

Analysis of a Loop Transmission System with Round-Robin Scheduling of Services

Abstract: A finite population, multi-queue model is developed for a loop transmission system. Approximate expressions for the state transition matrix and other system variables are derived in recursive forms. It is also shown that a number of useful system parameters, such as average message response time, average cycle time, and average response time conditioned on message length, can be obtained. The analytical results have been validated by simulation.

Introduction

In the literature, most system models involving many queues with a single server assume that customers are drawn from an infinite population, such that arrival and service processes are mutually independent. Published works on this type of model include Liebowitz [1], Kruskal [2], Cooper and Murray [3], Eisenberg [4], and Konheim and Meister [5]. These papers differ mainly in the service disciplines and the degree of generality of the models assumed. In some physical systems, however, the arrival and service processes are not independent. For instance, in an interactive environment, arrivals at a queue occur in batches and no new arrival occurs until the previous entry has been completely served. To our knowledge, only one author has considered a system model of this type: Runnenberg [6] assumes that the server patrols the queues in cyclic order, completely exhausting each before advancing to the next. In contrast, we assume a round-robin scheduling of services. That is, the server provides only one unit of service for each visit to a queue.

The motivation that brought this problem to our attention is its application to loop transmission systems. In this case the server is a processor that controls the loop; the batches are the messages, each of which consists of a number of characters to be transmitted; the queues are the terminals attached to the loop. We believe that the model described in this paper can be useful in a number of other applications also.

The model consists of N queues distributed around a loop (Fig. 1). The queues are served in cyclic order by a traveling server. The arrivals to the queues occur as messages, where the number of characters in a message varies according to a geometric distribution with mean R . The server "walks" from one queue to the next, servicing exactly one character for each visit to a queue.

The service time per character, t_s , and the walking time between adjacent queues, t_w , are both assumed to be constants. At each queue the time interval from the completion of service of one message to the arrival of the next is exponentially distributed with mean $1/\lambda$.

For this model, several recursive expressions have been developed to obtain the state transition matrix and other important system variables. Here, the state of the system is defined as the total number of non-empty queues in the system at the instants of termination of the service quanta. The choice of this definition is made in the interest of limiting the number of possible states to the total number of queues, N , because a more complete specification of the system would require 2^N possible states. Such a definition necessitates an approximation in the derivation of the recursive expressions. However, comparison with simulation results (Tables 1 and 2) shows that this approximation is quite acceptable.

From these recursive expressions, it is shown that a number of useful system parameters, such as average message response time, average cycle time, and average response time conditioned on message length, can be readily derived.

Theory

The solution for the system parameters is based on three stationary matrices:

State transition: $\mathbf{P} = [p_{ij}]$,

Walking time: $\mathbf{W} = [w_{ij}]$, and

Cycle time: $\mathbf{C}^1 = [c_{ij}^1]$.

The terms "walking time" and "cycle time" as used above are to be interpreted, respectively, as the elapsed times between successive service quanta and between

successive departures from a non-empty queue. The exact dependence of the system parameters on these matrices is discussed in detail in the third section.

Let T_k ($k = 1, 2, \dots, \infty$; $T_{k-1} < T_k < T_{k+1}$) be successive instants of termination of the service quanta, and S_{T_k} be the state of the system at instant T_k ; then

$$p_{ij} = \text{Prob} [S_{T_{k+1}} = j | S_{T_k} = i], \quad 0 \leq (i, j) \leq N;$$

$$w_{ij} = \text{Prob} [(T_{k+1} - T_k) = (t_s + jt_w) | S_{T_k} = i], \quad 0 \leq i \leq N; 1 \leq j;$$

$$c_{ij}^1 = \text{Prob} [(T_{k+j} - T_k) = (jt_s + Nt_w) | S_{T_k} = i, Q_{T_k} = 1], \quad 0 \leq (i, j) \leq N;$$

where $Q_{T_k} = 0$ if the server departs from an empty queue at instant T_k , and $Q_{T_k} = 1$ otherwise. The quantities N , t_s and t_w are as defined in the Introduction.

The state transition probabilities p_{ij} may be written as

$$p_{ij} = \sum_{r=0}^1 \text{Prob} [S_{T_{k+1}} = j | S_{T_k} = i, Q_{T_k} = r] \\ \times \text{Prob} [Q_{T_k} = r | S_{T_k} = i].$$

Let

$$\mathbf{P}^r = [p_{ij}^r]; p_{ij}^r = \text{Prob} [S_{T_{k+1}} = j | S_{T_k} = i, Q_{T_k} = r];$$

$$\zeta_{ij} = \begin{cases} \text{Prob} [Q_{T_k} = 0 | S_{T_k} = i] & \text{for } i = j \\ 0 & \text{otherwise;} \end{cases}$$

$$\mathbf{Z} = [\zeta_{ij}], 0 \leq (i, j) \leq N;$$

then,

$$\mathbf{P} = \mathbf{Z} \cdot \mathbf{P}^0 + (\mathbf{I} - \mathbf{Z}) \cdot \mathbf{P}^1, \quad (1)$$

where $\mathbf{I}$ is the unit matrix.

