

Printing Requisition / Graphic Services

Repl. No. 37824

Title or Description

Date Date Required Account

Signature Signing Authority

Department Room Phone Delivery
☐ Mail ☐ Vis Stores
☐ Pick-up ☐ Other

Reproduction Requirements		Number of Pages	Number of Copies	Cost: Time/Materials	Fun.	Prod. Un.	Prod. Cpn.	Ch.	No.	Mins.	Total
<input type="checkbox"/> Offset <input type="checkbox"/> Signs/Repro's <input type="checkbox"/> Xerox Type of Paper Stock <input type="checkbox"/> Bond <input type="checkbox"/> Book <input type="checkbox"/> Cover <input type="checkbox"/> Bristol <input type="checkbox"/> Supplied				Signs/Repro's	1						
Paper Size <input type="checkbox"/> 8 1/2 x 11 <input type="checkbox"/> 8 1/2 x 14 <input type="checkbox"/> 11 x 17				Camera	2						
Paper Colour <input type="checkbox"/> White <input type="checkbox"/> Other <input type="checkbox"/> Black				Correcting & Masking Negatives	3						
Printing <input type="checkbox"/> 1 Side <input type="checkbox"/> 2 Sides <input type="checkbox"/> Numbering to				Platemaking	4						
Binding/Finishing Operations <input type="checkbox"/> Collating <input type="checkbox"/> Corner Stitching <input type="checkbox"/> 3 Ring <input type="checkbox"/> Tape <input type="checkbox"/> Plastic Ring <input type="checkbox"/> Perforating				Printing	5						
Binding Finished Size Cutting Finished Size											
Special instructions				Bindery	6						
				Sub. Total Time							
Film Qty Size Plates Qty Size & Type				Sub. Total Materials							
Paper Qty Size Plastic Rings Qty Size				Prov. Tax							
Outside Services				Total							

1. Please complete unshaded areas on form as applicable. (A part so shaded required).
2. Distribute copies as follows: White, Canary and Pink—Printing Arts Library or applicable Dept. Office; Goldenrod—Retail.
3. On completion of order, pink copy will be returned with printed material. Canary copy will be posted and returned to requisitioner. Retain as a record of your charges.
4. Please direct enquiries, quoting requisition number, to Printing Graphic Services, Extension 3481.

ANALYSIS OF CLOSED QUEUEING NETWORKS WITH
PERIODIC SERVERS

by

Gaston H. Gonnet

and

David E. Morgan

Research Report CS-79-14

Computer Communications Network Group

and

Department of Computer Science

University of Waterloo

Waterloo, Ontario, Canada

Index terms for "Analysis of Closed Queueing Networks with Periodic Servers"

by G.H. Gonnet and D.E. Morgan

Queueing networks
Bounded variation periodic functions
Periodic servers
Computer system modelling
Computer networks
Time sharing systems
Eigenvalue and eigenvector problem

Analysis of Closed Queueing Networks with Periodic Servers

Gaston H. Gonnet
and
David E. Morgan

Computer Communications Network Group
and
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
March 1979

Abstract. A periodic network is a queueing network whose steady state behaviour is not constant in time, but repeats itself in a cycle. This behaviour may be caused by the introduction of periodic servers, e.g. paging drums. The model presented is a generalization of some other models of queueing networks, and provides a more general definition of steady state behaviour. A theoretical solution is presented. Examples of theoretical and approximate solutions are presented for a well known queueing network model of a computer system.

1. Introduction.

A Periodic network is a queueing network whose steady state behaviour is not constant in time, but repeats itself in a cycle. One important cause of this behaviour is the introduction of periodic servers in the network. A periodic server is a special kind of server that starts service of its customers in the queue only at certain points in time. A rotating memory device used for swapping equal size blocks, such as a paging drum or disk, is an example of a periodic server. Figure 1 shows a schematic graph on the behaviour of a periodic server. Coffman and Denning [5] and Fuller and Baskett [7] have comprehensively studied the behaviour of paging drums and disks. However the authors are not aware of any comprehensive work that analyzes the effect such periodic servers have on queueing networks that contain them.

The classic analyses of queueing networks containing disks or drums (Baskett et al [2], Kleinrock [12,13], Jackson [10,11], Gordon [9], etc.) are based on the assumption that the requests to servers are started at random (not synchronous) times. In this case an exponential server is a good approximation. The interservice time may be considered random when we have a very low arrival rate, or the lengths of the records (i.e. service time) are variable. At the other extreme, when there is a heavy swapping activity of equal size pages and the queue waiting for service is essentially never empty, we may assume that the service time is constant and approximate the behaviour of the devices with an M/D/1 server. However because the assumptions that support such approximations are different, they are not valid to describe situations between these two extreme cases. Considering the periodic nature of certain devices in a system allows us to model system behaviour better across a broad spectrum of cases while including both extremes as special cases.

The purpose of this study is to develop theoretical solutions of queueing networks containing periodic servers. Section 2 of the paper defines terms and notation. Section 3 provides a steady state solution for a periodic network. Section 4 presents an example of a queueing network analyzed with these techniques.

