1977

Operational Analysis of Queueing Networks

Peter J. Denning

Jeffrey P. Buzen

Report Number:
77-225

OPERATIONAL ANALYSIS OF QUEUEING NETWORKS

Peter J. Denning
Computer Science Department
Purdue University
West Lafayette, Indiana 47907

and

Jeffrey P. Buzen
BGS Systems, Inc.
Box 128
Lincoln, MA 01773

CSD-TR 225

March 1977

OPERATIONAL ANALYSIS OF QUEUEING NETWORKS[1]

Peter J. Denning[2]

Jeffrey P. Buzen[3]

March 1977

Abstract: In typical validations of computer performance models, analysist interpret the p(n) of queueing networks as time-proportions during which a given network state n is observed. They parameterize performance calculations with directly measured device service time functions and job device visit counts. Three operational assumptions constitute a minimal set of assumptions for calculating these p(n): the number of jobs observed to arrive at a 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; and the on-line service functions of devices are 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. Operational queueing network theory is weaker than Markovian queueing network theory.

---

(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.

1 INTRODUCTION

1.1 Background

Since they can represent multiple resource systems, queueing networks have become a common analytic tool for 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 systems with arbitrary state dependent service rates at all devices in the system [JACK63]. In 1967, Gordon and Newell extended this analysis to 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 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 among job steps within a job 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 underlined words illustrate concepts that the analyst must understand to be able to use the models confidently. Not only are some of these concepts difficult, but some can be disproved empirically -- for example, system parameters change over time, jobs are dependent, job steps do not follow Markov chains, systems are observable only for short intervals, service distributions seldom follow exponentials. 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 these computed values compare well with actual (measured) values, over many different observation periods, he will conclude that the model is good. (See Figure 1.) Thereafter, he may employ it confidently for predicting future behavior or evaluating proposed changes in the system.

The important observation is that many 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 necessary.

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

- All quantities must be measurable in <u>finite observation periods</u> -- there is no assumption of "stationarity" or "steady state".
- The system must be <u>work conservative</u> -- 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.

