THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 5, May–June 1982
Printed in U.S.A.

Moment Formulae for a Class of Mixed Multi-Job-Type Queueing Networks

By H. HEFFES

(Manuscript received August 26, 1981)

Queueing network models play an important role during each stage of a computer system's life cycle (from initial conception to system maturity), where in each stage broadly applicable performance analysis tools are needed. This paper presents new results which contribute to the foundations of a tool to support performance analysis and modeling activities. In dealing with some performance issues, it is important to be able to quantify distribution or moment information, because these quantities can influence system capacity and service and performance measures. It is also important that models include the effect of congestion adaptive I/O devices, in a stable and efficient manner, for this inclusion can significantly affect the outcome of studying certain performance issues. We address the problem of direct, recursive computation of moments of the queue size distributions at a class of service centers embedded in a mixed network of queues. The parameterized class includes state-dependent processing rates useful in modeling congestion adaptive I/O devices. We also present results for calculating moments of both the waiting time and virtual delay (work backlog) distributions at a class of service centers. In addition, we obtain a Little's Law type of relation between delay moments and queue size factorial moments. For a class of networks, an algorithm is given for the direct, recursive computation of the tail of the node delay distribution.

I. INTRODUCTION

Performance analysis and modeling activities are essential for answering key questions at various stages of a computer system's life-cycle, ranging from initial conception to maintaining and growing a mature system. Although both the questions asked and the analysis approaches may differ from stage to stage, in each of these stages broadly applicable performance analysis tools are needed to support such activities. This paper presents new results which contribute to the foundation of one such tool: algorithmic techniques for efficiently solving a class of queueing networks.

In dealing with some performance issues, it is important to be able to quantify distribution or moment information (e.g., delay variability as opposed to only the mean delay)[1] because these quantities can influence system capacity and service and performance measures. It is also important that models include the effect of congestion adaptive I/O devices, in a stable and efficient manner, for this inclusion can significantly affect the outcome of studying certain performance issues (e.g., the impact of multiprogramming).[2,3]

In this paper, we present results for the direct recursive computation of moments of the queue size distributions at a class of service centers embedded in a mixed network of queues. The class of service centers allows us to efficiently treat, in a stable manner, a parameterized class of state-dependent processing rates useful in modeling congestion adaptive I/O devices.[3,4,5] By dealing with mixed systems, we allow consideration of systems with workloads from a finite population (e.g., a collection of terminals), multiprogrammed systems, together with workloads from basically infinite customer populations. We present results for calculating moments of the waiting time and virtual delay distributions at a class of service centers that enable us to quantify the variability of node delays, as well as work backlogs.

The well-known class of multiple resource models, usually referred to as product form queueing networks,[6] have been used to address a wide range of performance issues, such as capacity estimation and planning, bottleneck identification, performance prediction,[7] memory interference, and software lockout in multiprocessor systems.[8] Usually the models used to address these issues have approximately included the factors of interest (e.g., priority processor scheduling disciplines).[9] A considerable amount of effort has been devoted to the study of this class of queueing networks, and efficient computational algorithms exist[1,8,10] that allow one to obtain, for example, mean values of the desired quantities. While, in principle, the entire network queue size state description can be obtained from the above, one may be interested in obtaining results, directly, for a more moderate level of detail, e.g., moments of queue sizes, as well as in quantifying variability of delays at a network node. Existing algorithms for calculating even only mean values can become much more complex and sometimes exhibit chaotic behavior.[11] This situation arises, for example, when a state-dependent service rate is used to model a class of devices, such as an efficiently scheduled disk or drum,[3,4,5]* whose efficiency is a function

* See Ref. 5 for examples of secondary storage units that employ a scheduling algorithm which attempts to minimize rotational latency and/or seek times.

of the number of queued requests. The complexity results from a requirement that the entire* marginal queue size distribution is needed at each step of the computation.[1]

The results in this paper overcome some deficiencies in present methods for analyzing networks of queues and enable us to quantify important performance measures for which, previously, no efficient computational methods existed. In Section 1.1, we discuss how a tool for evaluating networks of queues can fit into a performance analysis and modeling methodology, and in Section 1.2, we give a more specific definition of the problems treated and outline the remaining sections of this paper.

1.1 Use of network of queues models

To illustrate the aforementioned need for broadly applicable tools and how a queueing network analysis tool can fit into a performance analysis and modeling methodology, we consider the various stages of a system's life cycle. During the system's conception stage, where one is concerned with the services to be offered, broad objectives—performance, reliability, etc.—basic architecture, proposed components and initial sizing, questions involving initial feasibility arise. Initial feasibility studies generally require several tools to address the following type of question: Given a proposed architecture and assumptions concerning the way the system is to be used (e.g., obtained from a gross workload characterization tool which may indicate various usage scenarios), how well can the system be expected to perform? The answer to this question often leads to a modification of the proposed architecture and/or what the system is planned to do. During this stage, where system specification is often at a macroscopic level of detail, a tool based on a network-of-queues methodology can be useful for the performance prediction. Here, for example, use of a tool incorporating a central-server[12] queueing network model can be helpful in answering questions such as the effects of CPU size, number of disks, and the number of terminals that can be supported, while meeting broad performance objectives, as well as estimates of cost. The particular form of input that may be available from a workload characterization is particularly suitable for use by a network-of-queues methodology at this stage.

During a system design phase (when one is attempting design optimization), one may tend to focus more on subsystems, and models would tend to include more microscopic details, such as a disk schedule model or a CPU process schedule model. Although this stage generally requires more detail than is attainable with a network-of-queues model,

* Some simplifications result when the state dependence disappears above a given loading,[1,10] the so-called limited queue-dependent server.

issues such as the effect of the degree of multiprogramming*[2] and the effect of a disk schedule which uses a SCAN† algorithm versus a FIFO discipline, can be treated. We note that at this stage it is important to have models which include, at least parametrically, the effect of a device being congestion adaptive since this can significantly influence the performance that can be expected.‡ In this stage, distributional information can also be important, as opposed to just dealing with mean values, since performance criteria and service objectives may be in terms of the tail of a distribution.

During the stage when one is dealing with and maintaining a mature system—adding features, growing, doing capacity estimation and planning—a tool or methodology for viewing the overall system is desirable. Here, measurement tools, providing resource utilizations, workload characterizations, and response times, together with proposed changes in system use and growth, can be used with an overall system performance model to answer such questions as Is the system adequate? How much more load can it handle? or, What system changes are required to handle a further load increase? A capacity planning methodology which accepts, say, maximum allowable utilizations on various system resources and measurements of current utilizations and workloads, and predicts allowable increases in load could use the output of a network-of-queues tool to determine what the maximum resource utilizations should be. We note that the determination of these levels, and thus the system capacity, could very well be based on distribution information.

1.2 Outline of paper and summary

In this paper we address the problem of recursively computing moments of the queue size distribution at a service center embedded in a class of product form networks. Parameterizing the service center with respect to its state dependence allows us to treat, in an efficient manner, congestion adaptive models with either improved or degraded§ efficiency. We treat mixed systems where the population corresponding to a given job type may be fixed (i.e., a closed chain) and where system requests corresponding to other job types may arrive exogenously from basically an infinite population (i.e., an open chain). Closed chains

* Reference 2 also shows that the multiprogramming effect, representing an increase in potential throughput of up to 75 percent, can depend strongly on the disk scheduling algorithm.

† Actually Ref. 2 considers the LOOK algorithm which is similar to SCAN.

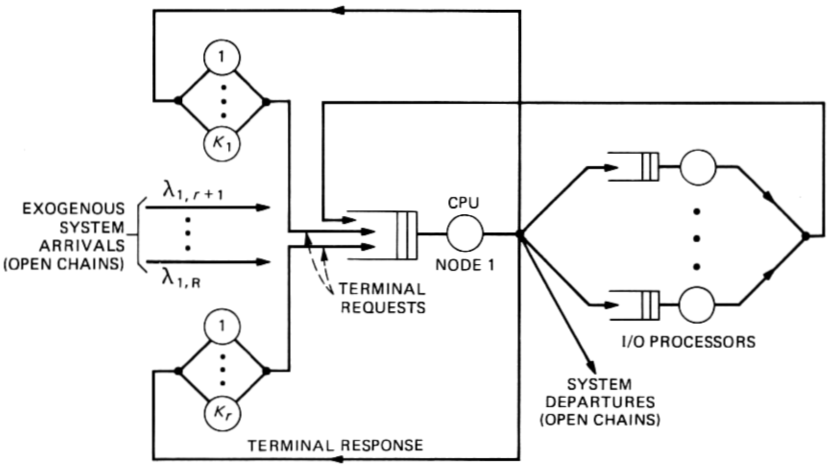
‡ Reference 3, which presents a performance comparison of two I/O access disciplines, notes that the relative comparisons, representing capacity improvements in excess of 100 percent, can depend strongly on whether the I/O device is congestion adaptive (e.g., a fixed head drum employing the shortest access time first schedule) or congestion independent (e.g., FIFO).

§ Resulting, for example, from increased overhead with increased processor queueing.

arise when dealing with finite population models where, for example, a finite number of terminals places requests into a computer system. They also arise in modeling multiprogrammed systems where the size of the finite population corresponds to the degree of multiprogramming.

Figure 1 shows an example of a mixed system where the closed chains correspond to each of the terminal groups, each group with possibly different think-time distributions and routing and service requirements (workloads). Figure 2 shows a closed network model representing requests arriving to a system, S, over a finite collection of access trunk groups. A request is blocked if all trunks in the group are occupied. Note that a trunk is held during the entire time the request is in S. When a trunk is available in group i, requests enter S from group i at rate λ_i (the Poisson arrival rate to group i). The population of chain i is K_i, the size of trunk group i, and the blocking probability at group i corresponds to 1−utilization of the node with service rate λ_i. In addition to throughputs and resource utilizations as the usual quantities of interest, other quantities of interest may be queue size moments, and the distribution, or moments, of the time from trunk activation to its first entry into the CPU (reaction time).

We treat the problem of recursively calculating moments of the waiting time and virtual delay distributions at first-come-first-served (FCFS) state-independent service centers, which enables us to get a



- r GROUPS OF TERMINALS (CLOSED CHAINS)
  GROUP s CONTAINS K_s TERMINALS (CHAIN POPULATION)
- R-r TYPES OF EXTERNALLY ARRIVING JOBS
  $\lambda_{1,s}$ ARRIVAL RATE FOR JOB TYPE s

Fig. 1—Mixed system illustration.

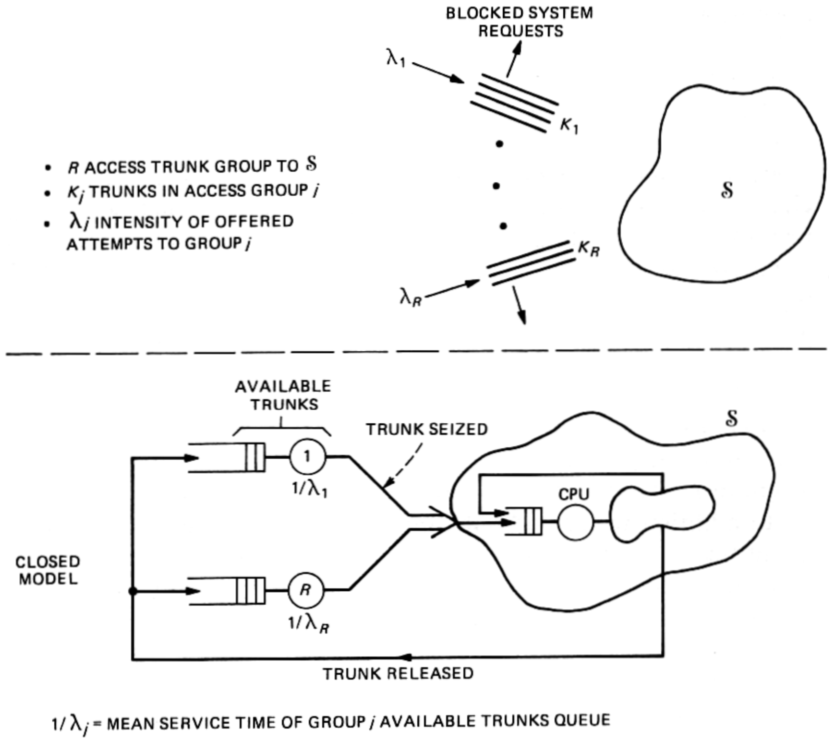


Fig. 2—Closed system illustration.

handle on the variability of nodal delays and to estimate the distribution of such delays. With knowledge of the virtual delays, we can compare performance, for example, as viewed by arriving customers to that viewed by an outside observer. We can thus estimate performance measures such as the system reaction time[13] distribution, which is a measure of the time between a terminal request and this request first getting the attention of the CPU. In another application, it could correspond to the time between a request coming into a system over an access trunk group and the start of processing.

