Cyclic-Service Systems with Probabilistically-Limited Service

Kin K. Leung, *Member, IEEE*

***Abstract*—We analyze an asymmetric cyclic-service system with a probabilistically-limited (P-L) service policy. In such a service policy, the maximum number of customers served at a queue during a server visit is determined by a probability, which is independent of system states. Exhaustive, limited-k, and Bernoulli service are special cases of the P-L policy. Customer service times and changeover times have general distributions. A numerical technique based on discrete Fourier transforms is proposed to solve for the queue-length distributions. Thus, the waiting and response time distributions are obtained. A set of numerical examples is presented to validate the approach.**

I. Introduction

A CYCLIC-SERVICE SYSTEM (also known as a *polling system* or *token-passing system*) is a set of queues served by a single server in a cyclic manner. Excellent surveys of results on cyclic-service systems with extensive lists of references have been presented by Takagi in [24] and [25]. Such considerable research attention is due to the wide applicability of these models in communication, computer, and production systems.

Various service disciplines for cyclic-service systems have been studied, which include *exhaustive*, *gated*, and *limited service* policies. The first two policies have been completely solved, in the sense that their waiting time distributions have been obtained (cf. [8], references in [25]), while the average waiting times can be computed by solving a set of linear equations (cf. [11], [22]). However, the systems with limited service are difficult to analyze and few exact results exist. In particular, for symmetric systems with limited-one service (also known as *alternating* or *nonexhaustive service*), the average waiting times have been obtained (cf. [12], [20]). Asymmetric systems with two queues and limited-one service have been solved in [3], [6], and [9]. Recently, a numerical solution for systems with Bernoulli service has been proposed in [1]. Because of the analytical difficulty, many researchers (cf. [4], [13], [14], [23]) approximate the average waiting times for various limited service policies.

In this paper, we consider an asymmetric cyclic-service system with a probabilistically-limited (P-L) service policy. In such a service policy, the maximum number of customers served at a queue during a server visit is determined by a probability, which is independent of system states and can be different for various queues.

The P-L policy is motivated by two major reasons. First, this policy allows a unified treatment in the analysis of systems that involve several commonly-used service policies such as exhaustive service, limited-k service (referred to as *E-limited service* in [10] and [13]), and Bernoulli service [23]. Such a unified approach is possible because these disciplines are the P-L policy with appropriate probability settings. Second, the P-L policy complements the inadequacy of existing policies. For example, these policies serve at least one customer from a nonempty queue at each server visit, while the P-L policy allows the server not to serve the queue. Such a situation occurs on a processor in the 5ESS® Switch developed by AT&T. On that processor, when a nonempty queue is polled, it may not serve the queue in that visit, depending on the state of other parts of the switch.

To study the system with the P-L policy, we adopt the approach introduced in [8] and consider imbedded Markov chains formed at the instants of *(customer) service beginning*, *service completion*, *(server) visit beginning*, and *visit completion*. (A visit is the period of time at which the server is continuously serving a particular queue. If no customer is served during a visit, the visit beginning and completion occur simultaneously. This situation occurs either if the queue is empty when it starts to be served, or the server chooses not to serve any customer despite the fact that the queue is nonempty. The instant of a visit beginning is also referred to as a *polling instant*.) We express the queue-length distributions as functions of the probability generating functions (pgf's) for the state probabilities observed at visit-completion instants. A numerical technique based on discrete Fourier transforms (DFT's) is used to solve for the pgf's, from which the response time and waiting time distributions are obtained.

In the following, Section II describes the model and its assumptions. The pgf for the marginal queue-length distribution and the Laplace transforms (LT's) for response and waiting times at each queue are also derived. This pgf involves other unknown pgf's, for which a functional equation is developed in Section III. In Section IV, a numerical technique based on DFT's is proposed to solve for the unknown pgf's. Section V discusses the estimates of the "maximum" queue lengths and correctness checks on the numerical results. A set of numerical examples is given in Section VI. Finally, Section VII presents our conclusions.

II. Response and Waiting Time Analysis

Consider a cyclic-service system with M queues, indexed by $i = 1, 2, \cdots, M$, which are served by a single server in a cyclic manner. Customers in a queue are served in the order of their arrival. Each queue is assumed to have infinite waiting room. At each queue i, customers arrive according to a Poisson process with a rate λ_i, and their service times have a general distribution with a mean denoted by $\bar{x}_i$. The offered traffic at

Manuscript received April 20, 1990; revised September 13, 1990. This paper was presented in part at the IEEE INFOCOM 90, San Francisco, CA, June 1990, and at the 14th International Symposium on Computer Performance Modeling, Measurement and Evaluation, Edinburgh, Scotland, September 1990.

The author is with AT&T Bell Laboratories, Holmdel, NJ 07733.

IEEE Log Number 9040839.

0733-8716/91/0200-0185$01.00 © 1991 IEEE

queue i is defined as $\rho_i = \lambda_i \bar{x}_i$. Thus, the total offered traffic in the system is $\rho = \sum_{i=1}^{M} \rho_i$.

The service discipline in use is the P-L policy, as described above. More precisely, each time the server polls queue i (i.e., at the visit beginning), the maximum number of customers to be served (preferred to as the *service limit*) during the visit is determined as follows. Each queue i has a set of *service limit probabilities* $\{a_i^j\}$ where $\sum_{j=0}^{\infty} a_i^j = 1$. The service limit for a server visit at queue i is equal to j with probability a_i^j. The choices of service limits are independent of the system states. During a visit, when the queue in service becomes empty or its service limit has been reached (whichever occurs first), the server switches to visit (serve) the next queue. We denote the *changeover time* incurred by the server to switch from queue i $-$ 1 to i by c_i, which has a general distribution. It is understood the subscript $i - 1$ must be replaced by M for $i = 1$.

Remark 1: If a_i^0 is nonzero and queue i is nonempty when it is polled, it is possible that no customer from the queue is served during the server visit. □

Remark 2: The P-L policy includes the common service disciplines such as exhaustive, limited-k, and Bernoulli service by appropriate probability settings. In fact, the P-L policy can be a mixed service discipline. To illustrate this, if queue i has exhaustive service, then $a_i^{\infty} = 1$ and $a_i^j = 0$ for $j < \infty$. If queue i has limited-k service, its service limit for each visit is a constant (denoted by K_i) and we have $a_i^{K_i} = 1$ and $a_i^j = 0$ for $j \neq K_i$. Further, if queue i receives Bernoulli service with a probability p_i, then $a_i^0 = 0$ and $a_i^j = (1 - p_i)\, p_i^{j-1}$ for $j = 1, 2, \cdots, \infty$. □

Given $\{a_i^j\}$, the service limit averaged over all server visits at queue i is $\bar{L}_i = \sum_{j=0}^{\infty} j a_i^j$. We assume that each queue in the system is stable and define the *cycle time* as the time interval between two polling instants at the same queue. Then, by the balancing argument [2], the average cycle time at steady state is

$$\bar{c} = \sum_{i=1}^{M} \bar{c}_i/(1 - \rho) \tag{1}$$

where $\bar{c}_i$ is the average of c_i. Under the P-L policy, one can extend the arguments in [2] to find that the necessary and sufficient conditions for all queues being stable are $\rho < 1$ and for each queue i, $\lambda_i \bar{c} < \bar{L}_i$.

