NUMERICAL SOLUTION OF SEMICLOSED EXPONENTIAL SERVER QUEUING NETWORKS

Martin Reiser and Hisashi Kobayashi*
IBM Research Laboratory
San Jose, California

Abstract

A class of exponential server queuing networks, called semiclosed networks, is introduced. In a semiclosed network, the number of customers K is a random variable such that $K^{-} \le K \le K^{+}$. Arrival rates may be variable and the service rates load dependent. A numerical solution, based on a multidimensional Horner scheme is given with an operation count of $O(NK^{+2})$ where N is the number of servers.

1. INTRODUCTION

The most important class of analytically solvable queuing networks are the exponential server networks. J. R. Jackson(1) discussed a rather general case with (1) Markovian routing, (2) load dependent processing rates, (3) variable arrival rate, and (4) constraints on the number of customers in the system and on the queue-sizes. Closed queuing networks and multiprocessor servers are special cases of Jackson's result. The solution is known as product form solution. Although quite simple mathematically, a numerical evaluation requires a summation of the product terms over the entire state space which exhibits a combinatorially exploding size.

The great interest in applying large queuing network models, especially in the area of computer network performance has led to several solutions of the computational problem.(2-5) Most of these publications are for closed networks. Most general is the algorithm of J. P. Buzen(3) which allows for load dependent servers. The same algorithm was also reported independently by M. Reiser and H. Kobayashi.(5)

It is the object of this paper to describe a novel algorithm which is based on a multidimensional Horner scheme. This algorithm, then, is applied to a rather general class of queuing networks, which we call semiclosed queuing networks. Although the new algorithm has the same asymptotic operation count as Buzen's, it turns out to be considerably faster (by more than a factor 2) in the domain of greatest interest (i.e., up to 20 servers and 50 customers).

2. SEMICLOSED QUEUING NETWORKS

We consider the following queuing network:

(1) N servers with K customers proceeding randomly through the network. Customers may arrive or depart.

(2) K is a random number such that $K^{-} \le K \le K^{+}$. If $K = K^{-}$, a departing customer is immediately replaced; if $K^{-} \le K < K^{+}$ customers arrive as a Poisson process with parameter $\lambda(K)$; if $K = K^{+}$ the arrivals stop.

(3) Service times are exponentially distributed with parameter $\mu_n(k_n)$ where n denotes the server and k_n its queue length. The queuing discipline is work conserving.

*Permanent address: T. J. Watson Research Center, Yorktown Heights, N.Y.

This network is essentially Jackson's with the newly introduced constraint $K \le K^+$. It may be viewed as a generalization of the closed queuing network to the situation where the number of customers is allowed to fluctuate randomly between K^- and K^+. We may therefore call it a semiclosed network. Note that the semiclosed network includes the closed network as a special case (i.e., $K^- = K^+$). Similar to the closed network, the semiclosed network is always ergodic. The open queuing network is obtained in the limit $K^+ \to \infty$. Multiserver stations can be treated by a special choice of the load dependent processing rates $\mu(k)$ (i.e., as a staircase function) as was shown by W. J. Gordon and G. F. Newell.(6)

The product form solution is

$$p(\vec{k}) = \pi p^*(\vec{k}) = \pi \left[\prod_{i=K^-}^{K^+-1} \lambda(i) \right] \prod_{n=1}^{N} \prod_{j=1}^{k_n} \frac{e_n}{\mu_n(j)} \qquad (1)$$

where $\vec{k}$ is the (non-negative) state vector of queue-lengths k_n, $p(\vec{k})$ is the joint queue size distribution, $K = \Sigma_i k_i$ is the number of customers in the system, e_n is the expected number of visits to server n made by a customer on its routing and π is a normalization constant which is obtained by summing the product terms $p^*(\vec{k})$ over the feasible state space $F = \{\vec{k}; \vec{k} \ge 0 \text{ and } K^- \le K \le K^+\}$. An example of F for N=2 is depicted in Fig. 1.

