and state resulting from the departure of the customer from station i from the system. S+(i,k) and X+(i,k,Y) are the occupancy and state which result from the arrival of a customer of class k into the i^{th} station. The customer will have an initial service time requirement of Y.

There will be situations when the ordering of the customers will be unimportant. For these cases let $N = (n_1,\dots,n_K)$ represent the state of the system where there are n_k customers of class k. For those situations in which a customer arrives or departs we use the following notation:

$$N-k = (n_1,\dots,n_k-1,\dots,n_K)$$
$$N+k = (n_1,\dots,n_k+1,\dots,n_K)$$

The system is in (state) N if it is in occupancy S and $n_k = \sum_{i=1}^{n} \delta_{k,k(i)}$ for $k=1,\dots,K$. We will use the notation N=N(S) to denote this correspondence. $\delta_{i,j}$ is the Kronecker delta function which takes value 1 for i=j and value 0 for $i \neq j$.

3.3 Queueing Disciplines

3.3.1 Definition

The queueing discipline can be specified by two families of parameters $a(i|k,S)$ and $r(i|S)$. A customer of class k arrives into the i^{th} station of the queue with probability $a(i|k,S)$, $i = 1,\dots,n+1$ given that the occupancy immediately prior to the arrival was S. Since all customers arrive into the queue, the relationship

$$\sum_{i=1}^{n+1} a(i|k,S) = 1$$

must hold true for all occupancies S and customer classes k.

A customer in station i receives service at rate $r(i|S)$ given that the system is in occupancy S. A customer in station i has its remaining service time X_i decreased at rate

$$\frac{dX}{dt}\, i = -r(i|S)$$

for $i=1,\dots,n$ given that the system is in occupancy S. The queue rate is $R(S) = \sum_{i=1}^{n} r(i|S)$. For most (but not all) queueing disciplines the queue rate is 1.

A discipline is considered to be class independent if the arrival rates $a(i|k,S) = a(i|n)$ and service rates $r(i|S) = r(i|n)$ are dependent only on the number of customers in the system and not on the class of the arriving customer. A <u>class independent discipline</u> (CID) treats all classes alike.

3.3.2 Feasibility

We must now consider the set of occupancies our system will be allowed to enter. We call this the set of <u>feasible</u> occupancies.

Obviously the occupancy $\emptyset$ where there are no customers in the system will be feasible. Occupancy S is feasible if (1) $S=S'+(i,k)$ for some feasible occupancy S', station i, customer class k, and $a(i|k,S')$ non-zero, or (2) $S=S'-i$ for some feasible occupancy S', station i, and nonzero $r(i|S')$. In other words, S is feasible if it can be reached by a finite sequence of arrivals and/or departures with nonzero probability from occupancy $\emptyset$. Infeasible states are assigned probability zero and are ignored in the rest of this work.

3.3.3 Examples

We present descriptions of several common queueing disciplines with the above notation. Table 3.2 contains their queueing parameter values.

First Come First Served (FCFS): A new customer enters the tail-end of the queue. Only the customer at the head of the queue receives service. When this customer departs all remaining customers advance one place in the queue. The parameter values for the FCFS discipline are $a(i|k,S) = \delta_{i,n+1}$ and $r(i|S) = \delta_{i,1}$.

Processor Sharing (PS): This discipline is the limiting case of a no-overhead round robin fixed quantum discipline as the quantum tends to zero [C1, B2]. Every customer in the queue is processed in parallel at equal rates. Each customer may also arrive into any station of the queue with equal probability. The queueing parameter values for the PS discipline are $a(i|k,S) = 1/(n+1)$ and $r(i|S) = 1/n$.

Last Come First Served Preemptive Resume (LCFSPR): This discipline is a priority preemptive resume discipline in which the priority of

the customer is the time in which he enters the queue. The customer with the highest priority (the customer who entered last) is the only one who receives service. Upon completion the customer with the next highest priority resumes service from the point of preemption. Preemption takes no time. Upon arrival a customer preempts the customer receiving service. The LCFSPR discipline has the queueing parameter values $a(i|k,S) = \delta_{i,1}$ and $r(i|S) = \delta_{i,1}$

Infinite Server (IS): In this discipline each customer receives service from one of infinitely many identical servers. The IS discipline parameter values are $a(i|k,S) = 1/(n+1)$ and $r(i|S) = 1$.

Fixed Priority Disciplines: We consider two priority non-CID's the PFP (Preemptive fixed priority) and the NFP (Non Preemptive Fixed Priority) disciplines. For both disciplines let $n_k{*}k$ be a sequence of n_k stations occupied by n_k customers of class $k(n_k \geq 0)$. The PFP discipline orders customers according to class restricting feasible occupancies to the form $(n_1{*}1,\ldots,n_K{*}K)$. The NFP discipline is defined analogously, except that the customer at the head of the queue is not preempted. A typical occupancy is $(k_1,n_1{*}1,\ldots,n_K{*}K)$ with a customer of class k_1 receiving service. In both NFP and PFP disciplines, customers within a class are treated in a FCFS fashion. The arrival probabilities for PFP are $a(i|k,S) = \delta_{i,1+\sum_{j=1}^{k} n_j}$, $i=1,\ldots,n+1$, and for NFP are $a(i|k,S) = \delta_{i,2+\sum_{j=1}^{k}(n_j-\delta_{j,k(1)})}$, $i=1,\ldots,n+1$. The service rates for both NFP and PFP are $r(i|S) = \delta_{i,1}$, $i=1,\ldots,n$.

Composite Queueing (CQ): This is a non-CID in which all customers of class k receive equal amounts of service, namely

$\frac{1}{n_k} \frac{P(N(S)-k)}{P(N(S))}$ where P is any positive valued function defined on N(S) for all feasible S. Finally customers may arrive into any station with equal probability $(a(i|k,S) = 1/(n+1))$.

Some common disciplines such as shortest job first (SJF) and longest job first (LJF) cannot be represented. SJF and LJF cannot be represented since neither the arrival probabilities nor the service rates in our notation can depend on the remaining service times of the customers in the queue.

Table 3.2 Queueing discipline parameters

Discipline	$a(i\|k,S)$	$r(i\|S)$	$R(S)$
FCFS	$\delta_{i,n+1}$	$\delta_{i,1}$	1
PS	$1/(n+1)$	$1/n$	1
LCFSPR	$\delta_{i,1}$	$\delta_{i,1}$	1
IS	$1/(n+1)$	1	n
PFP	$\delta_{i,1+\sum_{j=1}^{k} n_j}$	$\delta_{i,1}$	1
NFP	$\delta_{i,2+\sum_{j=1}^{k} (n_j-\delta_{j,k(1)})}$	$\delta_{i,1}$	1
CQ	$1/(n+1)$	$\frac{P(N(S)-k(i))}{n_{k(i)} P(N(S)}$	$\frac{\sum_{j=1}^{K} P(N(S)-j)}{P(N(S))}$

3.4 Differential Equations for the SQ System

3.4.1 State Probability Density Functions

The basis for all measures of behavior is the state probability density function (spdf). Let $p(t,X)$ be the spdf for the system being in state $X=(S,X_1,\ldots,X_n)$ at time t. If S is unfeasible, then $p(t,X)=0$. We can define a probability density function on the occupancy S as

$$p(t,S) = \int_0^\infty \ldots \int_0^\infty p(t,X)\{dXi\}_{i=1,\ldots,n}. \qquad (1)$$

Similarly we can define a probability function on N as

$$p(t,N) = \sum_{\substack{\text{all } S \\ N(S)=N}} p(t,S). \qquad (2)$$

Finally we have a normalization condition, namely

$$\sum_{\substack{\text{all} \\ \text{feasible } S}} p(t,S) = 1. \qquad (3)$$

3.4.2 Differential Equations

We can obtain equations describing the behavior of the system at time $t+\Delta t$ in terms of the possible states the system might have been in at time t. The system may transit to state X in the interval $[t,t+\Delta t]$, as a result of 1) service, 2) arrivals, or 3) departures. For small values of Δt multiple events occur with probability of order (Δt^2) and will be ignored. The notation we use will be in terms of the state X at time $t+\Delta t$.

(1) The system is in state $(S,\{X_i+r(i\mid S)\Delta t\}_{i=1,n})$ at time t and neither departures nor arrivals occur. The probability of no arrivals in Δt is $(1-(\sum_{k=1}^{K} \lambda_k \Delta t)$. The customer in station i will receive $r(i\mid S)\Delta t$ units of service during the interval $[t,t+\Delta t]$. The rate at which the system transits to X due to service is:

$$(1 - \sum_{k=1}^{K} \lambda_k \Delta t)p(t,(S,\{X_i+r(i\mid S)\Delta t\}_{i=1,n}))$$

This rate can be expanded as a Taylor series around each X_i with the resulting rate

$$p(t,X)+ [\sum_{i=1}^{n} \frac{\partial(t,X)}{\partial X_i} r(i\mid S) - \sum_{k=1}^{K} \lambda_k p(t,X)]\Delta t + 0(\Delta t^2).$$

(2) The system is in occupancy S-i at time t and a customer of class k(i) arrives within the interval $[t,t+\Delta t]$. This may occur for each station $i=1,\ldots,n$. Ignoring service as a multiple event, the rate at which the system transits from the state X-i summed over all stations is

$$\sum_{i=1}^{n} \lambda_{k(i)}\Delta t \quad f_{k(i)}(X_i)a(i\mid k(i),S-i) + 0(\Delta t^2)$$

(3) The system is in occupancy S+(i,k) at time t and the customer in station i has a remaining service time of $0<\tau<r(i\mid S+(i,k))\Delta t$. This may occur for each station $i=1,\ldots,n+1$. Ignoring service of other customers as contributing to multiple events, the rate at which the system transits from the state $X+(i,k,\tau)$ to state X is

$$\sum_{k=1}^{K} \sum_{i=1}^{n+1} \int_0^{r(i\mid S+(i,k))\Delta t} p(t,X+(i,k,\tau))d\tau.$$

Expanding the integrand in a Taylor series around 0 and integrating results in the expression

$$\sum_{k=1}^{K}\sum_{i=1}^{n+1} r(i|S+(i,k))\Delta t p(t,X+(i,k,0))+O(\Delta t^2).$$

The resulting difference equation is

$$\begin{aligned} p(t+\Delta t,X)-p(t,X)= & \sum_{i=1}^{n} \frac{\partial p(t,X)}{\partial X_i} r(i|S)\Delta t-\sum_{k=1}^{K}\lambda_k p(t,X)\Delta t \\ & +\sum_{i=1}^{n}\lambda_{k(i)}a(i|k(i),S-i)p(t,X-i)f_{k(i)}(X_i)\Delta t \\ & +\sum_{k=1}^{K}\sum_{i=1}^{n+1} r(i|S+(i,k))p(t,X+(i,k,0))\Delta t+O(\Delta t). \end{aligned}$$

Now the derivative of $p(t,X)$ is

$$\frac{dp(t,X)}{dt}=\lim_{\Delta t\to 0}\frac{p(t+\Delta t,X)-p(t,X)}{\Delta t}.$$

Applying this to the difference equation:

$$\begin{aligned} \frac{dp(t,X)}{dt} = & \sum_{i=1}^{n}\frac{\partial p(t,X)}{\partial X_i} r(i|S) \\ & -\sum_{k=1}^{K}\lambda_k p(t,X) \\ & +\sum_{i=1}^{n}\lambda_{k(i)}a(i|k(i),S-i)p(t,X-i)f_{k(i)}(X_i) \\ & +\sum_{k=1}^{K}\sum_{i=1}^{n+1} r(i|S+(i,k))p(t,X+(i,k,0)). \end{aligned} \quad (4)$$

A system is said to be in equilibrium if

$$\frac{dp(t,X)}{dX}=0$$

for all states X. The notation $p(X)$, $p(S)$, and $p(N)$ will be used for the equilibrium values of $p(t,X)$, $p(t,S)$, and $p(t,N)$. Some work has been done [M2] in obtaining transient solutions for particular SQ systems. In general, the problem is difficult and will not be considered within this work. The balance (equilibrium) equations for the system are

$$\sum_{i=1}^{n}\left[\frac{\partial p(X)}{\partial X_i} r(i|S)+\lambda_{k(i)}a(i|k(i),S-i)p(X-i)f_{k(i)}(X_i)\right] = \sum_{k=1}^{K}\left[\lambda_k p(X)-\sum_{i=1}^{n+1} r(i|k,S+(i,k))p(X+(i,k,0))\right] \quad (5)$$

for each state X.