A. Probability Generating Functions for System State Probabilities

To analyze the cyclic-service system with the P-L policy, we observe the system at these instants: service beginning, service completion, visit beginning, and visit completion. This results in a set of four imbedded Markov chains. In each chain, the system state is described by i, the queue where a service or visit takes place, and a vector $\underline{n} = (n_1, n_2, \cdots, n_M)$, where n_j is the number of customers present in queue j at an imbedded time epoch. Then, we recognize that the results of Section II in [8] are applicable to the system under consideration.

Remark 3: In fact, the analysis in Section II of [8] is so general that it is valid for any nonpreemptive service discipline, as long as each queue is stable. This is because no assumption involving the specifics of the service policy in use is made. The analysis has been applied to study the performance of disk storages in [5]. □

Before continuing our discussion, let us define some notation:

$X_i(s)$: LT for the customer service time at queue i,
$C_i(s)$: LT for the changeover time c_i,
$\underline{z} = (z_1, z_2, \cdots, z_M)$,
$X_i(\underline{z}) = X_i(\lambda_1 - \lambda_1 z_1 + \cdots + \lambda_M - \lambda_M z_M)$: pgf for the joint probabilities of the numbers of customers arriving at all queues during a queue-i customer service time,
$C_i(z) = C_i(\lambda_1 - \lambda_1 z_1 + \cdots + \lambda_M - \lambda_M z_M)$: pgf for the joint probabilities of the numbers of customers arriving at all queues during the changeover time c_i,
$P_{\pi}^i(\underline{n})$: state probability of the imbedded Markov chain formed at *service-completion* instants,

$$\pi^i(\underline{z}) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} z_1^{n_1} z_2^{n_2} \cdots z_M^{n_M} P_{\pi}^i(\underline{n}),$$

$P_{\alpha}^i(\underline{n})$: state probability of the imbedded Markov chain formed at *visit-beginning* instants,

$$\alpha^i(\underline{z}) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} z_1^{n_1} z_2^{n_2} \cdots z_M^{n_M} P_{\alpha}^i(\underline{n}),$$

$P_{\beta}^i(\underline{n})$: state probability of the imbedded Markov chain formed at *visit-completion* instants,

$$\beta^i(\underline{z}) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} z_1^{n_1} z_2^{n_2} \cdots z_M^{n_M} P_{\beta}^i(\underline{n}),$$

γ: the long-term ratio of the number of queue-i visit completions to the total number of (customer) service completions at all queues.

Using our notation and applying (19) in [8] to our system, we obtain

$$\pi^i(\underline{z}) = \frac{\gamma X_i(\underline{z})}{z_i - X_i(\underline{z})} \left[\beta^{i-1}(\underline{z}) C_i(\underline{z}) - \beta^i(\underline{z})\right] \tag{2}$$

for $i = 1, 2, \cdots, M$. This relates the pgf for state probabilities at a service completion to those for the state probabilities at the preceding and the subsequent visit completions. Note that (2) has two unknowns: the quantity γ and pgf's $\beta^i(\underline{z})$. Let us first find γ here. A recursive relation among $\beta^i(\underline{z})$'s is obtained and solve later.

Since the system is stable, γ can be obtained by observing the system behavior during an average cycle. Clearly, the average number of service completions in an average cycle equals $(\lambda_1 + \lambda_2 + \cdots + \lambda_M)\bar{c}$. This is because all arrivals are eventually served, as each queue is stable. Hence, the total arrival rate and the completion rate are equal at equilibrium. It is also clear that there is one and only one queue-i visit during each cycle. Combing these two facts yields

$$\gamma = 1/\left[(\lambda_1 + \lambda_2 + \cdots + \lambda_M)\bar{c}\right]. \tag{3}$$

B. Marginal Queue-Length, Response and Waiting Time Distributions

By the definition of $\pi^i(\underline{z})$, it is clear that $\pi^i(1, \cdots, 1, z_i, 1, \cdots, 1)$ is the pgf for the marginal queue-length probabilities at queue i observed at a service-completion epoch, and it happens that the service completion is a queue-i customer. In addition, $\pi^i(1, \cdots, 1)$ gives the probability that an arbitrary service completion is a service completion of a queue-i customer. Combining these two facts, the pgf, $N_i(z_i)$, for the marginal queue-length probabilities for queue i when a queue-i customer service is just completed is given by

$$N_i(z_i) = \pi^i(1, \cdots, 1, z_i, 1, \cdots, 1)/\pi^i(1, \cdots, 1). \tag{4}$$

Let us find $\pi^i(1, \cdots, 1)$. As mentioned above, the total arrival rate and the service completion rate are identical at equilibrium as the system is stable. Hence, the probability of an arbitrary service completion being a queue-i customer is simply the ratio of λ_i to the total arrival rate. That is, $\pi^i(1, \cdots, 1) = \lambda_i/(\lambda_1 + \cdots + \lambda_M)$. Substituting this into (4) and using (2)–(3), we obtain

$$N_i(z_i) = \frac{1}{\lambda_i \bar{c}} \cdot \frac{X_i(\lambda_i - \lambda_i z_i)}{z_i - X_i(\lambda_i - \lambda_i z_i)} \cdot \left[\beta^{i-1}(1, \cdots, 1, z_i, 1, \cdots, 1) \cdot C_i(1, \cdots, 1, z_i, 1, \cdots, 1) - \beta^i(1, \cdots, 1, z_i, 1, \cdots, 1)\right] \quad (5)$$

where $\bar{c}$ is given in (1). Since customers at each queue are served in the order of their arrival, the LT for the customer response time (waiting plus service time) at queue i is given by

$$T_i(s) = N_i(1 - s/\lambda_i). \quad (6)$$

As customer service is nonpreemptive, the LT for the waiting time at queue i is

$$W_i(s) = T_i(s)/X_i(s) = N_i(1 - s/\lambda_i)/X_i(s). \quad (7)$$

To obtain the queue length, response and waiting time distributions, our remaining task is to solve for $\beta^i(\underline{z})$'s in (5).

III. A Functional Equation for $\beta^i(\underline{z})$

In this section, we first establish a recursive relationship between $\beta^i(\underline{z})$ and $\beta^{i-1}(\underline{z})$ for $i = 1, \cdots, M$. Then, a functional equation for $\beta^i(\underline{z})$ is derived.