It is clear that $\mathbf{P}$ is both irreducible and aperiodic; hence, there exists a unique vector solution to the steady state equation:

$$\mathbf{p} = \mathbf{p} \cdot \mathbf{P},$$

where

$$\mathbf{p} = \{p_i\}; p_i = \text{Prob} [S_{T_k} = i]; \quad 0 \leq i \leq N; k \rightarrow \infty.$$

From Eq. (1) the steady state equation may be written as

$$\mathbf{p} \cdot [\mathbf{Z} \cdot \mathbf{P}^0 + (\mathbf{I} - \mathbf{Z}) \cdot \mathbf{P}^1] = \mathbf{p}. \quad (2)$$

In Appendix A we show that the matrices $\mathbf{P}^0$ and $\mathbf{P}^1$ may be obtained in recursive form. Still, vector $\mathbf{p}$ cannot be obtained directly from Eq. (2) because the matrix $\mathbf{Z}$ is also a function of $\mathbf{p}$, as can be seen from the definition of ζ_{ij} . The functional dependence, however, may be obtained through the following relation:

$$\text{Prob} [S_{T_k} = i + 1 | Q_{T_k} = 1] = \text{Prob} [S_{T_k} = i | Q_{T_k} = 0].$$

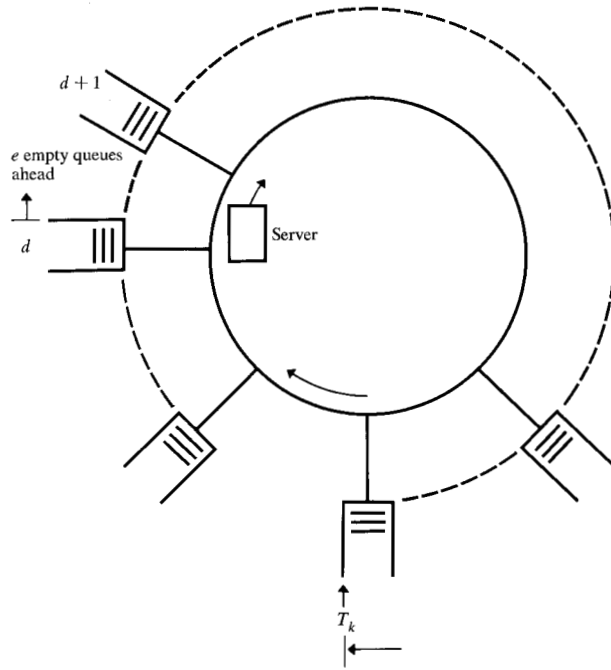


Figure 1 Schematic diagram of a loop model.

Because the message lengths are geometrically distributed with mean R , $\text{Prob} [Q_{T_k} = 0] = 1/R$.

Consequently,

$$\text{Prob} [S_{T_k} = i + 1, Q_{T_k} = 1] \\ = \frac{\text{Prob} [Q_{T_k} = 1]}{\text{Prob} [Q_{T_k} = 0]} \text{Prob} [S_{T_k} = i, Q_{T_k} = 0] \\ = (R - 1) \text{Prob} [S_{T_k} = i, Q_{T_k} = 0].$$

On the other hand,

$$\text{Prob} [S_{T_k} = i + 1, Q_{T_k} = 1] \\ + \text{Prob} [S_{T_k} = i + 1, Q_{T_k} = 0] = p_{i+1}.$$

Combining these equations yields the recursive relation

$$\text{Prob} [S_{T_k} = i + 1, Q_{T_k} = 0] \\ = p_{i+1} - (R - 1) \text{Prob} [S_{T_k} = i, Q_{T_k} = 0].$$

Observe that

$$\text{Prob} [S_{T_k} = 0, Q_{T_k} = 1] = 0;$$

therefore,

$$\text{Prob} [S_{T_k} = 0, Q_{T_k} = 0] = p_0.$$

Then the recursive relation generates the closed form solution below:

$$\text{Prob} [S_{T_k} = i, Q_{T_k} = 0] = \sum_{v=0}^i (1 - R)^{i-v} p_v. \quad (3)$$

Table 1 Comparison of theoretical results with simulation. The mean message length R is 10 and t_w/t_s is 0.05. The simulation result is the second value in each pair. There were 2000 messages processed in each simulation run.