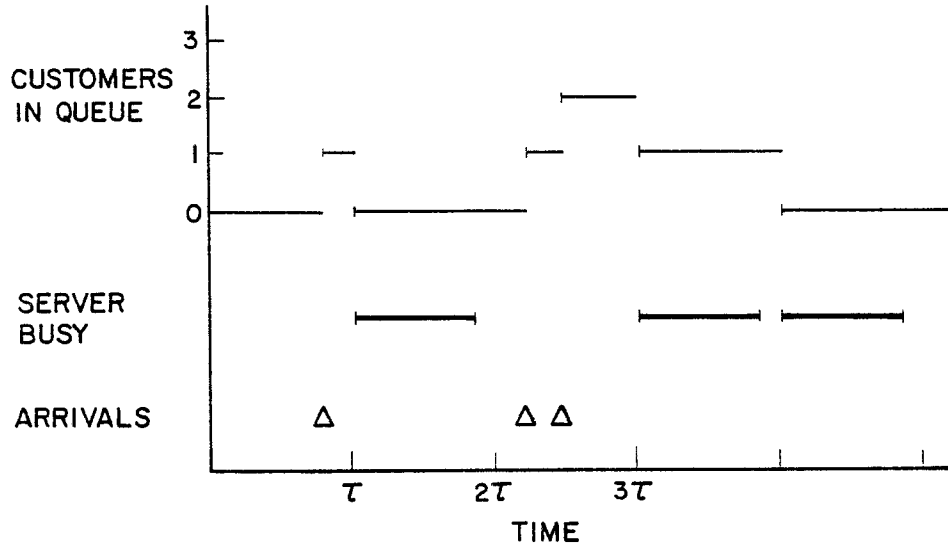


Figure 1

2. General Definitions and Assumptions.

Consistently we denote vectors and matrices in boldface capital letters (e.g. **A**, **W**, **P**). Elements of matrices or vectors are denoted by the corresponding lower case boldface, with the proper subscripts (e.g. **a_{ij}**). A superscript + or - indicates the limit to the quantity by the right or left respectively (e.g. τ^+). Superscript *T* (e.g. **P^T**) on a vector matrix means the transpose of that vector or matrix.

A *queueing network* will be defined by a directed graph in which each node represents a particular queue-server pair, and each edge represents a possible path for customers from one server to another.

We assume that there is a fixed number, *n*, of indistinguishable customers circulating through the network, i.e., it is a closed network. Let $k = k_e + k_p$ be the total number of servers (exponential + periodic) in the network. Each server has arbitrary mean service time. Associated with each server there is a queue of capacity at least *n*. The analysis given here is essentially the same for any simple queueing discipline, i.e. FIFO, LIFO, etc.

The state of the network will be defined by the number of customers in each exponential server, and by the number of customers in queue and in service in each periodic server. Notice that for a periodic server, the server may be idle although its queue may have customers awaiting service. If $R(k_e, k_p, n)$ is the number of possible distinguishable states, then we have [12]:

$$R(k_e, 0, n) = \binom{n + k_e - 1}{n} \quad (2.1)$$

if there are no periodic servers. The following recursion formula yields the number of distinguishable states, *r*, counting both exponential and periodic servers. The formula results from considering how the number of states changes when one replaces an exponential server with a periodic

server.

$$r = R(k_e, k_p, n) = 2R(k_e + 1, k_p - 1, n) - R(k_e, k_p - 1, n) \quad (2.2)$$

We find that

$$r \leq 2^{k_p} \binom{n+k-1}{n} \quad (2.3)$$

We will number each state with an unique integer from $1, 2, \dots, r$.

Let $\mathbf{P}(t)$ be any r -dimensional column vector that contains the probability that each of the r states is the state of the network at time t . Normally $\mathbf{P}(t)$ is a function of the topology of the network (reflected by the global balance equations), the initial state and time.

We say that the network is in *periodic steady state* if the probability state vector $\mathbf{P}_s(t)$, is a function of time which repeats its behaviour every τ units of time (e.g., see Figure II). τ will be called the *cycle* of the network. Because a network that is τ -periodic is also 2τ -periodic, we will take as *canonical form* the smallest possible τ . Thus we have:

$$\mathbf{P}_s(t) = \mathbf{P}_s(t + \tau) \quad [0 \leq t \leq \tau] \quad (2.4)$$

Note that this definition of steady state covers the normal notion, since given any $\mathbf{P}(t)$ for which $\mathbf{P}'(t) = 0 \Rightarrow \mathbf{P}(t) = \text{const} = \mathbf{P}(t + \tau)$.

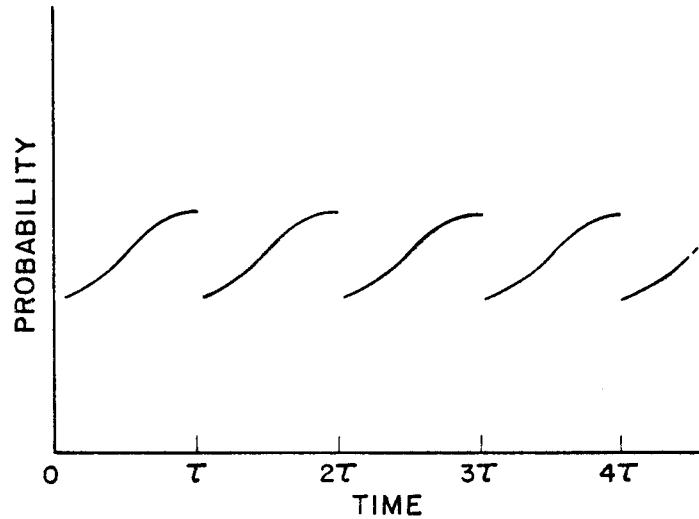


Figure II

3. Theoretical solution.

First we will consider only the time intervals where the periodic servers do not cause a change in state of the network; that is, those intervals that do not include points in which an instantaneous state transition occurs as a result of any periodic server activity. An *instantaneous state transition* at time T is a discontinuity of $\mathbf{P}(t)$ at $t = T$. An instantaneous state

transition may occur by the starting or completion of service in a synchronous server. In any of these intervals, the global balance equations yield:

$$\mathbf{P}'(t) = \mathbf{A}\mathbf{P}(t). \quad (3.1)$$

\mathbf{A} is an $r \times r$ matrix that defines all possible state transitions for the exponential servers. Each element \mathbf{a}_{ij} of the matrix \mathbf{A} is determined by the global balance equations (or flow-conservation equations, as in Kleinrock [12]) and results in: $\mathbf{a}_{ij}=0$ if there exists no single customer transition that takes the network from state j to state i ; $\mathbf{a}_{ij} = \mu_{ji}$ iff departures from state j to state i occur at a rate μ_{ji} and finally $\mathbf{a}_{ii} = -\sum_j \mu_{ij}$ (total rate of departure from state i).