A. A Recursive Relation for $\beta^i(\underline{z})$ and $\beta^{i-1}(\underline{z})$

Consider that queue i is in service. To find $\beta^i(\underline{z})$, we first condition that the service limit for the current visit is L. Then, one needs to keep track of the system state at each service completion at queue i. Further, one has to deal with the fact that the visit ends when either queue i becomes empty or the service limit L has been reached, whichever occurs first. To overcome this, we consider a modified system in which the system enters an absorbing state whenever queue i becomes empty and the server serves exactly L "customers" at queue i during the visit. If queue i becomes empty after $j(<L)$ customers have been served, the rest of $L - j$ "customers" served during the visit are not real customers, but rather represent a time interval at which the system state remains unchanged. Thus, the system state in the original system at the visit completion, regardless of its cause, is identical to that in the modified system. Finally, we uncondition L with its probabilities $\{a_i^j\}$. This is our approach in the following.

Let us begin with some additional notation:

$\underline{\psi}^j = (\psi_1(j), \psi_2(j), \cdots, \psi_M(j))$: numbers of customers at all queues immediately after j service completions at queue i since the visit beginning (note that the index i is omitted from this notation for brevity),

$P_{\underline{\psi}}^j(\underline{n})$: state probability that $\underline{\psi}^j = \underline{n}$ and queue i is in service,

$$\Psi^j(\underline{z}) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} z_1^{n_1} z_2^{n_2} \cdots z_M^{n_M} P_{\underline{\psi}}^j(\underline{n}),$$

$\underline{\tau}^i = (\tau_1, \tau_2, \cdots, \tau_M)$: numbers of customers arriving at all queues during a queue-i customer service time,

$\underline{1}^i = (0, 0, \cdots, 1, 0, \cdots, 0)$: a zero vector, except that the ith element is 1.

First, we note that the customers in the system at a queue-i visit beginning are those at the queue-$(i-1)$ visit completion plus the arrivals during the changeover time c_i. Written in terms of pgf's, we have $\alpha^i(\underline{z}) = \beta^{i-1}(\underline{z})C_i(\underline{z})$. By definition, $\Psi^0(\underline{z}) = \alpha^i(\underline{z})$. Thus, we also have

$$\Psi^0(\underline{z}) = \beta^{i-1}(\underline{z})C_i(\underline{z}). \quad (8)$$

Now, let us relate the system state at a service completion to that at the previous completion. Consider that j customers at queue i have completed their service during the current visit. Assume that $j < L$. Then, the system state at the $j+1$st service completion can be related to that at the service completion epoch of the jth customer by

$$\underline{\psi}^{j+1} = \begin{cases} \underline{\psi}^j & \text{if } \psi_i(j) = 0 \\ \underline{\psi}^j + \underline{\tau}^i - \underline{1}^i & \text{if } \psi_i(j) \geq 1 \end{cases} \quad (9)$$

for $j = 0, \cdots, L-1$. Now, we can use (9) to obtain a recursive relation between $\Psi^j(\underline{z})$ and $\Psi^{j+1}(\underline{z})$, as shown below.

Let I be the index of the queue at which the server is visiting. By definition, $P_{\underline{\psi}}^j(\underline{n}) = \Pr[\underline{\psi}^j = \underline{n}, I = i]$. Thus, we have

$$\Psi^{j+1}(\underline{z}) = E[z_1^{\psi_1(j+1)} z_2^{\psi_2(j+1)} \cdots z_M^{\psi_M(j+1)} | I = i] \Pr[I = i]$$
$$= \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} E[z_1^{\psi_1(j+1)} z_2^{\psi_2(j+1)} \cdots z_M^{\psi_M(j+1)} | \underline{\psi}^j = \underline{n}, I = i] \Pr[\underline{\psi}^j = \underline{n}, I = i]. \quad (10)$$

Depending on whether $\psi_i(j) = 0$ or not, replace $\underline{\psi}^{j+1}$ in (10) according to (9). After algebraic manipulations, we get

$$\Psi^{j+1}(\underline{z}) = \Psi^j(\underline{z})\big|_{z_i=0} + \frac{X_i(\underline{z})}{z_i}\left[\Psi^j(\underline{z}) - \Psi^j(\underline{z})\big|_{z_i=0}\right] \quad (11)$$

for $j = 0, 1, \cdots, L-1$. As discussed above, given that the service limit for the visit is L, the system state at the queue-i visit completion is characterized by $\Psi^L(\underline{z})$. Thus, unconditioning L with probability a_i^L in this relation yields

$$\beta^i(\underline{z}) = \sum_{j=0}^{\infty} a_i^j \Psi^j(\underline{z}). \quad (12)$$

It is important to recognize that $\beta^i(\underline{z})$ is related to $\beta^{i-1}(\underline{z})$, via (12), (11), and (8), for $i = 1, \cdots, M$. This forms a recursive relation between $\beta^i(\underline{z})$ and $\beta^{i-1}(\underline{z})$, which leads to a functional equation, as described next.

Remark 4: If each queue receives the limited-one service, one can obtain from (8), (11), and (12):

$$\beta^i(\underline{z}) = \left[1 - \frac{X_i(\underline{z})}{z_i}\right]\left\{\beta^{i-1}(\underline{z})C_i(\underline{z})\right\}\Big|_{z_i=0} + \frac{X_i(\underline{z})}{z_i}\beta^{i-1}(\underline{z})C_i(\underline{z}) \quad (13)$$

for all $i = 1, \cdots, M$. □

B. The Functional Equation

Let us focus on the numbers of customers at all queues, $\underline{n} = (n_1, \cdots, n_M)$. We observe that, given $\underline{n}$ at a visit completion of a particular queue, the state probabilities for $\underline{n}$ at the next visit completion of the same queue can be fully characterized (although it is very involved mathematically). Further, by steady-state arguments, the state probability for $\underline{n}$ at two consecutive instants of visit completion at the same queue are identically distributed. Formally, let F_i denote the mapping from $\beta^{i-1}(\underline{z})$ to $\beta^i(\underline{z})$, as defined by (8), (11), and (12). Hence, we have $\beta^i(\underline{z}) = F_i(\beta^{i-1}(\underline{z}))$ for $i = 1, \cdots, M$. Then, recursively replacing $\beta^{i-1}(\underline{z})$ in this relation by $F_{i-1}(\beta^{i-2}(\underline{z}))$ yields

$$\beta^i(\underline{z}) = F_i\Big(F_{i-1}\Big(\cdots F_1\big(F_M\big(\cdots F_{i+1}(\beta^i(\underline{z}))\cdots\big)\big)\cdots\Big)\Big). \tag{14}$$

This is the functional equation for $\beta^i(\underline{z})$ for any given $i = 1, \cdots, M$.

Apparently, it is difficult to solve for $\beta^i(\underline{z})$ from (14) analytically because the mapping functions are so complicated. However, (14) can be used as a basis for an iterative procedure as follows. One can choose an initial guess for $\beta^i(\underline{z})$ (e.g., corresponding to an empty system), which is input to (14) as the argument to generate a new result for $\beta^i(\underline{z})$. Then, this process is repeated by substituting the new result into (14) as the argument again. Since the system has a steady state, after a sufficiently large number of iterations, $\beta^i(\underline{z})$ converges to the steady-state solution, as one would expect. Once $\beta^i(\underline{z})$ for some i is obtained, all other $\beta^j(\underline{z})$'s also become known by applying the mappings on $\beta^i(\underline{z})$ appropriately. Thus, the remaining task is to solve for $\beta^i(\underline{z})$ from (14) numerically.