We can always obtain solutions to (5). However, if the queue is overloaded (for at least one class, the mean interarrival time is smaller than the mean service time), then it may not be possible to find a solution to both (5) and the normalization criterion (3). Such queues have no equilibrium spdf. We assume the arrival rates, service times and service rates R(S) are such that the queue is not overloaded. Furthermore, we consider only ergodic systems, so that the solution to (5) and (3) must be unique [D1].

3.5 Local Balance, Station Balance, and Product Form Solutions

3.5.1 Local Balance and Station Balance

The solution of (5) is in general difficult. Work has been done to obtain both closed form solutions [B2, C1, C4, J1, L2, M2, R1] and numerical solutions for systems with service time distributions with rational Laplace transforms. We will introduce the properties of local balance [B2, C1] and station balance [C4] and restrict ourselves to systems satisfying those properties.

Local balance: $p(X)$ satisfies local balance if the bracketed terms on the right hand side of the balance equation vanish, that is if

$$\lambda_k p(X) = \sum_{i=1}^{n+1} r(i)S+(i,k))p(X+(i,k,0)) \qquad (6)$$

for each state X and customer class k. Intuitively this means that the rate at which the system departs state X due to the arrival of a customer of class k is balanced by the rate at which the system enters state X due to the departure of a customer of class k. Note that because the system is in equilibrium the following must be true:

$$\sum_{i=1}^{n} [\lambda_{k(i)} a(i|k(i),S-i)p(X-i)f_{k(i)}(X_i)+r(i|S)\frac{\partial p(X)}{\partial X_i}]=0 \qquad (7)$$

i.e., the left hand side of the balance equation vanishes.

Station balance: $p(X)$ satisfies station balance if the bracketed terms on the left hand side of the balance equation vanish, that is if

$$\lambda_{k(i)} a(i|k(i),S-i)p(X-i)f_{k(i)}(X_i)=-r(i|S)\frac{\partial p(X)}{\partial X_i} \qquad (8)$$

for each state X and occupied station i. Intuitively this means

that the rate at which the system enters state X due to the arrival of a customer into station i is balanced by the rate at which the system leaves state X due to the servicing of that customer in station i.

These two properties form the basis from which the rest of this chapter and the next one are developed. Before continuing, two interesting results should be noted. First, local balance implies the "M → M property" [M3], which is that Markovian (Poisson) arrival processes lead to Markovian departure processes for each customer class. Since the SQ systems considered in this chapter have Poisson arrival processes, local balance implies that they have Poisson departure processes. The second result concerns the extensibility of our analysis to SQ systems with general arrival processes. If the properties of local and station balance are defined on such systems, it can be shown that such systems satisfying the property of local balance can only have Poisson arrival processes. Because of this result we have not considered more general arrival processes in this chapter. The above result can be found in Appendix A.

3.5.2 Product Form Solutions

If we look at the remaining service time X_i of a customer in station i at a random time, this time will be distributed according to the forward recurrence time probability density function $\mu_{k(i)}(1-F_{k(i)}(X_i))$ associated with its initial value distribution $F_{k(i)}$ [D1]. Therefore, if each X_i were distributed independently of all the others and of S, the equilibrium $p(X)$ would take the form

$$p(X)=p(S)\prod_{i=1}^{n}\mu_{k(i)}(1-F_{k(i)}(X_i)).$$

Experience indicates that p(S) is usually proportional to λ_k/μ_k for each customer of class k in the queue. If this is true, then

$$p(S)= C\cdot q(S)\prod_{i=1}^{n}\lambda_{k(i)}/\mu_{k(i)}$$

where C is a normalizing constant, $q(\emptyset)=1$ when $\emptyset$ is the empty state, and q(S) is algebraically independent of the λ_k and μ_k.

A system has a solution of the product form if the above is true, that is if

$$p(X)= C\cdot q(S)\prod_{i=1}^{n}\lambda_{k(i)}(1-F_{k(i)}(X_i)) \quad (9)$$

for all $X=(S,X_1,\ldots,X_n)$, where C is a constant, $q(\emptyset)=1$ and q(S) is algebraically independent of the various X_i, λ_k, and μ_k. Therefore varying λ_k or μ_k neither affects q(S) nor changes the product form properties of the solution. C may change due to the normalization constraint (3).

We now define product form versions of the three balance conditions. Assuming the product form, substituting it into the various balance equations, and simplifying yields the following product form versions:

<u>Product form balance:</u>

$$\sum_{k=1}^{K}\lambda_k[q(S)-\sum_{j=1}^{n+1}r(j|S+(j,k))q(S+(j,k))] = \sum_{i=1}^{n}\frac{f_{k(i)}(X_i)}{1-F_{k(i)}(X_i)}[a(i|k(i),S-i)q(S-i)-r(i|S)q(S) \quad (10)$$

for all $S=(k(1),\ldots,k(n))$,

<u>Product form local balance:</u>

$$q(S) = \sum_{j=1}^{n+1} r(j)S+(j,k))q(S+(j,k)) \quad (11)$$

for all $S=(k(1),\ldots,k(n))$ and $k=1,\ldots,K$, and

<u>Product form station balance:</u>

$$a(i|k(i),S-i)q(S-i)=r(i|S)q(S) \quad (12)$$

for all $S=(k(1),\ldots,k(n))$ and $i=1,\ldots,n$.

3.5.3 <u>Relationships Between Local Balance, Station Balance and the Product Form</u>

We are now in a position to explore relationships between systems exhibiting local balance and station balance properties with equilibrium state probability density functions of the product form. As a consequence, we will obtain solutions for some disciplines (LCFSPR, PS, IS, and CQ), and properties of solutions for other queueing disciplines (FCFS, PFP, and NFP). These relationships are presented in the following theorems.

Theorem 3.1: If a system is in station balance, then its equilibrium state probability density function is of the product form.

Proof: The proof is obtained by inducting on n, the number of customers. For n=0 the solution is trivially of the product form. Consider some n>0, the station balance relation (8) is

$$r(i|S)\frac{\partial p(X)}{\partial X_i} = -\lambda_{k(i)}a(i|k(i),S-i)p(X-i)]f_{k(i)}(X_i)).$$

Integrating over X_i yields

$$r(i|S)p(X)=[\lambda_{k(i)}a(i|k(i),S-i)p(X-i](A-F_{k(i)}X_i)) \qquad (13)$$

where A is a constant of integration. If p(X)=0 it is trivially of product form, so assume p(X)>0. Also since S is feasible, R(S)>0. Pick i such that r(i|S)>0. Then the bracketed term is non-zero. Since $\lim_{X_i\to\infty} p(X)=0$, we must have $\lim_{X_i\to\infty} (A-F_{k(i)}(X_i))=0$. This occurs only if A=1.

Substituting in for A and using the induction hypothesis to substitute the product form for p(X-i) yields

$$p(X)= C\cdot\frac{a(i|k(i),S-i)}{r(i|S)}\, q(S-i) \prod_{j=1}^{n} \lambda_{k(j)}(1-F_{k(j)}(X_j)).$$

Thus p(X) is in product form and q(S) satisfies (12) for any i such that r(i|S)q(S)>0. If r(i|S)q(S)=0, a(i|k(i),S-i)q(S-i)=0 and so (12) is true for all i and S.

Theorem 3.2: If a system satisfies station balance, then it satifies local balance.

Proof: Assume p(X) satisfies station balance. Then it is in product form and satisfies product form station balance. Since (12) holds for all S and i=1,...,n, it is satisfied for S+(j,k). Note that S+(j,k)-j=S. Equation (12) becomes

$$a(j|k,S)q(S)=r(j|S+(j,k))q(S+(j,k))$$

Summing over j we obtain equation (11), the product form local balance condition. This along with product form implies local balance.

Corollary 3.2.1: Station balance is a property of queueing disciplines regardless of load.

Proof: Equation (12) defines an infinite family of linear equations in q(S) with coefficients which depend only on the discipline. If station balance is satisfied for a given load, then a solution exists for (12). This solution may be used to construct a spdf which satisfies station balance for any load provided that a value for the normalizing constant C can be found.

Corollary 3.2.2: Any queueing discipline which satisfies station balance must begin to serve new customers immediately.

Proof: Let S result with non-zero probability from the arrival of a new customer at station i. Then a(i|k(i),S-i)q(S-i)>0. By station balance r(i|S)>0, which implies that the new customer receives immediate service.

Corollary 3.2.3: The FCFS and fixed priority disciplines do not satisfy station balance.

Proof: They fail to meet the immediate service criterion, for some occupancy S.

Theorem 3.3: If the spdf for a SQ system is in product form, then the system satisfies local balance. If additionally, all the distributions are non-exponential, the system also satisfies station balance.

Proof: Assume product form balance (10). The factor $f_{k(i)}(X_i)/(1-F_{k(i)}(X_i))$ is a constant function of X_i if $F_{k(i)}$ is negative exponential. When all the F_k's are non-exponential, each factor may be varied independently of the others by varying X_i. Therefore the bracketed term on the right side of (10) must vanish for each S and i, and the system is in station balance. Assume at least one of the $f_k(X_i)$ is negative exponential. In this case since q(S) is independent of λ_k, λ_k may be varied independently on the left hand side of (10) with no effect on the right hand side. Consequently the bracketed term on the left hand side vanishes identically and so the system is in local balance.

We now present a theorem which will specify a simple procedure for testing a class independent discipline for station balance. As a consequence the LCFSPR, PS, and IS disciplines will be shown to satisfy station balance.

Theorem 3.4: A class independent discipline satisfies station balance if and only if

$$r(i|n)=R(n)a(i|n-1) \tag{14}$$

for all n>0, and i=1,...,n.

Proof: We first show the "if" portion. To do so we need to determine a q(S) which will satisfy product form station balance. By the assumption of ergodicity, this will determine the spdf. Let

$$q(S) = 1/\prod_{m=1}^{n} R(m). \tag{15}$$

By multiplying both sides of (14) by q(S-i)/R(n) and substituting q(S)=q(S-i)/R(n), we obtain the equation for product form station balance.

We now show the "only if" portion.

Assume class independence and station balance. Then from Theorem 3.1

$$a(i|n-1)q(S-i)=r(i|n)q(S)$$

must be true for all S and i=1,...,n. Now let S_n consist of n customers of the same class. Substituting and summing over i yields

$$q(S_{n-1})=R(n)q(S_n)$$

This may be substituted back into the previous equation to yield (14).

Corollary 3.4.1: The LCFSPR, PS, and IS disciplines satisfy station balance.

Proof: All three disciplines satisfy the assumptions of Theorem 3.4.

To make our study of SQ systems with product form solutions complete, it would be good to show the necessary and sufficient conditions for the existence of the product form. The following theorem does just that for the set of class independent disciplines.

Theorem 3.5: For class independent disciplines, an equilibrium spdf is in product form if and only if either the discipline satisfies station balance or all the service distributions are negative exponential with the same mean.

Proof: We show the "if" portion first. By Theorem 3.1 station balance implies product form. Therefore we need only show that all negative exponential service distributions with the same means implies the product form. The product form balance equation for a CID is

$$\sum_{k=1}^{K} \lambda_k [q(S) - \sum_{j=1}^{n+1} r(j|n+1) q(S+(j,k))]$$
$$= \sum_{i=1}^{n} \frac{f_{k(i)}(X_i)}{1-F_{k(i)}(X_i)} [a(i|n-1)q(S-i) - r(i|n)q(S)].$$

Assume q(S) is of the form

$$q(S) = 1/\prod_{m=1}^{n} R(m). \qquad (15)$$

If we substitute $f_k(X_i)/(1-F_k(X_i))=\mu_k$ for all k, and q(S) of the form (15) into the product form balance equation, both sides vanish. Therefore, q(S) of the form (15) determines the unique solution spdf of a product form.