It follows immediately that the number of nonzero terms of each row of \mathbf{A} is bounded by the topology of the network (i.e., the bound is the number of outgoing edges from exponential server nodes + 1). Moreover this matrix \mathbf{A} is constant with respect to time, and the maximum norm, $\|\mathbf{A}\| \leq 2\sum \mu_k$, (where μ_k is the departure rate from exponential server k), is bounded by the characteristics of the network, but independent of the number of customers and possible states.

Let $e^{\mathbf{A}}$ be defined in the usual way (Frazer [6]) as:

$$e^{\mathbf{A}} = \mathbf{I} + \mathbf{A} + \frac{\mathbf{A}^2}{2} + \frac{\mathbf{A}^3}{3!} + \dots \quad (3.2)$$

In the intervals (t_0, t) where no activity in the periodic servers occurs, the solution of the system (3.1) is:

$$\mathbf{P}(t) = e^{\mathbf{A}t} \mathbf{P}(t_0), \quad (3.3)$$

where $\mathbf{P}(t_0)$ is the initial state vector. It is important to note that the behaviour of the $\mathbf{P}(t)$ vector in intervals with no periodic-server activity is completely characterized by $\mathbf{P}(t_0)$.

3.1. Single periodic server networks.

Suppose there is only one periodic server in the system, and it has both cycle and service time equal to τ . Let \mathbf{R} be the matrix that defines the state transition caused by the start or completion of service in the periodic server, based on the probability state vector before such an operation, hence:

$$\mathbf{P}(\tau^+) = \mathbf{R} \mathbf{P}(\tau^-) \quad (3.1.1)$$

\mathbf{R} is an $r \times r$ matrix consisting of ones and zeros only. \mathbf{R} describes the discontinuous transitions that occur in the network.

Now we will define $\mathbf{P}_s(0^+) = \mathbf{P}_s$, a probability state vector, that will characterize the steady state solution, $\mathbf{P}_s(t)$, of the periodic network. Using 2.4, 3.1.1 and 3.3 we obtain:

$$\mathbf{P}(0^+) = \mathbf{P}(\tau^+) = \mathbf{R} \mathbf{P}(\tau^-) = \mathbf{R} e^{\mathbf{A}\tau} \mathbf{P}(0^+) \quad (3.1.2)$$

Thus \mathbf{P}_s satisfy:

$$\mathbf{P}_s = \mathbf{R} e^{\mathbf{A}\tau} \mathbf{P}_s \quad (3.1.3)$$

Thus \mathbf{P}_s is the eigenvector of $\mathbf{R} e^{\mathbf{A}\tau}$ whose corresponding eigenvalue is 1). Since \mathbf{P}_s is a probability vector, we normalize \mathbf{P}_s in such a way that the sum of all its elements is 1.

Then:

$$\mathbf{P}_s(t) = e^{At} \mathbf{P}_s \quad [0 < t < \tau] \quad (3.1.4)$$

defines the periodic steady state. Note that if $\mathbf{R} \cdot e^{A\tau}$ contains no unit eigenvalue then there is no possible *steady state* solution for the network.

Let $\mathbf{Q}(t) = \mathbf{W} \cdot \mathbf{P}_s(t) = (w_1, w_2, \dots, w_r) \cdot \mathbf{P}_s(t)$ be a performance index that characterizes behaviour of the network that we want to evaluate (e.g. probability of certain configuration; mean queue length; utilization of certain server; or simply moments of marginal distributions). Clearly we are interested in the average behaviour of $\mathbf{Q}(t)$ rather than in its value for a certain t . Because of the periodic behaviour of $\mathbf{P}_s(t)$ and consequently $\mathbf{Q}(t)$, we only need to consider the average over a τ period of time. Then the averaged $\mathbf{Q}(t)$, denoted \mathbf{Q} , will be:

$$\mathbf{Q} = \tau^{-1} \int_0^\tau \mathbf{W} \cdot \mathbf{P}_s(t) dt = \mathbf{W} \cdot \tau^{-1} \int_0^\tau e^{At} dt \cdot \mathbf{P}_s \quad (3.1.5)$$

We will call

$$\tau^{-1} \int_0^\tau e^{At} dt \cdot \mathbf{P}_s \quad (3.1.6)$$

the *average probability state vector* (APSV). Notice that for a given network this vector does not depend on \mathbf{W} , so we may calculate any value \mathbf{Q} for any vector \mathbf{W} while only having to perform the APSV calculation once.

3.2. Multiple periodic servers.

Assume now that there is more than one periodic server, or that there is more than one discontinuous event in the cycle of the periodic server. Both situations imply that we will have more than one discontinuity in the probability state vector. If there is no rational relation between their periods, it is intuitively clear that there is no way to define a periodic steady state as before. Suppose then, that there exists a rational relation between their cycles, such that after a sequence $0 = \tau_0 < \tau_1 < \dots < \tau_j = \tau$ of transition points in time, we complete a period, and the network is in the same state as in time 0. Associated with the end of each interval $[\tau_{i-1}, \tau_i]$ there is a corresponding transition matrix \mathbf{R}_i . Let $\Delta_i = \tau_i - \tau_{i-1}$.