IV. Solving $\beta^i(\underline{z})$ by DFT Technique

In this section, we closely approximate $\Psi^j(\underline{z})$ and $\beta^i(\underline{z})$ by their corresponding DFT's, $\{\hat{\Psi}^j(\cdot)\}$ and $\{\hat{\beta}^i(\cdot)\}$, respectively. Then, an iterative algorithm is proposed to solve $\{\hat{\beta}^i(\cdot)\}$ from (14).

A. Approximation by DFT's

Clearly, each queue has a finite queue length at steady state, as all queues are stable. Thus, for each queue i, one can estimate its "maximum" queue length, including the one in service, to be N_i (such estimation is discussed later) and make the following approximation:

$$\Psi^j(\underline{z}) \approx \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\cdots\sum_{n_M=0}^{N_M-1} z_1^{n_1} z_2^{n_2}\cdots z_M^{n_M} P_\psi^j(\underline{z}). \tag{15}$$

Since the number of all possible states for $\underline{n}$ now becomes finite due to the approximation, one can use DFT's to represent $\{P_\psi^j(\underline{n})\}$, instead of using z-transform in (15), so that a useful property of DFT can be used in computation. To obtain the DFT's, we define $\omega_i = e^{-2\pi j/N_i}$ for $i = 1, \cdots, M$ where $j = \sqrt{-1}$. (Note that j is also used as an integer index.) Further, let $\underline{k} = (k_1, k_2, \cdots, k_M)$ where $k_i = 0, \cdots, N_i - 1$, and let S be the set of all possible $\underline{k}$. Let the DFT's for $\{P_\psi^j(\underline{n})\}$ be denoted by $\{\hat{\Psi}^j(\underline{k})\}$. By the definition of DFT, $\{\hat{\Psi}^j(\underline{k})\}$ is given by

$$\hat{\Psi}^j(\underline{k}) = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\cdots\sum_{n_M=0}^{N_M-1}\omega_1^{k_1 n_1}\omega_2^{k_2 n_2}\cdots\omega_M^{k_M n_M} P_\psi^j(\underline{n}) \tag{16}$$

for each $\underline{k} \in S$. Similarly, one can obtain the DFT's $\{\hat{\beta}^i(\underline{k})\}$ for $\{P_\beta^i(\underline{n})\}$ for $i = 1, \cdots, M$. Further, let the DFT's corresponding to $C_i(\underline{z})$ and $X_i(\underline{z})$ be denoted by $\{\hat{C}_i(\underline{k})\}$ and $\{\hat{X}_i(\underline{k})\}$, respectively.

One major difficulty in dealing with the z-transform is the computation of $\Psi^j(\underline{z})|_{z_i=0}$, and $\beta^{i-1}(\underline{z})|_{z_i=0}$ and $C_i(\underline{z})|_{z_i=0}$ for $j = 0$ in (11). These pgf's actually represent the cases where queue i is conditioned to be empty at the imbedded epochs. Such difficulty can be overcome by the appropriate use of the DFT's. For this, we define, for each $\underline{k} \in S$ and $j = 0, \cdots, \infty$,

$$\begin{aligned}\hat{\Psi}^j(k_1, \cdots, k_{i-1}, \theta, k_{i+1}, \cdots, k_M) &= \sum_{n_1=0}^{N_1-1}\cdots\sum_{n_{i-1}=0}^{N_{i-1}-1}\sum_{n_{i+1}=0}^{N_{i+1}-1}\cdots\sum_{n_M=0}^{N_M-1} \omega_1^{k_1 n_1}\cdots\omega_{i-1}^{k_{i-1} n_{i-1}}\omega_{i+1}^{k_{i+1} n_{i+1}}\cdots\omega_M^{k_M n_M} \\ &\quad \cdot P_\psi^j(n_1, \cdots, n_{i-1}, 0, n_{i+1}, \cdots, n_M).\end{aligned} \tag{17}$$

This notation is explained below. Note that queue i is empty, as indicated in the argument of $P_\psi^j(\cdot)$ on the RHS. With this definition, one can make use of the property of the DFT [21] to prove that

$$\hat{\Psi}^j(k_1, \cdots, k_{i-1}, \theta, k_{i+1}, \cdots, k_M) = \frac{1}{N_i}\sum_{k_i=0}^{N_i-1}\hat{\Psi}^j(k_1, \cdots, k_{i-1}, k_i, k_{i+1}, \cdots, k_M). \tag{18}$$

Now, let us clarify this notation. Certainly, one could use another notation to take the place of $\hat{\Psi}^j(k_1, \cdots, k_{i-1}, \theta, k_{i+1}, \cdots, k_M)$ in (17) and (18). However, to simplify our notation, it is used to show that k_j's, except k_i, are identical on both sides of (18). In addition, θ is a dummy argument which takes the ith position (to replace k_i) to indicate the fact that queue i is empty. Once the DFT's $\{\hat{\Psi}^j(\underline{k})\}$ are known, $\{\hat{\Psi}^j(k_1, \cdots, k_{i-1}, \theta, k_{i+1}, \cdots, k_M)\}$ are also known from (18).

Using a similar definition for $\hat{\Psi}^j(k_1, \cdots, k_{i-1}, \theta, k_{i+1}, \cdots, k_M)$, we can get

$$\hat{\beta}^{i-1}(k_1, \cdots, k_{i-1}, \theta, k_{i+1}, \cdots, k_M) = \frac{1}{N_i}\sum_{k_i=0}^{N_i-1}\hat{\beta}^{i-1}(k_1, \cdots, k_{i-1}, k_i, k_{i+1}, \cdots, k_M) \tag{19}$$

and

$$\begin{aligned}\hat{C}_i(k_1, \cdots, k_{i-1}, \theta, k_{i+1}, \cdots, k_M) &= \frac{1}{N_i}\sum_{k_i=0}^{N_i-1} C_i(\lambda_1 - \lambda_1\omega_1^{k_1} + \cdots + \lambda_i \\ &\quad - \lambda_i\omega_i^{k_i} + \cdots + \lambda_M - \lambda_M\omega_M^{k_M}).\end{aligned} \tag{20}$$

B. An Iterative Algorithm

Before proposing an iterative algorithm to solve for $\{\hat{\beta}^i(\underline{k})\}$, one needs to keep all possible service limits at each queue finite so that the iterations can converge in a finite amount of time. To achieve this, except for exhaustive service, one can deter-

mine a service limit for queue i, denoted by L, such that $\sum_{j=L+1}^{\infty} a_i^j$ is less than a very small value (e.g., $\leq 10^{-9}$). If queue i has exhaustive service, we set $L = \infty$. Then, the "maximum" service limit for queue i, denoted by L_i^m, can be chosen to be the minimum of $N_i - 1$ and L. (In some specific situations, one may choose $L_i^m \geq N_i$. See examples in Table III.)

