Measuring, Modelling and Evaluating Computer Systems,
H. Beilner and E. Gelenbe, (eds.)


# OPERATIONAL ANALYSIS OF QUEUEING NETWORKS[1]

Peter J. Denning[2]

Jeffrey P. Buzen[3]

The first step in analyzing a queueing network model is to obtain a set of equations which express $p(\underline{n})$, the steady state distribution of customers in the network, in terms of basic network parameters such as mean service times or routing probabilities. When applying these equations, the analyst usually identifies $p(\underline{n})$ with an operational (i.e., directly measurable) quantity, the proportion of time the system spends in state $\underline{n}$. The analyst also identifies network parameters with operational quantities; for example, he uses measured average service times as values for stochastic means, or relative transition frequencies for routing probabilities. In this paper, we show that the equations relating the operational values of $p(\underline{n})$ to the operational values of the queueing network parameters are considerably more general than in Markovian queueing network theory. In operational queueing network theory, these equations depend only on four assumptions: the number of jobs which are observed to arrive at a given device is (almost) the same as the number observed to depart; the number of transitions into a given system state is (almost) the same as the number out; the number simultaneous interdevice transitions is negligible; the on-line service functions of devices are (almost) the same as the off-line service functions. The last assumption, called "homogeneity", is the major approximation, on account of which queueing network results are not exact. It is closely related to the principle of decomposability.

## 1 INTRODUCTION

### 1.1 Background

Queueing networks have become a widely used analytic tool for multiple resource computer system performance studies. The theoretical results have been known for a long time. In 1957, Jackson published a paper showing the analysis of a multiple device system wherein each device contained one or more parallel servers and new jobs could enter or exit the system at any device [JACK57]. In 1963 Jackson extended his analysis to open and closed systems with arbitrary queue dependent service rates at all devices in the system [JACK63]. In 1967, Gordon and Newell simplified the notational structure of these results for the special case of closed systems, wherein the number of jobs was held fixed [GORD67]. In 1971, Buzen showed how to apply these models to computer systems [BUZE71]; he

(1) Supported in part by NSF Grant GJ-41289 at Purdue University.

(2) Computer Sciences Dept., Purdue University, W. Lafayette, IN 47907 USA.

(3) BGS Systems, Inc., Box 128, Lincoln, MA 01773 USA.

also developed efficient procedures for calculating performance quantities from these models [BUZE73]. Extensive validation since 1971 has verified that these models predict observed performance quantities with remarkable accuracy [BUZE75, GIAM76].

Most analysts have expressed puzzlement at the accuracy of queueing network models. The traditional approach to deriving them depends on a series of concepts from the theory of stochastic processes; for example:

- The system is modeled by a stationary stochastic process;
- Jobs are stochastically independent;
- Transitions from device to device follow a Markov Chain;
- The system is in stochastic equilibrium;
- The service time requirements at each device follow an exponential distribution; and
- The system is ergodic -- i.e., long term time averages converge to the mean values computed for stochastic equilibrium.

The theory of queueing networks based on these assumptions is usually called "Markovian queueing network theory" [KLEI76a]. The underlined words in this list of assumptions illustrate concepts that the analyst must understand to be able to use the models confidently. Some of these concepts are difficult. Others can be disproved empirically -- for example, system parameters change over time, jobs are dependent, device to device transitions do not follow Markov chains, systems are observable only for short intervals, service distributions are seldom exponential. It is no wonder that many people are surprised that these models succeed, when applied to systems that violate so many assumptions of the analysis!

Operational analysis explains these observations by showing a much weaker set of assumptions on which the validated results rely. (See BUZE76a,b,c;DENN75.)

## 1.2 Typical Form of Validations

Let $i = 1,\ldots,K$ denote a device in the system, $n_i$ denote the number of jobs present at the $i^{th}$ device, and $\underline{n} = (n_1,\ldots,n_K)$ denote a "state" of the system. In general, $\underline{n}$ changes over time as jobs move among the devices, or enter and exit the system. Let $p(\underline{n})$ denote the proportion of time during which the state is observed to be $\underline{n}$; the $p(\underline{n})$ sum to 1 over all possible values of $\underline{n}$.

An analyst normally uses a model -- whether simulation or analytic -- to define a method for computing, in terms of workload and device parameters, either $p(\underline{n})$ or quantities derived from $p(\underline{n})$. Three important derived quantities are the queue distributions, the mean queue lengths, and the device utilizations. The queue distribution $p_i(n)$ for device i measures the proportion of time $n_i = n$:

$$p_i(n) = \sum_{\underline{n},\, n_i=n} p(\underline{n}) .$$

The mean queue length at device i is

$$\bar{n}_i = \sum_{n>0} n\, p_i(n) .$$

The utilization of device i is the proportion of time $n_i > 0$:

$$U_i = \sum_{n>0} p_i(n) .$$

In a typical validation, the analyst will use physical properties of the devices, together with empirical data on request sizes, to determine the mean service time for one task at a device. He will use empirical data on the workload to determine how often jobs generate tasks for the various devices. He will use the model, applied to these parameters, to compute values for quantities like $U_i$ and $\bar{n}_i$. If, over many different observation periods, these computed values compare well with actual (measured) values, he will conclude that the model is good. (See Figure 1.) Thereafter, he will employ it confidently for predicting future behavior or evaluating proposed changes in the system.

The important observation is that most practical validations interpret model $p(\underline{n})$ as proportions of time rather than as probabilities. Though stochastic assumptions are sufficient to calculate the $p(\underline{n})$, they are stronger than needed for this purpose.

Three simple, operational, assumptions define weak conditions under which $p(\underline{n})$ can be computed from device and workload parameters:

- All quantities should be precisely measurable in finite observation periods -- the precision of results should not depend on an assumption of "stationarity" or "steady state".
- The system must be work conservative -- i.e., the number of entries to a given device (or system state) must be (almost) the same as the number of exits from that device (state) during the observation period.
- The system must be homogeneous -- i.e., the mean service time of each device for given queue length is the same whether the device is on line or off line. (When a device is off line, its output rate for given queue length is measured by subjecting it to constant load.)