The solution of this general system, using similar notation to that of Section 3.1 is as follows.

For each period of time between transitions we have:

$$\mathbf{P}'(t) = \mathbf{A} \cdot \mathbf{P}(t) \Rightarrow \mathbf{P}(t + \tau_i) = e^{A\Delta_i} \mathbf{P}(\tau_i^+) \quad [0 \leq t < \Delta_{i+1}]. \quad (3.2.1)$$

For each transition we have:

$$\mathbf{P}(\tau_i^+) = \mathbf{R}_i \mathbf{P}(\tau_i^-). \quad (3.2.2)$$

In the periodic steady state, putting this together we have:

$$\mathbf{P}_s = \mathbf{R}_j \cdot e^{A\Delta_j} \dots \mathbf{R}_1 \cdot e^{A\Delta_1} \mathbf{P}_s \quad (3.2.3)$$

\mathbf{P}_s now is calculated as the eigenvector of the product of all the matrices whose eigenvalue is 1, and is normalized so that its elements add up to 1. $\mathbf{P}_s(t)$ is sectionally defined by:

$$\mathbf{P}_s(t + \tau_i) = e^{At} \mathbf{P}_s(\tau_i^+) \quad [0 \leq t < \Delta_{i+1}] \quad (3.2.4)$$

$$\mathbf{P}_s(\tau^+) = \mathbf{R}_s e^{\mathbf{A}\Delta\tau} \mathbf{P}_s(\tau_{i-1}^+) \quad (3.2.5)$$

$$\mathbf{P}_s(0^+) = \mathbf{P}_s. \quad (3.2.6)$$

The average probability state vector is calculated as follows:

$$APSV = \tau^{-1} \sum_{i=1}^j \int_0^{\Delta\tau_i} e^{\mathbf{A}t} dt \mathbf{P}(\tau_{i-1}^+). \quad (3.2.7)$$

3.3. Complexity of the computations.

In the following evaluations we will assume that $k=o(r)$. The fact that \mathbf{A} is sparse becomes quite important for the numerical computation. The common Taylor expansion is best suited to compute the exponential forms. Since $\|\mathbf{A}\|$ is bounded, only a fixed number of terms of the expansion will be used. Since the number of non-zero elements in \mathbf{A} is $O(r)$, each power of \mathbf{A} can be calculated in $O(r^2)$ multiplications. Moreover, since the number of terms in the expansion is constant, the evaluation of $e^{\mathbf{A}}$ takes also $O(r^2)$ operations.

The same arguments apply for the evaluation of $\int_0^{\tau} e^{\mathbf{A}t} dt$.

Since \mathbf{R} is a matrix of r 1's, the product $\mathbf{R}e^{\mathbf{A}}$ will be performed in $O(r^2)$ operations.

The evaluation of the eigenvalue that determines \mathbf{P}_s is accomplished by the solution of a system of linear equations of dimension $r-1$. The solution of such a system of equations could be done in $O(r^2)$ operations depending on some conditions of the matrices that are not always true (Varga [16]). Unfortunately if we cannot use $O(r^2)$ techniques or if we have $j>1$ and we have to actually multiply the matrices in 3.2.3, the solution process becomes $O(r^3)$.

The actual size and, in some cases, exponential growth of the problem indicates that only small networks may be solved exactly.

3.4. General Approximate Solution of Periodic Networks.

To obtain an approximate solution we observe the fact that in periodic networks (as described in 3.2), all the components of the network oscillate with the same cycle. In this steady state, each node has an arrival probability distribution and probability state vector (of the node itself, ignoring the rest of the network) that repeats its behaviour every τ units of time.

In [8] we discuss a method of computing approximate solutions based on the above remark. These approximate solutions are most accurate when we have only one periodic server.

4. Example Results

In this section we will analyze three networks. The first is a simple loop containing two servers each with its queue, and the second and third are well known models of a time sharing system [1,14]. The first example will be used mainly to illustrate the mechanisms of the theoretical solution. The other examples will be computed for situations similar to the ones studied in [1,14], so we can compare results.

4.1. Simple Loop.

The following figure illustrates the topology of the first example.

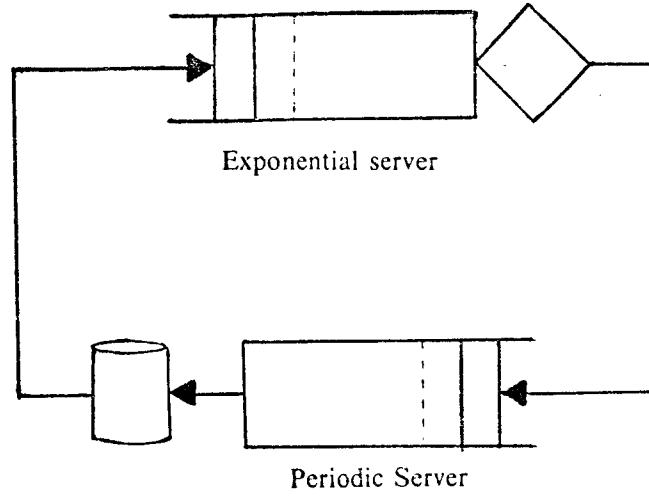


Figure III

A fixed number of customers, n , circulate through the network. For this example (using the exact derivation), assume that $n=2$. The number of possible states of the network, using (2.2), is

$$r = R(1, 1, 2) = 2R(2, 0, 2) - R(1, 0, 2) = 5. \quad (4.1.1)$$

A detailed description of each state is:

State	Periodic server	Exponential server
1	0-0	2
2	0-1	1
3	1-0	1
4	0-2	0
5	1-1	0

where the two numbers for the periodic server give the number of customers in service and in queue.