Since $\Psi^j(\underline{z})$ and $\beta^i(\underline{z})$ are proper pgf's, (8), (11), and (12) are valid as long as $|z_i| \leq 1$ for all $i = 1, \cdots, M$. Thus, one can replace each z_i in these equations by $\omega_i^{k_i}$ for any $k_i = 0, \cdots, N_i - 1$. As a result of using DFT's, the recursive relation given in terms of z-transforms can be converted into a set of relations in terms of DFT's. That is, for a given $i = 1, \cdots, M$, (8), (11), and (12) are equivalent to, for all $\underline{k} \in S$,

$$\hat{\Psi}^0(\underline{k}) = \hat{\beta}^{i-1}(\underline{k})\hat{C}_i(\underline{k}), \tag{21}$$

$$\hat{\Psi}^{j+1}(\underline{k}) = \left[1 - \frac{\hat{X}_i(\underline{k})}{\omega_i^{k_i}}\right]\hat{\Psi}^j(k_1, \cdots, k_{i-1}, \theta, k_{i+1}, \cdots, k_M) + \frac{\hat{X}_i(\underline{k})}{\omega_i^{k_i}}\hat{\Psi}^j(\underline{k}) \tag{22}$$

for $j = 0, \cdots, L_i^m - 1$ [note that the index i is omitted from $\hat{\Psi}^j(\cdot)$], and

$$\hat{\beta}^i(\underline{k}) = \sum_{j=0}^{L_i^m} a_i^j \hat{\Psi}^j(\underline{k}), \tag{23}$$

respectively. Now, (21)–(23) become the DFT version of the mapping function $F_i(\cdot)$ for $i = 1, \cdots, M$. Consequently, they allow us to solve for $\{\hat{\beta}(\underline{k})\}$ from (14).

Now, let us outline the iterative approach to solving for $\{\hat{\beta}^i(\underline{k})\}$. For brevity, let $\underline{k}^{(i)} = (k_1, \cdots, k_{i-1}, \theta, k_{i+1}, \cdots, k_M)$. For easy referencing during the iterations, one should precalculate $\{\hat{X}_i(\underline{k})\}$, $\{\hat{C}_i(\underline{k})\}$, and $\{\hat{C}_i(\underline{k}^{(i)})\}$ (from (20)) for $i = 1, \cdots, M$ and for each $\underline{k} \in S$ as they are fixed for a given set of system parameters. To start the iterations, we choose the initial guess for $\{\hat{\beta}^{i-1}(\underline{k})\}$ corresponding to an empty system. That is, $\hat{\beta}^{i-1}(\underline{k}) = 1$ for each $\underline{k} \in S$. Then, $\{\hat{\beta}^{i-1}(\underline{k}^{(i)})\}$ are obtained from (19). For each $\underline{k} \in S$, $\hat{\beta}^{i-1}(\underline{k})$, $\hat{\beta}^{i-1}(\underline{k}^{(i)})$, $\hat{X}_i(\underline{k})$, $\hat{C}_i(\underline{k})$, and $\hat{C}_i(\underline{k}^{(i)})$ are input into (21)–(23) to generate $\hat{\beta}^i(\underline{k})$. These DFT's $\{\hat{\beta}^i(\underline{k})\}$ are then used as arguments to generate $\{\hat{\beta}^{i+1}(\underline{k})\}$ according to (21)–(23) again. After going through all mappings in sequence as indicated in (14), the new results of $\{\hat{\beta}^i(\underline{k})\}$ are compared with the old ones. If for all $\underline{k} \in S$, their difference is less than a tolerable error (e.g., 10^{-8}), then the iteration stops. These DFT's $\{\hat{\beta}^i(\underline{k})\}$ represent the steady-state results. Once $\{\hat{\beta}^i(\underline{k})\}$ is obtained, we can input them $F_{i+1}(\cdot)$ and then the others, as defined in (21)–(23), in sequence to obtain other DFT's $\{\hat{\beta}^{\cdot}(\underline{k})\}$.

Using $\{\hat{\beta}^i(\underline{k})\}$ to find $N_i(z_i)$, we set $z_i = \omega_i^{k_i}$ for $k_i = 0, \cdots, N_i - 1$ in (5). Note that the DFT equivalences of $\beta^{i-1}(1, \cdots, 1, \omega_i^{k_i}, 1, \cdots, 1)$, $\beta^i(1, \cdots, 1, \omega_i^{k_i}, 1, \cdots, 1)$ and $C_i(1, \cdots, 1, \omega_i^{k_i}, 1, \cdots, 1)$ are $\hat{\beta}^{i-1}(0, \cdots, 0, k_i, 0, \cdots, 0)$, $\hat{\beta}^i(0, \cdots, k_i, 0, \cdots, 0)$, and $\hat{C}_i(0, \cdots, k_i, 0, \cdots, 0)$, respectively, which have been obtained from the iterations. For $k_i = 0, \cdots, N_i - 1$, setting $z_i = \omega_i^{k_i}$ and substituting the DFT's into (5) correspondingly yield the DFT's, $\{\hat{N}_i(k_i)\}$, for the marginal queue-length probabilities at queue i. By inversion of DFT's from $\{\hat{N}_i(k_i)\}$, we obtain the queue-length probabilities, $\{P_{k_i}^i\}$ for $k_i = 0, \cdots, N_i - 1$. Using these probabilities, $N_i(z_i)$ becomes

$$N_i(z_i) = \sum_{k_i=0}^{N_i-1} z_i^{k_i} P_{k_i}^i. \tag{24}$$

Substituting this into (6) and (7), the response and waiting time distributions are given by

$$T_i(s) = \sum_{k_i=0}^{N_i-1} P_{k_i}^i (1 - s/\lambda_i)^{k_i} \tag{25}$$

and

$$W_i(s) = \frac{1}{X_i(s)} \cdot \sum_{k_i=0}^{N_i-1} P_{k_i}^i (1 - s/\lambda_i)^{k_i}, \tag{26}$$

respectively. Differentiating (26) at $s = 0$ gives the average waiting time at queue i:

$$\bar{w}_i = \frac{1}{\lambda_i} \cdot \sum_{k_i=0}^{N_i-1} k_i P_{k_i}^i - \bar{x}_i. \tag{27}$$

Note that one can easily obtain the moments of response/waiting times from (25) and (26). However, since both series do not converge for all s with Re $(s) \geq 0$, the distribution functions may not be obtained from common methods of LT inversion. To find the response time distribution, we use the relation $T_i(\lambda_i - \lambda_i z_i) = N_i(z_i)$ where $N_i(z_i)$ is given by (24). The complementary cumulative function for the response time is closely approximated by a sum of Laguerre functions weighted by unknown coefficients [18]. Then, $T_i(\lambda_i - \lambda_i z_i)$ is expanded into a Taylor series at $z_i = 0$, in which the coefficients are linear functions of the unknown coefficients. Since the coefficient of $z_i^{k_i}$ in the expansion must be equal to $P_{k_i}^i$, the unknown coefficients can be recovered from solving a set of linear equations, thus the response time distribution is obtained. Similarly, one can also recover the waiting time distribution. If $\{P_{k_i}^i\}$ are accurate, this new inversion method yields the distributions very accurately. (This has been verified on other systems for which exact solutions are known.) However, due to the aliasing phenomenon of the DFT's [7] and roundoff errors, our experience shows that, using $\{P_k^i\}$ generated from the iterations, this method provides the response time distribution with an estimated relative error of a fraction of a percent. Since the LT for the waiting time involves $X_i(s)$ (i.e., a convolution in time domain) which requires $\{P_{k_i}^i\}$ to be more accurate, one can obtain the waiting time distribution by this method with a relative error of less than 1.5%.