	Server utilization = 30%		Server utilization = 90%	
	$N = 5$ $\lambda t_s = 0.66\%$	$N = 7$ $\lambda t_s = 0.46\%$	$N = 5$ $\lambda t_s = 5.05\%$	$N = 7$ $\lambda t_s = 3.16\%$
$\lambda E[T_m]$	0.1046 0.1032	0.08045 0.08181	1.805 1.749	1.457 1.505
$\lambda E[T_w]$	0.01549 0.01575	0.01083 0.01015	0.005592 0.005555	0.003497 0.003479
$\lambda E[T_c]$	0.002353 0.002336	0.002294 0.002337	0.1266 0.1273	0.1110 0.1115
$\lambda E[T_{c1}]$	0.01065 0.01061	0.008219 0.008243	0.1866 0.1844	0.1511 0.1504

Table 2 Comparison of theoretical results with simulation; the simulation result is the second value in each pair.

Server utilization (percent)		$N = 5$							
P	30	0.0708	0.6605	0.2161	0.0460	0.0063	0.0004		
		0.0721	0.6547	0.2170	0.0472	0.0087	0		
	90	0.0059	0.0694	0.1767	0.2932	0.3041	0.1506		
		0.0068	0.0737	0.1790	0.2956	0.3018	0.1429		
W	30	0.0897	0.0757	0.0663	0.0584	0.6394			
		0.0968	0.0763	0.0634	0.0628	0.6289			
	90	0.5941	0.2126	0.0905	0.0424	0.0548			
		0.6086	0.2063	0.0821	0.0413	0.0549			
C ¹	30	0	0.7055	0.2339	0.0525	0.0076	0.0005		
		0	0.7095	0.2322	0.0500	0.0078	0.0003		
	90	0	0.0574	0.1509	0.2734	0.3248	0.1935		
		0	0.0637	0.1536	0.2784	0.3259	0.1781		
Server utilization (percent)		$N = 7$							
P	30	0.0670	0.6278	0.2315	0.0601	0.0118	0.0017	0.0002	0.0000
		0.0646	0.6272	0.2301	0.0615	0.0141	0.0022	0	0
	90	0.0031	0.0366	0.0950	0.1757	0.2435	0.2422	0.1554	0.0485
		0.0033	0.0317	0.0852	0.1674	0.2452	0.2549	0.1605	0.0514
W	30	0.0715	0.0628	0.0565	0.0510	0.0462	0.0420	0.6034	
		0.0769	0.0647	0.0568	0.0506	0.0438	0.0438	0.5989	
	90	0.5525	0.2233	0.1017	0.0500	0.0264	0.0149	0.0282	
		0.5848	0.2241	0.0905	0.0364	0.0224	0.0147	0.0236	
C ¹	30	0	0.6665	0.2493	0.0678	0.0140	0.0021	0.0002	0.0000
		0	0.6626	0.2516	0.0687	0.0140	0.0026	0.0001	0
	90	0	0.0300	0.0791	0.1534	0.2292	0.2534	0.1863	0.0687
		0	0.0312	0.0807	0.1512	0.2371	0.2467	0.1911	0.0617

Realizing that

$$\text{Prob}[S_{T_k} = N, Q_{T_k} = 0] = 0,$$

We can write Eq. (3) in the matrix form

$$\mathbf{p} \cdot \mathbf{Z} = \mathbf{p} \cdot \mathbf{\Gamma}, \quad (4)$$

where

$$\mathbf{\Gamma} = [\gamma_{ij}], \quad 0 \leq (i, j) \leq N;$$

$$\gamma_{ij} = \begin{cases} (1-R)^{j-i} & \text{for } i \leq j; 0 \leq j < N; \\ 0 & \text{otherwise.} \end{cases}$$

Substituting Eq. (4) in Eq. (2) we obtain

$$\mathbf{p} \cdot \mathbf{X} = \mathbf{p} \quad (5)$$

where,

$$\mathbf{X} = \mathbf{\Gamma} \cdot \mathbf{P}^0 + (\mathbf{I} - \mathbf{\Gamma}) \cdot \mathbf{P}^1;$$

$\mathbf{p}$ may now be determined from Eq. (5).

Note that although Eqs. (2) and (5) have the same form, $\mathbf{X}$ may not be identified with the probability transition matrix $\mathbf{P}$ because the transformation matrix relating a pair of vectors is not necessarily unique. However, given vector $\mathbf{p}$, the matrix $\mathbf{Z}$ may be obtained from Eq. (4), which in turn relates $\mathbf{P}$ to $\mathbf{P}^0$ and $\mathbf{P}^1$ through Eq. (1).