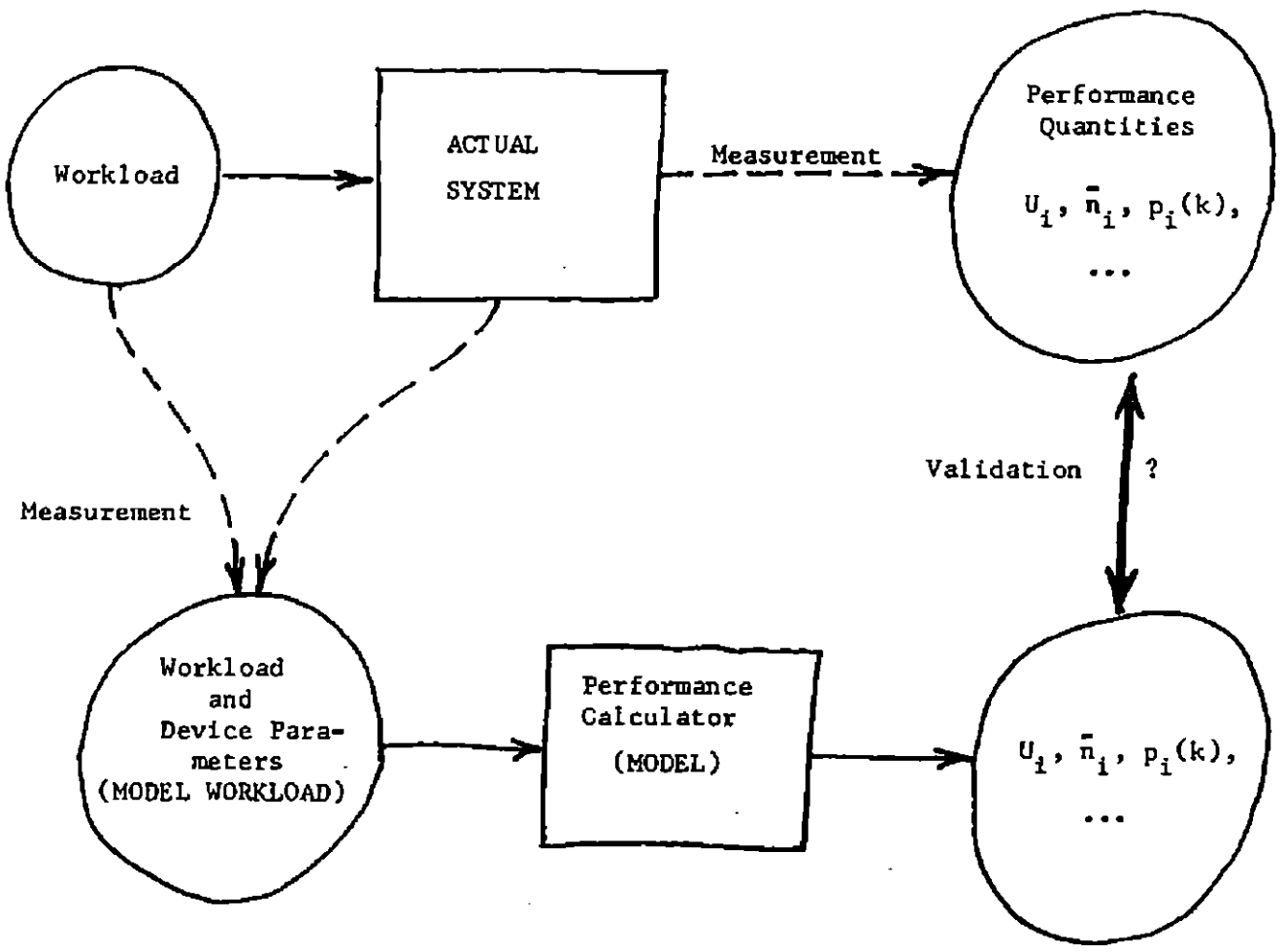


Figure 1. Typical validation scheme.

- The system must be homogeneous -- i.e., the mean output rate 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 the "local balance equations" of queueing network analysis. The usual product form solutions and computational procedures are then applicable. The conclusion is that (quantities derived from) the $p(\underline{n})$ actually depend only on the operational assumptions, which are weaker than the stochastic ones traditionally used.

The weaker assumptions of operational analysis restrict the set of questions that can be answered about queueing networks. The limitations of operational analysis will be 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 point 'IN'; whereupon it circulates through the network, waiting in queues and having job steps (tasks) served 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 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 <u>closed</u>; this is modeled by connecting the output back to the input. The system <u>output rate</u>, 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 ($i = 1,\ldots,K$):

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

$C_{ij}(n)$, number of times jobs start tasks at device j just after completing tasks 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 task was at device i when $N=n$; and

$C_{i0}(n)$, number of jobs whose last task was at device i when $n_i=n$.