V. Estimates of Maximum Queue Lengths and Correctness Checks

In this section, we consider systems with limited-k and Bernoulli service. The methods proposed in [13] or [23] can be used to approximate the average waiting time at each queue. Once the average waiting times are approximated, an estimate of the mean length of each queue i is known by Little's law. Let this quantity be $\tilde{n}_i$. Now, the behavior of each queue is approximated as an individual $M/M/1$ queue with a server utilization of $\tilde{\rho}_i$, such that its mean queue length matches $\tilde{n}_i$. By $M/M/1$ results, this yields $\tilde{n}_i = \tilde{\rho}_i/(1 - \tilde{\rho}_i)$. Thus, $\tilde{\rho}_i$ can be solved from this relation as $\tilde{n}_i$ is known. Since N_i has to be chosen such that the probability of reaching that queue length

TABLE I
AVERAGE WAITING TIMES FOR 3-QUEUE SYSTEMS WITH LIMITED-ONE SERVICE

Cases	(N_1, N_2, N_3)	$\bar{w}_1$	$\bar{w}_2$	$\bar{w}_3$	$\Sigma_{i=1}^{3} \rho_i[1 - \lambda_i \bar{c}]\bar{w}_i$	Conservation Constant
1.1	(51, 36, 36)	6.86406 (6.80)	5.45389 (5.38)	5.49541 (5.38)	3.78498	3.7848
1.2	(67, 54, 54)	11.40066 (10.72)	9.06770 (8.30)	9.10407 (8.30)	4.18990	4.1904
2.1	(98, 15, 15)	9.28629 (9.34)	1.87256 (1.89)	1.90727 (1.89)	2.96565	2.9656
2.2	(281, 26, 26)	58.46543 (55.70)	2.32686 (2.31)	2.36956 (2.31)	3.33234	3.3712

Note: Average waiting times in parentheses are simulation results adopted from [4] and [13].

has a very small value ϵ (e.g., $\leq 10^{-7}$), by the $M/M/1$ queue-length distribution, an estimate for N_i can be obtained from

$$(1 - \tilde{\rho}_i)\tilde{\rho}_i^{N_i - 1} \leq \epsilon. \tag{28}$$

This method provides a set of initial choices for $\{N_i\}$. If the correctness checks (to be discussed) indicate that the numerical results generated from these initial choices are not satisfactory, they can be revised accordingly.

One simple way to check the correctness of the numerical results—queue-length probabilities—is by observation. If $\{N_i\}$ is chosen properly and the computation is performed as expected, the queue-length probabilities decay to a very small value (e.g., on the order of 10^{-6}) as the queue length approaches its maximum value.

Another method to check the correctness of the results—average waiting times—is to make use of the waiting-time conservation (pseudoconservation) laws. If the system uses the limited-one or Bernoulli service policy, two conservation laws have been proved in [27] and [26], respectively. These laws simply state that the weighted sum of average waiting times equals a constant (referred to as the *conservation constant*). For the limited-k service, one can apply another conservation law given by (3) in [10]. Note that $g_i^{(2)}$ (the second factorial moment of the number of queue-i customers served during each visit) in this law is an unknown. Nevertheless, we know that $g_i^{(2)} \geq \max(0, \bar{g}_i^2 - \bar{g}_i)$ where $\bar{g}_i = \lambda_i \bar{c}$. Substituting this into the conservation law yields an upper bound for the weighted sum of average waiting times. If the computation is performed properly, the average waiting times obtained from the numerical approach should closely satisfy the conservation laws. For the cases of the limited-k service, our numerical results also show that the upper bound for the conservation constant is tight.

VI. NUMERICAL RESULTS

We considered four sets of numerical examples. These examples have been previously studied by other researchers, for which exact or approximation results have been obtained. To obtain high quality results, our computation was performed on a CRAY X-MP computer with 64-bit words. The iteration stopping criterion was 10^{-8}, except for the second set of examples where the criterion was 10^{-7} to reduce CPU time. The method in Section V was used to obtain the initial estimates of $\{N_i\}$ with $\epsilon = 2 \times 10^{-8}$ in (28).

A. Two-Queue Systems with Limited-One Service

First, we considered several two-queue systems with limited-one service for which the exact average waiting times have been obtained in Table I of [3]. Our algorithm was able to reproduce the average waiting times reported in the table and the CPU time consumption was about 0.1 s to 2 min.

B. Three-Queue Systems with Limited-One Service

Second, we considered a set of four 3-queue systems with limited-one service. These systems correspond to Cases 1.1, 1.2, 2.1, and 2.2 in [13]. Table I summarizes the average waiting times obtained from the numerical procedure. The weighted sums of these waiting times along with their conservation constants are also presented. Clearly, the average waiting times closely satisfy the conservation law. The CPU times consumed by the procedure for the first three cases was about 3, 15.5, and 2.5 min, respectively. However, due to the extremely heavy load in Case 2.2, it required about 5.5 h of CPU time. By the conservation law, the numerical results for Case 2.2 are not as "exact" as in other cases, but we were not able to further improve the results. This was so because a little increase of (N_1, N_2, N_3)—from (281, 21, 21) to (281, 26, 26)—increased the CPU time from 2.5 to 5.5 h. However, only marginal improvement was obtained. Nevertheless, the average waiting times for Case 2.2 shown in Table I appear to be more "accurate" than the simulation results in [4] and [13] as the former comes closer to satisfying the conservation law than the latter. The response time distributions for queues 1 and 2 in Case 1.1 are given in Fig. 1. The queue-length distributions have been reported in [16].