In addition to obtaining algorithmic results, we obtain a relation between moments of node delays and queue sizes at a class of FCFS service centers. We finish with an algorithm for the recursive computation of the tail of the nodal delay distribution at a FCFS service center embedded in a network consisting of either single-server, state-independent nodes or infinite server nodes, e.g., the central server model of multiprogramming.[12]

The organization of this paper is as follows: In Section II, we present the class of networks under consideration, including specification of the job-type characteristics and specification of the class of service centers. In Section III, we consider closed systems and present results for the factorial moments of queue size distributions, for different levels of customer aggregations. The aggregations considered are the total number of jobs of a given type at a network node and total number of jobs of all types at a network node, the former requiring consideration of joint factorial moments and correlations. Results for mixed systems are given in Section IV by considering a mixed system as the limiting case of a multi-job-type closed system, formed by augmentation, as the population increases and the augmented node becomes the bottleneck.

The recursions for nodal delay and flow time moments are presented in Section V for closed systems and in Section VI for mixed systems. Delay results are obtained as experienced by arriving customers or as experienced by an outside observer i.e., the virtual delay. Delay distribution results appear in Section VII, and the appendices contain details of the investigations.

II. CLASS OF NETWORKS: SERVICE CENTER AND JOB CHARACTERISTICS

We define the structure of the networks in terms of the types of service centers or nodes and the job-type characteristics in terms of their routing through the network, their service requirements per service center visit,* their populations for closed job types, and their exogenous arrival rates for open job types.

We consider a network with R types of customers (jobs, chains) where $s = 1, 2, \cdots, r$ correspond to closed chains and $s = r + 1, \cdots, R$ correspond to open chains. The nodes in the network are either single-server nodes, using a FCFS,† processor sharing or last-come-first-served preemptive resume (LCFS-PR) queueing discipline or infinite server nodes. At the single-server nodes, the processing rate can depend on the total number of customers present.

Customers of a given job type (chain) may change class membership as they traverse the network but always remain in the same chain. Allowing customer class change allows one to model a broad class of routing scenarios including a deterministic, fixed sequence of node visits and also allows for different visits to, for example, a processor

* Job characteristics can also be defined in terms of the workload requirements, the average resource usage/job lifetime for each resource where job lifetime is suitably defined for closed customer types.

† Other types of customer selection rules, not depending on actual customer service requirements, (e.g., random selection, LCFS nonpreemptive) can also result in product form.[14]

sharing node by a given customer to have different service requirement distributions.

For the open chains, $s = r + 1, \cdots, R$, we let $\lambda_{i,0,c,s}$ = rate of exogenous arrivals of chain s customers, with class membership c, to node i. Exogenous arrivals of customers, of a given chain-class pair, to a particular node are assumed to be Poisson.*

If we let $p^s_{kl,ic}$ = the probability that a chain s job completing service at node k as a class l customer will next enter node i with class membership c, then the actual node-class-chain flow rates $\lambda_{i,c,s}$ satisfy the so-called traffic equation,

$$\lambda_{i,c,s} = \lambda_{i,o,c,s} + \sum_{(k,l) \in I_s} \lambda_{k,l,s} p^s_{kl,ic};$$

$$s = r + 1, \cdots, R; \qquad (i, c) \in I_s, \tag{1}$$

where I_s is the set of feasible node-class pairs for chains s jobs.

We note that for each closed chain, the node-class-chain rates satisfy

$$\lambda_{i,c,s} = \sum_{(k,l) \in I_s} \lambda_{k,l,s} p^s_{kl,ic};$$

$$s = 1, \cdots, r; \qquad (i, c) \in I_s. \tag{2}$$

The closed chains are further specified by their population,

$$\mathbf{K} = (K_1, K_2, \cdots, K_r), \tag{3}$$

where K_s denotes the system population corresponding to the closed chain s. We denote the rate of customers of a given chain, s, flowing into a node as

$$\lambda_{i,s} = \sum_{c \in C_i(s)} \lambda_{i,c,s}, \tag{4}$$

where

$$C_i(s) = [c;\ (i, c) \in I_s].$$

We assume that for each open chain the traffic equation (1) has a unique solution† for the traffic rates

$$(\lambda_{ics};\ (i, c) \in I_s); \qquad s = r + 1, \cdots, R,$$

and that every closed chain has an irreducible routing matrix so that

---

* This does not preclude having a large class of state-dependent arrival processes, including examples where customer sources are turned off or blocked when their system population reaches a threshold. These can easily be transformed into a mixed network of the type being considered.

† This precludes certain pathological cases where customers enter the system and never exit.

the solution to (2) is unique up to a scalar.* Thus, the actual flow rates, for a given job type, can be specified by an arbitrary solution to (2), yielding relative arrival rates for classes in a closed chain, and a proportionality constant.

We note that the actual proportionality constant, for a given closed chain, say $\bar{\lambda}_s(\mathbf{K})$, depends on the population vector $\mathbf{K}$ and that the actual flow rates, $\bar{\lambda}_{ics}(\mathbf{K})$, can be written as

$$\bar{\lambda}_{ics}(\mathbf{K}) = \bar{\lambda}_s(\mathbf{K})\lambda_{ics}, \qquad (i, c) \in I_s. \tag{5}$$

Thus, the actual node-chain flow rates $\bar{\lambda}_{is}(\mathbf{K})$ can be written as

$$\bar{\lambda}_{is}(\mathbf{K}) = \bar{\lambda}_s(\mathbf{K})\lambda_{is}, \tag{6}$$

where λ_{is} is given by (4).

The assumptions on the service time distributions are those for which the product-form solution holds (see Ref 6). We denote

μ_{ics}^{-1} = mean amount of service required by a chain s class c customer at node i when only one customer is present at node i (note that at a FCFS node it is required that $\mu_{ics} = \mu_{ief}$ for all c, s, e and f),

and

$\mu_i(k)$ = the processing or service rate of node i when a total of k customers (regardless of type) are present.†

Clearly, the average service requirement of an arbitrary, with respect to class, chain s customer, denoted μ_{is}^{-1}, is given by the weighted average of the individual class average service requirements.

$$\mu_{is}^{-1} = \sum_{c \in C_i(s)} \frac{\lambda_{ics}}{\lambda_{is}} \mu_{ics}^{-1}. \tag{7‡}$$

While we allow the network to contain nodes with general state-dependent processing rates, we focus on getting queue size moments at nodes with a parameterized class of processing rates defined by

---

* This follows from the assumption of irreducibility of the routing matrix

$$P^s = [p^s_{kl,ic}]; \qquad kl, ic \in I_s$$

and precludes the pathological case where the set of communicating node-class pairs can be decomposed into disjoint subsets. We note that the matrix can be made irreducible by using enough chains.

† For example, the number of seconds of processing per second of elapsed time. Without loss of generally we have let $\mu_i(1) = 1$.

‡ Recall that these are not adjusted for state-dependent processing rates. Another interpretation of (7) is that μ_{is}^{-1} represents the average service time of an arrival to node i in a system whose only customer is a single chain s customer.

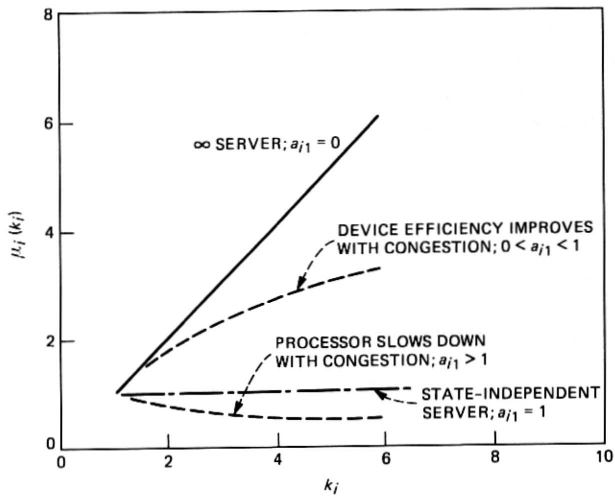


Fig. 3—A class of state-dependent service rates.

$$\mu_i(k) = \frac{k}{a_{i0} + a_{i1}k}, \tag{8}*$$

where $a_{i1} \geq 0$; $a_{i0} + a_{i1} = 1$. We note that $a_{i1} = 1$ corresponds to a single server with state-independent processing rate, while $a_{i1} = 0$ corresponds to an infinite-server node, often useful in modeling a finite population source—a collection of terminals—or a random delay. These are shown in Fig. 3, along with cases $0 < a_{i1} < 1$ and $a_{i1} > 1$ that can be used to represent congestion adaptive devices. We also note that the family of service rates may also be useful in approximating subsystems with restricted entry[15,16] by a single state-dependent node as is done in multiprogramming models.[17] What is needed here is the subsystem throughput as a function of the sizes of its populations, and recursive methods are naturally suitable for obtaining this.

III. QUEUE SIZE FACTORIAL MOMENTS—CLOSED SYSTEMS ($R = r$)

We present recursions for two different levels of customer aggregation: (*i*) for the total number of jobs of all types at a network node, and (*ii*) for the total number of jobs of a given type, the latter requiring consideration of joint factorial moments and correlations.

We denote

* The methods of this paper can be extended to include a generalization of this family of processing rates. The above family provides a good fit to the empirically obtained (by R. J. T. Morris) state dependence used in Ref. 2 for an efficiently scheduled (similar to SCAN) moving head disk over the multiprogramming range considered.

$$\beta_{ij}(\mathbf{K}) = E[k_i(k_i - 1) \cdots (k_i - j + 1); \mathbf{K}] \tag{9}$$

as the jth factorial moment of the total number of customers, k_i, at node i at an arbitrary time in equilibrium, for a system with population vector $\mathbf{K}$. Appendix A shows that $\beta_{ij}(\mathbf{K})$ satisfies the following recursions

$$\beta_{ij}(\mathbf{K}) = \sum_{s=1}^{R} \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}} \{a_{i1}\beta_{ij}(\mathbf{K} - \mathbf{e}_s) + [1 + (j-1)a_{i1}]\beta_{i,j-1}(\mathbf{K} - \mathbf{e}_s)\}, \tag{10}^*$$

with initialization $\beta_{ij}(\mathbf{0}) = 0; j > 0$ and $\beta_{io}(\mathbf{K}) = 1$. The quantity $\mathbf{e}_s$ is a unit vector in direction s. We thus obtain the jth moment at population $\mathbf{K}$ by updating the jth moments corresponding to systems with one less customer for each chain in addition to similarly including the effect of the $(j - 1)$st moments. The node-chain throughputs, $\bar{\lambda}_{is}(\mathbf{K})$, are available via standard mean-value analysis;[1] however, the standard analysis requires computations† of marginal probabilities at nodes with state-dependent processing rates. For networks, with nodes of the type under consideration, we can generalize mean-value analysis, in a stable manner, without ever computing marginal probabilities and, furthermore, obtain the higher order moments. When $j = 1$, (10) yields the generalization

$$\beta_{i1}(\mathbf{K}) = \sum_{s=1}^{R} \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}} [a_{i1}\beta_{i1}(\mathbf{K} - \mathbf{e}_s) + 1]. \tag{11}$$

Defining the mean node flow time for a chain s customer as the mean time a chain s customer spends at node i per visit‡ to the node i queue, $T_{is}(\mathbf{K})$, Appendix A shows that

$$T_{is}(\mathbf{K}) = \frac{1}{\mu_{is}} [1 + a_{i1}\beta_{i1}(\mathbf{K} - \mathbf{e}_s)] \tag{12}$$

The mean number of chain s customers at node i

$$\beta_{i,1,s}(\mathbf{K}) = E(k_{is}; \mathbf{K})$$

satisfies

$$\beta_{i1s}(\mathbf{K}) = \bar{\lambda}_{is}(\mathbf{K}) T_{is}(\mathbf{K}), \tag{13}^{\S}$$

where the (i, s) throughput satisfies [see (6)]

---

* In all recursions, terms with a negative population component are zero, e.g., $\mathbf{K} - \mathbf{e}_s$ with $\mathbf{K} = \mathbf{0}$.

† Those computations can become unstable.[11,18]

‡ If a customer is fed back to the same queue after a service completion, as in the central server model of multiprogramming, this corresponds to starting a new flow time.

§ The relation with (12) is via Little's Law for chain s customers at node i.

1. Initialize $\beta_{ij}(0) = 0\ j > 0$, $\beta_{io}(0) = 1$.
2. Loop on $\mathbf{K}$ until desired population $\mathbf{K}^*$.
3. Compute node flow time means, $T_{is}(\mathbf{K})$, from (12), $i \epsilon N(s)$, $s = 1,2,\ldots,R$.
4. Compute throughput proportionality constants, $\bar{\lambda}_s(\mathbf{K})$ from (15), $s = 1,2,\ldots,R$.
5. Compute node-chain throughputs, $\bar{\lambda}_{is}(\mathbf{K})$, $i \epsilon N(s)$, $s = 1,2,\ldots,R$, from (14).
6. Compute $\beta_{i1s}(\mathbf{K})$ from (13), $i \epsilon N(s)$, $s = 1,2,\ldots,R$ and $\beta_{i1}(\mathbf{K})$ from (16).
7. If do not desire $\beta_{ij}(\mathbf{K})\ j > 1$ or $\beta_{ijs}(\mathbf{K})\ j > 1$, go to 2.
8. If desire $\beta_{iJ}(\mathbf{K})\ J > 1$, compute $\beta_{ij}(\mathbf{K})$ from (10) at desired nodes for $1 < j < J$.
   If do not want $\beta_{ijs}(\mathbf{K})\ j > 1$, go to 2.
9. If desire $\beta_{iJs}(\mathbf{K}^*)$: If $\mathbf{K} = \mathbf{K}^* - J\mathbf{e}_s$, initialize $\gamma_{i,o,\ell,s}(\mathbf{K}) = \beta_{i\ell}(\mathbf{K})$, $\ell = 0,1,\ldots,J$.
   If $\mathbf{K} = \mathbf{K}^* - n\mathbf{e}_s\ n < J$, compute $\gamma_{i,J-n,\ell,s}(\mathbf{K})$ from (21) as shown in Fig. 5. Go to 2.