We now proceed to show the "only if" portion. Assume the product form. If any one of the F_k is non-exponential, the argument of Theorem 3.3 can be used to show that the discipline satisfies station balance. Assume all exponential distributions. Theorem 3.3 states that the discipline satisfies local balance. It is easy to show that q(S) of the form (15) satisfies local balance product form for all exponential distributions. Therefore by ergodicity it is unique and must determine the equilibrium spdf. Under the assumption of all F_k exponential, $f_k(X_i)/(1-F_K(X_i))=\mu_k$. Substituting this, equation (15), and the local balance product form equation into the product form balance equation yields

$$\sum_{i=1}^{n} \mu_{k(i)} [a(i|n-1)R(n) - r(i|n)] = 0$$

Consider this for two occupancies that differ only in the class of customer at station i and subtract, to obtain

$$(\mu_k - \mu_h)[a(i|n-1)R(n) - r(i|n)] = 0$$

where $k \neq h$. If $\mu_k \neq \mu_h$, then this implies station balance. The remaining case is that all F_k are negative exponential with the same mean.

3.5.4 Composite Queueing

This section will develop some relationships which will be useful in the development of Norton's Theorem to be discussed in the next chapter. We define R(k,N) to be the mean service rate at which all the customers of class k receive service when the system is in state N. R(k,N) is

$$R(k,N)=\sum_{\substack{\text{all } S\\ \text{s.t.} N(S)=N}} \left[\sum_{i=1}^{n} r(i|S)\delta_{k,k(i)}\right] \frac{n_1!\cdots n_k!}{n!}$$

It is easy to show that for systems with product form solutions, and $q(S)=q(N(S))$, that the following relation must hold for the spdf $p(N)$:

$$R(k,N)\mu_k p(N)=\lambda_k p(N-k). \qquad (16)$$

This relation and the results of the next theorem are important for the proof of Norton's Theorem.

Theorem 3.6: A system with a CQ discipline satisfies station balance.

Proof: If it satisfies station balance, it must have a spdf of product form and satisfy equation (12) for all S and $i=1,\ldots,n$. This is trivially true when

$$q(S)=\frac{p(N(S))}{n!}\prod_{k=1}^{K} n_k!$$

Under the assumption of ergodicity, this determines the only solution. Therefore the CQ discipline satisfies station balance.

3.6 Summary

We have completed our characterization of single queue systems. We have introduced the properties of local balance and

station balance and presented simple solutions for various systems (of practical and theoretical interest) exhibiting these properties. These results are both interesting in themselves and form the foundation of the solution techniques we present in the next two chapters.

CHAPTER IV

QUEUEING NETWORKS

4.1 Introduction

Queueing network models have become a standard tool in the evaluation of computer systems. In the last chapter we studied the components (SQ systems) that make up the network. We now consider the network structure, along with the results for SQ systems, to study the solutions and properties of various networks in this chapter. This work is based on that of Chandy, Howard, and Towsley [C4] with one major extension. We allow state dependent routing within the network. Previous work in this direction includes simulation studies by Fultz [F2], and simple numerical approximation by Chen [C5].

This chapter is structured in the following way. A queueing network definition is presented in section 4.2. The differential equations describing the behavior of a queueing network are found in section 4.3. We define the properties of network local balance and station balance. Furthermore, the network has a product form solution. We show that this set of networks includes a useful class of state dependent routing strategies. These results, along with application examples will be presented in section 4.4. Finally we present Norton's Theorem in section 4.5. Norton's Theorem for queueing networks is analogous to Norton's theorem in electrical circuit theory. Norton's Theorem was formulated by Chandy, Herzog, and Woo [C2] for a class of net-

Table 4.1 Notation for Chapter IV

Formal symbolism	Meaning	Section reference
K	number of customer classes	4.2
k	generic index for customer class	4.2
U	number of SQ systems in network	4.2
u,v	generic indices for systems	4.2
S_u	occupancy for system u	4.2
X_u	state of system u	4.2
$n(u,k)$	number of class k customers in system u	4.2
N_u	$(n(u,1),\ldots,n(u,K))$	4.2
$a_u(i\|k,S_u)$	arrival probability for system u	4.2
$r_u(i\|S_u)$	service rate at station i for system u	4.2
$R_u(S_u)$	service rate for system u	4.2
$F_{u,k}$	service distribution for system u	4.2
$f_{u,k}$	service density function for system u	4.2
$1/\mu_{u,k}$	mean service time for system u	4.2
K_o	number of open classes	4.2
$1/\lambda_k$	mean interarrival time for open classes	4.2
$X(u,i)$	remaining service time for customer at station i in system u	4.2
$k(u,i)$	class of customer at station i in system u	4.2
$\bar{S}$	$(S_1,\ldots,S_U)$	4.2
$\bar{X}$	$(X_1,\ldots,X_U)$	4.2
$\bar{S}+(u,i,k,Y)$	$(S_1,\ldots,S_u+(i,k),\ldots,S_U)$	4.2

Table 4.1 Notation for Chapter IV

Formal symbolism	Meaning	Section reference
$\bar{X}+(u,i,k,Y)$	$(X_1,\ldots,X_u+(i,k,Y),\ldots,X_U)$	4.2
$\bar{S}-(u,i)$	$(S_1,\ldots,S_u-i,\ldots,S_U)$	4.2
$\bar{X}-(u,i)$	$(X_1,\ldots,X_u-i,\ldots,X_U)$	4.2
$r(u,i\|\bar{S})$	$r_u(i\|S_u)$	4.2
$a(u,i\|k,\bar{S})$	$a_u(i\|k,S_u)$	4.2
$n(u)$	number of customers in system u	4.2
$\bar{N}$	a matrix with $n_{u,k} = n(u,k)$	4.2
$p(t,\bar{X})$	the spdf for the network	4.2
$\bar{N}+(u,k)$	identical to $\bar{N}$ except with $n_{u,k}+1$	4.2
$\bar{N}-(u,k)$	identical to $\bar{N}$ except with $n_{u,k}-1$	4.2
$R(u,k,N_u)$	service rate for all class k customers in system u	4.2
$b_{u,v,k}$	branching probability for class k from u to v	4.2
$y_{u,k}$	arrival rate of class k into system u	4.2
$p(\bar{X})$	equilibrium spdf for the network	4.3
$1/\lambda_{u,k}$	mean interarrival time for system u when in isolation	4.4
$S(U)$	set of all U subsystems	4.4.1
$S(K)$	set of all customer classes	4.4.1
α	generic index for subset of S(U)	4.4.1
β	generic index for subset of S(K)	4.4.1
$N(\alpha,\beta\|\bar{N})$	number of customers of classes in β residing in systems in α	4.4.1

Table 4.1 Notation for Chapter IV

Formal symbolism	Meaning	Section reference
$f_{\alpha,\beta}$	factor of $y_{u,k}$ where $u\epsilon\alpha$ and $k\epsilon\beta$	4.4.1
$d_{u,k}$		4.4.1
W	number of p-subnetworks in network	4.4.2
w	generic index for p-subnetworks	4.4.2
$e(w)$	entry system to p-subnetwork w	4.4.2
L_w	number of p-branches for p-subnetwork w	4.4.2
$m(w)$	set of systems in p-subnetwork w	4.4.2
$m(w,\ell)$	set of systems in ℓ^{th} p-branch of p-subnetwork w	4.4.2
$\ell(w,v)$	p-branch of p-subnetwork w containing v	4.4.2
p ,	factor of $b_{u,v,k}$ where v and k	4.4.2
g ,	factor of $b_{u,v,k}$ where v and k	4.4.2
$p'_{u,v,k}$		4.4.2
U_f	number of systems with functional $y_{u,k}$'s	4.5.1
*	convolution operator	4.5.1
P_u	marginal probability function for system u	4.5.1
G	normalization constant	4.5.1
$T_{u,k}$	throughput of class k customers from system u	4.5.1

works without state dependent routing strategies. For a given SQ system σ, Norton's Theorem determines a substitution of the remainder of the network by a single "composite queue" without affecting the behavior of the system σ. Table 4.1 contains the notation of chapter IV for quick reference.

4.2 Definition of a Queueing Network

A queueing network consists of SQ systems (queue and server combinations as defined in chapter III) and paths connecting the various systems together. Let a queueing network contain U SQ systems labelled u=1,...,U, and K customer classes k=1,...,K. The discipline of u can be described by the parameters $a_u(i|k,S_u)$, $r_u(i|S_u)$, and $R_u(S_u)$ and its service distributions by $F_{u,k}$, $f_{u,k}$, and $\mu_{u,k}$, as in chapter III.

We will define the occupancy $\bar{S}$ and state $\bar{X}$ of the network as Cartesian products of the individual systems. We may define the following terms analogously to the SQ system:

$\bar{S} = (S_1,\ldots,S_U)$

$\bar{X} = (X_1,\ldots,X_U)$

$\bar{S}+(u,i,k) = (S_1,\ldots,S_u+(i,k),\ldots,S_U)$

$\bar{X}+(u,i,k,Y) = (X_1,\ldots,X_u+(i,k,Y),\ldots,X_U)$

$\bar{S}-(u,i) = (S_1,\ldots,S_u-i,\ldots,S_U)$

$\bar{X}-(u,i) = (X_1,\ldots,X_u-i,\ldots,X_U)$

$r(u,i|\bar{S}) = r_u(i|S_u)$

$a(u,i|k,\bar{S}) = a_u(i|k,S_u)$

n(u) - the number of customers in system u

n(u,k)= the number of class k customers in system u

$\bar{N}$ is a matrix with $n_{u,k}=n(u,k)$ u=1,...,U;k=1,...,K

k(u,i)= the class of the customer at station i of system u.

X(u,i)= the remaining service time of the customer at station i of system u.

$p(t,\bar{X})$= the spdf for the network being in state $\bar{X}$ at time t

$N'=\bar{N}+(u,k)$ is a matrix with $n'_{v,j}=n(v,j)+\delta_{v,u}\cdot\delta_{j,k}$, u,v=1,...,U and j,k=1,...,K.

$N'=\bar{N}-(u,k)$ is a matrix with $n'_{v,j}=n(v,j)-\delta_{v,u}\cdot\delta_{j,k}$, u,v=1,...,U and j,k=1,...,K.

$N_u=(n(u,1),\ldots,n(u,K))$

$R(u,k,N_u)$ is the service rate for all customers of class k in system u when the state of u is N_u (section 3.5.4).

System u may possibly be fed both by other systems and by an outside source. We define $b_{u,v,k}(\bar{N}-(u,k))$ to be the branching probability that a customer of class k departing from system u will enter system v given that the network immediately prior to departure was in state $\bar{N}$. Note that the branching probabilities do not depend on the customer in transition.

The network may be open for customers of $K_o \leq K$ classes. Customers of class k=1,...,K_o arrive into the network at rate λ_k and enter the u^{th} system from an external Poisson source labelled 0 with probability $b_{0,u,k}(\bar{N})$. $\bar{N}$ is the state of the network prior to the arrival. A customer from the open class k may leave the network from the u^{th} system to an external sink also labelled 0 with probability $b_{u,0,k}$

$(\bar{N}-(u,k))$. We assume $b_{0,0,k}=0$ for $k=1,\ldots,K$. Clearly,

$$\sum_{u=0}^{U} b_{v,u,k}(\bar{N}-(u,k))=1 \text{ for all } v=0,\ldots,U \text{ and } k=1,\ldots,K_o. \quad (17)$$

A network is considered <u>open</u> if $K_o=K$. An example of an open network is given in Figure 4.1

The network may be closed for customers of $K-K_o$ classes. Customers of class k, $k=K_o+1,\ldots,K$, neither enter nor depart the network. A constant number N_k of each class cycle endlessly through the network. For each closed customer class

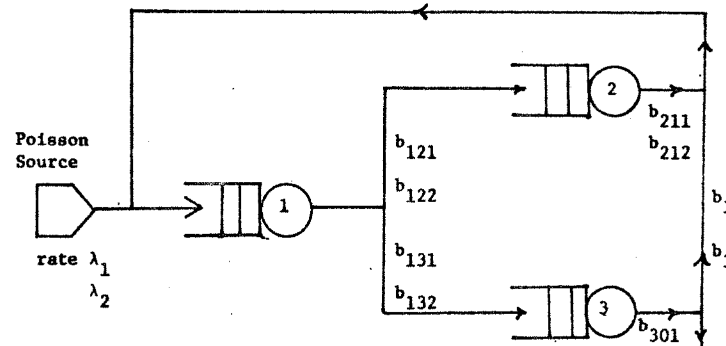
$$\sum_{u=1}^{U} b_{v,u,k}(\bar{N}-(u,k))=1 \text{ for } v=1,\ldots,U,\ k=K_o+1,\ldots,K, \text{ and all feasible } \bar{N} \text{ with } n(u,k)\geq 1. \quad (18)$$

For notational purposes, $b_{0,u,k}=b_{u,0,k}=0$ for $u=1,\ldots,U$ and $k=K_o+1,\ldots,K$. A network is <u>closed</u> if $K_o=0$. If a network is neither open nor closed $(0 < K_o < K)$, it is <u>mixed</u>. An example of a closed network is given in Figure 4.2.