From the definition of $\mathbf{W}$, the unconditional walking time distribution is given by $\mathbf{w} = \mathbf{p} \cdot \mathbf{W}$, where

$$\mathbf{w} = \{w_j\};$$

$$w_j = \text{Prob}[(T_{k+1} - T_k) = (t_s + jt_w)]; \quad 1 \leq j; k \rightarrow \infty.$$

The derivation of matrix $\mathbf{W}$ is given in Appendix B.

The cycle time distribution,

$$c^1 = \{c_j^1\};$$

$$c_j^1 = \text{Prob}[(T_{k+j} - T_k) = (jt_s + Nt_w) | Q_{T_k} = 1];$$

$$0 \leq j \leq N; k \rightarrow \infty,$$

may be obtained as follows. Let the vector

$$\mathbf{p}^1 = \{p_i^1\} \text{ for } 1 \leq i \leq N,$$

where

$$p_i^1 = \text{Prob}[S_{T_k} = i | Q_{T_k} = 1].$$

Therefore,

$$c^1 = \mathbf{p}^1 \cdot \mathbf{C}^1.$$

The matrix $\mathbf{C}^1$ is derived in Appendix C. By Bayes' rule,

$$p_i^1 = \frac{(1 - \zeta_{ii})p_i}{\text{Prob}[Q_{T_k} = 1]} = \frac{(1 - \zeta_{ii})p_i}{1 - 1/R}.$$

In vector notation, this becomes

$$\mathbf{p}^1 = \left(1 - \frac{1}{R}\right)^{-1} \mathbf{p} \cdot (\mathbf{I} - \mathbf{Z}).$$

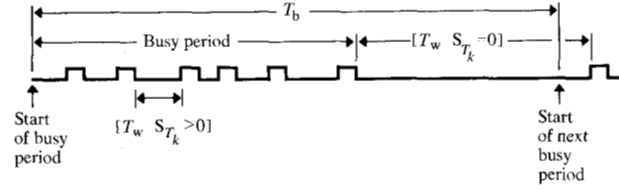


Figure 2 Plot of a busy-period cycle of the entire loop system.

Derivation of system parameters

• Average message response time

The message response time T_m is defined as the interval of time from the arrival of a message at a queue until the last character is served. It is shown below that the average message response time is given by

$$E[T_m] = NR\{E[T_w] + t_s\} - 1/\lambda, \quad (6)$$

where $T_w = [(T_{k+1} - T_k) - t_s]$ is the walking time between successive service quanta. Even though $\text{Prob}[T_w \rightarrow \infty] > 0$, the average walking time $E[T_w]$ may still be obtained in closed form as follows:

$$E[T_w] = p_0 E[T_w | S_{T_k} = 0] + (1 - p_0) E[T_w | S_{T_k} > 0].$$

An examination of the recursive relationship in Appendix B gives

$$E[T_w | S_{T_k} = 0] = t_w \left[\sum_{j=1}^N j w_{0j} + \frac{e^{-N\lambda t_w}}{1 - e^{-N\lambda t_w}} w_{0N} \right. \\ \left. \times \left(N + 1 + \frac{e^{-N\lambda t_w}}{1 - e^{-N\lambda t_w}} \right) \right];$$

$$E[T_w | S_{T_k} > 0] = t_w \left[\frac{1}{1 - p_0} \sum_{i=1}^N p_i \left(\sum_{j=1}^{N-i+1} j w_{ij} \right) \right].$$

Equation (6) may be derived with aid of Fig. 2. The arrival of messages at a queue has an average cycle time of $E[T_m] + 1/\lambda$; hence the rate of message arrival in the system is

$$\frac{N}{E[T_m] + 1/\lambda}.$$

We define the system as being busy when at least one of the queues is non-empty, and a busy period as a contiguous period of time that the system remains busy. Then $1/p_0$ is the average number of service quanta in a busy period. Let T_b be the inter-arrival time of busy periods; then

$$E[T_b] = \frac{1}{p_0} \{t_s + E[T_w]\}.$$

Because the average number of messages served in a busy period is $1/p_0 R$, the rate of message departure from the system is $1/p_0 E[T_b] R$.

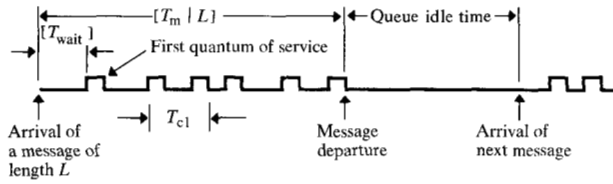


Figure 3 Plot of a typical busy-period cycle of one of the queues in the system.