Fig. 4—Algorithm for queue size moments—closed systems.

$$\bar{\lambda}_{is}(\mathbf{K}) = \lambda_{is}\bar{\lambda}_s(\mathbf{K}). \tag{14}$$

In terms of the flow times, the chain s proportionality constant is

$$\bar{\lambda}_s(\mathbf{K}) = \frac{K_s}{\sum_{i \in N(s)} \lambda_{is} T_{is}(\mathbf{K})}; \qquad s = 1, 2, \cdots, R, \tag{15}$$

where $N(s)$ is the set of nodes visited by chain s customers. To close the algorithmic loop, we use

$$\beta_{i1}(\mathbf{K}) = \sum_{s=1}^{R} \beta_{i1s}(\mathbf{K}). \tag{16}$$

The algorithm (12) → (15) → (14) → (13) → (16) (see Fig. 4), with initial condition $\beta_{i1}(\mathbf{0}) = 0$ and with the relative traffic rates, λ_{is}, as in the previous section,* represents a simple, stable,† modification of mean value analysis for the desired class of state dependencies. With the node-chain throughputs from (14), the aggregate higher order factorial moments are obtained, in a numerically stable manner, by (10). We note that, unlike the mean values, it is only necessary to compute higher order moments at those nodes of interest. Thus, at the desired node we have the algorithm (see Fig. 4) (12) → (15) → (14) → (13) → (16) → (10). We now turn our attention to obtaining the higher order moments for the lower level of aggregation.

To obtain the higher order (>1) factorial moments of the number of each type of customer at a node

* We recall the relative traffic levels need not be computed for each population vector since they do not depend on $\mathbf{K}$.

† Note that no subtractions appear in the computations and recall $a_{i1} \geq 0$.

$$\beta_{i,j,s}(\mathbf{K}) = E[k_{is}(k_{is}-1)\cdots(k_{is}-j+1); \mathbf{K}], \tag{17}$$

we consider the joint factorial moments

$$\gamma_{i,j,l,s}(\mathbf{K}) = E[k_{is}(k_{is}-1)\cdots(k_{is}-j+1) \cdot k_i(k_i-1)\cdots(k_i-l+1); \mathbf{K}]. \tag{18}$$

Note that

$$\gamma_{i,0,l,s}(\mathbf{K}) = \beta_{i,l}(\mathbf{K}), \tag{19}$$

the high level lth factorial moment defined by (9) and computable from (10), and

$$\gamma_{i,j,0,s}(\mathbf{K}) = \beta_{i,j,s}(\mathbf{K}) \tag{20}$$

the desired low level jth factorial moment.

From Appendix A, we have the recursion

$$\gamma_{i,j,l,s}(\mathbf{K}) = \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}} \{a_{i1}\gamma_{i,j-1,l+1,s}(\mathbf{K}-\mathbf{e}_s) + (1+2la_{i1})\gamma_{i,j-1,l,s}(\mathbf{K}-\mathbf{e}_s) + [1+(1+a_{i1})(l-1) + a_{i1}(l-1)^2]\gamma_{i,j-1,l-1,s}(\mathbf{K}-\mathbf{e}_s)\}; \qquad j>0. \tag{21}$$

For $j = 0$ the computation is made via (10) and the identification (19). We note that for $l = 0$, $j = 1$ (21)† reduces to (13). Making the identification (20), gives

$$\beta_{ijs}(\mathbf{K}) = \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}} [\beta_{i,j-1,s}(\mathbf{K}-\mathbf{e}_s) + a_{i1}\gamma_{i,j-1,1,s}(\mathbf{K}-\mathbf{e}_s)]. \tag{22}$$

We note that if we are interested in the Jth factorial moment of the number of chain s customers at node i for a target population K^*, i.e., $\beta_{iJs}(\mathbf{K}^*)$, then (21) and (22) are initialized at population $\mathbf{K}^* - J\mathbf{e}_s$ with

$$\gamma_{i,0,l,s}(\mathbf{K}^* - J\mathbf{e}_s) = \beta_{i,l}(\mathbf{K}^* - J\mathbf{e}_s), \qquad l = 0, 1, \cdots, J, \tag{23}$$

the simple high-level aggregation result. We further note that (21) and (22) only need be updated along parameter direction $\mathbf{e}_s$. This is shown schematically in Fig. 5 and the steps in the algorithm in Fig. 4. If, for example, one is interested in a mean and variance analysis, this specializes to

$$\beta_{i2s}(\mathbf{K}^*) = \frac{\bar{\lambda}_{is}(\mathbf{K}^*)}{\mu_{is}} [\beta_{i1s}(\mathbf{K}^*-\mathbf{e}_s) + a_{i1}\gamma_{i,1,1,s}(\mathbf{K}^*-\mathbf{e}_s)], \tag{24}$$

---

† Terms with a negative subscript are zero.

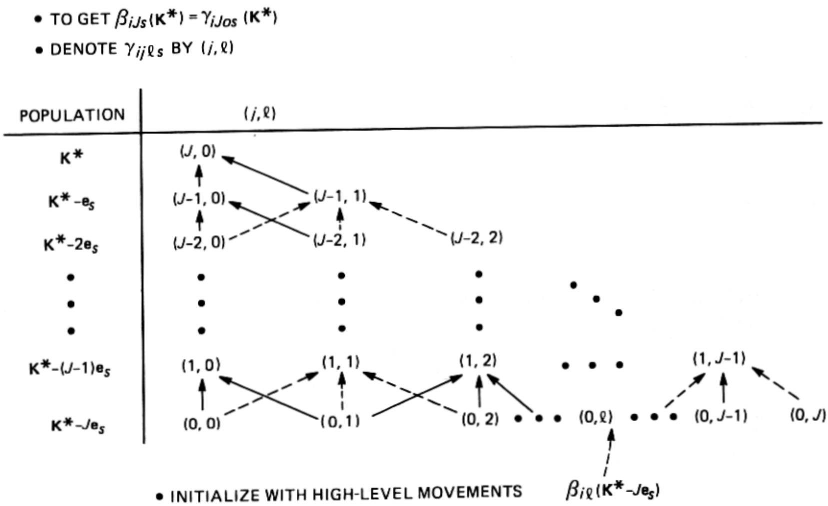


Fig. 5—Schematic for moments of low-level aggregation—closed systems.

where the correlation is given by

$$\gamma_{i,1,1,s}(\mathbf{K}^* - \mathbf{e}_s) = \frac{\bar{\lambda}_{is}(\mathbf{K}^* - \mathbf{e}_s)}{\mu_{is}} [a_{i1}\beta_{i2}(\mathbf{K}^* - 2\mathbf{e}_s) + (1 + 2a_{i1})\beta_{i1}(\mathbf{K}^* - 2\mathbf{e}_s) + 1]. \quad (25)$$

IV. QUEUE SIZE FACTORIAL MOMENTS—MIXED SYSTEMS

To obtain the mixed network results, we consider an augmented closed system which contains an external node for each open chain. The service rate of an external node corresponds to the system arrival rate for the corresponding open chain, i.e., $\lambda_{0s} = \sum_{i \in N_s} \lambda_{i,0,s}$; $s = r + 1, \cdots, R$. When customers depart in the original system, they are routed to the appropriate external node in the augmented system. A departure from a given external node, say, corresponding to the open chain s, is routed to node i as a class c customer with probability $\lambda_{i0cs}/\lambda_{0s}$. Taking the limit of the closed network results as the populations of chains $r + 1, \cdots, R \to \infty$, and assuming the external nodes become the bottlenecks,† we obtain the desired recursion for the high-level aggregate factorial moments

† Note that if an internal node becomes a bottleneck, the original mixed system is unstable.

$$\beta_{ij}(\mathbf{K}) = {}^{[1+(j-1)a_{i1}]} \frac{\rho_i^0}{1 - a_{i1}\rho_i^0} \beta_{i,j-1}(\mathbf{K}) + \sum_{s=1}^{r} \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}(1 - a_{i1}\rho_i^0)} \{a_{i1}\beta_{ij}(\mathbf{K} - \mathbf{e}_s) + [1 + (j-1)a_{i1}]\beta_{i,j-1}(\mathbf{K} - \mathbf{e}_s)\}, \quad (26)$$

where

$$\rho_i^0 = \sum_{s=r+1}^{R} \frac{\lambda_{is}}{\mu_{is}} \quad (27)$$

is the unadjusted utilization [unadjusted for state-dependent service rates $\mu_i(k) \neq 1$] at node i, corresponding to all open chains. We note that the stability condition is given by

$$\rho_i^0 a_{i1} < 1, \quad (28)$$

where we note that $\rho_i^0 a_{i1}$ is the limiting (as $k_i \to \infty$) utilization due to customers belonging to open chains. Equation (26) is initialized by

$$\beta_{i,0}(\mathbf{K}) = 1$$

and

$$\beta_{ij}(\mathbf{0}) = \frac{[1 + (j-1)a_{i1}]\rho_i^o}{1 - a_{i1}\rho_i^o} \beta_{i,j-1}(\mathbf{0}), \quad (29)$$

the open network factorial moments. For $j = 1$, (26) yields

$$\beta_{i1}(\mathbf{K}) = \frac{\rho_i^o}{(1 - a_{i1}\rho_i^o)} + \sum_{s=1}^{r} \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}(1 - a_{i1}\rho_i^o)} [1 + a_{i1}\beta_{i1}(\mathbf{K} - \mathbf{e}_s)], \quad (30)$$

with initial condition

$$\beta_{i1}(\mathbf{0}) = \frac{\rho_i^o}{1 - a_{i1}\rho_i^o}. \quad (31)$$

The required node-chain throughputs, $\bar{\lambda}_{is}(\mathbf{K})$, in (26) can be obtained via a standard type of mean analysis. However, as before, we are interested in a generalized analysis that does not involve marginal probabilities. These throughputs are obtained via the limiting argument which yields the recursions

$$\beta_{i1s}(\mathbf{K}) = \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}} [1 + a_{i1}\beta_{i1}(\mathbf{K} - \mathbf{e}_s)]; \qquad s \le r \quad (32)$$

for the closed chains and

$$\beta_{i1s}(\mathbf{K}) = \frac{\lambda_{is}}{\mu_{is}} [1 + a_{i1}\beta_{i1}(\mathbf{K})]; \qquad s > r, \quad (33)$$

for the open chains. We note that in (32), $\bar{\lambda}_{is}(\mathbf{K})$, the closed chain s-

node i throughput, is to be solved for, whereas in (33), λ_{is} is simply the solution to the traffic equation for open chain s. Summing (33) over all chains, we obtain

$$\beta_{i1}^{o}(\mathbf{K}) = \frac{\rho_i^o}{1 - a_{i1}\rho_i^o}[1 + a_{i1}\beta_{i1}^{c}(\mathbf{K})], \tag{34}$$

where

$$\beta_{i1}^{o}(\mathbf{K}) = \sum_{s=r+1}^{R} \beta_{i1s}(\mathbf{K}), \tag{35}$$

$$\beta_{i1}^{c}(\mathbf{K}) = \sum_{s=1}^{r} \beta_{i1s}(\mathbf{K}), \tag{36}$$

which relates the mean value of the total number of open customers at node i to the mean value of the total number of closed customers at node i. To obtain the desired recursion for β_{i1s}, $s \leq r$ we use (34) in (32), which results in

$$\beta_{i,1,s}(\mathbf{K}) = \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}(1 - a_{i1}\rho_i^o)}[1 + a_{i1}\beta_{i1}^{c}(\mathbf{K} - \mathbf{e}_s)], \qquad s \leq r; \tag{37}$$

and from Little's Law, the mean chain s flow times

$$T_{is}(\mathbf{K}) = \frac{1}{\mu_{is}(1 - a_{i1}\rho_i^o)}[1 + a_{i1}\beta_{i1}^{c}(\mathbf{K} - \mathbf{e}_s)], \qquad s \leq r. \tag{38}$$

The node chain throughputs are calculated from

$$\bar{\lambda}_{is}(\mathbf{K}) = \lambda_{is}\bar{\lambda}_s(\mathbf{K}), \quad s \leq r, \tag{39}$$

where, $\bar{\lambda}_s(\mathbf{K})$, the chain s proportionality constant (independent of i) is given by

$$\bar{\lambda}_s(\mathbf{K}) = \frac{K_s}{\sum_i \lambda_{is} T_{is}(\mathbf{K})}; \qquad s \leq r. \tag{40}$$