C. Three-Queue Systems with Limited-k Service

The third set of examples involve systems with limited-k service, which correspond to Cases 9.1, 9.3, 11.5, and 11.7 of the E-limited service in [13]. Table II presents the average waiting times obtained from the algorithm, which were validated by simulation. The weighted sums of waiting times and the upper bounds for their respective conservation constants are also presented. The close comparison of the weighted sums with the bounds shows that they are tight bounds. The CPU time consumption ranges from 0.5 to 7.6 min for these cases. The re-

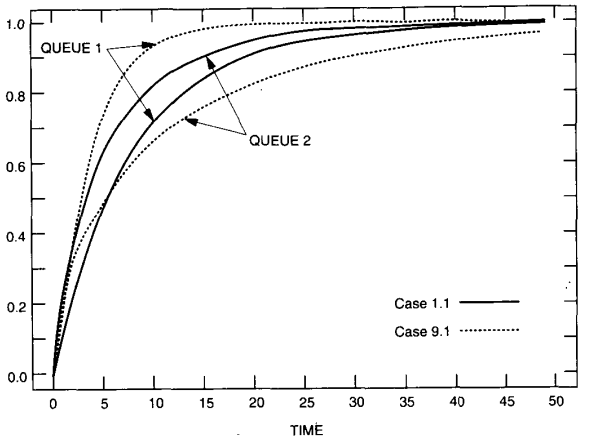


Fig. 1. Response time distributions.

TABLE II
AVERAGE WAITING TIMES FOR 3-QUEUE SYSTEMS WITH LIMITED-k SERVICE

Cases	(N_1, N_2, N_3)	$\overline{w}_1$	$\overline{w}_2$	$\overline{w}_3$	$\Sigma_{i=1}^{3} \rho_i \left[1 - \frac{\lambda_i \overline{c}}{K_i} \right] \overline{w}_i$	Upper Bound for Conservation Constant
9.1	(27, 63, 63)	2.54335 (2.5523)	10.42245 (10.0453)	10.48704 (10.1450)	3.62821	3.66959
9.3	(42, 56, 56)	3.71966 (3.7373)	6.84054 (6.8457)	6.97543 (6.9230)	3.57088	3.65039
11.5	(47, 23, 23)	3.05708 (3.0429)	4.58778 (4.6233)	4.62628 (4.5967)	2.72102	2.80239
11.7	(42, 30, 30)	2.04488 (2.0433)	5.69626 (5.6840)	5.80968 (5.7806)	2.71572	2.77295

Note: Average waiting times in parentheses are new simulation results.

TABLE III
AVERAGE WAITING TIMES FOR 2-QUEUE SYSTEMS WITH BERNOULLI SERVICE

Cases	(p_1, p_2)	(N_1, N_2)	(L_1^m, L_2^m)	$\overline{w}_1$	$\overline{w}_2$	$\Sigma_{i=1}^{2} \rho_i [1 - \lambda_i \overline{c}(1 - p_i)] \overline{w}_i$	Conservation Constant
B-1	(1, 0.5)	(28, 223)	(27, 30)	0.48983 (0.4897)*	5.76710 (5.7304)*	1.13032	1.13033
B-2	(1, 0.75)	(39, 98)	(38, 73)	0.84146 (0.8406)*	2.05188 (2.0513)*	1.04597	1.04596
B-3	(1, 1)	(116, 56)	(115, 55)	2.86369 (3.345)**	0.90698 (0.8361)**	0.96660	0.96158
B-4	(1, 1)	(116, 56)	(345, 165)	3.33215 (3.345)**	0.84059 (0.8361)**	0.96366	0.96158
B-5	(1, 1)	(231, 111)	(690, 330)	3.34494 (3.345)**	0.83615 (0.8361)**	0.96161	0.96158

Note: * and ** are new simulation and exact results, respectively.

sponse time distributions for Case 9.1 are shown in Fig. 1. The queue-length distributions are given in [17].

D. Two-Queue Systems with Bernoulli Service

Finally, we considered several two-queue systems with Bernoulli service. The approximate average waiting times for these systems have been presented in Fig. 2 and Table I of [23]. The system parameters are: $\lambda_1 = 2$, $\lambda_2 = 2.5$, $\overline{x}_1 = 0.05$ (exponential distribution), $\overline{x}_2 = 0.3$ (Erlang-3 distribution), and $\overline{c}_1 = \overline{c}_2 = 0.045$ (constant). Table III presents the average waiting times from the numerical approach for five cases that use these parameters and several Bernoulli probabilities, p_1 and p_2. Simulation or exact results for these systems along with the weighted sums of average waiting times and their conservation constants are also included in Table III. N_1 and N_2 for Cases *B*-1 to *B*-3 were obtained by the method in Section V. The maximum service

limits, (L_1^m, L_2^m), for the first two cases were chosen such that $L_1^m = N_1 - 1$ and $\Sigma_{j=L_2^m+1}^{\infty}(1 - p_2)p_2^{j-1} < 10^{-9}$. The procedure consumed about 22.5, 3, 0.3, 0.6, and 4.4 min of CPU time for Cases *B*-1 to *B*-5, respectively.

As shown in Table III, our numerical results for Cases *B*-1 and *B*-2 are correct, as they have been validated by both simulation and the conservation law. However, the results for Case *B*-3 are not satisfactory, despite that they closely satisfy the conservation law. This is because, as $p_1 = p_2 = 1$, the service policy become exhaustive service in this case. Due to the high variability of the server intervisit time under the exhaustive service scheme, the method in Section V no longer yields good estimates for $\{N_i\}$. To show the effects of maximum service limits, (L_1^m, L_2^m) for Case *B*-4 are three times those for Case *B*-3, while other parameters remain unchanged. Clearly, our results for Case *B*-4 come much closer to the exact values. For Case *B*-5, we double the (N_1, N_2) from that of Case *B*-3 and set the maximum service limits as three times these new maximum queue lengths. Then, the numerical results in Case *B*-5 are virtually identical to the exact results. This indicates that the proposed approach is capable of providing correct results, as long as $\{N_i\}$ and $\{L_i^m\}$ are chosen properly. Our study shows that the method in Section V generally yields good estimates for $\{N_i\}$, except for cases where most of the queues have exhaustive service. In those cases, one should repeat the algorithm with larger values for $\{N_i\}$ and $\{L_i^m\}$ to achieve accurate results.

VII. Conclusions and Future Work

An iterative numerical approach based on DFT's for asymmetric cyclic-service systems with a P-L service policy has been proposed. This techqniue has been validated by the waiting-time conservation laws (if applicable) and exact/simulation results. Since the memory and CPU time used by the algorithm are exponential functions of the number of queues, we currently can only solve relatively small systems if the offered traffic load is high. Nevertheless, our method is applicable to many applications where the number of queues involved is small. Further, results from the proposed approach—response and waiting time distributions which are often difficult to obtain via simulation—can be used to assess the accuracy of new approximate methods. For more general applications, this new technique is also applicable to solving imbedded Markov chains with a multidimensional state description (e.g., the disk performance problem analyzed in [5]).

This work can be further extended in a number of ways. First, the method in Section V for estimating $\{N_i\}$ needs to be generalized to consider the general P-L (mixed) service policy. Second, we note that the proposed approach is applicable to cyclic-service systems with compound Poisson arrivals and/or correlated arrivals [19]. This is so because $X_i(z)$ and $C_i(z)$ can be easily obtained according to these special arrival processes. And, the rest of the approach remains essentially unchanged. Further, we plan to apply the proposed technique to approximate the average waiting times for cyclic-service queues with a nonpreemptive time-limited service policy.