For each customer class k, there is associated with system u a customer arrival rate $y_{u,k}(\bar{N}-(u,k))$. These rates may depend on the state of the network prior to the arrival excluding the customer in transition $(\bar{N}-(u,k))$ as well as on the network structure. At equilibrium, if it exists, the arrival rate and departure for system u will be identical for customer class k and is $y_{u,k}$. Hence the $y_{u,k}$'s must satisfy the following conservation equations

$$y_{u,k}(\bar{N}-(u,k))=\sum_{v=0}^{U} y_{v,k}(\bar{N}-(u,k))\ b_{v,u,k}(\bar{N}-(u,k)) \text{ for } u=1,\ldots,U \text{ and } k=1,\ldots,K_o \quad (19)$$

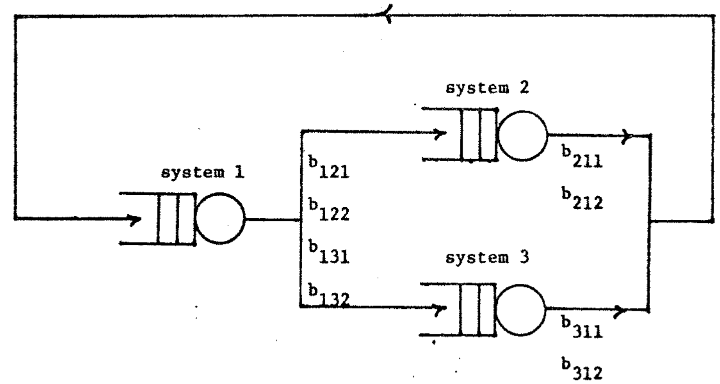
$$y_{0,k}(\bar{N}) = {}_k,\ k=1,\ldots,K_o,$$



If $b_{u,v,k}$ is missing for some u,v,k, $b_{u,v,k}=0$

An Open Queueing Network with Two Customer Classes

Figure 4.1



A Closed Queueing Network with Two Customer Classes

Figure 4.2

and

$$y_{u,k}(\bar{N}-(u,k)) = \sum_{v=1}^{U} y_{v,k}(\bar{N}-(u,k))b_{v,u,k}(\bar{N}-(u,k)) \text{ for } u=1,\ldots,U \text{ and } k=K_o+1,\ldots,K.$$

In the case of constant branching probabilities, these equations reduce to

$$y_{u,k} = \sum_{v=0}^{U} y_{v,k} b_{v,u,k} \text{ for } u=1,\ldots,U \text{ and } k=1,\ldots,K_o,$$

$$y_{0,k} = \lambda_k \text{ for } k=1,\ldots,K_o, \tag{20}$$

and

$$y_{u,k} = \sum_{v=1}^{U} y_{v,k} b_{v,u,k} \text{ for } u=1,\ldots,U \text{ and } k=K_o+1,\ldots,K.$$

For the closed customer classes, the $y_{u,k}$'s may be determined up to normalizing constants by solving the set of linear equations (19) or (20). These constants arise due to the fact that the equations for the closed classes are dependent, the $y_{u,k}$'s are relative, rather than absolute, arrival rates. On the other hand the $y_{u,k}$'s associated with open classes are exact because the equations for the open classes are independent and produce only one solution.

4.3 Queueing Network Differential Equations and Balance Equations

Using previously defined notation, the differential equations for $p(t,\bar{X})$ can be derived by arguments similar to those and used in chapter III. the differential equations are:

$$\frac{dp(t,\bar{X})}{dt} = \sum_{u=1}^{U}\left\{\sum_{i=1}^{n(u)}\left[\frac{\partial p(t,\bar{X})}{\partial X(u,i)} r(u,i|\bar{S}) + \right.\right.$$

$$\sum_{v=1}^{U}\sum_{j=1}^{n(v)+1} b_{v,u,k(u,i)}(\bar{N}-(u,k(u,i)))\, a(u,i|k(u,i),\bar{S}-(u,i))\cdot$$

$$f_{u,k(u,i)}(X(u,i))\; r(v,j|\bar{S}-(u,i)+(v,j,k(u,i)))\cdot$$

$$p(t,\bar{X}-(u,i)+(v,j,k(u,i)),0)+\lambda_{k(u,i)}\, b_{0,u,k(u,i)}(\bar{N}-(u,k(u,i)))\cdot$$

$$\left.\left. a(u,i|k(u,i),\bar{S}-(u,i))\; f_{u,k(u,i)}(X(u,i))\; p(t,\bar{X}-(u,i))\right]\right\} +$$

$$\sum_{k=1}^{K_o}\left[\sum_{u=1}^{U}\sum_{i=1}^{n(u)+1} p(t,\bar{X}+(u,i,k,0))\; r(u,i|\bar{S}+(u,i,k))\cdot\right.$$

$$\left. b_{u,0,k}(\bar{N}) - y_{0,k}(\bar{N})\; p(t,\bar{X})\right] . \tag{21}$$

We denote the equilibrium spdf by $p(\bar{X})$. By setting $\frac{dp(t,\bar{X})}{dt}$ to zero, we obtain the network balance equation

$$\sum_{u=1}^{U}\left\{\sum_{i=1}^{n(u)}\left[\frac{\partial p(\bar{X})}{\partial X(u,i)} r(u,i|\bar{S}) + \sum_{v=1}^{U}\sum_{j=1}^{n(v)+1} b_{v,u,k(u,i)}(\bar{N}-(u,k(u,i)))\cdot\right.\right.$$

$$a(u,i|k(u,i),\bar{S}-(u,i))\; r(v,j|\bar{S}-(u,i)+(v,j,k(u,i)))\cdot$$

$$f_{u,k(u,i)}(X(u,i))\; p(\bar{X}-(u,i)+(v,j,k(u,i)),0) + \lambda_{k(u,i)}\, f_{u,k(u,i)}(X(u,i))\cdot$$

$$\left.\left. b_{0,u,k(u,i)}(\bar{N}-(u,k(u,i)))\; a(u,i\; k(u,i),\bar{S}-(u,i))\; p(\bar{X}-(u,i))\right]\right\} =$$

$$\sum_{k=1}^{K_o}\left[\sum_{u=1}^{U}\sum_{i=1}^{n(u)+1} p(\bar{X}+(u,i,k,0))\; r(u,i|\bar{S}+(u,i,k))\, b_{u,0,k}(\bar{N}) - \right.$$

$$\left. y_{0,k}(\bar{N})\; p(\bar{X})\right] . \tag{22}$$

We assume the network is ergodic, and hence that the balance equation has a unique solution.

As with SQ systems, we limit our study to networks exhibiting the following two properties.

<u>Network local balance for system u:</u> $p(\bar{X})$ satisfies network local balance for system u if the term in braces $\{\;\}$ in equation (22) vani-

shes for all feasible $\bar{X}$, that is if

$$\sum_{i=1}^{n(u)} \frac{\partial p(\bar{X})}{\partial X(u,i)} r(u,i|\bar{S}) + \sum_{v=1}^{U} \sum_{j=1}^{n(v)+1} b_{v,u,k(u,i)}(\bar{N}-(u,k(u,i)))\cdot$$

$$a(u,i|k(u,i),\bar{S}-(u,i))\, r(v,j|\bar{S}-(u,i)+(v,j,k(u,i)))\cdot$$

$$f_{u,k(u,i)}(X(u,i))\, p(\bar{X}-(u,i)+(v,j,k(u,i)),0) + \lambda_{k(u,i)}\cdot$$

$$f_{u,k(u,i)}(X(u,i))\, b_{0,u,k(u,i)}(\bar{N}-(u,k(u,i)))\cdot$$

$$a(u,i|k(u,i),\bar{S}-(u,i))\, p(\bar{X}-(u,i)) = 0 \quad . \qquad (23)$$

Intuitively this means the rate at which the network enters state $\bar{X}$ due to arrivals into system u balances the rate at which the network leaves state $\bar{X}$ due to the servicing of customers in system u.

Network station balance for system u: $p(\bar{X})$ satisfies network station balance for system u if the term in brackets in equation(23) vanishes for $i=1,\ldots,n(u)$ and all feasible $\bar{X}$. Intuitively this means the rate at which the network enters state $\bar{X}$ due to an arrival at station i of system u is identical to the rate at which the network leaves state $\bar{X}$ due to the service of the customer in station i of system u, for each i.

4.4 Network Product Form

We are concerned with finding the relationship between network local balance and network station balance for a given system and the properties of local and station balance for the system when treated as a SQ system. In considering the behavior of a system in the network, we relate it to the system in isolation from the network.

System u is said to be in isolation if it has the same service discipline and distribution as in the network, but with customers of class k arriving in a Poisson manner with rates $\lambda_{u,k}$, $k=1,\ldots,K$ proportional to the arrival rates $y_{u,k}$ of the network. Thus $\lambda_{u,k}=A\, y_{u,k}$. The constant A is unimportant provided that it is positive and not so large that the isolated queue has no steady state.

4.4.1 Conditions for Product Form

We need also consider the functional forms that the branching probabilities $b_{u,v,k}(\bar{N}-(u,k))$ may take. We do it in the following manner. We consider what functional forms the arrival rates $y_{u,k}$ $(\bar{N}-(u,k))$ may take. In a later section we consider the functional forms the $b_{u,v,k}$'s can take to satisfy the constraints on the $y_{u,k}$'s. We need to first introduce some new notation. Let S(K) be the set of customer classes $1,\ldots,K$ and S(U) the set of subsystems $1,\ldots,U$ in the network. Let

$$N(\alpha,\beta|\bar{N}) = \sum_{u\in\alpha} \sum_{k\in\beta} n(u,k) \text{ for all } \alpha \subseteq S(U),\ \beta \subseteq S(K) \qquad (24)$$

when the network is in state $\bar{N}$. $N(\alpha,\beta|\bar{N})$ is the number of customers of the classes in β residing in the systems in α when the network is in state $\bar{N}$.

Using this notation, the arrival rates may have the following functional forms

$$y_{u,k}(\bar{N}) = d_{u,k} \prod_{\substack{\alpha \subseteq S(U)\\ u\in\alpha}} \prod_{\substack{\beta \subseteq S(K)\\ k\in\beta}} f_{\alpha,\beta}[N(\alpha,\beta|\bar{N})] \text{ for } u=1,\ldots,U,\ k=1,\ldots,K, \text{ all feasible } \bar{N}+(u,k) \qquad (25)$$

where $f_{\alpha,\beta}$ is an arbitrary positive real valued function and the $d_{u,k}$'s are arbitrary positive constants. For the case that $y_{u,k}$ is constant, $y_{u,k} = d_{u,k}$. $y_{u,k}$ can be thought of as a product of factors, each factor dependent on the number of customers of a subset of classes including k residing in a subset of systems including u. For simplicity let $f_{\emptyset,\emptyset} = 1$.