The algorithm,

$$(38) \rightarrow (40) \rightarrow (39) \rightarrow (37) \rightarrow (36),$$

(see Fig. 6) with initial condition

$$\beta_{i1}^{c}(\mathbf{0}) = 0,$$

is identical to the algorithm (see Fig. 4) for closed systems,

$$(12) \rightarrow (15) \rightarrow (14) \rightarrow (13) \rightarrow (16).$$

However, the closed chain service rates have been adjusted to account for the presence of the open chain customers, i.e.,

$$\mu_{is}' = \mu_{is}(1 - a_{i1}\rho_i^o). \tag{41}$$

1. Initialize β_{i1}(O) from (31) and β_{ij}(O) from (29) for $1 < j \leq$ desired order moment. $\beta^{c}_{i1}(O) = 0$.
2. Loop on **K** until desired population.
3. Compute node flow time means, $T_{is}(\mathbf{K})$, $s \leq r$ (i.e., for the closed chains) from (38) for $i \in N(s)$.
4. Compute throughput proportionality constants $\bar{\lambda}_s(\mathbf{K})$ from (40) for $s \leq r$.
5. Compute node-chain throughputs, $\bar{\lambda}_{is}(\mathbf{K})$, $s \leq r$, from (39), $i \in N(s)$.
6. Compute $\beta_{i1s}(\mathbf{K})$ from (37) for $s \leq r$, $i \in N(s)$ and $\beta^{c}_{i1}(\mathbf{K})$ from (36).
7. If desire $\beta^{o}_{i1}(\mathbf{K})$ or $\beta_{i1s}(\mathbf{K})$, for some $s > r$, use (34) for $\beta^{o}_{i1}(\mathbf{K})$ and (33) with $\beta_{i1}(\mathbf{K})$ given by the sum of $\beta^{c}_{i1}(\mathbf{K})$ and $\beta^{o}_{i1}(\mathbf{K})$.
8. If do not desire $\beta_{ij}(\mathbf{K})$ $j > 1$ or $\beta_{ijs}(\mathbf{K})$ $j > 1$, go to 2.
9. If desire $\beta_{iJ}(\mathbf{K})$ $J > 1$, compute $\beta_{ij}(\mathbf{K})$ from (26) for $1 < j \leq J$. If do not want $\beta_{ijs}(\mathbf{K})$ $j > 1$, go to 2.
10. If desire $\beta_{iJs}(\mathbf{K}^*)$ $s \leq r$: if $\mathbf{K} = \mathbf{K}^* - J\mathbf{e}_s$, initialize $\gamma_{io\ell s}(\mathbf{K}) = \beta_{i\ell}(\mathbf{K})$, $\ell = 0, 1, \ldots, J$. If $\mathbf{K} = \mathbf{K}^* - n\mathbf{e}_s$, $n < J$ Compute $\gamma_{i,J-n,\ell,s}(\mathbf{K})$ from (42) as shown in Fig. 5. Go to step 2.
11. At the desired population $\mathbf{K}^*$, if desire $\beta_{iJs}(\mathbf{K}^*)$ for $s > r$, initialize $\gamma_{io\ell s}(\mathbf{K}^*) = \beta_{i\ell}(\mathbf{K}^*)$ $\ell = 0, 1, \ldots J$. Compute $\gamma_{i,J-n,\ell,s}(\mathbf{K}^*)$ from (45) as shown in Fig. 7.

Fig. 6—Algorithm for queue size moments—mixed systems.

We note, however, that the adjustment is not directly related to the actual open-chain utilization (as in the state-independent case[11]), but rather to the limiting open-chain utilization.

The above algorithm, together with (33) and (34) describing the open chains, represents a simple, stable modification to a standard mean value analysis for mixed systems with the desired family of state dependencies.

With the node-chain throughputs from (39), the aggregate higher order factorial moments can be recursively computed in a numerically stable manner from (26). We note that, unlike the mean values, it is only necessary to compute higher order moments at those nodes of interest. Thus, at the desired node we have the algorithm (see Fig. 6)

$$(38) \rightarrow (40) \rightarrow (39) \rightarrow (37) \rightarrow (36) \rightarrow (26).$$

We now turn our attention to obtaining the higher order moments for the lower aggregation level.

To obtain the higher order factorial moments of the number of each type of customer at a node, $\beta_{ijs}(\mathbf{K})$ defined by (17), we require the joint factorial moments $\gamma_{ijls}(\mathbf{K})$ defined in (18), noting the relations (19) and (20). Treating the mixed system as a limiting closed system, we obtain

$$\gamma_{i,j,l,s}(\mathbf{K}) = \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}} \{a_{i1}\gamma_{i,j-1,l+1,s}(\mathbf{K} - \mathbf{e}_s) + (1 + 2la_{i1})\gamma_{i,j-1,l,s}(\mathbf{K} - \mathbf{e}_s) + [1 + (1 + a_{i1})(l-1) + a_{i1}(l-1)^2]\gamma_{i,j-1,l-1,s}(\mathbf{K} - \mathbf{e}_s)\}; \qquad j > 0; s \le r, \quad (42)$$

which is the same form as (21). For $l = 0, j = 1$ (42) reduces to (32). For $j = 0$, use (19) and (26). Making the identification (20), we get

$$\beta_{i,j,s}(\mathbf{K}) = \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}} [\beta_{i,j-1,s}(\mathbf{K} - \mathbf{e}_s) + a_{i1}\gamma_{i,j-1,1,s}(\mathbf{K} - \mathbf{e}_s)], \qquad s \le r. \quad (43)$$

If we are interested in $\beta_{i,J,s}(\mathbf{K}^*)$; $s \le r$, then (42) and (43) are initialized at population $\mathbf{K}^* - J\mathbf{e}_s$ with

$$\gamma_{i,0,l,s}(\mathbf{K}^* - J\mathbf{e}_s) = \beta_{il}(\mathbf{K}^* - J\mathbf{e}_s); \qquad l = 0, 1, \cdots, J, s \le r \quad (44)$$

the high-level aggregation result. Thus the computation proceeds as in the closed system case shown in Fig. 5.

For the open chains, the limiting system argument results in the relation

$$\gamma_{i,j,l,s}(\mathbf{K}) = \frac{\lambda_{is}}{\mu_{is}} \{a_{i1}\gamma_{i,j-1,l+1,s}(\mathbf{K}) + (1 + 2la_{i1})\gamma_{i,j-1,l,s}(\mathbf{K}) + [1 + (1 + a_{i1})(l-1) + a_{i1}(l-1)^2]\gamma_{i,j-1,l-1,s}(\mathbf{K})\} \qquad j > 0, s > r. \quad (45)$$

At $l = 0, j = 1$ (45) reduces to (33) and for $j = 0$, use (19) and (26). We note that (45) is not a recursion in $\mathbf{K}$ and can be evaluated as an aposteriori computation. Thus, if it is desired to compute $\beta_{i,J,s}(\mathbf{K})$; $s > r$ at a fixed desired population vector $\mathbf{K}^*$, all that is needed is the high-level aggregate $\beta_{il}(\mathbf{K}^*)$; $l = 0, 1, \cdots, J$, which starts off the (45) computation. This is shown in Fig. 7 and the steps in the algorithm in Fig. 6. As in the previous section, it is a straightforward matter to write down the expressions needed to do a mean and variance analysis.

V. DELAY AND FLOW TIME MOMENTS—CLOSED SYSTEMS

We consider the problem of obtaining recursions for the moments of the time a customer spends at a FCFS, single-server, state-independent node embedded in a closed network. As earlier, we define the node flow time as the length of time an arriving customer, either arriving from another node or being fed back from the node output, spends

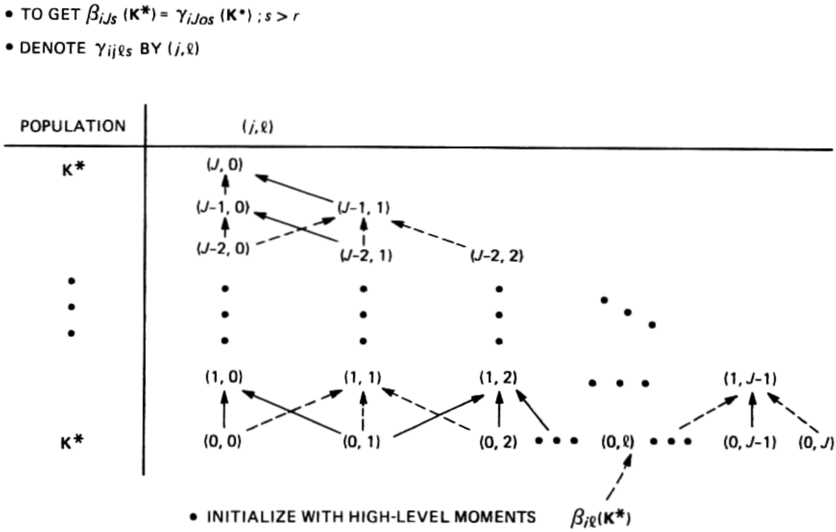


Fig. 7—Schematic for moments of low-level aggregation mixed systems—open chains.

until he next exits the server. The node delay is correspondingly that portion of the flow time spent in the queue. We also present a Little's Law type of relation between moments of the node flow times and queue size moments.

Denoting $T_{ijs}(\mathbf{K})$ as the jth moment of the flow time for an arbitrary chain s customer at node i, Appendix B shows that

$$T_{ijs}(\mathbf{K}) = \sum_{l=1}^{R} \frac{\bar{\lambda}_{il}(\mathbf{K} - \mathbf{e}_s)}{\bar{\mu}_i} T_{ijl}(\mathbf{K} - \mathbf{e}_s) + \frac{j}{\bar{\mu}_i} T_{i,j-1,s}(\mathbf{K}); \qquad j \geq 1, \quad (46)$$

where the quantity $\bar{\mu}_i^{-1}$ is the common mean service time at node i (i.e., $\bar{\mu}_i = \mu_{is} = \mu_{il}$).† From the initial conditions with respect to j

$$T_{i,o,s}(\mathbf{K}) = 1, \quad (47)$$

and (46), we have

$$T_{ijs}(\mathbf{e}_s) = \frac{j!}{\bar{\mu}_i^j}. \quad (48)$$

We note that an alternate form of (46), which only involves the flow time moments corresponding to the job type of interest, is obtained by recognizing that

$$T_{ijl}(\mathbf{K} - \mathbf{e}_s) = T_{ijs}(\mathbf{K} - \mathbf{e}_l).$$

† Required at a FCFS node to have a product form solution. Since node i has a state-independent service rate, $\mu_i(k) = 1$.

It can similarly be shown (using the approach discussed in Appendix B) that the jth moment of the delay distribution corresponding to a chain s customer satisfies

$$W_{ijs}(\mathbf{K}) = \sum_{l=1}^{R} \frac{\bar{\lambda}_{il}(\mathbf{K} - \mathbf{e}_s)}{\bar{\mu}_i} \cdot W_{ijl}(\mathbf{K} - \mathbf{e}_s) + \frac{j}{\bar{\mu}_i} W_{i,j-1,s}(\mathbf{K}); \qquad j > 1, \quad (49)^*$$

and

$$W_{i1s}(\mathbf{K}) = \sum_{l=1}^{R} \frac{\bar{\lambda}_{il}(\mathbf{K} - \mathbf{e}_s)}{\bar{\mu}_i} W_{i1l}(\mathbf{K} - \mathbf{e}_s) + \frac{1}{\bar{\mu}_i} \sum_{l=1}^{R} \frac{\bar{\lambda}_{il}(\mathbf{K} - \mathbf{e}_s)}{\bar{\mu}_i} \quad (50)$$

for $j = 1$.

As with the flow time moments, the delay moments at a given population are formed by updating the same order moment for reduced populations, together with including the effect of the lower order moment at the same population. We, thus, note that jth moment of the delay and node flow time distributions satisfy the same recursions, (46) and (49), for $j > 1$, the difference lying in the boundary equation (50), which results in

$$W_{i,j,s}(\mathbf{e}_l) = 0; \qquad j > 0. \quad (51)$$

It is *not* necessary, however, to have a separate computation for $W_{ijs}(\mathbf{K})$ since W_{ijs} is obtainable from T_{ijs} by the relation

$$W_{ijs}(\mathbf{K}) = \sum_{l=1}^{R} \frac{\bar{\lambda}_{il}(\mathbf{K} - \mathbf{e}_s)}{\bar{\mu}_i} T_{ijl}(\mathbf{K} - \mathbf{e}_s); \qquad j > 0. \quad (52)^\dagger$$

We note that the distributions of delays, as seen by arriving customers, will usually differ from the work backlog at a node at an arbitrary point in time (i.e., the virtual delay) as might be measured by an outside observer. If we define

$$V_{ij}(\mathbf{K}) = E(W_i^j(t); \mathbf{K}), \quad (53)$$

where $W_i(t)$ is the work present at node i at an arbitrary time in equilibrium, as the jth moment of the virtual delay, then it is shown in the next section that $V_{ij}(\mathbf{K})$ can be obtained from the moments of customer delay using the relations

---

* The computation of these delay moments or the preceding flow time moments are easily included in the algorithm shown in Fig. 4.

