Computing Transient and Steady-State Distributions in Polling Models by Numerical Transform Inversion

Gagan L. Choudhury
AT&T Bell Laboratories, Room 1L-238, Holmdel, New Jersey 07733-3030
email: gagan@buckaroo.att.com
Ward Whitt
AT&T Bell Laboratories, Room 2C-178, Murray Hill, New Jersey 07974-0636
email: wow@research.att.com

ABSTRACT: We show that probability distributions of performance measures in a large class of polling models can be effectively computed by numerically inverting transforms (generating functions and Laplace transforms). We develop new efficient iterative algorithms for computing the transform values and then use the Fourier-series method to perform the inversion. We also use this approach to compute moments. The algorithms apply to the transient behavior of stationary or nonstationary models as well as to the steady-state behavior of stationary models. Our main focus is on computing exact tail probabilities, but even for mean waiting times, our algorithm is faster than previous algorithms for large models. The computational complexity of our algorithm is $O(N^{\alpha})$ for computing performance measures at one queue and $O(N^{1+\alpha})$ for computing performance measures at all queues, where N is the number of queues and α is typically between 0.6 and 0.8. We demonstrate effectiveness by computing performance measures in an asymmetric 1000-queue system.

1. Introduction

Over the last thirty years a substantial literature on polling models has evolved, as can be seen from [2], [5]–[12]. Polling models have generated great interest because they have many applications to the performance analysis of computer and telecommunications systems. These applications include data transfer from terminals on multi-drop lines to a central computer, token passing schemes in local and wide area networks, job scheduling in telephone switching systems and scheduling moving arms in secondary storage devices.

Multidimensional transforms of performance measures of interest have been derived for several basic polling models with Poisson arrivals, general service-time and general switchover-time distributions. These transform expressions have been successfully exploited to derive means and sometimes second moments, but we are unaware of them being used to calculate the distributions themselves, higher moments or asymptotic parameters. (However, approximate distributions have recently been calculated by Federgruen and Katalan [5].)

In this paper we show that these polling transforms often can be quite easily computed and inverted numerically to calculate distributions and any desired moments. We develop a new efficient recursive algorithm for computing transform values. Our operation count for computing moments and distributions is $O(N^{\alpha})$ for one queue and is $O(N^{1+\alpha})$ for all queues of an N-queue system, where α is typically in the range 0.6 to 0.8. It appears that our algorithm is faster than other available algorithms for mean waiting time and queue lengths. To demonstrate, in Section 5 we consider an asymmetric 1000-queue system and compute the mean waiting time in less than 5 seconds, and several moments and tail probability values in a few minutes, using a SUN SPARC-2 workstation.

We demonstrate the power of numerical inversion by specifically considering the gated service model with unlimited waiting space, but our approach applies to many other models. In particular, it extends to all polling models where the joint queue length process at polling instants of a fixed queue is a multitype branching process with immigration; see Resing [9] and an expanded version of this paper to appear in *Performance Evaluation.* To perform the numerical inversion, we use the Fourier-series method in [1,3,4].

Computing full distributions instead of only means is important because in many performance analysis applications we really want to know high percentiles, such as the 95th or 99th. In emerging high-speed communication networks there is even great interest in very small tail probabilities, such as 10^{-9}, in order to provide an appropriate quality of service. Hence, the ability to compute full distributions should significantly increase the usefulness of polling models. Moreover, simulation is usually quite effective for computing means, but simulation has difficulty computing small tail probabilities.

We compute all moments by numerical inversion, so that our method is the same for the hundredth moment as it is for the first moment. By contrast, previous results for higher moments, such as by Konheim, Levy and Srinivasan [7], have been via analytical differentiation of the transform, which leads to cumbersome expressions. So far, analytical differentiation has only provided results for the first two moments, but we can easily compute even the hundredth moment. Moreover, these higher moments are very useful for performing numerical asymptotic analysis of the tail probabilities [3].

Unlike for the first-in first-out (FIFO) discipline, for polling models computing transient performance measures tends to be easier than computing steady-state performance

0-7803-2486-2/95 $4.00 © 1995 IEEE

measures, because we compute steady-state transforms by computing transient transforms for a suitably large number of queue visits. Here we compute *both* transient and steady-state performance measures for polling models. The transient performance measures are computed only at polling instants, whereas the steady-state performance measures are computed at both polling instants and arbitrary times. In the transient case, the model need not be stationary; i.e., the arrival rate, service-time distribution and switchover-time distribution can change at polling instants. The nonstationary model allows us to study the effect of a sudden overload. We are unaware of any previous work on time-varying polling models.