In the steady state, the rate of arrival must equal the rate of departure. From this, Eq. (6) follows.

• *Average cycle time*

Define T_c as the elapsed time between successive departures from a given queue by the server. In accordance with the definition in the second section, the cycle time is

$$T_{c1} = [T_c | Q_{T_k} = 1].$$

The distribution function c^1 of T_{c1} has been derived in the second section; hence, the average value is

$$E[T_{c1}] = Nt_w + (\mathbf{m} \cdot \mathbf{c}^1)t_s.$$

where the vector $\mathbf{m} = \{1, 2, 3, \dots, N\}$. The unconditional average cycle time $E[T_c]$ may be simply obtained by observing that the average number of queues served in a cycle is $Nt_w / E[T_w]$. Therefore,

$$E[T_c] = Nt_w \left\{ 1 + \frac{t_s}{E[T_w]} \right\}.$$

• *Average response time conditioned on message length*

For this loop model, the expected response time of a message is a linearly increasing function of the length of service demanded. From Fig. 3 it is clear that the average response time conditioned on message length L is

$$E[T_m | L] = E[T_{wait}] + t_s + (L - 1)E[T_{c1}],$$

where $E[T_{wait}]$ is the mean recurrence time. That is, $E[T_{wait}]$ is the mean waiting time from the arrival of the message until the server reaches that queue for the first time. Since

$$E[T_m] = E[E[T_m | L]] = E[T_{wait}] + t_s + (R - 1)E[T_{c1}],$$

$E[T_{wait}]$ may be obtained from this average relationship. Consequently,

$$E[T_m | L] = E[T_m] + (L - R)E[T_{c1}].$$

Summary

In this paper, some performance aspects of a multi-queue model under interactive input environments have

been studied. Such a model describes many terminal-oriented computer communication systems. In general, it requires 2^N states to completely specify an N -queue system, which usually results in excessive computational difficulty. However, it is shown in this paper that by adopting a certain approximation, the number of required states is drastically reduced to N and the accuracy of the desired performance parameters is only slightly degraded. Even though the expressions derived appear complex, they may be easily programmed due to their recursive structure.

Appendix A: State transition matrix

• *Stationary transition probabilities*

For $r = 0$ and $r = 1$:

$$\mathbf{P}^r = [p_{ij}^r];$$

$$\begin{aligned} p_{ij}^r &= \text{Prob} [S_{T_{k+1}} = j | S_{T_k} = i, Q_{T_k} = r] \\ &= \text{Prob} [M_{T_k} = (j - i) | S_{T_k} = i, Q_{T_k} = r] \\ &\quad \times \text{Prob} [Q_{T_{k+1}} = 1] \\ &\quad + \text{Prob} [M_{T_k} = (j - i + 1) | S_{T_k} = i, Q_{T_k} = r] \\ &\quad \times \text{Prob} [Q_{T_{k+1}} = 0], \quad 0 \leq (i, j) \leq N, \end{aligned}$$

where M_{T_k} is the number of message arrivals in the system during the interval $[T_k, T_{k+1}]$. It is clear that

$$\text{Prob} [Q_{T_{k+1}} = 1] = 1 - 1/R \quad \text{and}$$

$$\text{Prob} [Q_{T_{k+1}} = 0] = 1/R.$$

The probability $\text{Prob} [M_{T_k} = m | S_{T_k} = i, Q_{T_k} = r]$ may be obtained in recursive form as explained below.

• *Recursive algorithm*

Consider the instant when the server leaves the d th empty queue during the time interval $[T_k, T_{k+1}]$. Let $\Lambda_i^r(q, e, d)$ be the probability of q more messages arriving in the system before T_{k+1} , given that $S_{T_k} = i$, $Q_{T_k} = r$, and there are still e empty queues ahead (see Fig. 1). Then it is clear that

$$\text{Prob} [M_{T_k} = m | S_{T_k} = i, Q_{T_k} = r] = \begin{cases} \Lambda_i^r(m, N - i, 0) & \text{for } 0 \leq i \leq N \text{ and} \\ 0 \leq m \leq N - i & \text{except } i = m = 0; \\ 0 & \text{otherwise.} \end{cases}$$

By enumerating the probabilities of all possible outcomes, $\Lambda_i^r(q, e, d)$ may be represented by a recursive tree structure as shown in Fig. 4. Note that