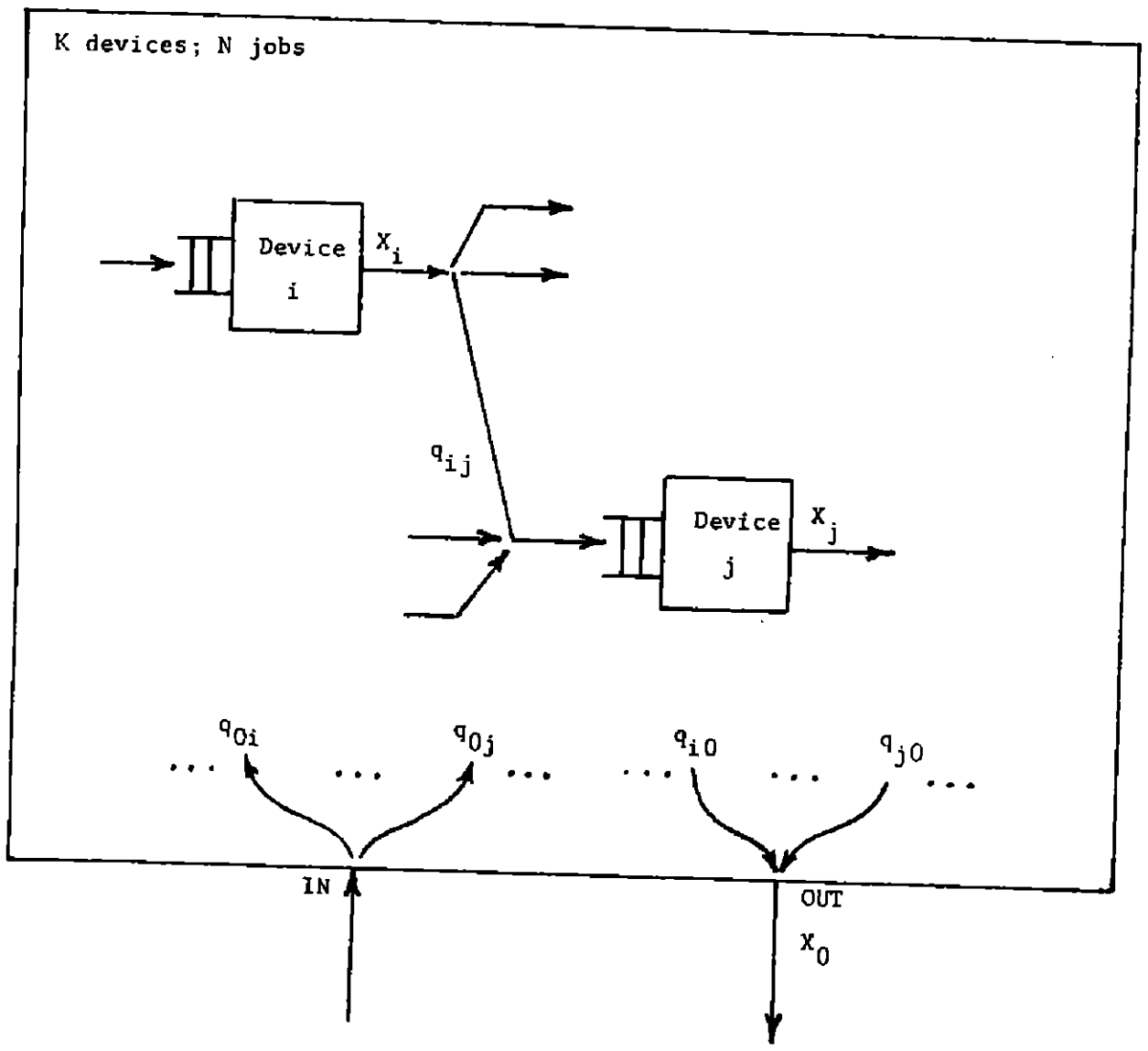


Figure 2. A queueing network.

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 <u>stratified sampling</u>. 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:

$X_i(n)$, job flow 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, $S_i(n) \simeq T_i(n)/C_i(n)$

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

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

and the <u>overall output rate</u> of device i to be

$$X_i = C_i/T .$$

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

$$S_i = B_i/C_i .$$

It is easily verified that the utilization satisfies

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

(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 task, $R_i = J_i/C_i$

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 the 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 that move immediately to 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 is the section on decomposability.

3 JOB FLOW ANALYSIS AND BOTTLENECKS

3.1 Job Flow Balance

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, the equations have no unique solution; the sum of the X_j-equations for $j = 1,\ldots,K$ is

$$\sum_{j=1}^{K} X_j = \sum_{i=0}^{K} X_i \sum_{j=1}^{K} q_{ij} = \sum_{i=0}^{K} X_i(1-q_{i0}) = \sum_{i=1}^{K} X_i + X_0 - \sum_{i=0}^{K} X_i q_{i0}$$

This implies

$$X_0 = \sum_{i=0}^{K} X_i \, q_{i0},$$

which is the equation for j=0. Since X_0 is unknown in a closed network, this shows that 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 considerable information of 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 than as many as $(K+1)^2$ values of q_{ij}.

3.2 Saturation and Bottlenecks in Systems of Load Independent Parameters

In a network whose parameters are load independent -- that is, $S_i(n) = S_i$ for all $n>0$ and the q_{ij} do not depend on the total load N -- job flow analysis yields enough information to deduce throughputs under light and heavy loads. The following results are the operational counterparts of results obtained by Muntz and Wong for Markovian networks [MUNT74, MUNT75; also DENN75].

In general, the ratio of any two throughputs is given by the ratio of the visit counts:

$$X_i/X_j = V_i/V_j, \qquad \text{for all N.}$$

Since $U_i = X_i S_i$, a similar property holds for utilizations:

$$U_i/U_j = V_i S_i/V_j S_j, \qquad \text{for all N, } j \neq 0.$$

These properties were first observed by Chang and Lavenberg for Markovian networks [CHAN72].