To understand tail probabilities such as for a steady-state waiting time W, it is useful to exploit asymptotics such as

$$P(W > x) \sim \alpha x^{\beta} e^{-\eta x} \quad \text{as } x \to \infty , \tag{1.1}$$

where $f(x) \sim g(x)$ means that $f(x)/g(x) \to 1$ as $x \to \infty$. In the expanded version of this paper we study the asymptotics (1.1) for polling models by *numerically* deriving the parameters α, β and η, using the moment approach in [3]. We show that the asymptote (1.1) is often quite accurate and that it reveals important properties of the polling disciplines. Our experience indicates that, unlike FIFO, $\beta \neq 0$ for M/G/1 polling models. Indeed, the pure-exponential asymptotics with $\beta = 0$ typically holding with the FIFO discipline seems to be the exception rather than the rule for other disciplines. When $\beta \neq 0$, we also do two-term asymptotic expansions, which significantly improve the accuracy.

To indicate the kind of insights we can obtain, consider a symmetric system (gated or exhaustive) with N queues, mean switchover time r per queue and a fixed total server utilization. First, for $r = 0$, if we increase N, then the mean waiting time remains constant, but often the asymptotic parameter η in (1.1) decreases, and the tail probabilities increase. Next, for a fixed N, if we increase r, then the mean waiting time increases significantly, but the asymptotic decay rate η decreases (and the tail probabilities increase) only slightly. Finally, for $r > 0$ and $N > 1$, the mean waiting time is bigger for the gated discipline than for the exhaustive discipline. However, the reverse is true for higher-order moments and for small tail probabilities.

Here is how the rest of this paper is organized. In Section 2 we develop new efficient algorithms for computing transient and steady-state transforms for the gated service discipline. We also show how the computational complexity (in the case of mean waiting times) compares to that of previous algorithms. In Section 3 we show how to extend our approach to zero switchover times. There we derive a new iterative algorithm for computing the steady-state transform values. In Section 4 we indicate how we verify the accuracy of our computations. In Section 5 we present an example with 1000 queues. In Section 6 we discuss how the number of iterations to reach steady state depends on the number of queues, their traffic intensities and the target precision. Finally, in Section 7 we present an example in which we analyze time-dependent behavior in a nonstationary model.

2. The Gated Service Model

There are N queues indexed by i, $0 \leq i \leq N-1$. There is a single server that moves successively from queue i to queue $i+1 \bmod N$, i.e., from i to $i+1$ for $i \leq N-2$ and from $N-1$ to 0. We assume that a *gated policy* is in use, so that the server serves all customers at a queue that it finds upon arrival there, but no new customers that arrive after the server.

Customers arrive at queue i according to a Poisson process with the rate λ_i. Each customer at queue i requires a service time that has *Laplace-Stieltjes transform* (LST) $\hat{B}_i(s) \equiv \int_0^{\infty} e^{-st} dB_i(t)$. The server has a switchover (reply) time to go from queue i to queue $i+1 \bmod N$ with LST $\hat{R}_i(s)$. The customer service times, server switchover times and customer arrival processes are all mutually independent.

2.1 Transient Analysis

We assume that the system starts at time 0 with some initial distribution of customers at the various queues and with the server about to begin service at some queue. Let $L_{i,m} \equiv (L_{i,m,0}, \ldots, L_{i,m,N-1})$ represent the vector of queue lengths at the instant the server is about to begin service at queue i after having visited m queues. For analysis, it is significant that $\{L_{i,m} : m \geq 0\}$ is a discrete-time Markov chain. Let $p_{i,m} \equiv (p_{i,m}(k_0, \ldots, k_{N-1}))$ be the N-dimensional probability mass function (pmf) of $L_{i,m}$ and let $\hat{p}_{i,m}$ be its N-dimensional generating function, i.e.,

$$\begin{aligned} \hat{p}_{i,m}(\mathbf{z}) &= E \prod_{j=0}^{N-1} z_j^{L_{i,m,j}} \qquad (2.1) \\ &= \sum_{k_0=0}^{\infty} \cdots \sum_{k_{N-1}=0}^{\infty} \prod_{j=0}^{N-1} z_j^{k_j} \, p_{i,m}(k_0, \ldots, k_{N-1}) \end{aligned}$$

for an N-dimensional vector of complex variables $\mathbf{z} = (z_0, \ldots, z_{N-1})$. It is known that we can express $\hat{p}_{i,m}(\mathbf{z})$ in terms of $\hat{p}_{i-1,m-1}(\mathbf{z})$, e.g., see p. 105 of Takagi [12]. The following is the transient analog of the steady-state equations given in (5.231) of [12], which hold by the same argument. For the statement, let w be the wrap-around function $w(k) = k \bmod N$.

Proposition 2.1. *The generating function of the pmf* $p_{i,m}$ *is*

$$\hat{p}_{i,m}(\mathbf{z}) = \hat{R}_{w(i-1)}\Big(\sum_{j=0}^{N-1} (\lambda_j - \lambda_j z_j)\Big) \hat{p}_{w(i-1),m-1}(\tilde{z}) \tag{2.2}$$

where

$$\tilde{z} = \Big(z_0, \ldots, z_{i-2}, \hat{B}_{w(i-1)}\Big(\sum_{j=0}^{N-1} (\lambda_j - \lambda_j z_j)\Big), z_i, \ldots, z_{N-1}\Big) .$$

Using (2.2) recursively m times, we arrive at the following proposition. (It is justified by mathematical induction.)