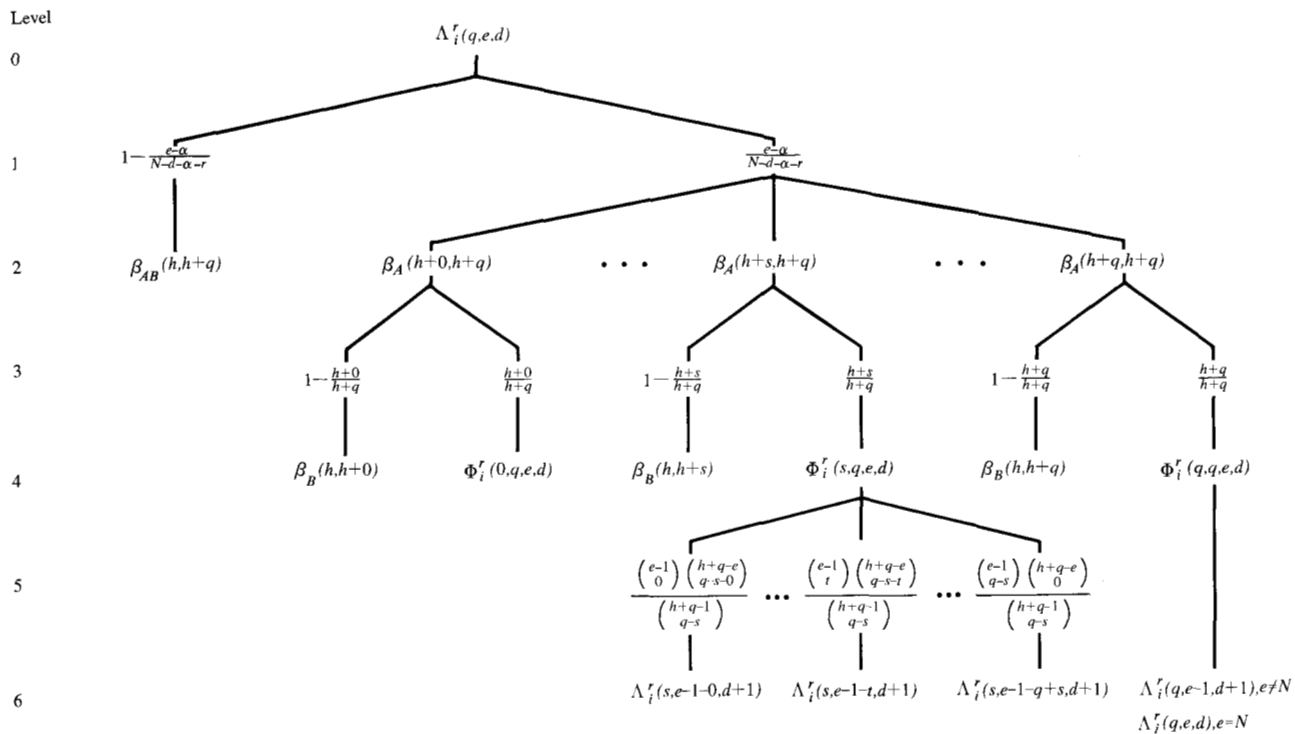


Figure 4 Transition probability tree.

$$h = N - S_{T_{k+1}} = N - (i + m);$$

$$\alpha = \begin{cases} 1 & \text{if } r = d = 0, \\ 0 & \text{otherwise;} \end{cases}$$

$$A = \exp(-\lambda t_w); B = \exp(-\lambda t_s);$$

$$\beta_\xi(\nu, \eta) = \binom{\eta}{\nu} \xi^\nu (1 - \xi)^{\eta - \nu};$$

$$\beta_{AB}(h, h + q) = \text{Prob}[q \text{ arrivals to the system during } t_w + t_s];$$

$$\beta_A(h + s, h + q) = \text{Prob}[q - s \text{ arrivals to the system during } t_w];$$

$$\beta_B(h, h + s) = \text{Prob}[s \text{ arrivals to the system during } t_s];$$

$$\frac{e - \alpha}{N - d - \alpha - r} = \text{Prob} \left[\begin{array}{l} (d + 1) \text{th queue is empty at the} \\ \text{instant that the server leaves the} \\ d \text{th queue} \end{array} \right];$$

$$\frac{h + s}{h + q} = \text{Prob} \left[\begin{array}{l} (d + 1) \text{th queue is empty at the instant} \\ \text{that the server arrives at the } (d + 1) \text{th} \\ \text{queue, given that } q - s \text{ arrivals occur} \\ \text{during the preceding interval } t_w \end{array} \right];$$

$$\frac{\binom{e-1}{t} \binom{h+q-e}{q-s-t}}{\binom{h+q-1}{q-s}} = \text{Prob} \left[\begin{array}{l} t \text{ messages arrive ahead of} \\ \text{the server, given that } q - s \\ \text{arrivals occur during the} \\ \text{preceding interval } t_w \end{array} \right];$$

It should be observed that, in stating the probabilities depicted at levels 1 and 5 of Fig. 4, it is implicitly assumed that the probability of an empty queue receiving a message is independent of the time elapsed since last examined by the server. However, comparisons with simulation (Tables 1 and 2) indicate that the approximations are entirely justified.