† Note that for $j = 1$, this simply states the known result $W_{i1s}(\mathbf{K}) = \beta_{i1}(\mathbf{K} - \mathbf{e}_s)/\bar{\mu}_i$. Relations (49), (50), and (52) can be written in a form containing only moments corresponding to the job type of interest by using $W_{ijl}(\mathbf{K} - \mathbf{e}_s) = W_{ijs}(\mathbf{K} - \mathbf{e}_l)$.

$$V_{ij}(\mathbf{K}) = \sum_{l=1}^{R} \frac{\bar{\lambda}_{il}(\mathbf{K})}{\bar{\mu}_i} W_{ijl}(\mathbf{K}) + \frac{j}{\bar{\mu}_i} V_{i,j-1}(\mathbf{K}); \qquad j > 1 \tag{54}^*$$

and

$$V_{i1}(\mathbf{K}) = \sum_{l=1}^{R} \frac{\bar{\lambda}_{il}(\mathbf{K})}{\bar{\mu}_i} \left(\frac{1}{\bar{\mu}_i} + W_{i1l}(\mathbf{K}) \right). \tag{55}$$

Having obtained results both for the queue size factorial moments, $\beta_{ij}(\mathbf{K})$, in Section III and the ordinary moments of the flow time distribution $T_{ijs}(\mathbf{K})$ (see above), we now present a relationship between them analogous to a well-known Little's Law type of relationship for the $M/G/1$ queue in isolation. Appendix B shows that these quantities are related via the j-fold summation

$$\beta_{ij}(\mathbf{K}) = \sum_{s_1=1}^{R} \cdots \sum_{s_j=1}^{R} \bar{\lambda}_{is_1}(\mathbf{K}) \bar{\lambda}_{is_2}(\mathbf{K} - \mathbf{e}_{s_1}) \cdots \bar{\lambda}_{is_j}\left(\mathbf{K} - \sum_{q=1}^{j-1} \mathbf{e}_{s_q}\right) T_{ijs_j}\left(\mathbf{K} - \sum_{q=1}^{j-1} \mathbf{e}_{s_q}\right).$$

The special, single chain, case $R = 1$ gives

$$\beta_{ij}(K) = \left(\prod_{l=K-j+1}^{K} \bar{\lambda}_i(l) \right) T_{ij}(K - j + 1), \tag{56}$$

where $\bar{\lambda}_i(l)$ is the node i arrival rate when our closed system contains l customers and $T_{ij}(K - j + 1)$ is the jth moment of the node i flow time when our closed system contains $K - j + 1$ customers. Relation (56) is analogous to a known result[19] for an $M/G/1$ queue in isolation which states

$$\beta_j = \lambda^j T_j. \tag{57}$$

Noting that λ^j is replaced by the product of flow rates, each with a different population, we see that (56) generalizes (57) in the sense of holding for a queue embedded in a closed network with the corresponding, complicated arrival process.

VI. DELAY AND FLOW TIME MOMENTS—MIXED SYSTEMS

To obtain the mixed network results, we consider the augmented system as in Section IV. We obtain recursions for the delay and flow time moments at a FCFS node with a single state-independent server

* As pointed out in Section VI, the virtual work results also apply to a limited class of processor sharing or LCFS-PR nodes; the limitations being that the service times are exponential with $\mu_{is} = \mu_{il}$ (i.e., the same assumptions being made for the FCFS nodes). Inclusion of this computation in the closed network algorithm shown in Fig. 4 is straightforward.

embedded in a mixed network. Results are obtained for delays experienced by customers in each of the closed chains and for customers in any* of the open chains, thus, yielding moments for the virtual delay or unfinished work at a node.

Denoting $\bar{\mu}_i^{-1}$ as the common mean service time at node i (i.e., $\bar{\mu}_i = \mu_{is} = \mu_{il}$)† and ρ_i^o as the actual utilization at node i due to customers belonging to open chains,

$$\rho_i^o = \sum_{l=r+1}^{R} \frac{\lambda_{il}}{\bar{\mu}_i}, \tag{58}$$ ‡

we obtain

$$T_{ijs}(\mathbf{K}) = \sum_{l=1}^{r} \frac{\bar{\lambda}_{il}(\mathbf{K}-\mathbf{e}_s)}{\bar{\mu}_i(1-\rho_i^o)} T_{ijl}(\mathbf{K}-\mathbf{e}_s) + \frac{j}{\bar{\mu}_i} T_{i,j-1,s}(\mathbf{K}) + \frac{j\rho_i^o}{\bar{\mu}_i(1-\rho_i^o)} T_{i,j-1}^o(\mathbf{K}-\mathbf{e}_s); \qquad s \le r, \tag{59}$$

where $T_{i,j}^o(\mathbf{K})$ is the jth moment of the flow time experienced by customers belonging to any open chain at node i and satisfies

$$T_{ij}^o(\mathbf{K}) = \sum_{l=1}^{r} \frac{\bar{\lambda}_{il}(\mathbf{K})}{\bar{\mu}_i(1-\rho_i^o)} T_{ijl}(\mathbf{K}) + \frac{j}{\bar{\mu}_i(1-\rho_i^o)} T_{i,j-1}^o(\mathbf{K}). \tag{60}$$

The recursion (60) is initialized with

$$T_{ij}^o(\mathbf{0}) = \frac{j}{\bar{\mu}_i(1-\rho_i^o)} T_{i,j-1}^o(\mathbf{0}) = \frac{j!}{[\bar{\mu}_i(1-\rho_i^o)]^j}, \tag{61}$$

the open network flow time moments, and (59) is initialized with

$$T_{ijs}(\mathbf{e}_s) = T_{ij}^o(\mathbf{0}). \tag{62}$$

The initialization (62) can be obtained by recognizing that the distribution of the number of customers found at node i by the single closed chain customer is identical to the distribution of the number at node i at an arbitrary time point, in equilibrium, in a system with no customers belonging to closed chains, i.e., $\mathbf{K} = \mathbf{0}$. Since the distribution at an arbitrary point in time is identical with that seen by an arbitrary open customer, we have (62).§

The results for the moments of the delay distributions can be found using a similar argument. For the closed chains, this results in

---

* We note that from Ref. 20 it is easy to show that delays experienced by customers belonging to different open chains have the same moments.

† Recall that this condition is required at a FCFS node in order to have a product form solution.

‡ Since node i has a state-independent processing rate, $\mu_i(k) = 1$ and ρ_i^o is the actual utilization as indicated.

§ An alternate way of obtaining (62) is by use of (59) at $\mathbf{K} = \mathbf{e}_s$, use of (61) and induction.

$$W_{ijs}(\mathbf{K}) = \sum_{l=1}^{r} \frac{\bar{\lambda}_{il}(\mathbf{K}-\mathbf{e}_s)}{\bar{\mu}_i(1-\rho_i^o)} W_{ijl}(\mathbf{K}-\mathbf{e}_s) + \frac{j}{\bar{\mu}_i} W_{i,j-1,s}(\mathbf{K})$$
$$+ \frac{j\rho_i^o}{\bar{\mu}_i(1-\rho_i^o)} W_{i,j-1}^o(\mathbf{K}-\mathbf{e}_s); \qquad s \le r; \quad j > 1, \quad (63)$$

where $W_{i,j}^o(\mathbf{K})$, the jth moment of the delay experienced by customers belonging to the open chains, is also the jth moment of the virtual delay or unfinished work at node i at an arbitrary point in time. The corresponding result for $j = 1$ is

$$W_{i1s}(\mathbf{K}) = \sum_{l=1}^{r} \frac{\bar{\lambda}_{il}(\mathbf{K}-\mathbf{e}_s)}{\bar{\mu}_i(1-\rho_i^o)} \left(\frac{1}{\bar{\mu}_i} + W_{i1l}(\mathbf{K}-\mathbf{e}_s)\right)$$
$$+ \frac{\rho_i^o}{\bar{\mu}_i(1-\rho_i^o)}; \qquad s \le r; j = 1. \quad (64)$$

The moments of the virtual delay, or delays experienced by open customers, are given by

$$W_{ij}^o(\mathbf{K}) = \sum_{l=1}^{r} \frac{\bar{\lambda}_{il}(\mathbf{K})}{\bar{\mu}_i(1-\rho_i^o)} W_{ijl}(\mathbf{K})$$
$$+ \frac{j}{\bar{\mu}_i(1-\rho_i^o)} W_{i,j-1}^o(\mathbf{K}); \qquad j > 1, \quad (65)$$

and for $j = 1$

$$W_{i1}^o(\mathbf{K}) = \sum_{l=1}^{r} \frac{\bar{\lambda}_{il}(\mathbf{K})}{\bar{\mu}_i(1-\rho_i^o)} \left(\frac{1}{\bar{\mu}_i} + W_{i1l}(\mathbf{K})\right) + \frac{\rho_i^o}{1-\rho_i^o}\frac{1}{\bar{\mu}_i}; \qquad j = 1. \quad (66)$$

We note that for both the open and closed chains, the jth moment of the delay and node flow time distributions satisfy the same recursions, (59) and (63), for the closed chains and (60) and (65) for the open chains, the difference lying in the boundary equations (64) and (66). The initial conditions

$$W_{i1l}(\mathbf{0}) = 0; \qquad l \le r,$$

together with (66) and (65), give

$$W_{ij}^o(\mathbf{0}) = \frac{j}{\bar{\mu}_i}\frac{\rho_i^o}{1-\rho_i^o} W_{i,j-1}^o(\mathbf{0}) = j!\left[\frac{\rho_i^o}{\bar{\mu}_i(1-\rho_i^o)}\right]^j, \quad (67)$$

the open network result. Using the same argument as was used for the closed flow time moments, we have

$$W_{ijs}(\mathbf{e}_s) = W_{ij}^o(\mathbf{0}) \quad (68)$$

which can be used to initialize (63) and (64). It is not necessary to have a separate computation for $W_{ijs}(\mathbf{K})$ if the flow time moments have been computed since they are related by

$$W_{ijs}(\mathbf{K}) = \sum_{l=1}^{r} \frac{\bar{\lambda}_{il}(\mathbf{K} - \mathbf{e}_s)}{\bar{\mu}_i} T_{ijl}(\mathbf{K} - \mathbf{e}_s)$$

$$+ \rho_i^o T_{ij}^o(\mathbf{K} - \mathbf{e}_s); \qquad s \le r \quad (69)$$

and

$$W_{ij}^o(\mathbf{K}) = \sum_{l=1}^{r} \frac{\bar{\lambda}_{il}(\mathbf{K})}{\bar{\mu}_i} T_{ijl}(\mathbf{K}) + \rho_i^o T_{ij}^o(\mathbf{K}); \qquad s > r \qquad (70)*$$

for the open chains or virtual work.

To obtain the moments of virtual delay for closed systems, denoted by $V_{ij}(\mathbf{K})$, we observe that

$$V_{ij}(\mathbf{K}) = \lim_{\rho_i^o \to o} W_{ij}^o(\mathbf{K}),$$

which results in the relations given in (66) and (67).

As a final note here we observe that in addition to applying to the FCFS nodes as stated, the virtual work results, W_{ij}^o, V_{ij}, apply to a limited class of processor sharing or LCFS-PR nodes† the limitation being that service times are exponential with rates $\mu_{is} = \mu_{il} = \bar{\mu}_i$. This follows from the insensitivity of the stationary queue size distribution to the queueing discipline for this case and the memoryless property of the exponential distribution.

VII. DELAY DISTRIBUTIONS

We consider a multi-job-type closed‡ network containing either state-independent service centers or infinite server nodes, e.g., central server model of multiprogramming, and present results for the tail of the delay distribution experienced by a type s job arrival to a FCFS service center embedded in a product-form network. Appendix C, which uses multidimensional generating functions and the known relation between the stationary distributions and those seen by customer arrivals to a node,[21] contains the details of the investigation.

We denote

$$\bar{W}_{is}(t; \mathbf{K}) = P(d_{is} > t; \mathbf{K}) \qquad (71)$$

as the probability that a chain s arrival to node i is delayed in excess of t in a system with population $\mathbf{K}$. Figure 8 shows a single-chain example where the delay shown corresponds to the time between a

* The inclusion of the results of this section into the mixed network algorithm shown in Fig. 6 is straightforward.
† Or other types not affecting the stationary queue-size distributions, e.g., random selection or LCFS nonpreemptive. The identification as actual delay moments is, however, lost.
‡ It is a straightforward matter to treat the limiting mixed network, as well as a class of state-dependent nodes (e.g., multiprocessor nodes) by the techniques presented.