Our interest in this paper is showing how the operational assumptions are employed to set up the familiar "local balance equations" of queueing network analysis. The usual product form solutions and computational procedures are then

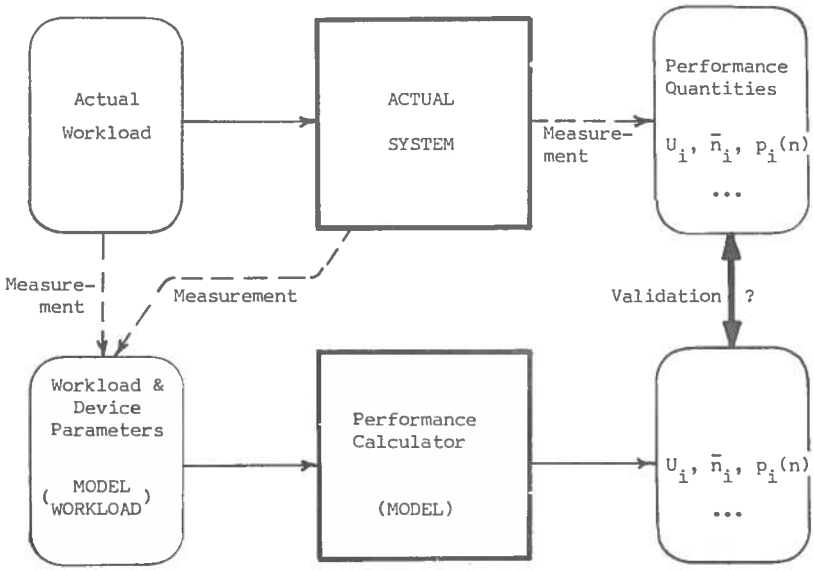


FIGURE 1. Typical validation scheme.

applicable. The conclusion is that the $p(\underline{n})$, and quantities derived from them, actually depend only on the operational assumptions, which are weaker than the stochastic ones traditionally used.

The weaker assumptions of operational queueing network theory allow the $p(\underline{n})$ to be interpreted only as proportions of time. The stronger assumptions of Markovian queueing network theory are required to answer questions in which the $p(\underline{n})$ are interpreted as probabilities. The limitations of operational analysis are discussed at the end of the paper.

## 2 OPERATIONAL QUANTITIES IN NETWORKS

### 2.1 Basic Device and Routing Measures

Figure 2 shows two of the K devices in a multiple resource network. A device may depend on load to the extent that its work completion rate is a function of $n_i$, the number of jobs present there. All jobs of this system are of one class -- i.e., they exhibit similar patterns of demand. A job enters the system at the

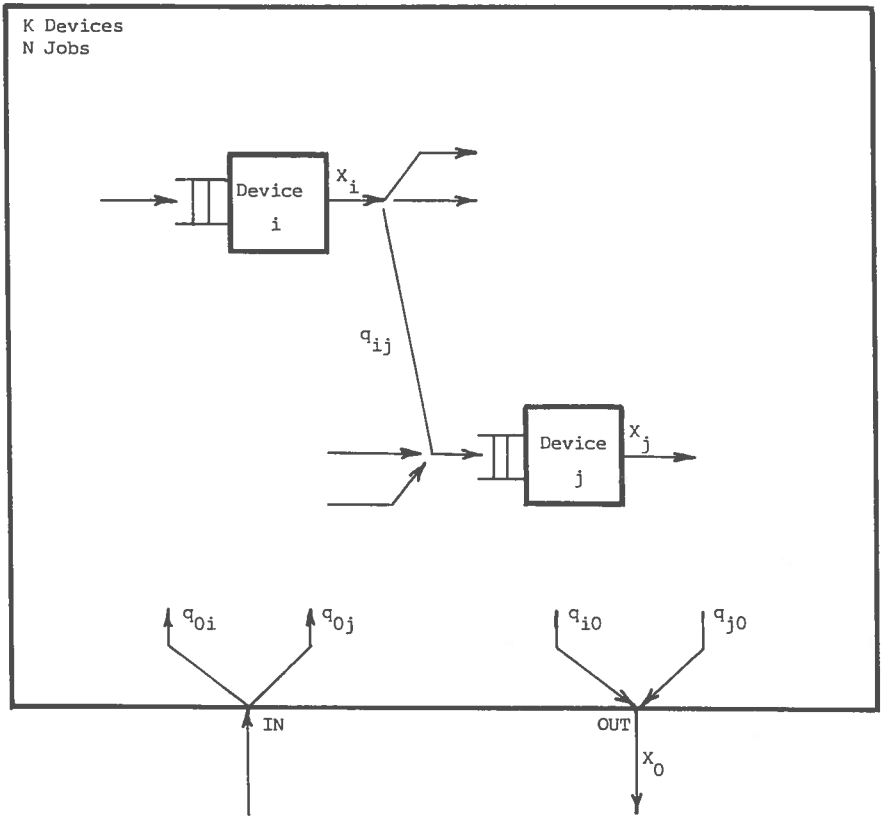


FIGURE 2. Portion of a queueing network.

point 'IN'; whereupon it circulates through the network, waiting in queues and having service requests processed at various devices; when done, it exits at 'OUT'.

The model assumes no job overlaps its use of different devices. In practice, few applications ever achieve more than 2 or 3 per cent overlap between central processor (CPU) and input/output (I/O) devices: the error introduced by this model assumption is usually not significant.

If $n_i$ is the number of jobs present at device i, then $N = n_1+\ldots+n_K$ is the total in the system. If N is fixed, the system is closed; this is modeled by connecting the output back to the input. The system output rate, $X_0$, is the number of jobs per unit time leaving the system; it is a function of N.

Suppose the system is observed for a time interval [0,T], wherein these data are collected (i = 1,...,K):

$A_i(n)$, number of arrivals at device i when $n_i = n$;[+]

$C_{ij}(n)$, number of times a job requests service at device j immediately after completing a service request at device i, when $n_i=n$;[+] and

$T_i(n)$, total time during which $n_i = n$.