The global balance equations define the matrix \mathbf{A} so that:

$$\mathbf{P}'(t) = \begin{pmatrix} \mathbf{p}'_1 \\ \mathbf{p}'_2 \\ \mathbf{p}'_3 \\ \mathbf{p}'_4 \\ \mathbf{p}'_5 \end{pmatrix} = \begin{pmatrix} -\mu & 0 & 0 & 0 & 0 \\ \mu & -\mu & 0 & 0 & 0 \\ 0 & 0 & -\mu & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{p}'_1 \\ \mathbf{p}'_2 \\ \mathbf{p}'_3 \\ \mathbf{p}'_4 \\ \mathbf{p}'_5 \end{pmatrix}$$

In this case we compute $e^{\mathbf{A}t}$ explicitly, i.e.

$$e^{At} = \begin{pmatrix} e^{-\mu t} & 0 & 0 & 0 & 0 \\ \mu t e^{-\mu t} & e^{-\mu t} & 0 & 0 & 0 \\ 0 & 0 & e^{-\mu t} & 0 & 0 \\ 1-(1+\mu t)e^{-\mu t} & 1-e^{-\mu t} & 0 & 1 & 0 \\ 0 & 0 & 1-e^{-\mu t} & 0 & 1 \end{pmatrix}$$

The matrix \mathbf{R} is derived from the transitions that occur at τ caused by the periodic server. This in words is: for the periodic server, when the customer in the server (if any) completes its service, one of the customers in the queue (if any) starts service.

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

If we denote $\alpha = \mu\tau$, then the matrix that describes a complete transition, from time 0^+ to time τ^+ , using [3.1.2], is:

$$\mathbf{R}e^{A\tau} = \begin{pmatrix} e^{-\alpha} & 0 & e^{-\alpha} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \alpha e^{-\alpha} & e^{-\alpha} & 1-e^{-\alpha} & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1-(1+\alpha)e^{-\alpha} & 1-e^{-\alpha} & 0 & 1 & 0 \end{pmatrix}$$

The eigenvector whose eigenvalue is 1, that defines the steady state solution, is:

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix} = \begin{pmatrix} \frac{1}{e^{\alpha} + 1 - (1+\alpha)e^{-\alpha}} \\ 0 \\ \frac{e^{\alpha} - 1}{e^{\alpha} + 1 - (1+\alpha)e^{-\alpha}} \\ 0 \\ \frac{1 - (1+\alpha)e^{-\alpha}}{e^{\alpha} + 1 - (1+\alpha)e^{-\alpha}} \end{pmatrix}$$

Clearly this is not the way one would normally compute solutions; instead, they would be numerically computed.

4.2. Resource Loop Model of a Time Sharing System.

The following figure illustrates the network with which we intend to model part of a time sharing system. This model has been carefully described and analyzed in [1,3,14], among others. The model consists of three servers: CPU, Secondary Memory and other I/O in the form of a file disk. The secondary memory is likely to behave like a periodic server since we will probably use a drum or disk to swap fixed size pages in and out. In this model the secondary memory plays a decisive role in the behaviour of the overall performance, so an inaccurate representation in its modelling may invalidate the results.

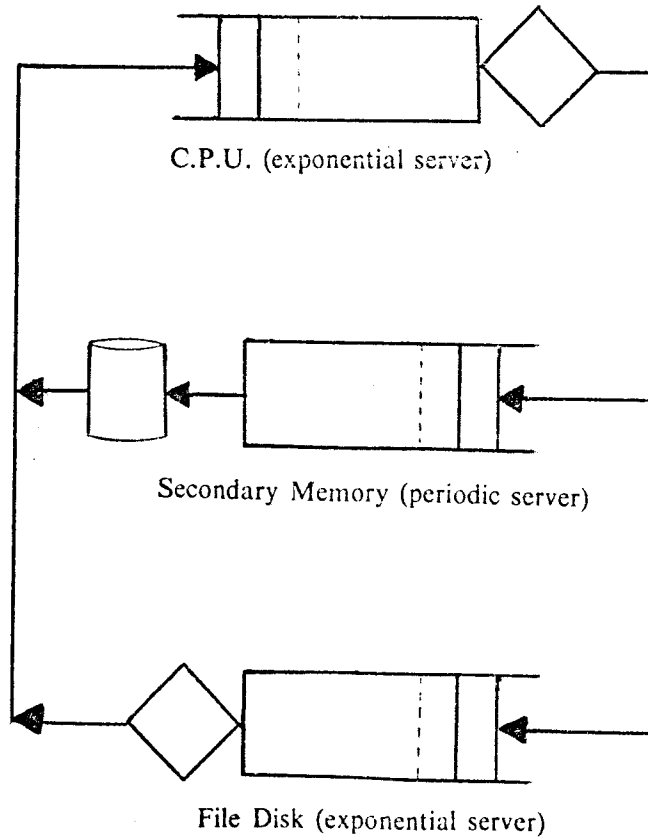


Figure IV

In this case we find 16 different states. They are indicated by (customers in CPU, in periodic server - in periodic queue, file disk).

1	(0,0-0,3)	5	(1,0-0,2)	9	(2,0-1,0)	13	(0,1-2,0)
2	(0,0-1,2)	6	(1,0-1,1)	10	(3,0-0,0)	14	(1,1-0,1)
3	(0,0-2,1)	7	(1,0-2,0)	11	(0,1-0,2)	15	(1,1-1,0)
4	(0,0-3,0)	8	(2,0-0,1)	12	(0,1-1,1)	16	(2,1-0,0)

The following paragraph shows a typical output of the program that computes the exact solution.