Device i is saturated if its utilization reaches 100%. In this case the formula $U_i = X_i S_i$ implies

$$X_i = 1/S_i,$$

which is the maximum throughput achieveable at device i. (In general, $U_i \leq 1$ and $X_i \leq 1/S_i$.) To achieve $U_i=1$, device i must have a long queue; for this reason it is called a "bottleneck". Every system has at least one bottleneck. We use the subscript b for any device capable of being a bottleneck. Thus $U_b = 1$ and $X_b = 1/S_b$ will be observed if N becomes large enough.

Since the ratios U_i/U_j are fixed, the device i with the largest value of $V_i S_i$ will be the first to achieve 100% utilization as N increases; thus

$$V_b S_b = \max \{ V_1 S_1, \ldots, V_K S_K \} .$$

Since $V_b = X_b/X_0$, and since $X_b = 1/S_b$ is saturation,

$$X_0 = 1/V_b S_b$$

is the maximum value of system throughput. Since $V_i S_i$ is the total service time requirement of a job at device i, the sum

$$R = V_1 S_1 + \ldots + V_K S_K$$

is the minimum possible value of mean response time. In fact, R is the mean response time when N=1. This implies that $X_0 = 1/R$ when N=1.

These properties of X_0 are summarized in Figure 3. As a function of N, X_0 rises monotonically from $X_0(1) = 1/R$ to asymptote $1/V_b S_b$. It stays below the line of slope 1/R eminating from the origin: job interference via queueing when N=k prevents throughput from reaching k/R.

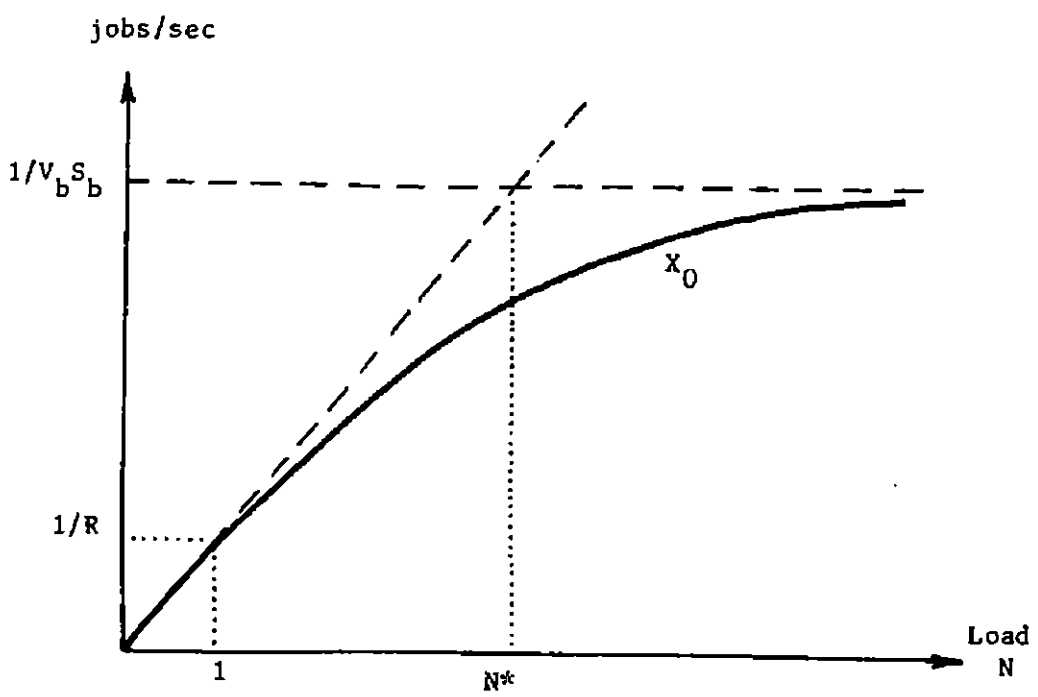


Figure 3. System throughput function.