In summary, the tree structure may be written as

$$\begin{aligned} \Lambda_i^r(q, e, d) &= \frac{N - d - e - r}{N - d - \alpha - r} \beta_{AB}(h, h + q) \\ &+ \frac{e - \alpha}{N - d - \alpha - r} \left\{ \sum_{s=0}^q \beta_A(h + s, h + q) \right. \\ &\times \left[\frac{q - s}{h + q} \beta_B(h, h + s) \right. \\ &+ \frac{h + s}{h + q} \sum_{t=0}^{q-s} \frac{\binom{e-1}{t} \binom{h+q-e}{q-s-t}}{\binom{h+q-1}{q-s}} \\ &\times \left. \Lambda_i^r(s, e - 1 - t, d + 1) \right] \Big\} \text{ for } e \neq N. \end{aligned}$$

For $e = N$, the extreme right-hand branch in Fig. 4 is infinitely recursive; hence, it can be seen that

$$\Lambda_i^r(q, N, d) = \frac{1}{1 - \beta_A(h + q, h + q)} \left\{ \Lambda_i^r(q, N, d) \right\}_{s \text{ from } 0 \text{ to } q-1}$$

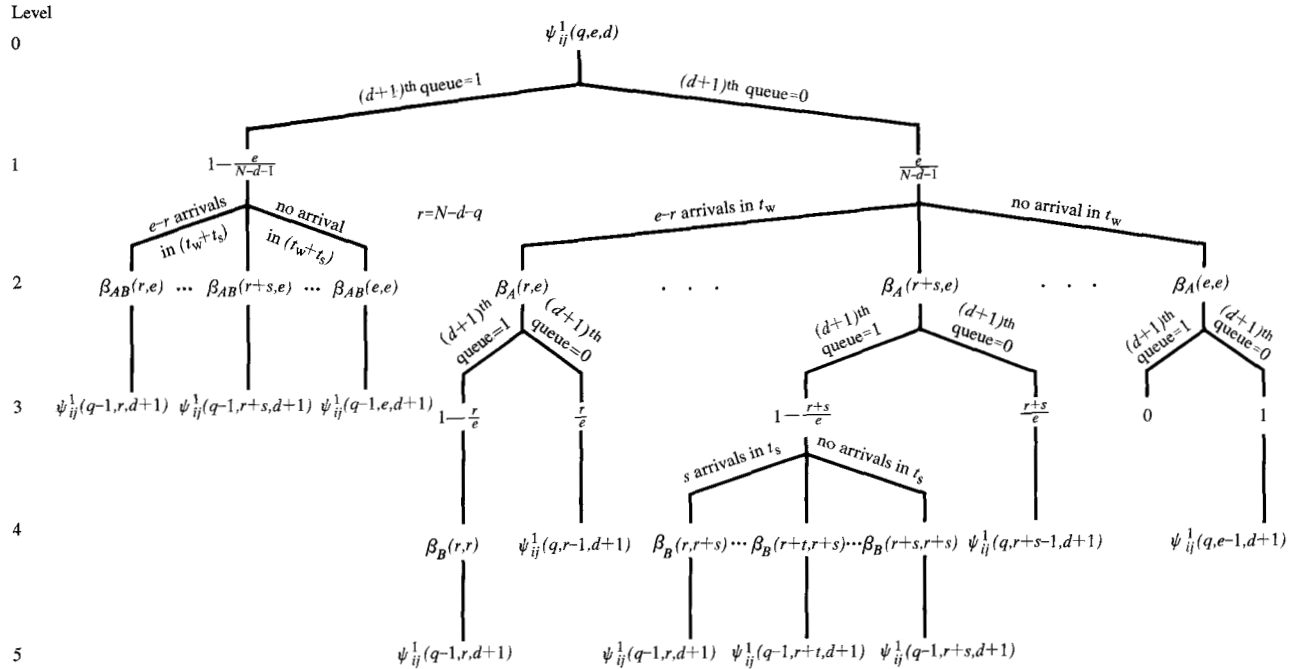


Figure 5 Cycle-time probability tree.