Finally, unless approximations are made or new techniques are developed, the order of magnitude of memory and CPU time required by the proposed approach is intrinsically $O(\Sigma_{i=1}^{M} L_i^m \Pi_{j=1}^{M} N_j)$. As an analogy to NP-complete problems, this exponential use of memory and CPU time provides additional insight into why cyclic-service systems with limited service are so difficult to solve. If the modeling problem under consideration is indeed an NP-type problem, only small systems are tractable and can be solved as the proposed technique is capable of doing. For systems with a moderate or large number of queues, we should be more convinced than ever before to pursue approximate solutions. Nevertheless, it appears that one can base on the proposed approach to develop new approximation techniques for analyzing cyclic-service systems with limited service.

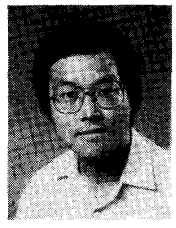
Acknowledgment

The author would like to thank M. Eisenberg and W. S. Wong for their discussions. Thanks are also due to Y. T. Wang for performing the simulation in Table II. The author also thanks E. G. Coffman, Jr., and H. Takagi for their comments.

References

[1] J. P. C. Blanc, "A numerical approach to cyclic-service queueing models," *Queueing Syst.*, vol. 6, pp. 173–188, 1990.
[2] O. J. Boxma and W. P. Groenendijk, "Pseudo-conservation laws in cyclic-service systems," *J. Appl. Prob.*, vol. 24, pp. 949–964, 1987.
[3] O. J. Boxma and W. P. Groenendijk, "Two queues with alternating service and switching times," *Queueing Theory and Its Applications—Liber Amicorum for J. W. Cohen*, O. J. Boxma and R. Syski, Eds. Amsterdam, The Netherlands: Elsevier North-Holland, 1988, pp. 261–282.
[4] O. J. Boxma and B. Meister, "Waiting-time approximations for cyclic-service systems with switchover times," *Perform. Eval.*, vol. 7, pp. 299–308, 1987.
[5] E. G. Coffman, Jr., and M. Hofri, "On the expected performance of scanning disks," *SIAM J. Comput.*, vol. 11, pp. 60–70, 1982; in *Stochastic Analysis of Computer Storage*, by O. I. Aven, E. G. Coffman, Jr., and Y. A. Kogan. The Netherlands: Reidel, 1987.
[6] J. W. Cohen and O. J. Boxma, "The M/G/1 queue with alternating service formulated as a Riemann-Hilbert problem," *Perform. '81*, F. J. Kylstra, Ed. Amsterdam, The Netherlands: Elsevier North-Holland, 1981, pp. 181–199.
[7] J. N. Daigle, "Queue length distributions from probability generating functions via discrete Fourier transforms," *Oper. Res. Lett.*, vol. 8, pp. 229–236, 1989.
[8] M. Eisenberg, "Queues with periodic service and changeover times," *Oper. Res.*, vol. 20, pp. 440–451, 1972.
[9] ——, "Two queues with alternating service," *SIAM J. Appl. Math.*, vol. 36, pp. 287–303, 1979.
[10] D. Everett, "A note on the pseudoconservation laws for cyclic service systems with limited service disciplines," *IEEE Trans. Commun.*, vol. 37, pp. 781–783, 1989.
[11] M. J. Ferguson and Y. J. Aminetzah, "Exact results for nonsymmetric token ring systems," *IEEE Trans. Commun.*, vol. 33, pp. 223–231, 1985.
[12] S. W. Fuhrmann, "Symmetric queues served in cyclic order," *Oper. Res. Lett.*, vol. 4, pp. 139–144, 1985.
[13] S. W. Fuhrmann and Y. T. Wang, "Analysis of cyclic service systems with limited service: Bounds and approximations," *Perform. Eval.*, vol. 9, pp. 35–54, 1988.
[14] O. C. Ibe and X. Cheng, "Approximate analysis of asymmetric single-service token-passing systems," *IEEE Trans. Commun.*, vol. 37, pp. 572–577, 1989.
[15] P. Kuehn, "Multiqueue systems with nonexhaustive cyclic service," *Bell Syst. Tech. J.*, vol. 58, pp. 671–698, 1979.
[16] K. K. Leung, "Waiting time distributions for token-passing systems with limited-one service via discrete Fourier transforms," in *Proc. IEEE Infocom '90*, San Francisco, CA, 1990, pp. 1111–1118.

[17] ——, "Waiting time distributions for token-passing systems with limited-k service via discrete Fourier transforms," *Performance'90*, P. J. B. King, I. Mitrani, and R. J. Pooley, Eds. Amsterdam, The Netherlands: Elsevier North-Holland, 1990, pp. 333–347.
[18] K. K. Leung and M. Eisenberg, "A single-server queue with vacations and gated time-limited service," *IEEE Trans. Commun.*, vol. 38, pp. 1454–1462, 1990.
[19] H. Levy and M. Sidi, "Polling systems with correlated arrivals," in *Proc. IEEE Infocom'89*, Ottawa, Canada, 1989, pp. 907–913.
[20] M. Nomura and K. Tsukamoto, "Traffic analysis on polling systems," *Trans. Inst. Elect. Commun. Eng. Japan*, vol. J61-B, pp. 600–607, 1978.
[21] A. V. Oppenheim and R. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[22] D. Sarkar and W. I. Zangwill, "Expected waiting time for nonsymmetric cyclic queueing systems—Exact results and applications," *Manag. Sci.*, vol. 35, pp. 1463–1474, 1989.
[23] L. D. Servi, "Average delay approximation of M/G/1 cyclic service queues with Bernoulli schedules," *IEEE J. Select. Areas Commun.*, vol. 4, pp. 813–822, 1986.
[24] H. Takagi, *Analysis of Polling Systems*. Cambridge, MA: M.I.T. Press, 1986.
[25] ——, "Queueing analysis of polling systems," *ACM Comput. Survey*, vol. 20, pp. 5–28, 1988.
[26] Tedijanto, "Exact results for the cyclic-service queue with a Bernoulli schedule," *Perform. Eval.*, vol. 11, pp. 107–115, 1990.
[27] K. S. Watson, "Performance evaluation of cyclic service strategies: A survey," *Performance'84*, E. Gelenbe Ed. Amsterdam, The Netherlands: Elsevier North-Holland, 1984, pp. 521–533.



Kin K. Leung (S'78-M'85-S'85-M'86) received the B.S. degree with first class honors in electronics from the Chinese University of Hong Kong, Hong Kong, in 1980, and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles, in 1982 and 1985, respectively.

While at UCLA, he served as a Teaching Assistant from 1981 to 1983 and as a Post-Graduate Research Engineer from 1983 to 1985. Since 1986, he has been a Member of Technical Staff at AT&T Bell Laboratories, Holmdel, NJ. His research interests include stochastic modeling, computer networks, distributed processing and databases, and neural networks.