3. SUMMATION ALGORITHM

The sums we are dealing with have the form of homogeneous multinomials in N×K variables which may be written in matrix form $X = [x_{k,n}]$, k=1,2,...,K and n=1,2,...,N, viz.

$$R_{N,K}(X) = \sum_{\vec{k} \in D} \prod_{n=1}^{N} \prod_{k=1}^{k_n} x_{k,n} \qquad (2)$$

where $D = \{\vec{k}; \vec{k} \ge 0 \text{ and } \Sigma_i k_i = K\}$. The following is an example with N=3 and K=3 which illustrates Horner's rule(7) applied to the last variable. For convenience we denote the variables by $\vec{u}$, $\vec{v}$ and $\vec{w}$, i.e., $X = [\vec{u}, \vec{v}, \vec{w}]$:

$$\begin{aligned} R_{3,3}(X) &= u_1u_2u_3+u_1u_2v_1+u_1u_2w_1+u_1v_1v_2+u_1v_1w_1 \\ &\quad +u_1w_1w_2+v_1v_2v_3+v_1v_2w_1+v_1w_1w_2+w_1w_2w_3 \\ &= (u_1u_2u_3+u_1u_2v_1+u_1v_1v_2+v_1v_2v_3) \\ &\quad +w_1[(u_1u_2+u_1v_1+v_1v_2)+w_2[(u_1+v_1)+w_3]] \\ &= R_{2,3}(X)+w_1[R_{2,2}(X)+w_2[R_{2,1}(X)+w_3]] \ . \qquad (3) \end{aligned}$$

The R-expressions in the last line of equation (3) have one less variable (dimension) and we now may apply the same procedure again until we are left with terms in one variable only, which are

$$R_{1,K}(X) = \prod_{i=1}^{K} x_{i,1} \quad .$$

However, such a recursive procedure, which is easily implemented as computer program, exhibits an exponentially growing operation count. Examination of the recursion tree shows, that this exponential growth is due to repetitive re-evaluation of the same subexpressions $R_{n,k}(X)$. Elimination of such re-evaluation leads to a row-wise construction of the array $R_{n,k}(X)$ for n=1,2,...,N and k=1,2,...,K as follows:

(1) Initialize first row by

$$R_{1,k} = \prod_{i=1}^{k} x_{i,1} \quad \text{for } k=1,2,\dots,K \quad . \qquad (4)$$

(2) Compute row-wise (n=2,3,...,N) the values $R_{n,k}$ (k=1,2,...,K) from previously computed values $R_{n-1,k}$ by means of Horner's rule.

$$\begin{aligned} R_{n,k} &= R_{n-1,k}+x_{1,n}[R_{n-1,k-1}+\dots \\ &\quad +x_{k-1,n}[R_{n-1,1}+x_{k,n}]]\dots] \quad . \qquad (5) \end{aligned}$$

The diagram in Fig. 2 illustrates this algorithm. The operation count is

$$1/2(N-2)K(K-1)+2(K-1)=0(NK^2) \qquad (6)$$

essential operations (i.e., multiplications or divisions). The storage requirement is 2K words.

It is well known that Horner's rule not only minimizes the necessary multiplications but also the roundoff errors. In the case of load independent servers, the computational demand reduces to O(NK).[5]

4. COMPUTATION OF QUEUE STATISTICS BY MEANS OF THE R-FUNCTION

This section describes the application of the summation algorithm to the semiclosed queuing network. The $R_{N,K}$-function basically does the summation for a closed network with N servers and K customers. The solution of the semiclosed network can be expressed as a superposition of closed network solutions with $K=K^-, K^-+1, \ldots, K^+$ with weights $\Lambda_k=\lambda(K^-)\lambda(K^-+1)\ldots\lambda(k-1)$ (see Fig. 1). The argument for the R-function is the matrix $T=[\tau_{k,n}]$, $k=1,2,\ldots,K$ and $n=1,2,\ldots,N$ where

$$\tau_{k,n} = e_n/\mu_n(k) \tag{7}$$

The proof of the following expressions is a straightforward exercise for which there is no space in this paper. For convenience, we define $R_{0,K}(X)=1$.