Were we to hypothesize that k jobs always manage to avoid delaying each other, so that $X_0 = k/R$, the saturation asymptote requires that $k/R \leq 1/V_b S_b$, or

$$k \leq N^* = \frac{R}{V_b S_b} = \frac{V_1 S_1 + \ldots + V_K S_K}{V_b S_b} \leq K .$$

In other words, $k > N^*$ would imply with certainty that some device were saturated. Since N^* thus represents a load beyond which queueing is certain to occur somewhere in the system, we call N^* the "saturation point" of the system.

To summarize: the workload parameters or the job visit equations allow the analyst to determine the visit counts V_i. Device characteristics allow him to determine the mean service time per visit S_i. The largest of the products $V_i S_i$ determines the bottleneck device b. The sum of the products determines the smallest possible mean response time R. The system throughput is $1/V_b S_b$ in saturation and the saturation point is $N^* = R/V_b S_b$.

An analysis leading to a sketch like Figure 3 may give some gross guidance on the effects of proposed changes. For example, reducing $V_i S_i$ for a device i which is not a bottleneck (e.g., by reducing the service time or the visit count) will not affect the bottleneck; it will make no change in the asymptote and will produce at best a small change in minimal response time. Reducing the product $V_i S_i$ for all the bottleneck devices will remove the bottleneck; it will raise the asymptote and reduce minimal response time. However, this effect will be noticed only as long as $V_b S_b$ remains the largest of the $V_i S_i$: too much improvement at device b will cause the bottleneck to move elsewhere.

4 SOLUTIONS FOR STATE OCCUPANCIES

4.1 State Space 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 $A(\underline{n},\underline{m})$ 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, $A(\underline{n},\underline{n}) = 0$. We make the approximation,

> Principle of State 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:

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

The only error in these equations is a +1 (-1) term missing on the right side if $\underline{n}$ is the final (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:

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

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

reexpressed as

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

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

State Space Balance Equations

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

for all $\underline{n}$ in which each $B(\underline{n},.)$ 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 $B(\underline{n},\underline{m})$. Were an analyst prepared to measure system states, he would hardly use these equations to "solve" for the $p(\underline{n})$. The analyst is instead interested in how to express the $B(\underline{n},\underline{m})$ in terms of device and workload parameters, so that he can obtain (unique) values for the $p(\underline{n})$ without actually measuring any system states.

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 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 use this approximation:

> 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. This assumption usually 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. The state space balance equations reduce to (for all $\underline{n}$):

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

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

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 $B(\underline{n},\underline{m})$.

To obtain solutions of these equations from device and workload parameters, analysts frequently combine routing frequencies with 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

$$\sum_{i,j} p(\underline{n}_{ij})\frac{q_{ij}I_j}{S_i(n_i+1)} + \sum_i p(\underline{n}_{i0})\frac{q_{i0}}{S_i(n_i)} + \sum_j p(\underline{n}_{0j})X_0 q_{0j} I_j$$

$$= p(\underline{n})\left(\sum_i \frac{I_i}{S_i(n_i)} + X_0\right), \qquad \text{all } \underline{n}$$

These equations are identical in form to the "local balance equations" of Markovian queueing networks [KLEI76]. The analyst can solve them for the $p(\underline{n})$ without measuring the state space. Since the solution is

Table I. Homogeneous Transition Rates.

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

approximate -- mainly because of the homogeneity assumption -- the results require validation. Practical experience is good.

The solution of the homogenized balance equations is known to be 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, KLEI76.) 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 observed, determines $(N+1)^K$ values of $p(\underline{n})$.

The normalizing constant of an open system 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

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

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]. It can also be deduced from the "generalized birth death" analysis, which is the operational counterpart of Markovian birth-death analysis [BUZE76a,b]. No similar property holds for closed networks.