If we treat the "outside world" as device "0" we can define also

$C_{0i}(n)$, number of jobs whose first service request is for device i when $N = n$;[+] and

$C_{i0}(n)$, number of jobs whose last service request is for device i when $n_i = n$.[+]

Note that $C_{00}(n) = 0$ for all n. The number of completions at device i is computed as

$$C_i(n) = \sum_{j=0}^{K} C_{ij}(n) , \qquad i = 1,\ldots,K.$$

The number of arrivals to the system when N=n is

$$A_0(n) = \sum_{i=1}^{K} C_{0i}(n) .$$

The method of partitioning the data according to time intervals in which $n_i=n$ is called stratified sampling. The sets of intervals in which $n_i=n$ are sometimes called the "strata" of the sample. This technique aggregates data in the same stratum.

In terms of the (stratified) data, these operational quantities are defined:

$X_i(n)$, request completion rate from device i when $n_i=n$, $X_i(n)=C_i(n)/T_i(n)$

$p_i(n)$, proportion of time when $n_i=n$, $p_i(n)=T_i(n)/T$

$S_i(n)$, mean service time when $n_i=n$, $S_i(n)=T_i(n)/C_i(n)$

(None of these quantities is defined if its denominator is 0.) Define the total number of completions at device i to be

$$C_i = \sum_{n>0} C_i(n) ,$$

and the overall request completion rate of device i to be

$$X_i = C_i/T .$$

[+]More precisely, these counters register the number of times t at which the given event occurred, such that $n_i(t^-) = n$, or $N(t^-) = n$.

It is easily verified from the definitions that

$$X_i = \sum_{n>0} p_i(n)\, X_i(n) .$$

Define the total busy time of device i to be

$$B_i = \sum_{n>0} T_i(n) .$$

The mean service time over all tasks completed at device i is

$$S_i = B_i/C_i .$$

These definitions imply the operational utilization formula:

$$U_i = X_i S_i , \qquad i = 1,\ldots,K.$$

(See also BUZE76c.)

Let $J_i$ denote the total job-seconds accumulated at device i, that is,

$$J_i = \sum_{n>0} n\, T_i(n) .$$

Two more operational quantities follow:

$\bar{n}_i$ = mean queue length, $\bar{n}_i = J_i/T$

$R_i$ = mean response time of a request, $R_i = J_i/C_i$

These definitions imply the operational Little's Formula:

$$\bar{n}_i = R_i X_i , \qquad i = 1,\ldots,K.$$

(See also BUZE76c.)

In the special case of a load independent system, the load parameter, n, can be dropped from the service times and work rates; thus $S_i(n) = S_i$, and $X_i(n) = X_i$. In this case, data collection is simpler because the data do not need to be stratified.

Congestion in a queueing network depends not only on the service functions $S_i(n)$ of devices, but also on the frequencies at which jobs generate tasks for the devices. We define a routing frequency as

$$q_{ij} = \frac{1}{C_i} \sum_{n>0} C_{ij}(n) ,$$

which is the fraction of the completions at device i which are followed immediately by requests for device j. In most cases the routing frequencies depend only on intrinsic job characteristics; they are independent of queue lengths. Thus quantities like $q_{ij}(n) = C_{ij}(n)/C_i(n)$ are of no interest. In some systems, the routing frequencies depend on the total load, N; for example, the relative frequency of swapping requests will increase as N increases in a multiprogrammed memory fixed in size [DENN76]. We will not consider this case further here.

## 2.2 On Line and Off Line Behavior

The method of stratified sampling defines a (load dependent) service function, $S_i(n)$, for each device i. It is defined so that $X_i(n) = 1/S_i(n)$ is the number of tasks per unit time leaving device i, over all time periods in which $n_i = n$. We call this the on line service function of the device.

The analyst can also measure an off line service function, $S_i^*(n)$. He does this with a "constant load" controlled experiment -- in which, for given n, he maintains $n_i=n$. The rule of the experiment is, simply, that a new job of the given class is added to the device's queue just after a previous job completes service. If, during T seconds of such an experiment, the analyst observes C jobs leaving the device, he assigns

$$S_i^*(n) = T/C .$$

Off line behavior is often easier to determine than on line behavior because, off line, the device is isolated from possible interactions with the rest of the system. Off line behavior can often be determined from simple analysis or simulation. Analysts frequently use off line characteristics as approximations to the true behavior when a device is on line.

The concept of off line behavior can be extended to an entire subsystem. We will return to this in the section on decomposability.

# 3 JOB FLOW AND BOTTLENECK ANALYSIS

Suppose that we know the overall mean service times ($S_i$) and the routing frequencies ($q_{ij}$); how much can we determine about overall device output rates ($X_i$)? This question is usually approached through the approximation known as the

> Principle of Job Flow Balance. For each device i, $X_i$ is the same as the total input rate to device i.

This principle will give a good approximation when the difference between arrivals and completions, $A_i - C_i$, is small compared to $C_i$. When it holds, we refer to the $X_i$ as device throughputs. Expressing it as an equation,

$$C_j = A_j = \sum_{i=0}^{K} C_{ij} \qquad j = 0,\ldots,K .$$

(The dependence of $C_{ij}$ and $A_i$ on $n_i$ has been removed by summing over all observed values of $n_i$.) The definition $q_{ij} = C_{ij}/C_i$ allows writing

$$C_j = \sum_{i=0}^{K} C_i \, q_{ij} .$$

Employing the definition $X_i = C_i/T$, we obtain

**Job Flow Balance Equations**

$$X_j = \sum_{i=0}^{K} X_i \, q_{ij} \qquad j = 0,\ldots,K$$

If the network is open, $X_0$ will have a value determined by the environment and these equations will have a unique solution for the unknowns $X_i$. However, if the system is closed, $X_0$ is unknown and the equations have no unique solution; it is easy to verify that the sum of the $X_j$-equations for $j = 1,\ldots,K$ reduces to the equation for $j=0$. In a closed network, there are K independent equations and K+1 unknowns.