4.1 NORMALIZATION CONSTANT

With the above definitions, the normalization constant π is found as

$$\pi^{-1} = \sum_{i=K^-}^{K^+} \Lambda_i R_{N,i}(T) \tag{8}$$

Note that all the R-values are found as intermediate results for the computation of $R_{N,K^+}(T)$.

4.2 MARGINAL DISTRIBUTION

The marginal queue size distribution $P_n(k)$ at server n is found by summing the joint distribution over $\{\vec{k};\vec{k}\geq 0$ and $K^-\leq\Sigma\vec{k}\leq K^+$ and $k_n=k\}$. If the common factor

$$\prod_{i=1}^{k} \tau_{i,n}$$

is taken outside the products, then the remainder is of the same form as equation (1), thus

$$P_n(k) = \pi\left\{\prod_{j=1}^{k} \tau_{j,n}\right\}\sum_{i=I^-}^{K^+-k} \Lambda_{i+k} R_{N-1,i}(\theta_n T) \tag{9}$$

where $I^-=\max(0,K^--k)$ and $\theta_n T$ is obtained from T by deleting the n-th column. The operation count for computing the marginal distribution is $O(NK^2)$, the same as for π.

4.3 MOMENTS AND SERVER UTILIZATION

Often only some low order moments are of interest. The moments may be obtained with slightly less computational effort directly by the R-function, viz.

$$E[k_n^m] = \pi \sum_{i=K^-}^{K^+} \Lambda_i\left[R_{N,i}(T^*)-R_{N-1,i}(\theta_n T)\right] \tag{10}$$

where the modified argument T^* is obtained from T by replacing the n-th column by $\tau^*_{k,n}=(k/k-1)^m\tau_{k,n}$. Note that all R-values can be computed simultaneously if n=N, a condition which can always be achieved by permutation of the arguments (the R-function is invariant to such permutations). The utilization of the n-th server is obtained at little extra cost from the R-values used in equation (9) by

$$U_n = 1-P_n(0) = \pi \sum_{i=K^-}^{K^+} \Lambda_i R_{N-1,i}(\theta_n T) \quad . \tag{11}$$

4.4 AVERAGE RESPONSE TIME

Assuming work conserving schedule discipline,[8] the average queuing time $\bar{t}_n$ at server n can be found via Little's formula, viz. $\bar{t}_n=\bar{k}_n/(\bar{\lambda}e_n)$ where $\bar{k}_n$ is the average queue size and $\bar{\lambda}$ the average arrival rate, given by

$$\bar{\lambda} = \pi \sum_{i=K^--1}^{K^+-1} \lambda(i)R_{N,i}(T) \quad . \tag{12}$$

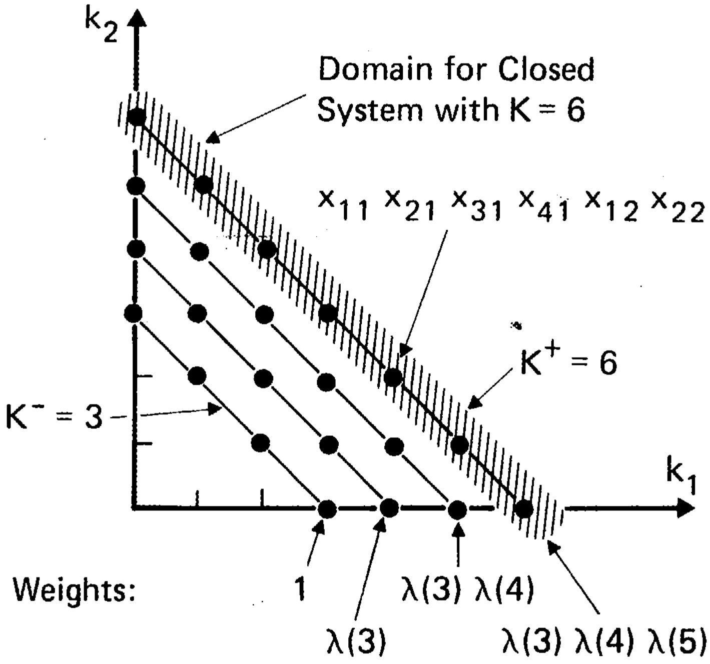
Again, note that the R values are the same as those for the computation of π.