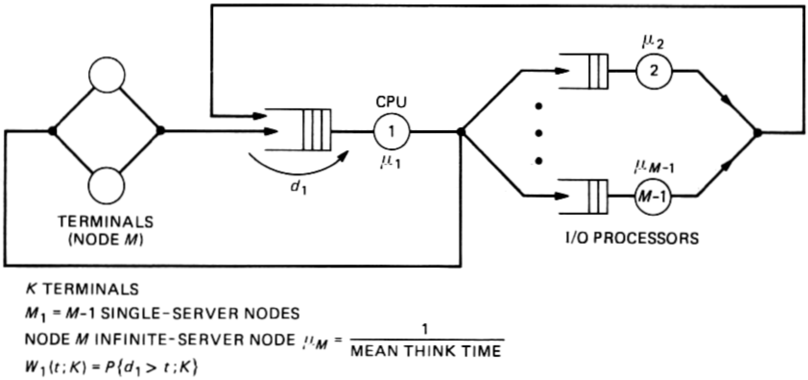


Fig. 8—Closed system illustration—delay distribution.

terminal request and the time the request first gets the attention of the CPU in a system with K terminals. References 22 and 23 contain a study of the response time distribution (queueing and service) for a single node being fed traffic from a collection of terminals, the classical machine repair problem with multiple repairmen. This could correspond to a multiprocessor version of Fig. 8, but without the I/O processors. In Ref. 24 the asymptotic behavior, as the number of terminals increases, is studied. We note that the methods described in this paper can easily be used to obtain results for the response time distribution (delay plus service time) for the above example.* In general, the state of the art for obtaining response time distributions for multiple resource systems is quite limited.†

Figure 9 shows a representation of two loosely coupled systems with shared mass storage devices modeled as two central server models with some shared I/O queues.[29] For I/O requests served on a FCFS basis in each I/O queue, the delay distribution of interest shown corresponds to the delay in accessing a disk, either dedicated or shared, for each of the component systems and to the delay in getting each of the CPU's for FCFS scheduling algorithms. We note that the population vector here, $\mathbf{K} = (K_1, K_2)$, could correspond to the degree of multiprogramming for each component system.

We start our presentation of results with the single-chain case and then generalize to the multichain case. For the single-chain case, letting

$$\bar{W}_i(t; K) = P(d_i > t; K), \tag{72}$$

* The single node being a limited queue-dependent server.[10]

† We note that these response times could involve several visits to a given resource, as well as visits to other resources. This is studied in Ref. 25 for the single resource queue with feedback. Some other recent work is discussed in Section VIII.[26,27,28]

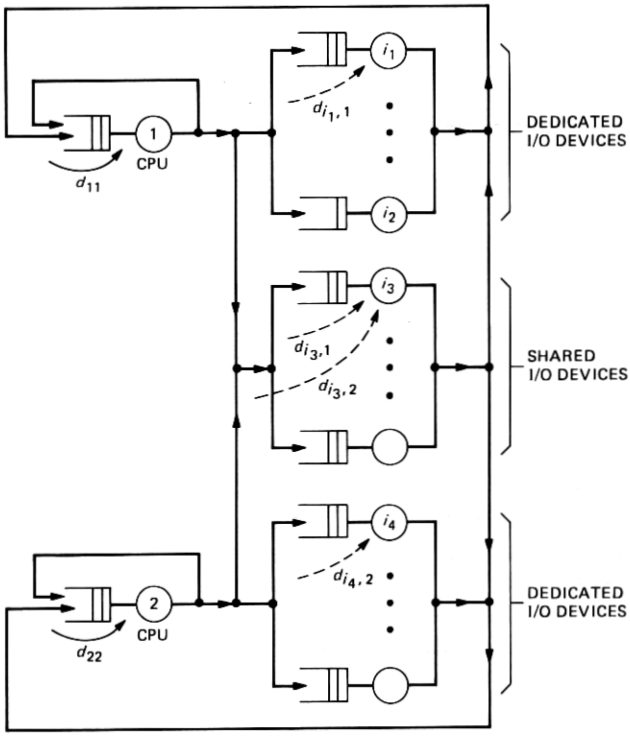


Fig. 9—Two systems sharing mass storage devices.

it is shown in Appendix C that $\bar{W}_i(t; K)$ satisfies the M_1^{th} order recursion

$$\bar{W}_i(t; K) = y_i(t; K-1) - \sum_{j=1}^{M_1} \bar{\alpha}_j(K-1)\bar{W}_i(t; K-j), \tag{73}$$

where $M_1 \leq M$ is the number of single-server nodes in the network. The coefficients $\bar{\alpha}_j(K-1)$ are related to the coefficients of Z^j, α_j, in

$$P(Z) = \prod_{i=1}^{M_1}\left(1 - \frac{\lambda_i}{\mu_i}Z\right) = \sum_{j=0}^{M_1} \alpha_j Z^j, \qquad (74)^*$$

where λ_i is the relative arrival rate to node i.

Clearly,

$$\alpha_j = (-1)^j \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq M_1} \rho_{i_1}\rho_{i_2} \cdots \rho_{i_j}, \tag{75}$$

where ρ_{i_j} is the relative utilization λ_{i_j}/μ_{i_j}. The coefficients of (73) $\bar{\alpha}_j(K-1)$ are obtained from

* We have arbitrarily labeled nodes 1 through M_1 to correspond to the single-server nodes. Since these nodes are state-independent service centers, in this section we denote the common mean service time at node i as μ_i^{-1}, $i = 1, 2, \cdots, M_1$.

$$\bar{\alpha}_j(K-1) = (-1)^j \sum_{1\le i_1<\cdots<i_j\le M_1} \bar{\rho}_{i_1}(K-1)$$

$$\cdot\bar{\rho}_{i_2}(K-2)\cdots\bar{\rho}_{i_j}(K-j), \quad (76)$$

which corresponds to (75) with the actual utilizations

$$\bar{\rho}_{i_j}(K-j) = \frac{\bar{\lambda}_{i_j}(K-j)}{\mu_{i_j}} \quad (77)^*$$

evaluated at the appropriate population replacing the relative utilizations ρ_{i_j}. Recall that the actual utilizations are available via standard analysis. The forcing function on difference equation (73), $y_i(t; K-1)$, can be obtained recursively (see Appendix C) from

$$y_i(t; K) = \frac{\mu_i t}{K-1}\bar{\rho}_i(K)\left(1+\frac{a_\infty}{\lambda_i t}\right) y_i(t; K-1) \quad (78)$$

and the initial condition

$$y_i(t; 1) = e^{-\mu_i t}\bar{\rho}_i(1). \quad (79)$$

The quantity a_∞ represents the relative loading on all the infinite-server nodes

$$a_\infty = \sum_{j=M_1+1}^{M} \frac{\lambda_j}{\mu_j}. \quad (80)$$

We note that for the network example of Fig. 8,

$$y_1(t; K) = \frac{1}{K-1}\bar{\rho}_1(K)(pr+\mu_1 t)\, y_1(t; K-1),$$

where p is the fraction of CPU requests feedback to the terminals,

$$r = \frac{\text{mean think time}}{\text{mean CPU service time}},$$

and $\mu_i t$ represents the point at which we are evaluating the tail of the delay distribution in units of CPU service times.

For the multi-job-type networks, the tail of the node i delay distribution as seen by a chain s arrival is shown in Appendix C to satisfy the multidimensional recursion

$$\bar{W}_{is}(t; \mathbf{K}) = y_i(t; \mathbf{K}-\mathbf{e}_s) - \sum_{(0<j_1+j_2+\cdots+j_R\le M_1)} \bar{\alpha}_{j_1,j_2,\cdots j_R}(\mathbf{K}-\mathbf{e}_s)$$

$$\cdot\bar{W}_{is}(t; \mathbf{K}-j_1\mathbf{e}_1-j_2\mathbf{e}_2-\cdots-j_R\mathbf{e}_R). \quad (81)$$

The quantities $\bar{\alpha}_{j_1,\ldots,j_R}(\mathbf{K}-\mathbf{e}_s)$ are the multidimensional analogues of $\bar{\alpha}_j(K-1)$ in (76), i.e., they are related to the coefficients of $Z_1^{j_1}\cdot$

* Here $\bar{\lambda}_i(K)$ denotes the actual arrival rate to node i for a system with population K.

$Z_2^{j_2} \cdots Z_R^{j_R}$, $\alpha_{j_1,\ldots,j_R}$, in

$$P(\mathbf{Z}) = \prod_{i=1}^{M_1} \left(1 - \sum_{s=1}^{R} \frac{\lambda_{is}}{\mu_{is}} Z_s\right). \tag{82}$$

Analogous to (76), we have

$$\bar{\alpha}_{j_1 \cdots j_R}(\mathbf{K} - \mathbf{e}_s) = (-1)^{j_1 + \cdots j_R} \sum_{\{i_{pq}\} \in L} \{[\bar{\rho}_{i_{1,1};1}(\mathbf{K} - \mathbf{e}_s) \cdots \bar{\rho}_{i_{j_1,1};1}(\mathbf{K} - \mathbf{e}_s - (j_1 - 1)\mathbf{e}_1)] \cdot \cdots [\bar{\rho}_{i_{1R};R}(\mathbf{K} - \mathbf{e}_s - j_1\mathbf{e}_1 - \cdots j_{R-1}\mathbf{e}_{R-1}) \cdots \bar{\rho}_{i_{j_R,R};R}(\mathbf{K} - \mathbf{e}_s - j_1\mathbf{e}_1 - \cdots -(j_R - 1)\mathbf{e}_R)]\},$$

where

$$0 < \sum_{r=1}^{R} j_r \leq M_1 \tag{83}$$

and $\{i_{pq}\} \in L$ corresponds to

$$\begin{pmatrix} i_{11} < i_{21} < \cdots < i_{j_1 1} \\ \vdots \\ i_{1R} < i_{2R} < \cdots i_{j_R R} \end{pmatrix}; \qquad i_{mn} \neq i_{jk}; \tag{84}$$

i.e., $\bar{\alpha}_{j_1, j_2, \cdots j_R}(\mathbf{K} - \mathbf{e}_s)$ is a sum of products of utilizations with $\bar{\rho}_{i_{p,q};r}$ evaluated at the population $(\mathbf{K} - \mathbf{e}_s - j_1\mathbf{e}_1 - j_2\mathbf{e}_2 - \cdots j_{r-1}\mathbf{e}_{r-1} - (p-1)\mathbf{e}_r)$. The forcing function is obtained recursively from

$$y_i(t; \mathbf{K}) = \begin{cases} \dfrac{(\bar{a}_{s\infty}(\mathbf{K}) + \bar{\lambda}_{is}(\mathbf{K})t)}{K_s} y_i(t; \mathbf{K} - \mathbf{e}_s); \\ \quad s \notin \mathscr{R}(i),\ s \in \mathscr{R}^*(i),\ K_r = 0\ \forall r \notin \mathscr{R}^*(i), \\ \dfrac{(\bar{a}_{s\infty}(\mathbf{K}) + \bar{\lambda}_{is}(\mathbf{K})t)\delta(\mathbf{K})}{K_s\delta(\mathbf{K} - \mathbf{e}_s)} y_i(t; \mathbf{K} - \mathbf{e}_s); \\ \quad s \in \mathscr{R}(i),\ K_r = 0\ \forall r \notin \mathscr{R}^*(i),\ \delta(\mathbf{K} - \mathbf{e}_s) > 0^{\dagger} \\ 0; \qquad K_r > 0 \quad \text{for some} \quad r \notin \mathscr{R}^*(i), \end{cases} \tag{85}$$

with initial condition

$$y_i(t; \mathbf{e}_r) = \begin{cases} \bar{\rho}_{ir}(\mathbf{e}_r)e^{-\mu_i t}; & r \in \mathscr{R}(i) \\ 0 & r \notin \mathscr{R}(i). \end{cases} \tag{86}$$

$\mathscr{R}(i)$ denotes the set of chains passing through node i, and $\mathscr{R}^*(i)$ denotes the set of chains which either pass through node i or through an infinite-server node. The quantity

$$\delta(\mathbf{K}) = \sum_{r \in \mathscr{R}(i)} \frac{K_r}{\left(1 + \dfrac{a_{r\infty}}{\lambda_{ir}t}\right)}, \tag{87}$$

† If $\delta(\mathbf{K} - \mathbf{e}_s) = \mathbf{0}$ and $\mathbf{K} - \mathbf{e}_s \neq \mathbf{0}$ [if $\mathbf{K} - \mathbf{e}_s = \mathbf{0}$, use initial condition (86)] $y(t; \mathbf{K})$ can be computed from the first relation for some $s \notin \mathscr{R}(i)$, $s \in \mathscr{R}^*(i)$ corresponding to $K_s > 0$.

the relative chain r loading on the infinite-server nodes

$$a_{r\infty} = \sum_{j=M_1+1}^{M} \frac{\lambda_{jr}}{\mu_{jr}}, \tag{88}$$

and the actual chain r infinite-server loading

$$\bar{a}_{r\infty}(\mathbf{K}) = \sum_{j=M_1+1}^{M} \frac{\bar{\lambda}_{jr}(\mathbf{K})}{\mu_{jr}} = \bar{\lambda}_r(K) a_{r\infty}. \tag{89}$$

We note that it is only necessary to compute $y_i(t, \mathbf{K})$ in the subspace spanned by chains passing through node i or any infinite-server node. For the special case where $M_1 = M$ (e.g., the central server model) $\mathscr{R}(i) = \mathscr{R}^*(i)$ and (85) becomes

$$y_i(t, \mathbf{K}) = \begin{cases} \dfrac{\mu_i t}{K_s} \dfrac{|\mathbf{K}|}{|\mathbf{K}| - 1} \bar{\rho}_{is}(\mathbf{K})\, y_i(t, \mathbf{K} - \mathbf{e}_s); \\ \qquad s \in \mathscr{R}(i),\ K_r = 0\ \forall r \notin \mathscr{R}(i),\ |\mathbf{K}| > 1, \\ 0; \qquad K_r > 0 \quad \text{for some } r \notin \mathscr{R}(i), \end{cases} \tag{90}$$

where

$$|\mathbf{K}| = \sum_{r=1}^{R} K_r. \tag{91}$$

VIII. SUMMARY

This paper has presented contributions to the foundations of a tool to support performance analysis and modeling activities aimed at answering some key questions at various stages of a computer system's life cycle. The emphasis here has been on presenting easily and efficiently computable results for calculating distributional information and a stable, efficient method for dealing with congestion adaptive devices.† Mixed systems have been considered to allow us the generality of dealing with traffic sources which are fundamentally different in their behavior. By obtaining results for different levels of customer aggregation, we allow one to consider a macroscopic or more microscopic level of detail. The virtual delay results allow us to quantify differences between service as perceived by an arriving customer and that perceived by a measuring device.