Even when the job flow equations cannot be solved for a unique set of $X_i$, they still contain information of considerable value. Define

$$V_i = X_i/X_0 ,$$

which is the job flow through device i relative to the system throughput. Our definitions imply that $V_i = C_i/C_0$, which is the number of completions at device i for each completion at the system: $V_i$ is the mean number of requests per job for device i. We refer to $V_i$ as the visit count of a for device i. Substituting into the job flow balance equations, we obtain the

**Job Visit Count Equations**

$$V_0 = 1$$

$$V_j = q_{0j} + \sum_{i=1}^{K} V_i \, q_{ij} \qquad j = 1,\ldots,K$$

A unique solution of these equations is always possible. If $X_0$ is known, we can compute $X_i = V_i X_0$.

The solution of the $p(\underline{n})$ of a queueing network will, as we shall see, require knowledge of the visit counts, $V_i$, and of the service functions, $S_i(n)$. The routing frequencies are used in the proofs to show that this is so. In practice, the analyst needs only to extract the K visit counts from workload data, rather then as many as $(K+1)^2$ values of $q_{ij}$.

If, besides job flow balance, we assume that the service time and routing parameters are all independent of load, we can prove that system throughput $X_0$ increases in N toward the asymptote 1/W, where

$$W = \max\{V_1S_1,\ldots,V_KS_K\} .$$

This property was first observed by Buzen for the special case of central server networks with exponential service times [BUZE71]. It was shown to hold under very general conditions by Chang and Lavenberg [CHAN72]. Muntz and Wong used it in bottleneck analysis of general queueing networks, to compute response time asymptotes and to evaluate effects of device speed-up [MUNT74; also DENN75, KLEI76b, MUNT75].

## 4 SOLUTIONS FOR STATE OCCUPANCIES

### 4.1 State Transition Balance

Let $T(\underline{n})$ denote the total time during which state $\underline{n} = (n_1,\ldots,n_K)$ is observed in a network over an interval [0,T]; the $T(\underline{n})$ sum to T over all $\underline{n}$. The time proportion for $\underline{n}$ is $p(\underline{n}) = T(\underline{n})/T$.

In the following discussion, $\underline{k}$, $\underline{n}$, and $\underline{m}$ denote distinct system states. Let $Q(\underline{n},\underline{m})$ denote the number of one-step transitions observed from $\underline{n}$ to $\underline{m}$; since the system's remaining in a state is not counted as a transition, $Q(\underline{n},\underline{n}) = 0$. We make the approximation,

> Principle of State Transition Balance. The number of entries to every state is the same as the number of exits from that state during the observation period.

With this, we can write "conservation of transition" equations:

$$\sum_{\underline{k}} Q(\underline{k},\underline{n}) = \sum_{\underline{m}} Q(\underline{n},\underline{m}) , \qquad \text{all } \underline{n} .$$

The only error in these equations is a +1 (or -1) term missing on the right side if $\underline{n}$ is the final (or initial) state of the system for the observation period. This error is not significant if the initial and final states are visited frequently; it is zero if the initial and final states are the same. For given $\underline{n}$ both sides of the equation are zero if and only if $T(\underline{n}) = 0$.

The transition rate from $\underline{n}$ to $\underline{m}$ is the number of transitions per unit time $\underline{n}$ is occupied:

$$H(\underline{n},\underline{m}) = Q(\underline{n},\underline{m})/T(\underline{n}), \qquad T(\underline{n}) \neq 0;$$

it is not defined if $T(\underline{n}) = 0$. The conservation equations can be reexpressed as

$$\sum_{\underline{k}} T(\underline{k})\, H(\underline{k},\underline{n}) = T(\underline{n}) \sum_{\underline{m}} H(\underline{n},\underline{m}) ,$$

for all $\underline{n}$ in which $H(\underline{n},\underline{m})$ is defined; note $T(\underline{n})=0$ when $H(\underline{n},\underline{m})$ is not defined. If we substitute $T(\underline{n}) = p(\underline{n})T$ and cancel T, we obtain the

State Space Balance Equations

$$\sum_{\underline{k}} p(\underline{k})\, H(\underline{k},\underline{n}) = p(\underline{n}) \sum_{\underline{m}} H(\underline{n},\underline{m})$$

for all $\underline{n}$ in which each $H(\underline{n},.)$ is defined.

Because the $T(\underline{n})$ sum to T, we can augment these equations with the normalizing condition

$$\sum_{\underline{n}} p(\underline{n}) = 1 ,$$

which will guarantee that only one set of $p(\underline{n})$ can satisfy them. (Our definitions imply $p(\underline{n}) = 0$ for states $\underline{n}$ not included in the balance equations.)

## 4.2 Solving the Balance Equations

The state space balance equations are nothing more than algebraic identities on the operational definitions of $p(\underline{n})$ and $H(\underline{n},\underline{m})$. An analyst would hardly use these equations to "solve" for the $p(\underline{n})$. He would instead express the $H(\underline{n},\underline{m})$ in terms of device and workload parameters, and seek a unique solution for the $p(\underline{n})$ in terms of these parameters.

The system state space contains a large number, L, of possible $\underline{n}$ values. If N is the maximum number of jobs ever observed in any queue in the system, L may be as large as $(N+1)^K$ in an open system, and as large as $\binom{N+K-1}{K-1}$ in a closed system.

To render the balance equations more manageable, analysts often make this assumption:

> One Step Behavior. The only observable state changes result from single jobs either entering the system, or moving between pairs of devices in the system, or exiting from the system.