5. CONCLUSION

An efficient and numerically stable summation algorithm has been described and applied to the general case of Jackson's exponential server network model. An interactive APL program is available for solution of semiclosed queuing network models. It is an additional advantage of the R-function approach that once the R-array is computed for a given routing and given processor speeds, results for various arrival rates, K^- and K^+ may be obtained at little extra cost from those stored R-values. The operation count for computation of all N marginal distributions is $O(N^2K^2)$. This is only a moderate effort and the APL program solves for fairly large problems (e.g., N=20, K=50) within few seconds (on an IBM 360/67). The storage requirement is O(5NK). This APL program has already proven to be a useful tool in systems design and analysis.

REFERENCES

1. J. R. Jackson, Management Sci. 10, 131 (1963).
2. F. R. Moore, IBM J. Res. & Dev. 16, 567 (1972).
3. J. P. Buzen, CACM 16, 527 (1973).
4. H. Kobayashi, IBM Res. Report RC 3943 and RC 4054, 1972.
5. M. Reiser and H. Kobayashi, IBM Res. Report RC 4254, 1973.
6. W. J. Gordon and G. F. Newell, Operations Res. 15, 254 (1967).
7. D. E. Knuth, The Art of Computer Programming, Vol. 2, Addison-Wesley Publ. Co., 1969.
8. R. W. Wolff, J. Appl. Prob. 7, 327 (1970).



k_2
Domain for Closed System with K = 6
$x_{11}\ x_{21}\ x_{31}\ x_{41}\ x_{12}\ x_{22}$
$K^+ = 6$
$K^- = 3$
k_1
Weights:
1
$\lambda(3)\ \lambda(4)$
$\lambda(3)$
$\lambda(3)\ \lambda(4)\ \lambda(5)$

FIGURE 1. Feasible state space for the example N=2, K^-=3, K^+=6. The R-function sums the terms along the diagonal lines which correspond to closed queuing networks. The solution for the semiclosed case is obtained by a superposition of closed network solutions with the indicated weights.

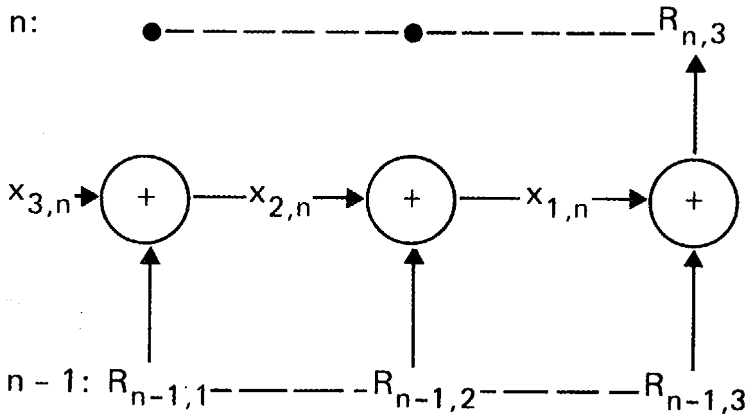


FIGURE 2. Diagram of the computation of $R_{n,3}$ from $R_{n-1,1}$, $R_{n-1,2}$ and $R_{n-1,3}$ according to the multidimensional Horner's rule.