SOLUTION OF RESOURCE LOOP MODEL

degree of multiprograming (number of customers) = 3
number of available memory pages= 128
mean compute time between i/o requests= 20.0
mean total compute time= 1000.0
locality and memory management parameter= 1.50
secondary memory service time= 5.0
file-disk mean service time= 30.0

DERIVED VALUES

mean time between memory faults= 2.787

rate of cpu= 0.4098
rate of file disk= 0.0333
probability of exit from the resource loop= 0.00244
probability of memory fault= 0.87555
probability of i/o fault= 0.12201
Average Probability State Vector (APSV)
0.08146 0.06053 0.01127 0.00189 0.05430 0.01513 0.00332 0.02179 0.00634
0.00959 0.10023 0.16283 0.17247 0.10645 0.15199 0.04040

probability of 0,1,..., 3 customers at node 0
0.590669011 0.331203296 0.068537984 0.009589709
expected number of customers= 0.4970484 variance= 0.666787

probability of 0,1,..., 3 customers at node 1
0.167146126 0.329084019 0.329416604 0.174353252
expected number of customers= 1.5109770 variance= 0.965856

probability of 0,1,..., 3 customers at node 2
0.386003520 0.317475401 0.215064009 0.081457069
expected number of customers= 0.9919746 variance= 0.962721

CPU UTILIZATION 0.4093310

The following table and graph show the exact values for several degrees of multiprogramming. These results compare very well to those obtained by [1,14]. The graph shows a slight shift to the right that may be interpreted in terms of the behaviour of the secondary memory server, which is a periodic server.

Degree of multiprogramming	CPU Utilization	Queue length at Second. Mem.
1	0.330	0.174
2	0.410	0.662
3	0.409	1.511
4	0.339	2.780
5	0.257	4.143
6	0.197	5.391

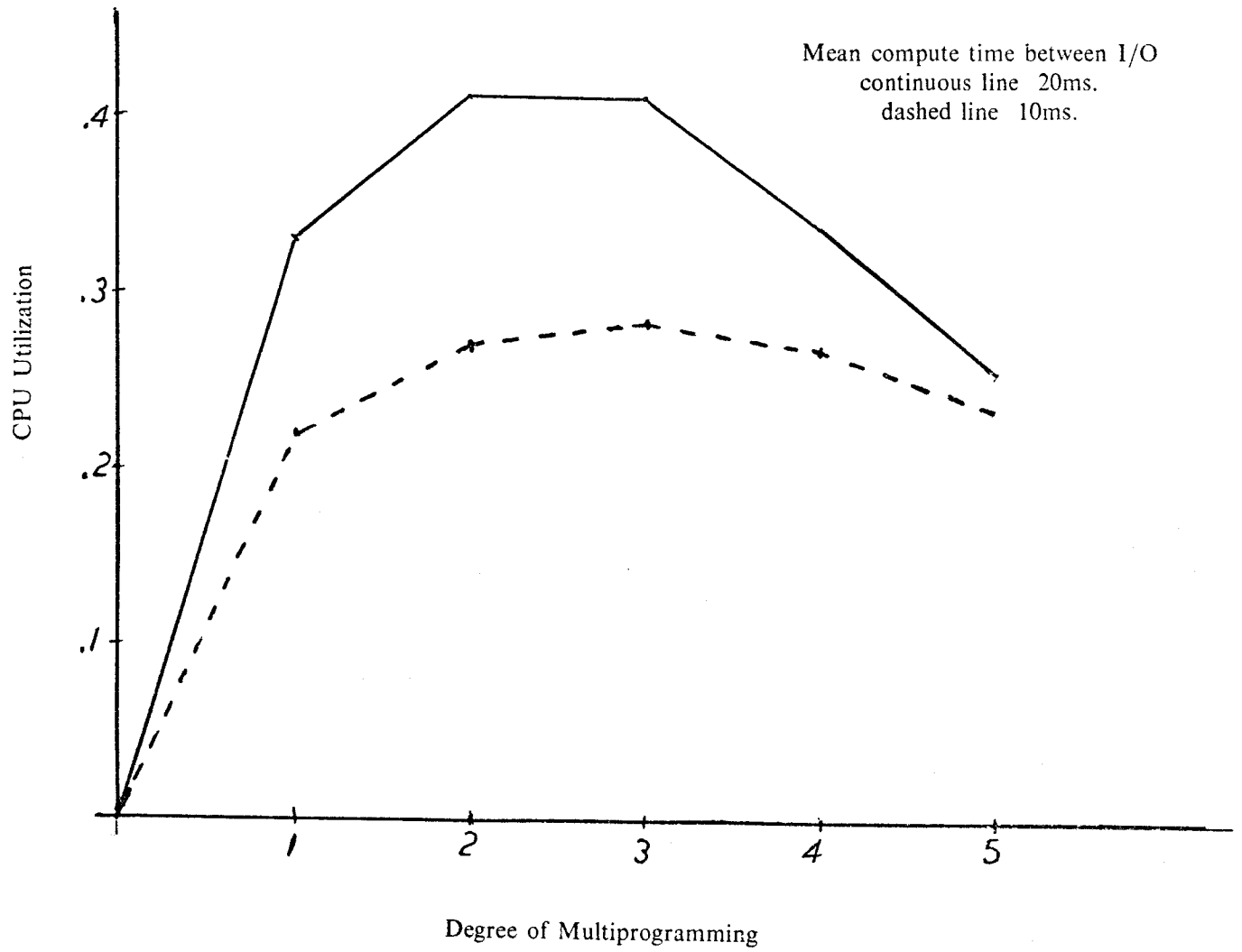


Figure V

4.3. Resource Loop Model of a Timesharing System with 2 Periodic Servers.

Figure VI shows the topology of a particular timesharing model with two secondary memory devices modelled as periodic servers. The service time of these servers are $\tau_1=7.5ms$ and $\tau_2=15ms$ respectively. Note that the total throughput of the secondary memory is the same as for the previous example. In this case we use the solution described in 3.2.