Appendix B: Walking time matrix

As with the transition matrix, $\mathbf{W}$ may be decomposed into

$$\mathbf{W} = \mathbf{Z} \cdot \mathbf{W}^0 + (\mathbf{I} - \mathbf{Z}) \cdot \mathbf{W}^1, \text{ where}$$

$$\mathbf{W}^r = [w_{ij}^r],$$

$$w_{ij}^r = \text{Prob}[(T_{k+1} - T_k) = (t_s + jt_w) | S_{T_k} = i, Q_{T_k} = r].$$

It is easy to see that w_{ij}^r may also be obtained by tracing the probability transition tree except under the following circumstances:

1. Up to $N - i$ message arrivals may occur during the walking time (i.e., $h = 0$).
2. For $d < j$, take only the right-hand branches from level 0 to level 1 and from level 2 to level 3.
3. For $d = j$, take only the left-hand branches from level 2 to level 3.

Thus, in analogy to $\Lambda_i^r(q, e, d)$, we define

$$\begin{aligned} \Omega_{ij}^r(q, e, d) &= \frac{e - \alpha}{N - d - \alpha - r} \left\{ \sum_{s=0}^q \frac{s}{q} \beta_A(s, q) \right. \\ &\quad \times \left[\sum_{t=0}^{q-s} \frac{\binom{e-1}{t} \binom{q-e}{q-s-t}}{\binom{q-1}{q-s}} \right] \\ &\quad \times \Omega_{ij}^r(s, e-1-t, d+1) \Big\} \\ &\quad \text{for } d < j, e \neq N; \end{aligned}$$

$$\begin{aligned} \Omega_{ij}^r(q, N, d) &= \frac{1}{1 - \beta_q(h+q, h+q)} \left\{ \Omega_{ij}^r(q, N, d) \right\}_{s \text{ from } 0 \text{ to } q-1} \\ &\quad \text{for } d < j, e = N; \end{aligned}$$

$$\begin{aligned} \Omega_{ij}^r(q, e, j) &= \frac{N - i - e - r}{N - j - \alpha - r} + \frac{e - \alpha}{N - j - \alpha - r} \\ &\quad \times \left\{ \sum_{s=0}^q \left(1 - \frac{s}{q} \right) \beta_A(s, q) \right\}, \\ &\quad \text{for } d = j. \end{aligned}$$

Then it is clear that

$$\begin{aligned} w_{ij}^r &= \Omega_{i,j-1}^r(N - i, N - i, 0), \\ &\quad 0 < i \leq N, j < N - i + 1; \\ w_{0j}^r &= \beta_A^{j-N}(N, N) w_{0N}^r, \\ &\quad j \geq N + 1. \end{aligned}$$

Appendix C: Cycle time matrix

To compute

$$\begin{aligned} c_{ij}^1 &= \text{Prob}[(T_{k+j} - T_k) = (jt_s + Nt_w) | S_{T_k} = i, Q_{T_k} = 1], \\ &\quad 0 \leq (i, j) \leq N, \end{aligned}$$

consider the instant when the server leaves the d th queue during the time interval $[T_k, T_{k+j}]$. Let $\Psi_{ij}^1(q, e, d)$ be the probability of serving q more queues before T_{k+j} , given that $S_{T_k} = i, Q_{T_k} = 1, j - q$ queues have been served since T_k , and there are still e empty queues ahead. Then it is clear that

$$c_{ij}^1 = \Psi_{ij}^1(j, N - i, 0).$$

Following logic similar to that used in the development of the probability transition tree, $\Psi_{ij}^1(q, e, d)$ may be represented by a recursive structure, as shown in Fig. 5. Note that this recursive process terminates due to the following condition:

$$\Psi_{ij}^1(q, e, N) = 1.$$

Again, the approximation in level 1 of Fig. 4 is made here and, as before, simulation data indicate that this approximation is justified.

References

1. M. A. Liebowitz, "An Approximate Method for Treating a Class of Multiqueue Problems," *IBM J. Res. Develop.* 5, 204 (1961).
2. J. B. Kruskal, "Work-Scheduling Algorithms: A Non-probabilistic Queueing Study (with Possible Application to No. 1 ESS)," *Bell Syst. Tech. J.* 48, 2963 (1969).
3. R. B. Cooper and G. Murray, "Queues Served in Cyclic Order," *Bell Syst. Tech. J.* 48, 375 (1969).
4. M. Eisenberg, "Queues with Periodic Service and Change-over Time," *Oper. Res.* 20, 440 (1972).
5. A. G. Konheim and B. Meister, "Service in a Loop System," *J. ACM* 19, 92 (1972).
6. J. Th. Runnenburg, "Machine Served by a Patrolling Operator," *Math. Centrum Amsterdam. Statist. Afdeling Rep. S 221 (VP13)*, 1957.

Received April 17, 1974; revised March 2, 1975

R. M. Wu is located at the IBM System Communications Division, Kingston, New York 12401. Yen-Ben Chen is located at the IBM Data Processing Division, Poughkeepsie, New York 12601.