Theorem 4.1: If the network satisfies the above constraint (25) and each system satisfies local balance when isolated, then:

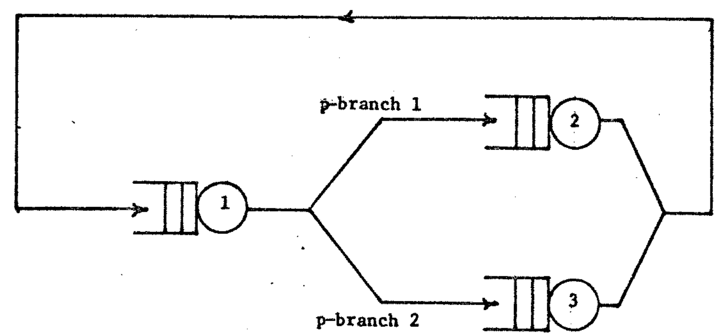
1. the equilibrium spdf of the network takes the product form

$$p(\bar{X}) = \frac{1}{G} \prod_{\alpha \subseteq S(U)} \prod_{\beta \subseteq S(K)} \prod_{i=1}^{N(\alpha,\beta|\bar{N})-1} f_{\alpha,\beta}(i) \prod_{u=1}^{U} p_u(X_u) \quad (26)$$

   where G is a normalizing constant and $p_u(X_u)$ is the spdf of system u in isolation with arrival rate $d_{u,k}$,
2. the network is locally balanced, and
3. each system satisfying station balance when isolated satisfies network station balance.

4.4.2 Parallel Subnetworks

A parallel subnetwork (p-subnetwork) is a union of a finite number L of mutually disjoint sets of systems termed parallel branches (p-branches). If system v is part of p-branch ℓ, it cannot be part of p-branch m, m≠ℓ. If u and v are systems in p-branch ℓ, then $b_{u,v,k}$ may be nonzero for k=1,...,K. Customers may enter the p-sub-



p-subnetwork: $e(1)=1$ $L_1=2$
$m(1,1)=\{2\}$, $m(1,2)=\{3\}$
$m(1)=\{2,3\}$

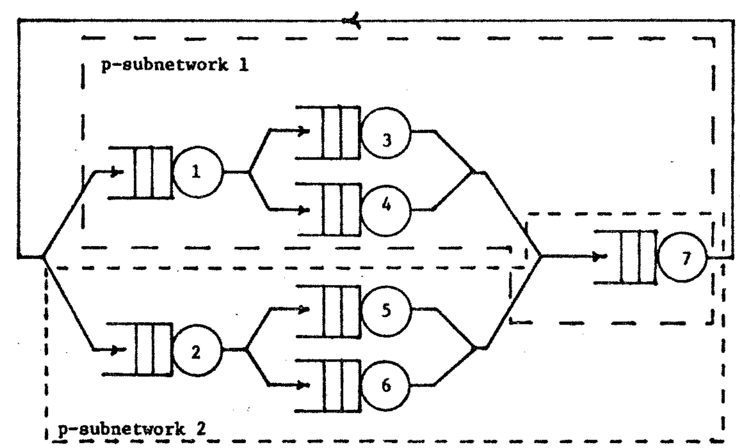
A Simple P-subnetwork

Figure 4.3

network from one and only one entry system u_1 which is not a member of the p-subnetwork. Similarly, customers may exit the p-subnetwork through one and only one exit system u_2 also not a member of the p-subnetwork. If v is any member of the p-subnetwork, then the branching probabilities $b_{u_1,v,k}$ may be nonzero. However, if system $u \neq u_1$ is not a member of the p-subnetwork then $b_{u,v,k}=0, k=1,\ldots,K$. Similarly, if v is a member of the p-subnetwork, then $b_{v,u_2,k}$ may be nonzero. But, if $u \neq u_2$ is not a member of the p-subnetwork, then $b_{v,u,k}=0, k=1, \ldots,K$. Once a customer enters p-branch ℓ, $\ell=1,\ldots,L$, it can only exit through system u_2. Hence, if u is a member of p-branch ℓ, and v is a member of p-branch $m \neq \ell$, then $b_{u,v,k}=0, k=1,\ldots,K$. Note: $b_{u,v,k} > 0$ for at least one system v in p-branch ℓ, $\ell=1,\ldots,L$.

Two p-subnetworks may be related to each other in the following ways: (1) they share neither member systems, entry system, nor exit system, (2) they share either the same entry or exit systems but not both (Figure 4.4), (3) they share both the same entry and exit systems (Figure 4.5), or (4) one of the p-subnetworks is nested within the other (Figure 4.6). We only allow a p-subnetwork to be nested within one p-branch of the other p-subnetwork. Figure 4.7 shows a prohibited nesting.

If system u is not the entry to any p-subnetwork, then the branching probabilities $b_{u,v,k}$ are constrained to be constant for $v=1, \ldots,U, k=1,\ldots,K$. Therefore, only the entries to p-subnetworks will be allowed to have state dependent branching probabilities. We introduce the following notation to aid in the definition of the functional forms.



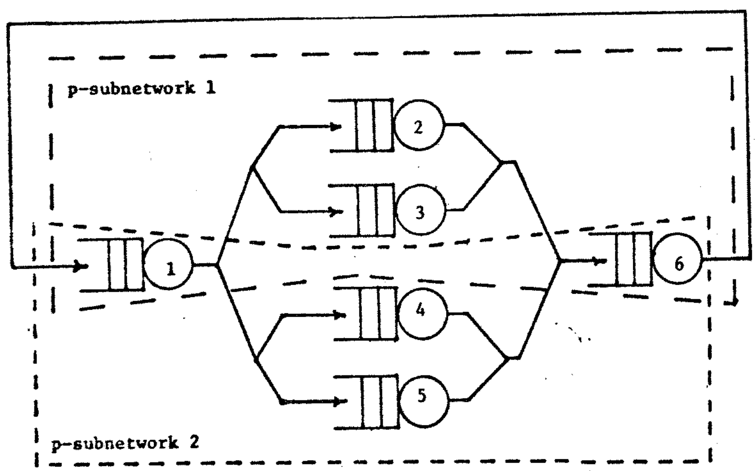
p-subnetwork 1: $e(1)=1$ $L_1=2$
$m(1,1)=\{3\}$, $m(1,2)=\{4\}$
$m(1)=\{3,4\}$

p-subnetwork 2: $e(2)=2$, $L_2=2$
$m(2,1)=\{5\}$, $m(2,2)=\{6\}$
$m(2)=\{5,6\}$

p-subnetwork 3: $e(3)=7$, $L_3=2$
$m(3,1)=\{1,3,4\}$, $m(3,2)=\{2,5,6\}$
$m(3)=\{1,2,3,4,5,6\}$

p-subnetworks 1 and 2 share the same exit system, 7

Figure 4.4



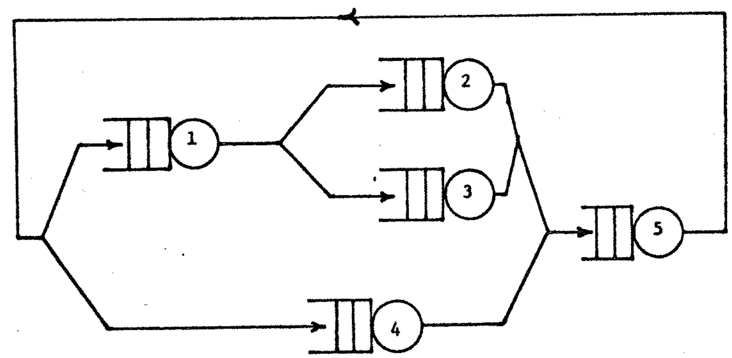
p-subnetwork 1: $e(1)=1$, $L_1=2$
$m(1,1)=\{2\}$, $m(1,2)=\{3\}$
$m(1)=\{2,3\}$

p-subnetwork 2: $e(2)=1$, $L_2=2$
$m(2,1)=\{4\}$, $m(2,2)=\{5\}$
$m(2)=\{4,5\}$

p-subnetwork 3: $e(3)=1$, $L_3=2$
$m(3,1)=\{2,3\}$, $m(3,2)=\{4,5\}$
$m(3)=\{2,3,4,5\}$

p-subnetworks 1 and 2 share the same entry and exit

Figure 4.5

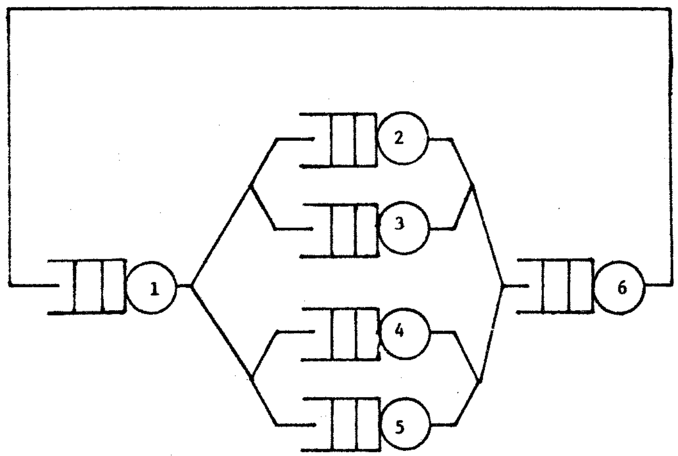


p-subnetwork 1: $e(1)=1$ $L_1=2$
$m(1,1)=\{2\}$, $m(1,2)=\{3\}$
$m(1)=\{2,3\}$

p-subnetwork 2: $e(2)=5$ $L_2=2$
$m(2,1)=\{1,2,3\}$ $m(2,2)=\{4\}$
$m(2)=\{1,2,3,4\}$

p-subnetwork 1 is nested within p-subnetwork 2

Figure 4.6



p-subnetwork 1: $e(1)=1$, $L_1=2$

$m(1,1)=\{2\}$, $m(1,2)=\{3\}$

$m(1)=\{2,3\}$

p-subnetwork 2: $e(2)=1$, $L_2=4$

$m(2,1)=\{2\}$, $m(2,2)=\{3\}$,

$m(2,3)=\{4\}$, $m(2,4)=\{5\}$

$m(2)=\{2,3,4,5\}$

p-subnetwork 1 is nested within two p-branches of p-subnetwork 2. This is prohibited.

Figure 4.7

Let there be W p-subnetworks within the queueing network ordered $w=1,\ldots,W$ satisfying the previously stated relationships. Let $e(W)$ be the entry system to p-subnetwork w. Let L_w be the number of branches in p-subnetwork w ordered $\ell=1,\ldots,L_w$. Define $m(w)$ to be the set of all systems in p-subnetwork w. Define $m(w,\ell)$ to be the set of all systems in p-branch ℓ belonging to p-subnetwork w. If $\ell < 0$ or $\ell > L_w$, then $m(w,\ell) = \emptyset$. Define $\ell(w,v)$ to be the p-branch of p-subnetwork w which includes system v. If $v \notin m(w)$, then $\ell(w,v)=0$. Note: if $v \in m(w)$, then $m(w,\ell(w,v))$ has as a value the set of all systems within the p-branch which includes v.

We can use Figure 3.4 to exemplify the above notation. This network has two p-subnetworks, $w=1,2$. The number of p-branches in p-subnetworks 1 and 2 are $L_1=2$ and $L_2=2$ respectively. The entry systems for the two p-subnetworks are $e(1)=1$ and $e(2)=2$. Considering only p-subnetwork 1, the sets of systems in each of the two branches are $m(1,1) = \{3\}$ and $m(1,2) = \{4\}$. Finally $m(1) = \{3,4\}$.