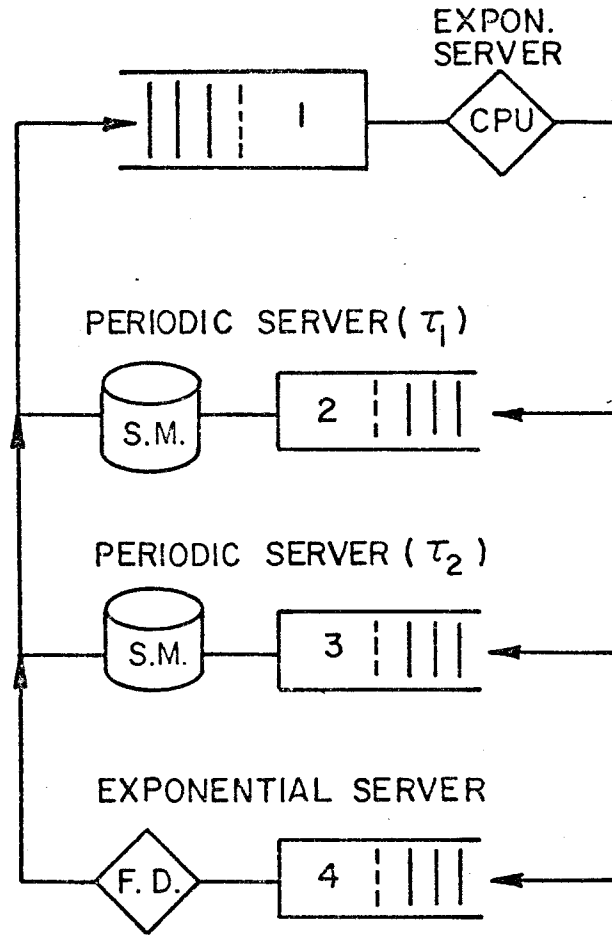


Figure VI

The following table shows the average number of customers (\bar{n}_i) and utilization factor (μ_i) of each server for various numbers of customers.

Degree of Multiprogramming	μ_1	\bar{n}_1	μ_2	\bar{n}_2	μ_3	\bar{n}_3	μ_4	\bar{n}_4
1	0.2725	0.2725	0.0993	0.0993	0.2198	0.2198	0.4048	0.4048
2	0.3015	0.3539	0.2777	0.3041	0.6114	0.7764	0.4506	0.5655
3	0.2822	0.3567	0.4400	0.5511	0.8593	1.5368	0.4112	0.5554
4	0.2369	0.3074	0.5452	0.7882	0.9631	2.4535	0.3375	0.4509