This assumption reduces the number of nonzero transition rates to about $K^2$ in a load-independent system, and to about $NK^2$ in a load-dependent system. In most computer modeling, this assumption introduces little or no error. Let

$$\begin{aligned} \underline{n}_{ij} &= (n_1,\ldots,n_i+1,\ldots,n_j-1,\ldots,n_K) \\ \underline{n}_{i0} &= (n_1,\ldots,n_i+1,\ldots,n_K) \\ \underline{n}_{0j} &= (n_1,\ldots,n_j-1,\ldots,n_K) \end{aligned}$$

denote states which are "neighbors" of $\underline{n}$ relative to the one step assumption. Under this assumption, the state space balance equations reduce to (for all $\underline{n}$):

$$\sum_{i,j} p(\underline{n}_{ij})H(\underline{n}_{ij},\underline{n}) + \sum_{i} p(\underline{n}_{i0})H(\underline{n}_{i0},\underline{n}) + \sum_{j} p(\underline{n}_{0j})H(\underline{n}_{0j},\underline{n})$$

$$= p(\underline{n})\left(\sum_{i,j} H(\underline{n},\underline{n}_{ji}) + \sum_{i} H(\underline{n},\underline{n}_{0i}) + \sum_{j} H(\underline{n},\underline{n}_{j0})\right)$$

The first terms on left and right correspond to jobs making (i,j) transitions within the system; the second terms on left and right correspond to jobs exiting the system from device i; the third terms on left and right correspond to jobs entering the system at device j. All sums on i and j use values 1,...,K. (For a closed system, the second and third terms on left and right are dropped, and $q_{ij}$ is increased by $q_{i0}q_{0j}$.) Relative to the one step assumption, these equations are algebraic identities over the $p(\underline{n})$ and $H(\underline{n},\underline{m})$.

To obtain solutions of these equations from device and workload parameters, analysts frequently use routing frequencies and off line device characteristics to determine the transition rates. Substituting the off line characteristics for the on line is a major approximation. In doing it, the analyst is asserting

> Homogeneity. The off line service function, $S_i^*(n)$, of each device i is the same as its on line service function, $S_i(n)$.

The substitutions implied by this assumption are summarized in Table I. We have defined the binary indicator variable, $I_i$, to be 1 when $n_i > 0$ and 0 when $n_i = 0$; this variable sets transition rates between pairs of states to zero when one of the states is illegitimate. Under the substitutions of Table I, together with the

identities $q_{01}+\ldots+q_{0K} = 1$ and $q_{i0}+q_{i1}+\ldots+q_{iK} = 1$, the balance equations reduce to

Homogenized Balance Equations

$$\sum_{i,j} p(\underline{n}_{ij}) \frac{q_{ij}I_j}{S_i(n_i+1)} + \sum_i p(\underline{n}_{i0}) \frac{q_{i0}}{S_i(n_i)} + \sum_j p(\underline{n}_{0j})X_0 q_{0j} I_j = p(\underline{n}) \left( \sum_i \frac{I_i}{S_i(n_i)} + X_0 \right), \quad \text{all } \underline{n}$$

These equations are identical in form to the "local balance equations" of Markovian queueing networks [KLEI76a]. The analyst can solve them for the $p(\underline{n})$ without measuring the state space. Note that the solution is exact if the assumptions of job flow balance, state transition balance, one step behavior, and homogeneity are all precisely satisfied. In practice, these assumptions may

Table I. Homogeneous Transition Rates.

| Type of Job Transition | Type of State Transition | Homogeneous Rate |
|---|---|---|
| $i \to j$ | $\underline{n}_{ij} \to \underline{n}$ | $H(\underline{n}_{ij},\underline{n}) = q_{ij}I_j/S_i(n_i+1)$ |
| | $\underline{n} \to \underline{n}_{ji}$ | $H(\underline{n},\underline{n}_{ji}) = q_{ij}I_i/S_i(n_i)$ |
| $i \to 0$ | $\underline{n}_{i0} \to \underline{n}$ | $H(\underline{n}_{i0},\underline{n}) = q_{i0}/S_i(n_i+1)$ |
| | $\underline{n} \to \underline{n}_{0i}$ | $H(\underline{n},\underline{n}_{0i}) = q_{i0}I_i/S_i(n_i)$ |
| $0 \to j$ | $\underline{n}_{0j} \to \underline{n}$ | $H(\underline{n}_{0j},\underline{n}) = X_0 q_{0j} I_j$ |
| | $\underline{n} \to \underline{n}_{j0}$ | $H(\underline{n},\underline{n}_{j0}) = X_0 q_{0j}$ |

only be satisfied approximately, whereupon the solution may only be approximate. Homogeneity is the assumption most likely to be violated in practice. Experience has been good: errors are usually small, the assumptions reasonable.

As shown by Jackson [JACK63], the solution of the homogenized balance equations is of the "product form"

$$p(\underline{n}) = \frac{1}{G} \prod_{i=1}^{K} F_i(n_i).$$

The term corresponding to device i is

$$F_i(n) = \begin{cases} 1, & n = 0 \\ X_i S_i(n) F_i(n-1), & n > 0 \end{cases}$$

The $X_i$ are a solution of the job flow balance equations and G is a normalizing constant. (See COFF73, GELE76, KLEI76a.) Efficient procedures are available for computing G and the queue distributions $p_i(n)$ [BUZE73, GELE76].

Our assumptions -- queueing network connectedness, job and state flow balance, and homogeneity -- imply a nonzero transition rate in and out of every possible state $\underline{n}$ of the network. The model will therefore assign nonzero values to all $p(\underline{n})$ even though the actual system may not enter all its possible states. The model of a closed system thus determines $\binom{N+K-1}{K-1}$ values of $p(\underline{n})$; the model of an open system, with a maximum of N jobs observable in any queue, determines $(N+1)^K$ values of $p(\underline{n})$.

Assuming a maximum of N jobs in any queue of an open system, the normalizing constant can be expressed as a product of normalizing constants:

$$G = \sum_{n_1=0}^{N} \cdots \sum_{n_K=0}^{N} \prod_{i=1}^{K} F_i(n_i) = \prod_{i=1}^{K} \sum_{n_i=0}^{N} F_i(n_i) = \prod_{i=1}^{K} G_i$$

Now: the solution of a network containing only device i, and having throughput $X_i$, is

$$p_i(n_i) = F_i(n_i)/G_i \qquad G_i = \sum_{n_i=0}^{N} F_i(n_i)$$

(See also BUZE76a,b.) This implies that, for an open system,

$$p(\underline{n}) = \prod_{i=1}^{K} p_i(n_i).$$

In other words, $p(\underline{n})$ is the product of the (marginal) queue distributions of the devices, the marginal distribution being determined as if the device were off line with job flow $X_i$ identical to the job flow it experiences on line. This is the operational counterpart of Jackson's Theorem [JACK63; also GELE76]. No similar property holds for closed networks.