Recalling the definition of $N(\alpha,\beta|\bar{N})$ as defined in (24), the functional forms for the branching probabilities for system u when u is the entry to at least one p-subnetwork are

$$b_{u,v,k}(\bar{N}) = p'_{u,v,k} \prod_{\substack{\beta \subseteq S(K) \\ k \in \beta}} \prod_{\substack{w=1 \\ u=e(w)}}^{W} P_{m(w,\ell(w,v)),\beta}[N(m(w,\ell(w,v)),\beta|\bar{N})] \cdot g_{m(w),\beta}[N(m(w),\beta|\bar{N})], \text{ for } k=1,\ldots,K$$

and all feasible $\bar{N}+(u,k)$ prior to transition. (27)

$p_{\alpha,\beta}$ and $g_{\alpha,\beta}$ are non-negative valued functions and $p'_{u,v,k}$ is a non-negative constant. For simplicity we assume $p_{\emptyset,\emptyset}$ and $g_{\emptyset,\emptyset}$ have values 1 for all $\overline{N}$. Thus when v is not a member of any p-subnetwork for which u is an entry, $b_{u,v,k}$ will be constant. These branching probabilities must satisfy the constraints of equations (17) and (18). For every p-network u is an entry to and v is a member of, $b_{u,v,k}$ has a factor which is a product of terms. Each term may be a function of the number of customers of some subset of the customer classes containing k which reside in the ℓ^{th} p-branch which contains system v or in the entire p-subnetwork.

Theorem 4.2: Queueing networks with branching probabilities of the form (27) have subsystem arrival rates of the form (25).

Proof: See Appendix C.

Corollary 4.2.1: If a network satisfies equation (27) and each subsystem satisfies local balance when isolated, then:

1. The equilibrium spdf of the network takes the product form of equation (26),
2. the network is locally balanced, and
3. each queue satisfying station balance when isolated satisfies network station balance.

Proof: The proof is a simple consequence of Theorems 3.1 and 3.2.

A consequence of Theorem 4.2 is the determination of the factors $f_{\alpha,\beta}$ which were introduced in equation (25). If $u \in m(w)$, then $y_{u,k}$ has the factors $f_{\alpha,\beta}=p$, where $\alpha=m(w,\ell(w,u))$ and $f_{\alpha,\beta}=g_{\alpha,\beta}$ where

$\alpha=m(w)$; β may be any subset of S(K) including k. All other factors would have value 1.

For the remainder of section 4.4 we consider networks with only one customer class. We delete any index pertaining to customer classes such as the index k or β from our notation throughout the remainder of this section. For example, $f_{\alpha,\beta}$ becomes f_α, $b_{u,v,k}$ becomes $b_{u,v}$, and $\overline{N}+(u,k)$ becomes $\overline{N}+u$. Additionally, the occupancy information S_u is no longer necessary and so X_u becomes $(X(u,1),\ldots,X(u,n(u)))$.

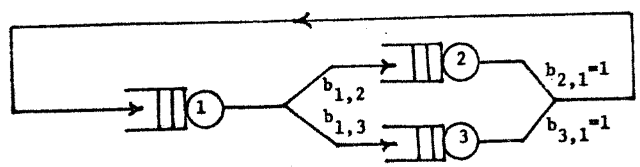
Consider the central server network in Figure 4.8. The two I/O's form a p-subnetwork. Assume all times are exponentially distributed and all disciplines are FCFS. Assume the branching probabilities $b_{1,2}$ and $b_{1,3}$ are of the form tabulated in Figure 4.8. Consider N even for the following discussion. Note that the definition of $b_{1,2}$ and $b_{1,3}$ precludes the feasibility of any state with $n(2)>N/2$ or $n(3)>N/2$. The arrival rates y_2 and y_3 have the functional forms

$$y_u(\overline{N}) = p_{\{u\}}[n(u)]\cdot g_{\{2,3\}}[n(2)+n(3)], \quad u=2,3, \text{ all feasible } \overline{N}+1$$

$$= (N/2-n(u))/(N-n(2)-n(3)), \quad u=2,3 \text{ all feasible } \overline{N}+1$$

If we assume N=4 and $\overline{X}=(\emptyset,(X(2,1),X(2,2)),(X(3,1),X(3.2))$ with no customers in system 1 and two customers in each of systems 2 and 3, then

$$P(\overline{X})=\frac{1}{G}\cdot\prod_{i=0}^{1}(N/2-i)\prod_{i=0}^{1}(N/2-i)\cdot\prod_{i=0}^{3}1/(n-i)\cdot\prod_{i=1}^{2}e^{-\lambda X(2,i)}\cdot\prod_{i=1}^{2}e^{-\lambda X(3,i)}$$
$$=\frac{1}{6\cdot G}\prod_{i=0}^{2}e^{-\lambda X(2,i)}\prod_{i=0}^{2}e^{-\lambda X(3,i)}\quad .$$



n(u) - number of customers in processor u

N - number of customers in network

$b_{1,2}=P_{\{2\}}[n(2)]\cdot g_{\{2,3\}}[n(2)+n(3)]$

$b_{1,3}=P_{\{3\}}[n(3)]\cdot g_{\{2,3\}}[n(2)+n(3)]$

For even N

$P_{\{u\}}[n(u)] = N/2-n(u) \qquad u=2,3$

$g_{\{2,3\}}[n(2)+n(3)] = 1/(N-n(2)-n(3))$

For odd N

$P_{\{u\}}[n(u)] = (N+1)/2-n(u) \qquad u=2,3$

$g_{\{2,3\}}[n(2)+n(3)] = 1/(N+1-n(2)-n(3))$

Figure 4.8

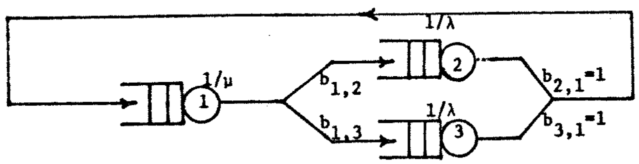
In this case $p(\overline{N})$ has the value

$$p(\overline{N}) = \frac{1}{6\cdot G\cdot\lambda^4} .$$

4.4.3 Examples

We present two application examples of the results of this section. First we present an example showing the effectiveness of the analytic forms (27) and (26) in modeling real world routing strategies. The closed central server network in Figure 4.9 will be used for this example. Let the routing strategy under study be the following: a job leaving processor 1 goes to the processor with the shortest queue length. If both have the same queue length, it enters either with equal probability. For an exact solution one must resort to a numerical solution of a discrete-state continuous-transition Markov model. We will label this model 1. This model is discussed further in appendix D. The network can also be modelled approximately in two ways: first as a network with identical constant branching probabilities, and second, as a network with functional branching probabilities of the forms tabulated earlier in Figure 4.8. These networks make up models 2 and 3 respectively. The model descriptions are summarized in Figure 4.9.

For the above system, the models were compared by looking at the utilization and throughput of processor 1. The results cited are the percentage differences between the models with independent parameters μ/λ and N (the number of customers in the system). The graph in Figure 4.10 gives this percentage difference between model 1 and 2. The percentage difference is with respect to model 1. Note that as



$1/\mu$ - mean service time for processor 1

$1/\lambda$ - mean service times for processors 2 and 3

All times are exponential

All service disciplines are FCFS

$n(u)$ - number of customers in processor u

N - number of jobs in system

Model 1 - exact solution

$$b_{1,2} = \begin{cases} 1 & \text{if } n(3) > n(2) \\ 0 & \text{if } n(3) < n(2) \\ 1/2 & \text{if } n(2) = n(3) \end{cases}$$

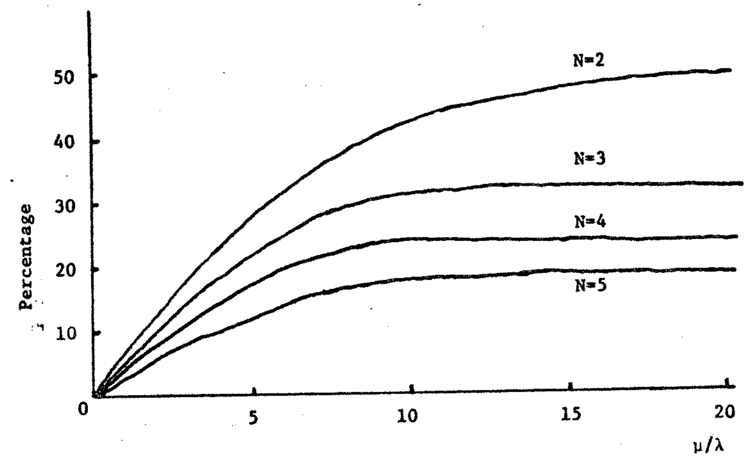
$$b_{1,3} = 1 - b_{1,2}$$

Model 2 - analytical with constant branching probabilities

$$b_{1,2} = b_{1,3} = 1/2$$

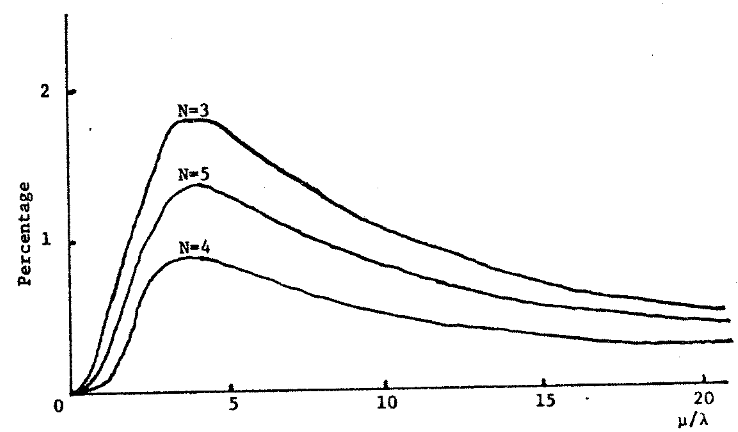
Model 3 - analytical with functional branching probabilities of the forms tabulated in Figure 4.8

Figure 4.9



Percentage Difference in Throughput Between Model 1 and Model 2

Figure 4.10



Percentage Difference in Throughput Between Model 1 and Model 3

Figure 4.11

μ/λ becomes large, the difference becomes appreciable. For a large range of the parameter μ/λ, model 2 is a poor approximation. The graph in figure 4.11 compares model 1 and model 3. Again, the difference is with respect to model 1. For the values of N used the difference is bounded by 2%.

It is conjectured that for larger values of N the difference will decrease. From these results model 3, with its use of functional branching probabilities, is an excellent approximation for the system under consideration.

The second example shows a simple application of the models of this chapter to the analysis of a computer system. Consider the computer system configuration in Figure 4.12 where jobs cycle through the CPU to one of several I/O operations: tape activity with probability .1, disk activity (excluding system disk activity) with probability .4, and system disk activity with probability .5. All service disciplines are FCFS. Note that there are two sets of system files, each residing on two separate disk banks. The service times are assumed to be exponential and can be found in Figure 4.12.

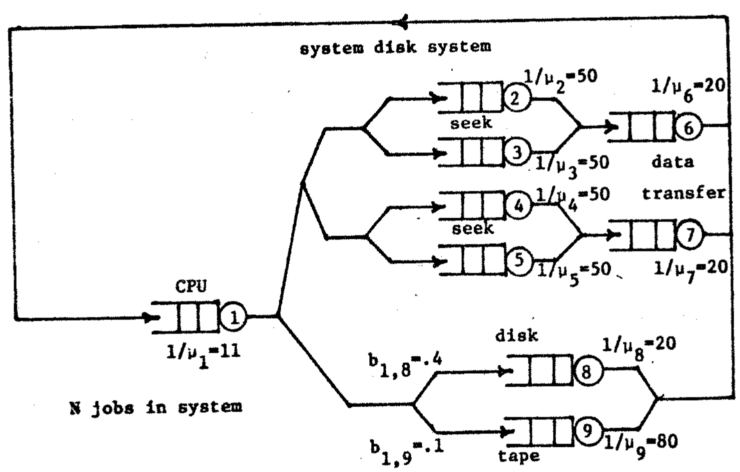
Each system disk bank consists of two identical disks and one dedicated controller-channel combination. An I/O request for system activity queues up for a disk. When it obtains the disk, it then requires a controller to initiate a seek. Once the seek is initiated, the controller is relinquished. Thus the controller can initiate a second seek or transfer data (R/W activity). Consequently, seeks and/or data transfer can occur in parallel in one disk bank with only one

controller. Modeling seeks with separate queues from the data transfer has the disadvantages that (1) a seek and data transfer may occur in parallel for the same disk, and (2) a seek may be initiated even when the controller is busy transferring data. These and other limitations are considered in detail in [B4]. We assume that the mean service times are degraded to account for the above flaws. We assume that jobs enter either system disk bank with equal probability and either disk also with equal probability. We want to determine what improvement a state dependent routing strategy can produce. In particular we consider the strategy by which a job goes to the system disk with the shortest queue length.