Table I

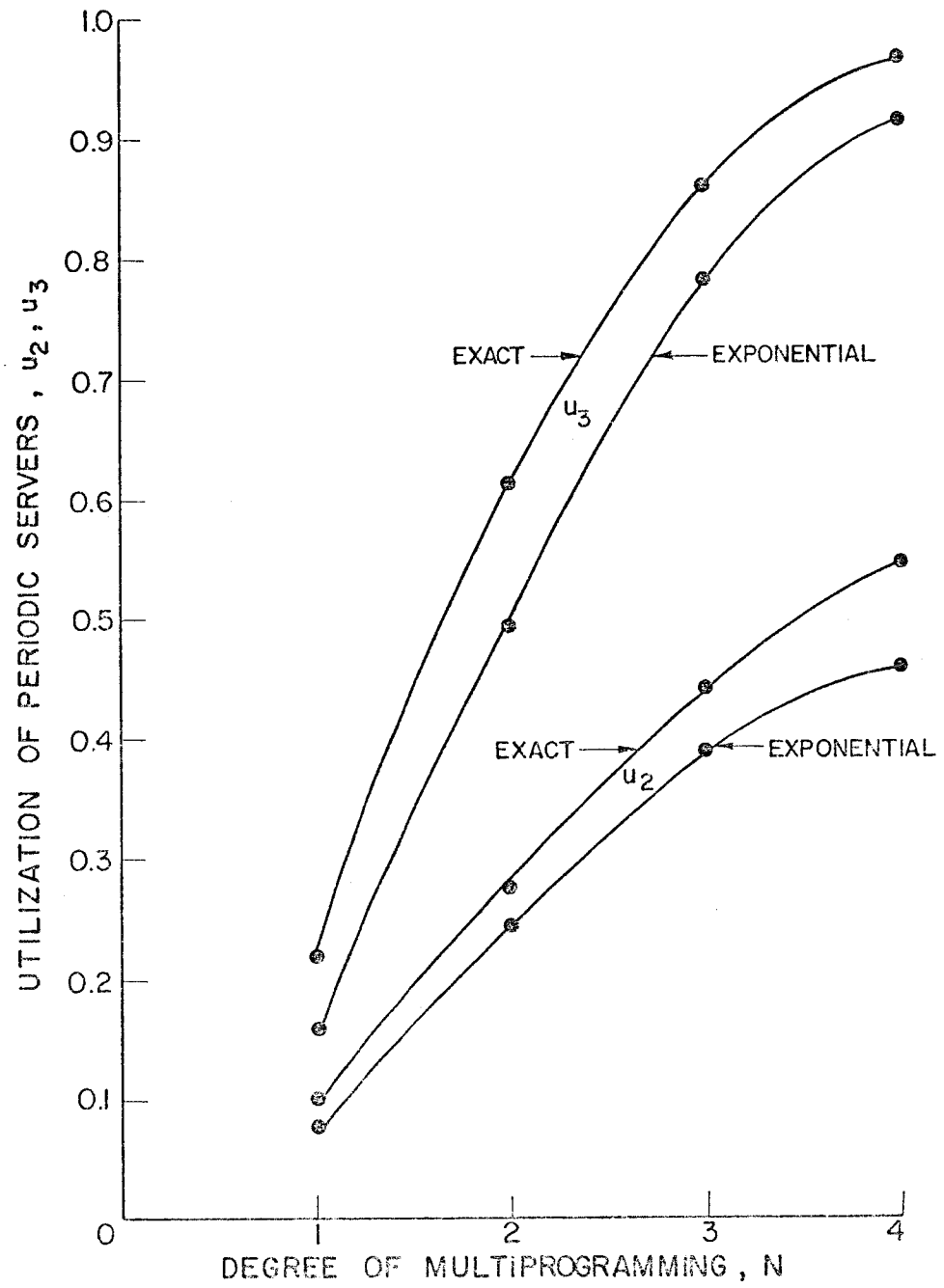


Figure VII

5. Conclusions and further work.

In this paper we present a method for the exact solution of queueing networks that contain periodic servers. The method is primarily applicable to networks with a small number of servers.

The example applications presented here deal with computer systems, featuring values of the period τ in the order of milliseconds. Neither the problem statement, nor the assumptions, nor the methods of solution presented depend on the periods of the periodic server been small. Thus, the method is applicable to any kind of system that can be modelled as a queueing network having either periodic or exponential servers, regardless of the values of τ . A wide variety of kinds of systems can be modelled, from banks to supermarkets to computer networks. Note that any daily process is by nature periodic.

A number of example queueing networks were analyzed using these methods, and the results were compared with simulations. Discrepancies were found to be very small. No attempt has yet been made to compare measurements of a real system with results of using these methods. The necessary measurement tools are available [15], and plans exist to make such comparisons in the near future.

The work extends the class of functions that can be used as service distributions to include the class of all bounded variation periodic functions.

It appears possible to extend this work to provide solution of open networks, i.e., when external arrivals and departures are allowed. Further research should be done to see if it is possible to generalize the class of non-periodic service distributions (e.g., M/G/1) that can be included with periodic servers in queueing networks.

6. Acknowledgments.

The authors wish to acknowledge the helpful comments of Professors J.D. Lawson, J. Wong, and J. Majithia. The authors are grateful for the support of the National Research Council and Defense Research Board of Canada, and the University of Waterloo. The programming assistance of M.Unsoy and the very constructive criticisms of the referees are gratefully acknowledged.

7. References.

- [1] M. Badel, E. Gelenbe, J. Leroudier, and D. Potier, "Adaptive Optimization of a Time-Sharing System's Performance", *Proceedings of the IEEE*, vol. 63-6, June 1975, pp. 958-965.
- [2] F. Baskett, K.M. Chandy, R.R. Muntz, and F.G. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers", *Journal of the A.C.M.*, vol. 22-2, Apr. 1975, pp. 248-260.
- [3] A. Brandwajn, "A Model of a Time Sharing Virtual Memory System Solved Using Equivalence and Decomposition Methods", *Acta Informatica*, vol. 4, 1974, pp. 11-47.
- [4] K.M. Chandy, U. Herzog, and L. Woo, "Parametric Analysis of Queueing Networks", *IBM Journal of Research and Development*, vol. 19-1, Jan. 1975, pp. 36-42.
- [5] E.G. Coffman, and P.J. Denning, *Operating Systems Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1973, ch. 5.
- [6] R.A. Frazer, W.J. Duncan, and A.R. Collar, *Elementary Matrices* London: Cambridge University Press, 1965.

- [7] S.H. Fuller, and F. Baskett, "An Analysis of Drum Storage Units", *Journal of the A.C.M.*, vol. 22-1, Jan. 1975, pp. 83-105.
- [8] G.H. Gonnet, and D.E. Morgan, "Analysis of Closed Queueing Networks with Periodic Servers", CCNG Report E-58, University of Waterloo, Waterloo, Ontario, Canada.
- [9] W.J. Gordon, and G.F. Newell, "Closed Queueing Systems with Exponential Servers", *Operations Research*, vol. 15, 1967, pp. 254-265.
- [10] J.R. Jackson, "Networks of Waiting Lines", *Operations Research*, vol. 5, 1957, pp. 518-521.
- [11] J.R. Jackson, "Jobshop-like Queueing Systems", *Management Science*, vol. 10-1, Oct. 1963, pp. 131-142.
- [12] L. Kleinrock, *Queueing Systems*, New York, NY: John Wiley, 1974, vol. I.
- [13] L. Kleinrock, *Queueing Systems*, New York, NY: John Wiley, 1976, vol. II.
- [14] J. Leroudier, and M. Parent, "Discrete Event Simulation Modelling of Computer System for Performance Evaluation", Rapport de Recherche 177, IRIA, France, June 1976.
- [15] D.E. Morgan, W. Banks, D.P. Goodspeed, and R. Kolanko, "A Computer Network Monitoring System", *IEEE Transactions on Software Engineering*, vol. SE-1, Sep. 1975, pp. 299-311.
- [16] R.S. Varga, *Matrix Iterative Analysis*, Englewood Cliffs, NJ: Prentice Hall, 1962.