## 4.3 An Example

Figure 3 illustrates a simple system with K=2 and N=2. The timing diagram shows a possible behavior that can be observed. The numbers within the diagram

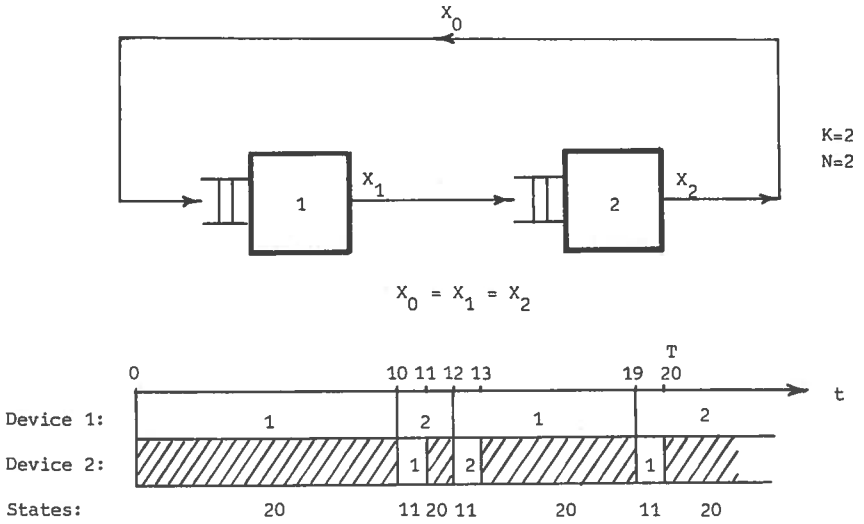


FIGURE 3. Two device system and observed behavior.

show which job is using the device, and shaded portions indicate idleness. The observed states ($n_1 n_2$) are shown below the timing diagram. The devices are load independent. The observation period is [0,20].

We will compare the model solutions with the actual behavior of this system. The basic operational quantities are

$$S_1 = B_1/C_1 = 20/3 \qquad U_1 = B_1/T = 1 \qquad X_1 = C_1/T = 3/20$$

$$S_2 = B_2/C_2 = 1 \qquad U_2 = B_2/T = 3/20 \qquad X_2 = C_2/T = 3/20$$

The proportions of time of state occupancy are

$$p(20) = T(20)/T = 17/20 \qquad p(11) = T(11)/T = 3/20$$

The transition rates are

$$H(20,11) = Q(20,11)/T(20) = 3/17$$

$$H(11,20) = Q(11,20)/T(11) = 1$$

The balance equations are

$$p(20)(3/17) = p(11)(1)$$
$$p(11)(1) = p(20)(3/17)$$
$$p(11) + p(20) = 1$$

It is easily verified that the observed $p(\underline{n})$ satisfy these equations.

The system is not homogeneous. Homogeneity assigns transition rates as follows:

$$H(20,11) = 1/S_1 = 3/20 \qquad H(11,20) = 1/S_2 = 1$$
$$H(11,02) = 1/S_1 = 3/20 \qquad H(02,11) = 1/S_2 = 1$$

These rates allow state 02 to be occupied, which is not observed in the actual system. The balance equations become

$$p(11)(1) = p(20)(3/20)$$
$$p(20)(3/20) + p(02)(1) = p(11)(1 + 3/20)$$
$$p(11)(3/20) = p(02)(1)$$
$$p(20) + p(11) + p(02) = 1$$

For which the solution is

$$p(20) = 400/469 \qquad p(11) = 60/469 \qquad p(02) = 9/469$$

This solution differs from the observed $p(\underline{n})$. The predicted utilizations are:

$$U_1 = p(20) + p(11) = 460/469$$
$$U_2 = p(11) + p(02) = 69/469$$

which yield $X_1 = X_2 = U_1/S_1 = 69/469$. The error between these predictions are the true values is under 2%: homogeneity enabled a solution agreeing closely with the observations.

Since $X_0 = X_1 = X_2$, the visit counts are $V_1 = V_2 = 1$. The product form solution specifies

$$p(n_1n_2) = (V_1S_1)^{n_1}(V_2S_2)^{n_2}/G = (20/3)^{n_1}(1)^{n_2}/G = (20/3)^{n_1}/G$$

where

$$G = (20/3)^0 + (20/3)^1 + 20/3)^2 = 469/9.$$

Then, as before,

$$p(20) = (20/3)^2/G = 400/469$$
$$p(11) = (20/3)^1/G = 60/469$$
$$p(02) = (20/3)^0/G = 9/469 .$$

## 5 DECOMPOSABILITY

The concept of decomposability is often used to simplify the analyses of stochastic processes that model real systems. The concept is straightforward. If a subsystem interacts weakly with its environment, the transient behavior of the subsystem (following an interaction with the environment) will have little effect on the long run dynamics of the total system. Thus, very little error will be introduced by supposing that the subsystem is in equilibrium for the entire interval between two interactions with the environment. The principle of decomposability allows an analyst to decouple a subsystem from its environment, determine its equilibria in isolation, then substitute the equilibria for the true behaviors when the subsystem is embedded in its environment. It is a powerful approximation tool. (See COUR75; COUR77.)

Operationally, decomposability allows an analyst to begin an analysis of a complex system by studying one of its subsystems in isolation. To do this, he subjects the subsystem in question to a series of controlled experiments. In one of these experiments, the subsystem is operated under a constant load (n jobs of the given type) for some time interval of length T. Immediately after each completion in the controlled experiment, the analyst adds another job, to keep the load equal to n. The analyst counts the number of completions, C, and assigns S(n) = T/C. The subsystem is then replaced by a single load dependent device of service function S(n), thereby simplifying the overall problem. If indeed the subsystem interacted weakly with its environment, the principle of decomposability holds that the marginal distribution $p_i(n_i)$ of any device in the environment will not be significantly affected by this replacement.