The models of this chapter allow the determination of a lower bound on the increase in CPU utilization due to the routing strategy over the same system with random routing. We will use the functional branching probabilities in Figure 4.12 to determine the approximate performance measures for the model with the routing strategy. Figure 4.13 contains a graph of the percentage improvement relative to the original system. For a small expense in software, the routing strategy can increase performance between 4% and 7%, depending on the level of multiprogramming.

4.5 Norton's Theorem

Many queueing network models are used to study the variation of certain system measures (response time, system throughput) as a function of the network structure and service times. In particular, many cases occur where one is interested in varying the parameters of



Branching Probabilities $b_{1,2}$, $b_{1,3}$, $b_{1,4}$, $b_{1,5}$

First Model

$$b_{1,2} = b_{1,3} = b_{1,4} = b_{1,5} = .125$$

Second Model

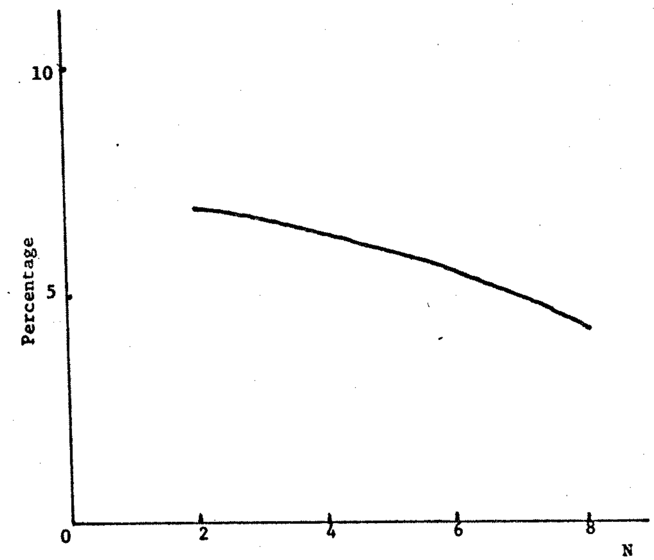
$$b_{1,2} = \begin{cases} \dfrac{.25(N/2-n(2)-n(3)-n(6))}{(N-n(2)-n(3)-n(4)-n(5)-n(6)-n(7))}, & N \text{ even} \\[2ex] \dfrac{.25\ ((N+1)/2-n(2)-n(3)-n(6)}{(N+1-n(2)-n(3)-n(4)-n(5)-n(6)-n(7))}, & N \text{ odd} \end{cases}$$

$$b_{1,3} = b_{1,2}$$

$$b_{1,4} = .25-b_{1,2}$$

$$b_{1,5} = b_{1,4}$$

Figure 4.12



Percentage Improvement in Throughput Between System with State Dependent Routing and the System with Random Routing

Figure 4.13

only one system σ in a network, Figure 4.14. It is desirable to construct an equivalent network in which all the queues except the one of interest are replaced by a single composite queue, Figure 4.15. For the class of queueing networks we have looked at, it can be shown that for certain system measures, the behavior of system σ in the equivalent network is the same as in the original network. This is the result of Norton's Theorem. Norton's Theorem was formulated by Chandy, Herzog, and Woo [C2]. We will present it in an extended form for networks with functional branching probabilities.

4.5.1 Notation and Computational Techniques

It is necessary to present some notation and review a computational technique developed by Buzen [B5] and extended by Chandy, et al. [C2] for the efficient calculation of the normalization constant G. We restrict ourselves to closed queueing networks. We assume that the systems have been ordered such that the first U_f systems have non-constant arrival rates $y_{u,k}$'s of the form in equation (25).

Since we are considering networks whose systems satisfy local balance when in isolation, the spdf $p(\overline{X})$ is of the form

$$p(\overline{X}) = \frac{1}{G}\left[\prod_{\alpha \subseteq S(U_f)} \prod_{\beta \subseteq S(K)} \prod_{i=1}^{N(\alpha,\beta|\overline{N})-1} f_{\alpha,\beta}(i)\right] \prod_{u=1}^{U} p_u(X_u).$$

It is easy to show that the spdf $p(\overline{N})$ has the form

$$p(\overline{N}) = \frac{1}{G}\left[\prod_{\alpha \subseteq S(U_f)} \prod_{\beta \subseteq S(K)} \prod_{i=1}^{N(\alpha,\beta|\overline{N})-1} f_{\alpha,\beta}(i)\right] \prod_{u=1}^{U} p_u(N_u). \quad (28)$$

Because G depends on N_k, k=1,...,K, we refer to $G(N_1,\ldots,N_K)$

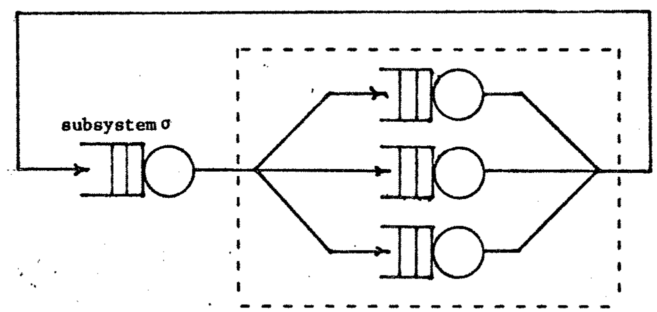


Figure 4.14

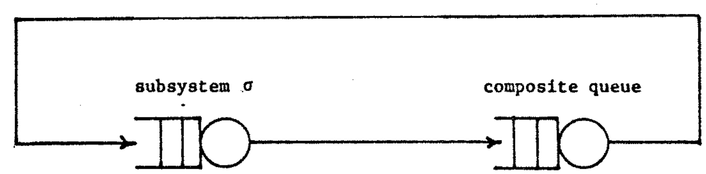


Figure 4.15

$G(N_1,\ldots,N_K)$ is defined as

$$G(N_1,\ldots,N_K) = \sum_{\substack{\Sigma n_{uk}=N_k \\ \text{for } k=1,\ldots,K \\ \text{and } n_{uk}\geq 0}} \prod_{\alpha\subseteq S(U_f)} \prod_{\beta\subseteq S(K)} \prod_{i=1}^{N(\alpha,\beta|\bar{N})-1} f_{\alpha,\beta}(i) \prod_{u=1}^{U} p_u(N_u). \quad (29)$$

G may be thought of as an array with the k^{th} subscript varying from 0 to N_k.

We now present an efficient technique for calculating G.

If $U_f > 0$, we define

$$G_{U_f}(N_1,\ldots,N_K) = \sum_{\substack{\Sigma n_{uk}=N_k \\ \text{for } k=1,\ldots,K \\ u=1,\ldots,U_f, \\ \text{and } n_{uk}\geq 0}} \prod_{\alpha\subseteq S(U_f)} \prod_{\beta\subseteq S(K)} \prod_{i=1}^{N(\alpha,\beta|\bar{N})-1} f_{\alpha,\beta}(i) \prod_{u=1}^{U_f} p_u(N_u). \quad (30)$$

We can think of $p_u(N_u)$ also as an array with the k^{th} subscript varying from 0 to N_k. Finally, we define a convolution operator * such that

$$G = G_{U_f} * p_{U_f+1}(N_{U_f+1}) * \ldots * p_U(N_U).$$

The operator * is associative and commutative. For convenience

$$G_u = G_{u-1} * p_u(N_u) \quad \text{for } u = \max\{2, U_f+1\},\ldots,U.$$

If $U_f=0$, $G_1=p_1$; otherwise G_{U_f} is defined by equation (30). The operator * is defined as follows:

$$G_u(n_1,\ldots,n_K) = \sum_{i_1=0}^{n_1}\cdots\sum_{i_K=0}^{n_k} G_{u-1}(i_1,\ldots,i_K)p_u(n_1-i_1,\ldots,n_K-i_K).$$

From this definition of G we can calculate the following measures.

(1) Let $P_U(n_1,\ldots,n_K)$ be the marginal probability that there are n_k customers of class k in queue U. P_u can be determined as follows,

$$P_U(n_1,\ldots,n_K) = G_{U-1}(N_1-n_1,\ldots,N_K-n_K)/G \cdot p_U(n_1,\ldots,n_K). \quad (31)$$

The marginal probability for any given queue with constant $y_{u,k}$ can be obtained by reordering the queues so that the given queue is indexed by U.

(2) Let T_{Uk} be the throughput of class k customers from the system U. We will derive a simple expression for the case in which $p_U(X_U)$ is a product form and $q_U(S_U)=q_U(N_U)$. $q_U(S_U)$ is as defined in section 3.5.2. T_{Uk} is the rate at which customers get service and leave the queue:

$$T_{Uk} = \sum_{n_1=0}^{N_1}\cdots\sum_{n_k=1}^{N_k}\cdots\sum_{n_K=0}^{N_K} P_U(n_1,\ldots,n_K)R_U(k,N_U)\mu_{Uk}. \quad (32)$$

From equation (16)

$$p_U(n_1,\ldots,n_K)R_U(k,N_U)\mu_{Uk} = p_U(n_1,\ldots,n_k-1,\ldots,n_K)y_{U,k}. \quad (33)$$

From equations (31), (32), and (33)

$$T_{Uk} = \sum_{n_1=0}^{N_1}\cdots\sum_{n_k=1}^{N_k}\cdots\sum_{n_K=0}^{N_k} y_{U,k}\, P_U(n_1,\ldots,n_k-1,\ldots,n_K).$$

$$G_{U-1}(N_1-n_1,\ldots,N_K-n_K)/G = y_{U,k}\cdot G_U(N_1,\ldots,N_k-1,\ldots,N_K)/G \quad (34)$$

4.5.2 Formulation of Norton's Theorem

Consider the closed network of U systems of the last section where system U has a constant arrival rate $y_{U,k}$. We construct an equivalent network consisting of system U and a composite queue c with the CQ discipline, exponential service time distributions with mean 1, and service rate for each customer of class k of $T_k(n_1,\ldots,n_K)/n_k$; n_k is the number of class k customers in the composite queue, $n_k = 0,\ldots,N_k$, $k=1,\ldots,K$. Set T_k to the throughput of system U when there are n_k class k customers in the network, $k=1,\ldots,K$, and the service time of system U is set to zero. Let $P_u'(n_1,\ldots,n_K)$ be the marginal probability of the queue length for system U in the equivalent network.