4.3 An Example

Figure 4 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 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

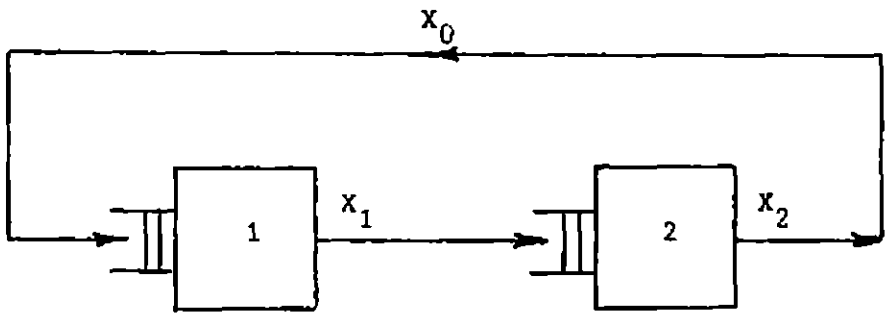
$$B(20,11) = A(20,11)/T(20) = 3/17$$
$$B(11,20) = A(11,20)/T(11) = 1$$

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:



$X_0 = X_1 = X_2$

$K = 2 \qquad N = 2$

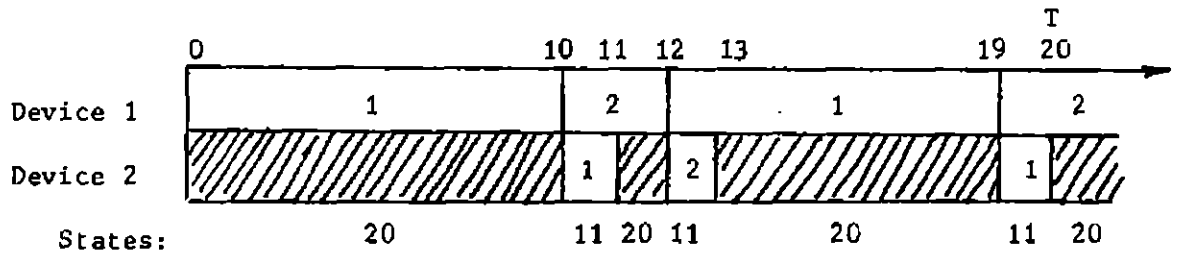


Figure 4. Two device system and observed behavior.

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

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

If a subsystem interacts weakly with its environment, the transient behaviors of the subsystem will have little effect on the long run dynamics of the environment. 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.)

Operationally, decomposability allows an analyst to conduct a series of controlled experiments on the subsystem in question. He subjects it to a constant load, n jobs of the given type, for some time period of T seconds. Just after each completion in the controlled experiment, he adds another job to keep the load at n. He counts the number of completions, C, and assigns $S(n) = T/C$. In the environment, he replaces the subsystem by a load dependent device of service function S(n). If indeed the subsystem interacted weakly with the 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 a subsystem can be replaced by a single load dependent device, whose service function is obtained by studying the subsystem off line, with no effect on the marginal distribution $p_i(n_i)$ of any device outside the subsystem. This theorem is a property of the product form solution; consequently it works for operational analysis. In other words, a network of homogeneous devices is itself homogeneous relative to the environment in which it is embedded.

The decomposability principle permits studying a nonhomogeneous subsystem using operational analysis. Regardless of its internal behavior, a subsystem may successfully be represented by a homogeneous device, as long as it interacts weakly with its environment. The off line behavior of the equivalent device can be obtained by a direct controlled experiment on the subsystem, by a simulation, or by an analysis.

6 LIMITATIONS OF OPERATIONAL ANALYSIS

Operational analysis specifies a weakest set of assumptions necessary to compute the proportions of time $p(\underline{n})$ a queueing network occupies each state $\underline{n}$, when only the 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 analyses do not produce exact answers. The principles of job flow balance and state flow balance are not met exactly in actual systems during most finite intervals; however, the error introduced by these assumptions 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. Homogeneity employs no information about the shapes of service distributions, which do influence the results.

Operational assumptions 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 quantities derivable from the time proportions $p(\underline{n})$. The Markovian assumptions in the stochastic queueing network theory considerably broaden the set of answerable questions. For example:

- Operational analysis has nothing to set about the effect of the shape of the service distributions on the $p(\underline{n})$. Using the method of stages, Markovian assumptions allow studying almost any service distribution encountered in practice. (See BASK75, GELE76, KLEI76.)
- Operational analysis 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 $\underline{n}(t_1)$.

To answer such questions, operational assumptions must be augmented by some or all of the stochastic assumptions.

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. The examples of the previous paragraph, which show important questions not answerable in operational analysis, disprove this assertion. Moreover, operational analysis can be applied in finite time periods; steady-state Markovian analysis cannot. Operational analysis permits using measured parameters directly; Markovian analysis requires careful estimation of stochastic parameters.