We note that many open questions exist in the areas of obtaining results related to the distribution of total time a customer spends in a subnetwork consisting of several nodes with feedback (e.g., Fig. 1,

† In a recent paper,[30] a modification of mean value analysis is introduced to eliminate numerical instabilities when dealing with general state-dependent service rates. The method involves analysis of complementary systems and evaluation of marginal queue size distributions.

where the time of interest is the response time).* This is an area of active research in the literature (e.g., see Refs. 26 and 28); however, the results available are fairly restrictive with respect to the network topology† or customer paths, and do not apply to, for example, the network of Fig. 1. Approximating the moments of the distribution of the overall time to transit a network from the individual node flow time moments is one possible approach which would have to be studied and evaluated. The problem arises because of statistical dependence of a given customer's flow times as he sojourns the network. For open Jackson networks, Reiman[33] uses a heavy traffic limit theorem to obtain a diffusion approximation for the network sojourn times. Another area of importance relates to the inclusion of priorities in, for example, the CPU schedule. We note that an approximation technique (based on utilization adjustments) does exist[9] for handling a class of priority disciplines and can perform quite satisfactorily in many cases.‡ The approximation is such that it enables us to compute performance measures using results in this paper.

APPENDIX A

Recursions for Queue Size Factorial Moments

For closed systems, it is known[1] that the marginal, stationary probability distributions satisfy

$$p(k_i = k; \mathbf{K}) = \frac{1}{\mu_i(k)} \sum_{s=1}^{R} \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_{is}} \cdot p(k_i = k-1; \mathbf{K} - \mathbf{e}_s); \qquad k > 0. \quad (92)$$

While conceptually, the desired moments could be computed from recursively computed marginal probability distributions, it is not recommended. Other, more computable, approaches could involve the use of generating functions.[35] Our approach is to directly obtain a recursive relation for the moments. We use (92), (8), and (9) to obtain (10), with the indicated initial conditions in a straightforward manner. We obtain the required node-chain throughputs, $\bar{\lambda}_{is}(\mathbf{K})$, by considering the lower level aggregation

* Or, for that matter, at a processor sharing node embedded in a general closed network. The waiting time distribution for a specific closed network consisting of a single processor sharing node fed by a single finite population class is treated in Ref. 31.

† In Ref. 27 a computational methodology is given for obtaining upper and lower bounds where an arbitrary network topology is allowed. When applied to two $M/M/1$ queues in tandem (for which the exact solution is known) the upper and lower bounds are close; however, over 30,000 states were used in the computation at 80 percent occupancy. Computational aspects are presented in Ref. 32.

‡ Reference 34, which obtains a closed form solution for a two-node closed network with priorities, proposes a criterion under which the approximation technique would be expected to perform well.

$$\mathbf{k}_i = (k_{i1}, k_{i2}, \cdots, k_{iR}),$$

where k_{ir} represents the number of chain r customers at node i. Denoting k_i as the total number of customers at node i, we can write the recursive relation

$$p_i(\mathbf{k}_i; \mathbf{K}) = \frac{k_i}{k_{is}} \frac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_i(k_i)\mu_{is}} p_i(\mathbf{k}_i - \mathbf{e}_s; \mathbf{K} - \mathbf{e}_s); \qquad k_{is} > 0, \tag{93}$$

which is obtainable from the product form solution. From (93), (8), the standard Little's law argument at each node and about the entire system, and the irreducibility of the routing chains we get (12) through (15). To obtain the desired recursion for the joint factorial moments, we use (8) and (93) in (18), make the appropriate identification corresponding to a system with reduced population and obtain (21).

The mixed system result (26) is obtained directly from (10) by considering the augmented system with population

$$\mathbf{K}' = (\mathbf{K} | K_{r+1}, \cdots, K_R),$$

denoting

$$\beta_{ij}(\mathbf{K}) = \lim_{K_{r+1}, \cdots, K_R \to \infty} \beta_{ij}(\mathbf{K}')$$

and decomposing the sum over the open and closed chains. The lower level aggregation results are obtained by use of

$$\lim_{K_{r+1}, \cdots, K_R \to \infty} p_i(\mathbf{k}_i; \mathbf{K}') = \begin{cases} \dfrac{k_i}{k_{is}} \dfrac{\lambda_{is}}{\mu_i(k_i)\mu_{is}} p_i(\mathbf{k}_i - \mathbf{e}'_s; \mathbf{K}); & s > r \\[2ex] \dfrac{k_i}{k_{is}} \dfrac{\bar{\lambda}_{is}(\mathbf{K})}{\mu_i(k_i)\mu_{is}} p_i(\mathbf{k} - \mathbf{e}'_s; \mathbf{K} - \mathbf{e}_s); & s \le r, \end{cases} \tag{94}$$

where $\mathbf{e}'_s$ is an R-dimensional unit vector in direction s as opposed to $\mathbf{e}_s$, the corresponding r dimension vector for $s \le r$.

APPENDIX B

Recursions for Node Delay and Flow Time Moments

Denoting $\bar{W}_{is}(t, \mathbf{K})$ as the complementary delay distribution experienced by a chain s customer at node i when the system population is $\mathbf{K}$, we can write

$$\bar{W}_{is}(t, \mathbf{K}) = \sum_{k=1}^{|\mathbf{K}|-1} \frac{(\mu_{is} t)^{k-1}}{(k-1)!} e^{-\mu_{is} t} p(k_i(t_{is}^-) \ge k; \mathbf{K}), \tag{95}$$

where $\mu_{is} = \mu_i$ at the FCFS state-independent node under consideration, $k_i(t_{is}^-)$ represents the number of customers at node i seen by a chain s

arrival to node i, and $|\mathbf{K}| = K_1 + K_2 + \cdots + K_R$. For the class of closed* systems, we are initially considering

$$p(k_i(t_{is}^-) = k; \mathbf{K}) = p(k_i = k; \mathbf{K} - \mathbf{e}_s), \tag{96}$$

i.e., the distribution as seen by an arriving chain s customer is equal to the distribution at an arbitrary point in time in equilibrium (i.e., the stationary distribution) for a system with one less chain s customer.[20] Using (96) in (95), we can obtain the Laplace-Stieltjes transform of the flow time distribution for node-chain pair (i, s)

$$T_{is}(\eta, \mathbf{K}) = \frac{\mu_i}{\eta + \mu_i} - \frac{\eta}{\mu_i} \sum_{k=1}^{|\mathbf{K}|-1} p(k_i \geq k; \mathbf{K} - \mathbf{e}_s) \left(\frac{\mu_i}{\eta + \mu_i}\right)^{k+1} \tag{97}$$

The jth moment, obtained by differentiation, satisfies

$$T_{ijs}(\mathbf{K}) = \frac{j}{\mu_i} T_{i,j-1,s}(\mathbf{K}) + \frac{j}{\mu_i^j} \sum_{k=1}^{|\mathbf{K}|-1} k(k+1) \cdots (k+j-2) \cdot p(k_i \geq k; \mathbf{K} - \mathbf{e}_s), \tag{98}$$

and the summation in (98) can be written as

$$\sum_{q=o}^{|\mathbf{K}|-2} (q+1) \cdots (q+j-1) \sum_{l=1}^{R} \frac{\bar{\lambda}_{il}(\mathbf{K} - \mathbf{e}_s)}{\mu_i} \cdot p(k_i \geq q; \mathbf{K} - \mathbf{e}_l - \mathbf{e}_s), \tag{99}$$

where we have used (92) to get the one-step recursion on the tail of the marginal queue size distribution. Inserting (99) in (98) and making the appropriate identification, we get (46), with initial and boundary conditions as indicated.

To obtain the Little's Law type moment relation, we use the definition of $\beta_{ij}(\mathbf{K})$ and summation by parts to get

$$\beta_{ij}(\mathbf{K}) = j \sum_{l=0}^{|\mathbf{K}|-j} (l+1)(l+2) \cdots (l+j-1) p(k_i \geq l+j; \mathbf{K}), \tag{100}$$

which upon repeated application of (92) results in

$$\beta_{ij}(\mathbf{K}) = \sum_{s_1=1}^{R} \cdots \sum_{s_j=1}^{R} \bar{\lambda}_{is_1}(\mathbf{K}) \bar{\lambda}_{is_2}(\mathbf{K} - \mathbf{e}_{s_1}) \cdots \bar{\lambda}_{is_j}\left(\mathbf{K} - \sum_{q=1}^{j-1} \mathbf{e}_{s_q}\right) \cdot \frac{j}{\mu_i^j} \sum_{l=0}^{|\mathbf{K}|-j} (l+1) \cdots (l+j-1) p\left(k_i \geq l \colon \mathbf{K} - \sum_{q=1}^{j} \mathbf{e}_{s_q}\right). \tag{101}$$

Identifying the last term, we obtain

* The mixed system results obtained by the usual limiting system argument are reported in Section VI.

$$\beta_{ij}(\mathbf{K}) = \sum_{s_1=1}^{R} \cdots \sum_{s_j=1}^{R} \lambda_{is_1}(\mathbf{K})\lambda_{is_2}(\mathbf{K} - \mathbf{e}_{s_1}) \cdots \lambda_{is_j}\left(\mathbf{K} - \sum_{q=1}^{j-1} \mathbf{e}_{s_q}\right)$$

$$\times T_{ijs_j}\left(\mathbf{K} - \sum_{q=1}^{j-1} \mathbf{e}_{s_q}\right), \quad (102)$$

which for the special, single chain, case $R = 1$ gives the moment relation

$$\beta_{ij}(K) = \left(\prod_{l=K-j+1}^{K} \lambda_i(l)\right) T_{ij}(K - j + 1). \quad (103)$$

APPENDIX C

Recursions for the Tail of the Node Delay Distributions

We consider a closed system of M_1 state-independent, single-server nodes and $M - M_1$ infinite-server nodes and obtain an M_1^{th} order difference equation for the tail of the customer delay distribution at a FCFS node. Denoting

$$\bar{W}_{is}(t; \mathbf{K}) = P(d_{is} > t: \mathbf{K})$$

as the probability that a chain s arrival to node i is delayed in excess of t in a system with population $\mathbf{K}$, we can write

$$\bar{W}_{is}(t; \mathbf{K}) = \sum_{k=1}^{|\mathbf{K}|-1} \frac{(\mu_i t)^{k-1}}{(k-1)!} e^{-\mu_i t} p(k_i \geq k; \mathbf{K} - \mathbf{e}_s). \quad (104)$$

We denote the product-form normalization constant* as

$$G(\mathbf{K}) = \sum_{\sum_1^M \mathbf{k}_j = \mathbf{K}} g_1(\mathbf{k}_1) g_2(\mathbf{k}_2) \cdots g_M(\mathbf{k}_M),$$

where

$$g_j(\mathbf{k}_j) = \frac{|\mathbf{k}_j|!}{k_{j1}! \cdots k_{jR}!} \left(\frac{\lambda_{j1}}{\mu_{j1}}\right)^{k_{j1}} \cdots \left(\frac{\lambda_{jR}}{\mu_{jR}}\right)^{k_{jR}};$$

$$j = 1, 2, \cdots, M_1,$$

$$g_j(\mathbf{k}_j) = \frac{\left(\frac{\lambda_{j1}}{\mu_{j1}}\right)^{k_{j1}} \cdots \left(\frac{\lambda_{jR}}{\mu_{jR}}\right)^{k_{jR}}}{k_{j1}! \cdots k_{jR}!}; \qquad j = M_1 + 1, \cdots M, \quad (105)$$

and we have labeled nodes $j = 1, \cdots, M_1$ as the single-server nodes. Multiplying (104) by $G(\mathbf{K} - \mathbf{e}_s)$ and obtaining the generating function, we have

---

* Our final result will not involve computation of the normalization constant, the calculation of which can result in numerical problems.