Operationally, decomposability asserts that off line behavior of a subsystem or device is nearly the same as its on line behavior: interactions are too weak to alter the off line behavior substantially. The homogeneity assumption is nothing more than an assertion of perfect decomposability.

Chandy, Herzog, and Woo proved a theorem for systems whose $p(\underline{n})$ satisfy the "local balance equations" (homogenized balance equations) [CHAN75]. Their theorem implies that, under local balance, a subsystem can be replaced by a single load dependent device, whose service function is obtained by studying the subsyst

off line; this replacement has no effect on the marginal distribution $p_i(n_i)$ of any device outside the subsystem. Actually, this theorem only depends on the product form of $p(\underline{n})$; consequently it works in the operational case as well. In other words, a network of homogeneous devices is itself homogeneous relative to the environment in which it is embedded.

## 6 LIMITATIONS OF OPERATIONAL ANALYSIS

Using a relatively weak set of assumptions, operational analysis makes it possible to derive the proportions of time $p(\underline{n})$ a queueing network occupies each state $\underline{n}$, when only the empirical mean service functions of devices and the job visit counts are known. To the extent that operational assumptions resemble practical conditions more closely than Markovian assumptions, they explain the success of typical queueing network validations. To the extent that operational assumptions are intuitive, more analysts can use the queueing network models with confidence and understanding.

Operational queueing network theory has been well validated. All the reported queueing network validations use measured mean service time functions and measured job visit counts to compute $p(\underline{n})$, which are then compared against actual proportions of time the system spends in state $\underline{n}$. (See, e.g., BUZE75, GIAM76.) Analysts have, all along, been validating operational analyses.

The operational results of this paper need not produce exact answers. This is because the principles of job flow balance, state transition balance, one step behavior, and homogeneity are not met exactly in actual systems during finite intervals. The error introduced by assuming that the first three principles are exact is generally not significant. The greatest error is introduced by the homogeneity principle. In practice, devices do interact; their on line service functions, measured by stratified sampling, may differ significantly from their service functions measured off line under fixed load. Homogeneity predicts that all model states will be occupied, even if some actual system states are not. The principle of homogeneity can be extended, with improvements in accuracy. Shum and Buzen, for example, show how off line characteristics of M/G/1 systems can be used to determine on line service functions with improved accuracy [SHUM77]. Cox's method of stages can be used to deal with very general service distributions in an operational context. (See BASK75, GELE76, KLEI76a.)

Operational assumptions do restrict the set of questions that can be answered about queueing networks. These assumptions produce a theory of queueing networks just powerful enough to answer questions about the $p(\underline{n})$ interpreted only as proportions of time. The Markovian assumptions in the stochastic queueing network

theory considerably broaden the set of answerable questions by allowing the $p(\underline{n})$ to be interpreted as probabilities.

Operational analysis, for example, has nothing to say about the state of the system at time t (except to the extent that $p(\underline{n})$ is the probability of observing state $\underline{n}$ at a "random" time t). Markovian assumptions allow constructing differential equations relating state probabilities $p(\underline{n},t)$. These equations can, in principle, be solved for the transient behavior of the system; they can be used to study $p(\underline{n},t_2)$ given $\underline{n}(t_1)$.

Operational analysis is sometimes criticized on the grounds that the homogeneous assumption "hides" a Markovian assumption -- with the implication that it is equivalent to Markovian queueing network theory. That operational queueing network theory cannot answer questions about transient behavior, or about system state correlations, disproves this assertion. Moreover, operational analysis can be applied with full precision in finite time periods; steady-state Markovian analysis cannot. Operational analysis assumes measured parameters are used directly; Markovian analysis assumes stochastic parameters are known which, in practice, often cannot be done without difficult methods of parameter estimation.

Operational analysis is also criticized on the grounds that the lack of "stochastic regularity" makes the models useless in performance prediction. To study this assertion, consider a typical scheme of prediction, shown in Figure 4. The analyst begins with a model and model workload validated against an actual system (as in Figure 1). He constructs a projected set of workload and device parameters under the future conditions -- e.g., the same system with a new workload at a future time, or the same workload in a different system. He applies the same model to calculate projected performance quantities. If the modified system is ever built, he validates the predictions by comparing the actual workload against the projection (#1), and the actual performance quantities against the projected (#2). Serious errors in validation #2 almost always result from errors in workload prediction. After all, previous validations established the ability of the model to compute performance quantities when applied to measured parameters

The central point here is that the difficulty in performance prediction is not the model. It is, rather, predicting the workload. This is a very important problem, but it has nothing to do with whether the analyst uses operational or stochastic assumptions to derive relations between the $p(\underline{n})$ and device or workload parameters.

Operational analysis defines a mathematical system weaker than stochastic analysis. Because it is weaker, it applies to a larger class of systems; but it answers fewer questions. Even as there is a hierarchy of algebraic systems in mathematics -- semigroups, groups, fields -- so there is a hierarchy of