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 5. 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.

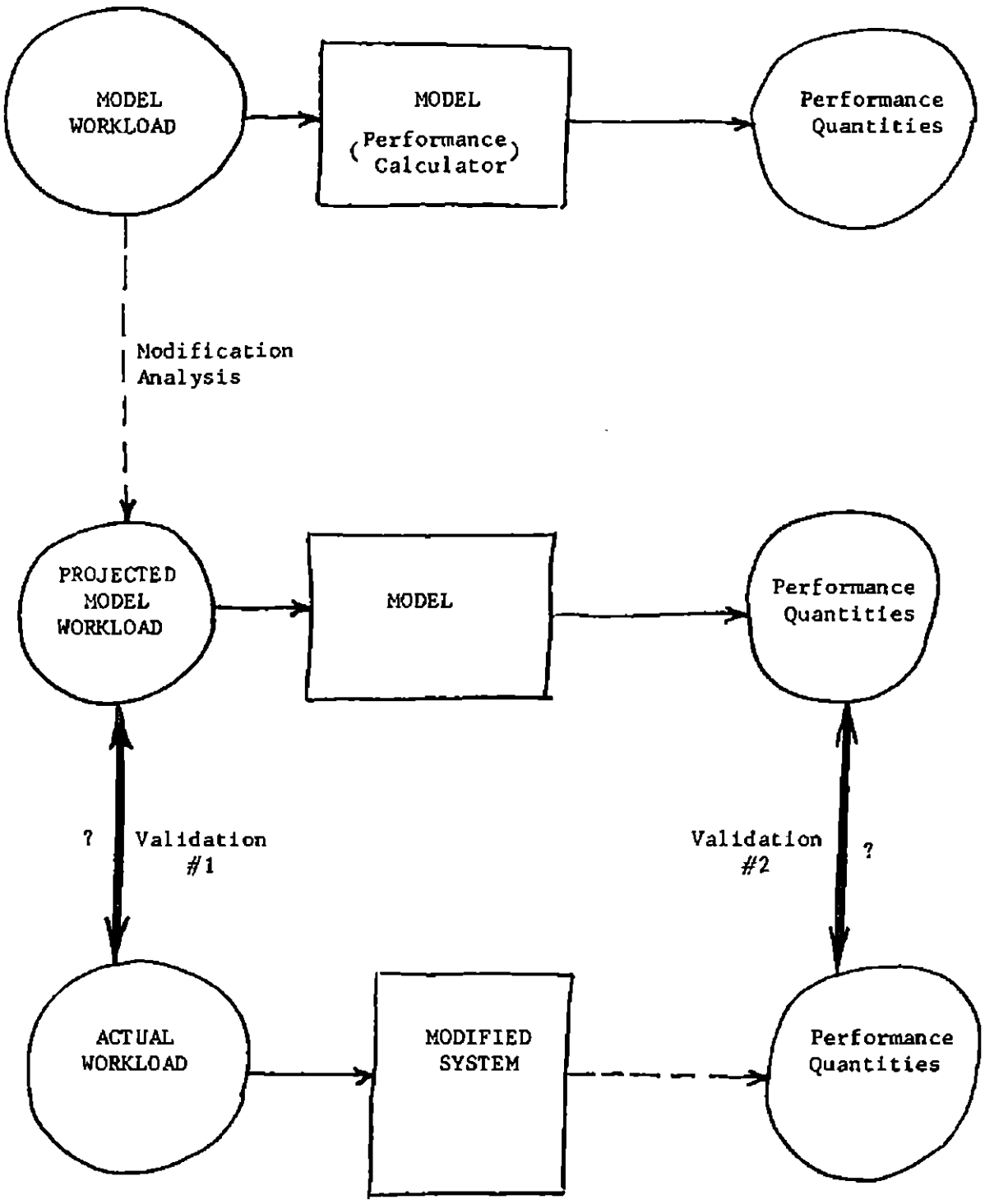


Figure 5. Typical performance prediction scheme.

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 has nothing to do with operational analysis.

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 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, 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, 3 (July 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 effect 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.

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.

KLEI76 Kleinrock, L., Queueing Systems, Vol. I, Wiley (1976).

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.