Theorem 4.4: The queue length distribution for system U in the equivalent network is the same as in the given network. In other words,

$$P_U'(n_1,\ldots,n_K) = P_U(n_1,\ldots,n_K),\quad n_k=0,\ldots,N_k,\ k=1,\ldots,K. \tag{35}$$

Proof: When the service time is 0, the system trivially has a product form solution $p_U(\emptyset)=1$ and $q_U(S)=q_U(N)$. Thus

$$T_k(n_1,\ldots,n_K) = y_{U,k} G_{U-1}(n_1,\ldots,n_k-1,\ldots,n_K)/G_{U-1}(n_1,\ldots,n_K),$$
$$n_k=1,\ldots,N_k,\ k=1,\ldots,K,\ n_\ell=0,\ldots,N_\ell,\ \ell\neq k,\ \ell=1,\ldots,K. \tag{36}$$

The service rate of the composite queue c is

$$r_c(i|S_c) = \frac{1}{n(c,k(i))}\cdot y_{U,k(i)}\, G_{U-1}(N-k)/G_{U-1}(N).$$

Since the CQ discipline satisfies station balance

$$p_c(X_c) = \frac{G_{U-1}(N_c)}{n_c!}\prod_{k=1}^{K} n(c,k)!\left(\frac{1}{y_{U,k}}\right)^{n(c,k)} \prod_{i=1}^{n(c)} (1-F_{c,k(c,i)}(X(c,i)))$$

It is easy to show

$$p_c(N_c) = G_{U-1}(N_c)\prod_{k=1}^{K}\left(\frac{1}{y_{U,k}}\right)^{n(c,k)}$$

Now

$$P_U'(n_1,\ldots,n_K) = p_U'(n_1,\ldots,n_K)\ \ p_c(N_1-n_1,\ldots,N_K-n_K),\ n_k=0,\ldots,N_k,\ k=1,\ldots,K$$

where

$$p_U'(n_1,\ldots,n_K) = p_U(n_1,\ldots,n_K)\prod_{k=1}^{K}\left(\frac{1}{y_{U,k}}\right)^{n(U,k)},\quad n_k=0,\ldots,N_k,\ k=1,\ldots,K.$$

Consequently

$$P_U'(n_1,\ldots,n_K) \propto p_U(n_1,\ldots,n_K)\prod_{k=1}^{K}\left(\frac{1}{y_{U,k}}\right)^{N_k} G_{U-1}(N_1-n_1,\ldots,N_K-n_K)$$
$$n_k=0,\ldots,N_k,\ k=1,\ldots,K.$$

This simplifies to

$$P_U'(n_1,\ldots,n_K) \propto p_U(n_1,\ldots,n_K)\ G_{U-1}(N_1-n_1,\ldots,N_K-n_K),$$
$$n_k=0,\ldots,N_k,\ k=1,\ldots,K.$$

From equation (31) we also know

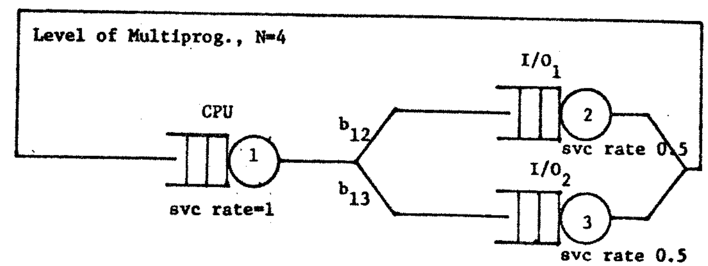
$$P_U(n_1,\ldots,n_K) \propto p_U(n_1,\ldots,n_K)\ G_{U-1}(N_1-n_1,\ldots,N_K-n_K),$$
$$n_k=0,\ldots,N_k,\ k=1,\ldots,K.$$

Therefore,

$$P'_U(n_1,\ldots,n_K) = P_U(n_1,\ldots,n_K),\ n_k=0,\ldots,N_k,\ k=1,\ldots,K.$$

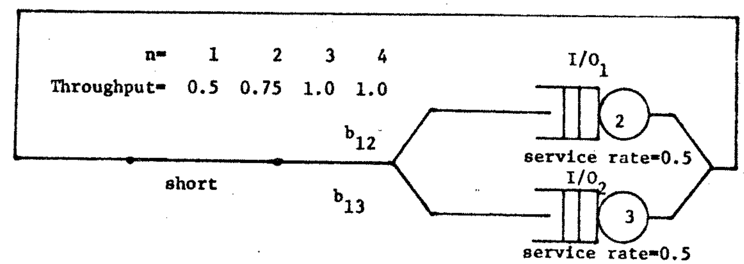
4.5.3 An Example

Consider the central server model shown in Figure 4.16 with a CPU and two I/O processors with FCFS disciplines and exponential service times. Let there be four identical customers using the I/O processors with the specified state dependent branching probabilities. The service times for both I/O's are 1. We are interested in the CPU utilization, queue length and throughput as the CPU service time is varied. We would like to solve a simpler two queue model. First, we determine the throughput of the network when the CPU service time is reduced to zero, Figure 4.17, with n customers in the system, n=0,..,N. The throughputs as a function of n are shown in a table in Figure 4.17. The equivalent network is shown in Figure 4.18, with the service rates of the composite queue set to the throughputs in Figure 4.17. The mean service time for the composite queue is set to one. As a result of Norton's Theorem, we know that the behavior of the CPU in the equivalent network is identical to that in Figure 4.16.



$$b_{12} = \frac{2-N_2}{4-N_2-N_3} \qquad b_{13} = 1-b_{12}$$

Central Server Model
Figure 4.16



n=	1	2	3	4
Throughput=	0.5	0.75	1.0	1.0

Levels of Multiprogramming, n=1, 2, 3, 4
Figure 4.17 Central Server Model (CPU Shorted)

n=	1	2	3	4
r(i|n)=	0.5	.375	0.33	0.25
R(n)=	0.5	.750	1.00	1.00

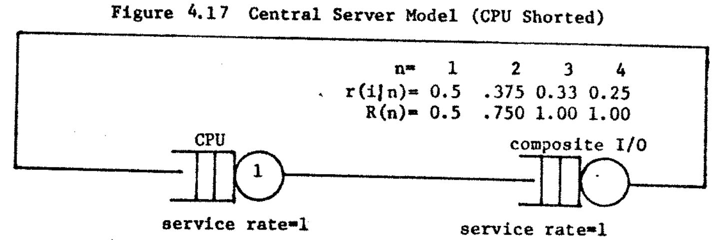


Figure 4.18 Central Server Model (Equivalent Network)

CHAPTER V

CENTRAL SERVER MODELS OF CPU-I/O AND I/O-I/O PROCESSING OVERLAP

5.1 Introduction

Central server network models are widely used in the analysis of computing systems [B1, B5, G2, S1]. A fixed number of customers (programs) traverse a closed network consisting of the central processor (CPU) and the input/output (I/O) devices. A customer alternatively receives service from the CPU and one of the I/O devices. After completing service at the CPU, a customer selects an I/O device with a probability which may depend on the state of all the I/O devices and the particular customer. The service time of a customer on a device depends on the customer, the device, and the queue lengths for that device. Figure 5.1 illustrates a central server network with 3 I/O devices.

The results of the last two chapters can be used to solve a large set of models. Approximate models with priority disciplines and FCFS disciplines for arbitrary generalized Erlang distributions can be solved rapidly but approximately with good accuracy [C3, S1]. All these models, though, neglect the capability for a program to overlap CPU and I/O processing or to overlap multiple I/O requests that exist in many present day computing systems. Let us broaden our description of a central server network to allow a customer to obtain ser-

vice at more than one device simultaneously.

In spite of the existing capability of many computing systems to allow CPU-I/O or I/O-I/O processing overlap, little work has been done to model such systems. The presently occurring development of such specialized devices as array processors and parallel CPU's increases the need for a capability to rapidly and accurately model the overlap of processing on several devices within a single job or program.

Hellerman and Smith [H1] considered a single-programmed system where all service times are constant to obtain optimal improvements for CPU-I/O processing and CPU-input-output (C-I-O) processing overlap. Cotten and Abd-Alla [C8] considered a single-programmed system with exponential service times to study the improvement of compute-output overlap processing over sequential processing. In neither case were the effects of overlap processing on a multiprogrammed system studied. The models developed in this chapter for CPU-I/O overlap will give exact solutions for two queue networks and good approximate solutions for arbitrary central server networks with more than two queues. The model developed for I/O-I/O overlap give good approximate solutions for arbitrary central server models. All models will assume only one customer class.

We consider a class of discrete state continuous transition ergodic Markovian models with finite state spaces. The equilibrium state probabilities can be determined from the balance equations for the state spaces [D1]. Let P(i) be the equilibrium probability of state i, i=1,...,M and $p_{i,j}$ be the rate at which the system transits from state i to j, i, j=1,...,M. Then the balance equation for state i, i=1,...,M will be

$$\sum_{j=1}^{M} P(i)p_{i,j} = \sum_{j=1}^{M} P(j)p_{j,i} \quad .$$

Because the above set of M equations are dependent, it is necessary to substitute the normalization criterion

$$P(1)+\ldots+P(M) = 1$$

for one of the balance equations. We can solve the resulting set of equations for the equilibrium state probabilities. From the P(i) we can determine model statistics such as device utilization, throughput, queue length, and wait time. In general, the solution of these models by direct numerical techniques may require excessive memory and computation requirements. For example, if a central server network has U queues and N jobs, the CPU-I/O overlap model and I/O-I/O overlap model will generate $(U+1)\binom{U+N+1}{N}$ and $\sum_{n=1}^{N} n[\binom{U+2N}{2N}+\binom{U+2N-n}{2N-n}]$ states respectively. Direct solution of a network with 8 queues and 8 jobs, requires solving a system of $\sim 10^4$ equations for the CPU-I/O overlap model, and $\sim 10^6$ equations for the I/O-I/O overlap model.

Herzog, Chandy, and Woo [H2] have developed a general approach to the numerical solution of Markovian models based on the structure of the state space. Many of the problems possess the following feature. There exists a subset of the state probabilities

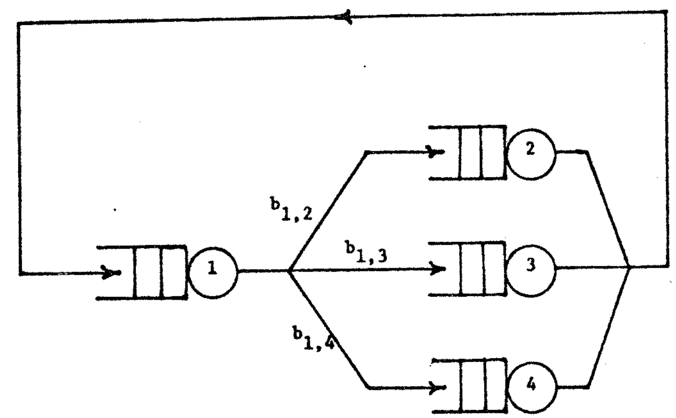


Figure 5.1

which they refer to as boundaries, and if their values are known, the solution of the rest of the probabilities can be carried out recursively with great efficiency. Chandy, Herzog, and Woo [C3], and Sauer [S1] use this approach to develop algorithms for determining approximate solutions to central server models exhibiting various non-local balance features (priority scheduling, non-exponential distributions).

We illustrate this approach for a very simple model in section 5.2. The next two sections develop algorithms based on the exact solution of one customer class two queue models with CPU-I/O and I/O-I/O overlap processing capabilities. These algorithms will form the basis of approximate solution algorithms for central server networks with overlap processing. In sections 5.5 and 5.6 we compare results of our techniques with results of simulations and actual trace data from a CDC 6400 system. Finally, in section 5.7 we consider the utility of CPU-I/O overlap in different multiprogramming environments.

5.2 Two Exponential Queues with No Overlap

Consider a system with two FCFS queues, one customer class, and exponentially distributed service times at each queue. After completing service at one queue, a customer proceeds to the other queue for service. Let the mean service time for queue 1 be $1/\mu$ and the mean service time for queue 2 be $1/\lambda$. Let the number of customers in the system be N. This system is depicted in Figure 5.2. The state of the system can be determined by the number of customers in the first queue. Let P(i) be the probability that the state of the system has i customers in queue 1, i=0,1,...,N. The state transitions for

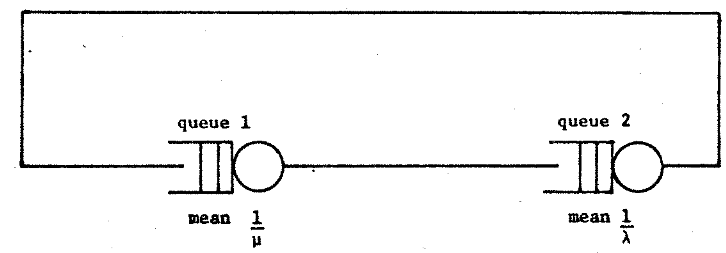


Figure 5.2

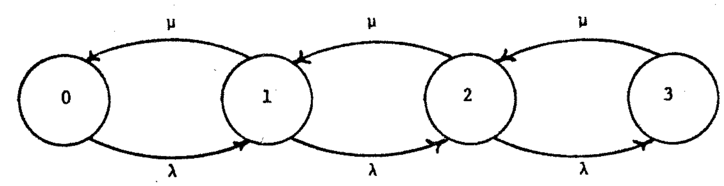


Figure 5.3