$$L_{is}(\mathbf{Z}) = \frac{S_i(\mathbf{Z})\left(\sum_{r=1}^{R}\frac{\lambda_{ir}}{\mu_i}Z_r\right)}{\left(1-\sum_{r=1}^{R}\frac{\lambda_{ir}}{\mu_i}Z_r\right)}\exp\left[-\mu_i t\left(1-\sum_{r=1}^{R}\frac{\lambda_{ir}}{\mu_i}Z_r\right)\right], \quad (106)$$

where

$$L_{is}(\mathbf{Z}) = \sum_{K_1=0}^{\infty}\cdots\sum_{K_R=0}^{\infty}\bar{W}_{is}(t;\mathbf{K}+\mathbf{e}_s)G(\mathbf{K})Z_1^{K_1}\cdots Z_R^{K_R}, \quad (107)$$

and

$$S_i(\mathbf{Z}) = \sum_{K_1=0}^{\infty}\cdots\sum_{K_R=0}^{\infty}\mathcal{G}_i(\mathbf{K})Z_1^{K_1}\cdots Z_R^{K_R}, \quad (108)$$

where $\mathcal{G}_i(\mathbf{K})$ is the normalization constant for a reduced network (node i absent)

$$\mathcal{G}_i(\mathbf{K}) = \sum_{\sum_{j\neq i}\mathbf{k}_j=\mathbf{K}} g_1(\mathbf{k}_1)\cdots g_{i-1}(\mathbf{k}_{i-1})\,g_{i+1}(\mathbf{k}_{i+1})\cdots g_R(\mathbf{k}_R). \quad (109)$$

Inserting

$$S_i(\mathbf{Z}) = \prod_{j\neq i}\bar{S}_j(\mathbf{Z}), \quad (110)$$

where

$$\bar{S}_j(\mathbf{Z}) = \begin{cases} \dfrac{1}{\left(1-\sum_{r=1}^{R}\frac{\lambda_{jr}}{\mu_{jr}}Z_r\right)}; & j\leq M_1, \\ e^{\sum_{r=1}^{R}\frac{\lambda_{jr}}{\mu_{jr}}Z_r}; & j > M_1. \end{cases} \quad (111)$$

into (106), we get

$$P(\mathbf{Z})L_{is}(\mathbf{Z}) = e^{-\mu_i t}\left(\sum_{r=1}^{R}\frac{\lambda_{ir}}{\mu_i}Z_r\right)e^{\sum_{r=1}^{R}(a_{r\infty}+\lambda_{ir}t)\mathbf{Z}_r} \triangleq H_i(t,\mathbf{Z}), \quad (112)$$

where $P(\mathbf{Z})$ is the polynomial given by (82) and $a_{r\infty}$ given by (88). Inversion of (112) and division by the appropriate normalization constant results in

$$\sum_{\substack{j_1=0 \\ j_1+\cdots j_R\leq M_1}}^{M_1}\cdots\sum_{j_R=0}^{M_1}\alpha_{j_1,\cdots j_R}\frac{l_{is}(\mathbf{K}-\mathbf{e}_s-j_1\mathbf{e}_1-\cdots-j_R\mathbf{e}_R)}{G(\mathbf{K}-\mathbf{e}_s)}$$

$$=\frac{h_i(t,\mathbf{K}-\mathbf{e}_s)}{G(\mathbf{K}-\mathbf{e}_s)} \triangleq y_i(t,\mathbf{K}-\mathbf{e}_s), \quad (113)$$

where $\alpha_{j_1,\dots j_R}$ is the coefficient of $Z_1^{j_1}Z_2^{j_2}\cdots Z_R^{j_R}$ in the polynomial (82), $h_i(t;\mathbf{K})$ is the inverse generating function of $H_i(t;\mathbf{Z})$ given by

$$h_i(t,\mathbf{K}) = e^{-\mu_i t}\left[\prod_{r\in\mathscr{R}^*(i)} \frac{(a_{r\infty}+\lambda_{ir}t)^{K_r}}{K_r!}\right]\sum_{r\in\mathscr{R}(i)} \frac{\lambda_{ir}K_r}{\mu_i(a_{r\infty}+\lambda_{ir}t)}; \qquad (114)$$

$\mathscr{R}(i)$ denotes the set of chains passing through node i and $\mathscr{R}^*(i)$ denotes the set of chains which either pass through node i or an infinite server node. The quantity $l_{is}(\mathbf{K})$ in (113) satisfies

$$l_{is}(\mathbf{K}) = \bar{W}_{is}(t,\mathbf{K}+\mathbf{e}_s)G(\mathbf{K}). \qquad (115)$$

As it stands (113) represents an M_1^{th} order recursion for the distribution tail; however, the coefficient and forcing function involve the normalization constant. Upon use of (115) in (113), and recognizing that

$$\frac{G(\mathbf{K}-\mathbf{e}_s-j_1\mathbf{e}_1-\cdots-j_R\mathbf{e}_R)}{G(\mathbf{K}-\mathbf{e}_s)}$$

can be written as a product of node throughput proportionality constants

$$\left[\prod_{l_1=0}^{j_1-1}\bar{\lambda}_1(\mathbf{K}-\mathbf{e}_s-l_1\mathbf{e}_1)\right]\cdot\left[\prod_{l_2=0}^{j_2-1}\bar{\lambda}_2(\mathbf{K}-\mathbf{e}_s-j_1\mathbf{e}_1-l_2\mathbf{e}_2)\right]\cdots$$
$$\left[\prod_{l_R=0}^{j_R-1}\bar{\lambda}_R(\mathbf{K}-\mathbf{e}_s-j_1\mathbf{e}_1-\cdots-j_{R-1}\mathbf{e}_{R-1}-l_R\mathbf{e}_R)\right],$$

we obtain (81) and (83).

The recursions (85) for the forcing function

$$y_i(t;\mathbf{K}) = \frac{h_i(t,\mathbf{K})}{G(\mathbf{K})} \qquad (116)$$

follow from

$$h_i(t,\mathbf{K}) = \frac{(a_{s\infty}+\lambda_{is}t)}{K_s}\frac{\delta(\mathbf{K})}{\delta(\mathbf{K}-\mathbf{e}_s)}h_i(t,\mathbf{K}-\mathbf{e}_s); \qquad s\in\mathscr{R}(i), \qquad (117)$$

where $\delta(\mathbf{K})$ is given by (87), and from

$$h_i(t,\mathbf{K}) = \frac{(a_{s\infty}+\lambda_{is}t)}{K_s}h_i(t,\mathbf{K}-\mathbf{e}_s); \qquad s\notin\mathscr{R}(i),\ s\in\mathscr{R}^*(i). \qquad (118)$$

REFERENCES

1. M. Reiser and S. S. Lavenberg, "Mean Value Analysis of Closed Multichain Queueing Networks," J. ACM, *27*, No. 2 (April 1980), pp. 313–22.
2. A. A. Fredericks, "Analysis of the Impact of Multiprogramming on a Transaction Oriented System," IFIP Proc. Performance 1981, Amsterdam: North Holland, November 1981, pp. 489–502.
3. R. J. T. Morris and J. S. Kaufman, unpublished work.
4. P. J. Denning, "A Note on Paging Drum Efficiency," Computing Surveys, *4*, No. 1 (March 1972), pp. 1–3.

5. E. G. Coffman, Jr. and P. J. Denning, *Operating Systems Theory*, Englewood Cliffs, N.J.: Prentice Hall, 1973.
6. F. Baskett et al., "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," J. ACM, *22*, No. 2 (April 1975), pp. 248–60.
7. J. P. Buzen, "A Queueing Network Model of MVS," ACM Computing Surveys, *10*, No. 3 (September 1978), pp. 319–31.
8. H. Kobyashi, "System Design and Performance Analysis Using Analytic Models," *Current Trends in Programming Methodology, Volume III, Software Modeling*, eds. K. M. Chandy and R. T. Yeh, Englewood Cliffs, N.J.: Prentice Hall, 1978, pp. 72–114.
9. K. C. Sevcik, "Priority Scheduling Disciplines in Queueing Network Models of Computer Systems," *Proceedings of IFIP Congress 77*, Amsterdam, The Netherlands: North Holland, 1977, pp. 565–70.
10. M. Reiser and C. H. Sauer, "Queueing Network Models: Methods of Solution and Their Program Implementation," *Current Trends in Programming Methodology, Volume III, Software Modeling*, eds. K. M. Chandy, R. T. Yeh, Englewood Cliffs, N.J.: Prentice-Hall, 1978, pp. 115–67.
11. K. M. Chandy and C. H. Sauer, "Computational Algorithms for Product Form Queueing Networks," Proc. Performance 80—The 7th IFIP W. G. 7.3 International Symposium on Computer Performance Modeling, Measurement and Evaluation, Toronto, May 1980, Extra Papers, pp. 1–10.
12. J. P. Buzen, "Analysis of System Bottlenecks Using a Queueing Network Model," Proc. *ACM-SIGOPS Workshop on System Performance Evaluation*, New York: ACM April 1971, pp. 82–103.
13. D. Ferrari, *Computer System Performance Evaluation*, Englewood Cliffs, N.J.: Prentice-Hall, 1978.
14. F. P. Kelly, "Reversibility and Stochastic Networks," Chichester: John Wiley, 1979.
15. M. Reiser and A. G. Konheim, "Finite Capacity Queueing Systems with Applications in Computer Modeling," Siam J. Computing, *7*, No. 2 (May 1978), pp. 210–29.
16. G. Latouche, "Algorithmic Analysis of a Multiprogramming Multiprocessor Computer System," J. ACM *28*, No. 4 (October 1981), pp. 662–79.
17. B. Avi-Itzhak and D. P. Heyman, "Approximate Queueing Models for Multiprogramming Computer Systems," Operations Research, *21*, No. 6 (1973), pp. 1212–30.
18. C. H. Sauer and K. M. Chandy, *Computer Systems Performance Modeling*, Englewood Cliffs, N.J.: Prentice Hall, 1981.
19. D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, New York: John Wiley, 1974.
20. K. C. Sevcik and I. Mitrani, "The Distribution of Queueing Network States at Input and Output States," 4th Int. Symp. on Modeling and Performance Analysis of Computer Systems, Vienna (February 1979), pp. 319–35.
21. S. S. Lavenberg and M. Reiser, "Stationary State Probabilities at Arrival Instants for Closed Queueing Networks with Multiple Types of Customers," J. Appl. Prob., *17*, No. 4 (December 1980), pp. 1048–61.
22. A. Sekino, "Performance Evaluation of Multiprogrammed Time-Shared Computer System," Project MAC, M.I.T., MAC-TR-103, September 1972.
23. A. Sekino, "Response Time Distribution of Multiprogrammed Time-Shared Computer Systems," Proc. Sixth Annual Princeton Conf. on Information Sciences and Systems, pp. 613–9, 1972.
24. J. W. Wong, "Response Time Distribution of the $M/M/m/N$ Queueing Model," Operations Research, *27*, No. 6 (November–December 1979), pp. 1196–202.
25. S. S. Lam and A. U. Shankar, "Response Time Distributions for a Multi-class Queue with Feedback," Proc. of Performance 80—The 7th IFIP W. G. 7.3 Int. Symp. Computer Performance Modeling Measurement and Evaluation, Toronto (May 1980), pp. 225–34.
26. B. Melamed, unpublished work.
27. B. Melamed and M. Yadin, "Randomization Procedures in the Computation of Cumulative Time Distribution Over Discrete State Markov Processes," Dept. of Industrial Engineering, Northwestern University, Department of Industrial Engineering, Technion, Israel Institute of Technology, June, 1980.
28. J. Walrand and P. Varaiya, "Sojourn Times and the Overtaking Condition in Jacksonian Networks," Adv. Appl. Prob., *12* (December 1980), pp. 1000–18.
29. J. P. Buzen, "Modeling Computer System Performance," Proc. 7th Int. Conf. of the Computer Measurement Group, Atlanta (November 16–19), 1976, pp. 230–8.
30. M. Reiser, "Queue-Dependent Servers in Closed Queueing Networks," Performance Evaluation, *1* (January 1981) pp. 7–18.

31. D. Mitra and R. Pinsky, "Waiting Time Distributions from Closed Queueing Network Models of Interactive, Shared-Processor Systems," IFIP Proc. Performance 81, Amsterdam: North Holland, November 1981, pp. 113–31.
32. B. Melamed and M. Yadin, "Numerical Computation of Sojourn Time Distributions in Queueing Networks," Dept. of Ind. Eng. and Math. Sci., Northwestern Univ., Dept. of Ind. Eng., Technion, Israel Institute of Technology, 1981.
33. M. I. Reiman, "The Diffusion Approximation for Sojourn Times in Jackson Networks," Proc. Appl. Prob.—Computer Sci.—The Interface ORSA/TIMS Conf., Boca Raton, January 1981, ed. R. L. Disney and T. J. Ott, Boston: Birkhauser, 1982.
34. R. J. T. Morris, "Priority Queueing Networks," B.S.T.J., *60*, No. 8 (October 1981), pp. 1745–69.
35. M. Reiser and H. Kobyashi, "Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms," I.B.M. J. Res. Dev. *19*, No. 3 (May 1975), pp. 283–94.