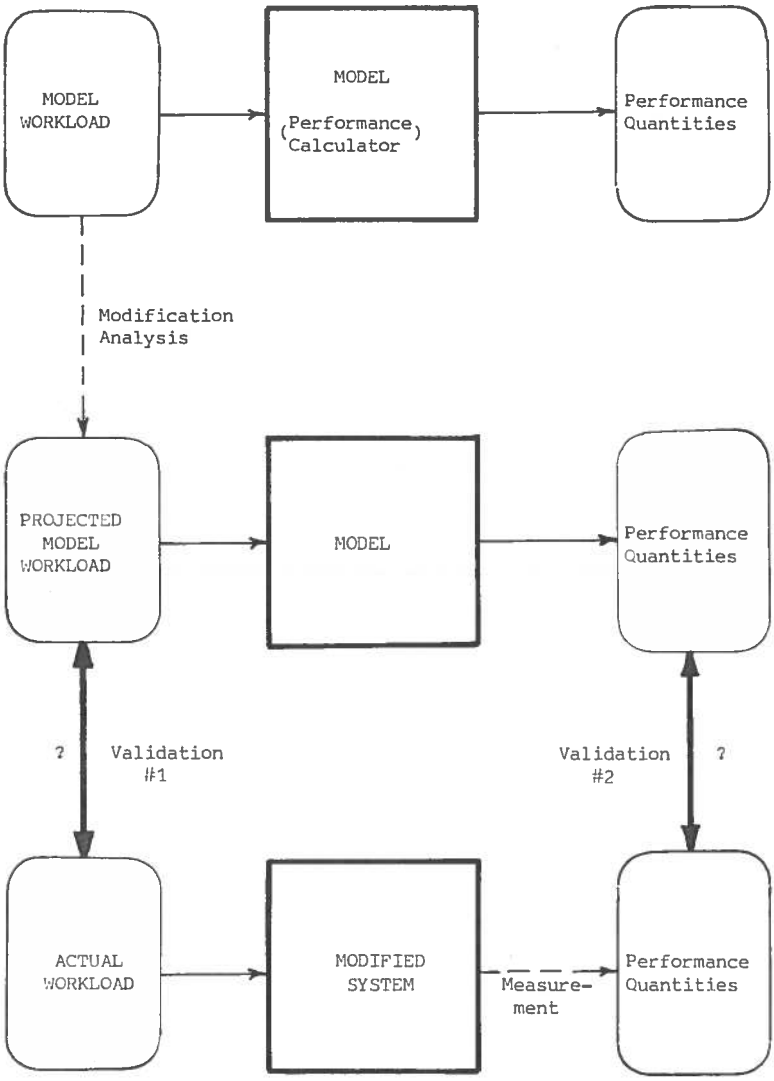


FIGURE 4. Typical performance prediction scheme.

mathematical systems for performance analysis. At the lowest level is bottleneck analysis, which assumes only that the visit counts and service functions are known and that job flow is conserved. At the next level is the network state space analysis of this paper, which adds the assumptions of state transition conservation and device homogeneity. At the highest level is Markovian queueing network analysis.

REFERENCES


BASK75 Baskett, F., Chandy, M., Muntz, R., and Palacios, J., "Open, closed, and mixed networks of queues with different classes of customers," J. ACM 22, 2 (April 1975), 248-260.

BUZE71 Buzen, J. P., "Analysis of system bottlenecks using a queueing network model," Proc. ACM SIGOPS Workshop on System Performance Evaluation (April 1971), 82-103.

BUZE73 Buzen, J. P., "Computational algorithms for closed queueing networks with exponential servers," Comm. ACM 16, 9 (September 1973), 527-531.

BUZE75 Buzen, J. P., "Cost effective analytic tools for computer performance evaluation," Proc. IEEE Compcon (September 1975), 293-296.

BUZE76a Buzen, J. P., "Operational analysis: the key to the new generation of performance prediction tools," Proc. IEEE Compcon (September 1976).

BUZE76b Buzen, J. P., "Operational analysis: an alternative to stochastic modeling," Technical report, BGS Systems, Inc., Box 128, Lincoln, MA 01773 (October 1976).

BUZE76c Buzen, J. P., "Fundamental operational laws of computer system performance," Acta Informatica 7, 2 (1976), 167-182.

CHAN72 Chang, A., and Lavenberg, S., "Work rates in closed queueing networks with general independent servers," IBM Research Report RJ989 (1972).

CHAN75 Chandy, M., Herzog, U., and Woo, L., "Parametric analysis of queueing networks," IBM J R & D 19, 1 (January 1975), 36-42.

COFF73 Coffman, E. G., Jr., and Denning, P. J., Operating Systems Theory, Prentice-Hall (1973).

COUR75 Courtois, P. J., "Decomposability, instabilities, and saturation in multiprogrammed systems," Comm. ACM 18, 7 (July 1975), 371-377.

COUR77 Courtois, P. J., Decomposability, ACM Monograph Series, Academic Press (1977).

DENN75 Denning, P. J., and Kahn, K. C., "Some distribution free properties of throughput and response time," Computer Sciences Dept., Purdue University, W Lafayette, IN 47907 USA, TR-159 (May 1975).

DENN76 Denning, P. J., Kahn, K. C., Leroudier, J., Potier, D., and Suri, R., "Optimal multiprogramming," Acta Informatica 7, 2 (1976).

GELE76 Gelenbe, E., and Muntz, R., "Probability models of computer systems - Part I (Exact results)," Acta Informatica 7, 1 (1976), 35-60.

GIAM76 Giammo, T., "Validation of a computer performance model of the exponential queueing network family," Acta Informatica 7, 2 (1976), 137-152.

GORD67 Gordon, W. J., and Newell, G. F., "Closed queueing systems with exponential servers," Operations Research 15 (1967), 254-265.

JACK57 Jackson, J. R., "Networks of waiting lines," Operations Research 5 (1957), 518-521.

JACK63 Jackson, J. R., "Job shop like queueing systems," Management Science 10 (1963), 131-142.

KLEI76a Kleinrock, L., Queueing Systems, Vol. I, Wiley (1976).

KLEI76b Ibid., Vol. II.

MUNT75 Muntz, R., "Analytic modeling of interactive systems," Proc. IEEE 63 (June 1975), 946-953.

MUNT74 Muntz, R., and Wong, J., "Asymptotic properties of closed queueing network models," Proc. 8th Princeton Conf. on Infor. Scis. and Systs., Dept EECS, Princeton University, Princeton, NJ 08540 USA (March 1974), 348-352.

SHUM77 Shum, A., and Buzen, J. P., "The EPF Technique: a method for obtaining approximate solutions to closed queueing networks with general service times," Proc. 3rd Int'l Symp. on Modelling and Performance Evaluation of Computer Systems, Bonn, Germany (October 1977).