Proposition 2.2. *The generating function in (2.2) can be expressed as*

$$\hat{p}_{i,m}(\mathbf{z}) = \prod_{k=1}^{m} \hat{R}_{w(i-k)} \left[\sum_{j=0}^{N-1} (\lambda_j - \lambda_j z_j^{(k-1)}) \right] \hat{p}_{w(i-m),0}(\mathbf{z}^{(m)}) \tag{2.3}$$

where

$$z_l^{(k)} = \begin{cases} z_l^{(k-1)} & \text{for } l \neq w(i-k) \\ \hat{B}_l(\sum_{j=0}^{N-1} (\lambda_j - \lambda_j z_j^{(k-1)}) & \text{for } l = w(i-k) \end{cases} \tag{2.4}$$

and $\mathbf{z}^0 = \mathbf{z}$.

The quantity $\hat{p}_{w(i-m),0}(\mathbf{z}^{(m)})$ in (2.3) is the transform of the initial distribution and $w(i-m)$ is the index of the first queue polled. If the system is empty at $t = 0$, then $\hat{p}_{w(i-m),0}(\mathbf{z}) = 1$ for all $\mathbf{z}$. If $\mathbf{k}_0 = (k_{0,0}, k_{0,1}, \ldots k_{0,N-1})$ represents a deterministic vector of initial queue lengths, then

$$\hat{p}_{w(i-m),0}(\mathbf{z}) = \prod_{j=0}^{N-1} z_j^{k_{0,j}} . \tag{2.5}$$

To compute each term of the m-fold product in (2.3), the operation count is $O(N)$ due to the presence of the two N-fold sums in (2.3) and (2.4). We show below that the operation count may be reduced to $O(1)$ by introducing the auxiliary variable

$$y^{(k)} = \sum_{j=0}^{N-1} (\lambda_j - \lambda_j z_j^{(k)}) . \tag{2.6}$$

We replace (2.3) and (2.4) by the following new set of recursions:

$$\hat{p}_{i,m}(\mathbf{z}) = \prod_{k=1}^{m} \hat{R}_{w(i-k)}(y^{(k-1)}) \hat{p}_{w(i-m),0}(\mathbf{z}^{(m)}) , \tag{2.7}$$

$$z_l^{(k)} = \begin{cases} z_l^{(k-1)} & \text{for } l \neq w(i-k) \\ \hat{B}_l(y^{(k-1)}) & \text{for } l = w(i-k) , \end{cases} \tag{2.8}$$

$$y^{(k)} = y^{(k-1)} + \lambda_{w(i-k)} (z_{w(i-k)}^{(k-1)} - z_{w(i-k)}^{(k)}) , \tag{2.9}$$

with initial conditions $\mathbf{z}^{(0)} = \mathbf{z}$ and

$$y^{(0)} = \sum_{j=0}^{N-1} (\lambda_j - \lambda_j z_j) .$$

Note that the computation of (2.9) and $\hat{R}_{w(i-k)}(y^{(k-1)})$ in (2.7) require $O(1)$ operations, and so does the computation of (2.8) since only one z variable changes. The total operation count in computing $\hat{p}_{i,m}(\mathbf{z})$ is thus $O(m)$ and the total storage requirement is $O(N)$ since we have to store the vector $\mathbf{z}^{(k)}$ and the scalar $y^{(k)}$ in each step. By contrast, the total operation count for the more straightforward recursions in (2.3) and (2.4) is $O(Nm)$ and storage requirement is $O(N)$.

Once we get $\hat{p}_{i,m}(\mathbf{z})$, we can compute distributions and moments by numerical inversion. In particular, by inverting $\hat{p}_{i,m}(1, 1, \ldots, z_j, \ldots 1)$ we get the marginal distribution at queue j. This is one-dimensional inversion and is pretty fast. So we can get marginal distributions at all queues even for pretty large N. (See Section 2.3 on the computational complexity). We can also compute joint distributions of, say, queues 0, 1 and 2 by inverting $\hat{p}_{i,m}(z_0, z_1, z_2, 1, 1, \ldots 1)$. However, for joint distributions the computational complexity grows exponentially with the number of dimensions [4].

For transient analysis we can even allow the system to be time-varying. In particular we can allow each of the service time LSTs $\hat{B}_l(\cdot)$, switchover time LSTs $\hat{R}_l(\cdot)$ and arrival rates λ_l to change as a function of m.

We can also get (and numerically invert) transforms related to $\hat{p}_{i,m}(\mathbf{z})$. For this purpose, let $\tilde{p}_{i,m}(z) = \hat{p}_{i,m}(\tilde{\mathbf{z}})$ where $\tilde{z}_k = 1$ for $k \neq i$ and $\tilde{z}_i = z$. Let $\hat{S}_{i,m}(s)$ be the LST of the time spent by the server at queue i after having visited m queues. By (5.11) of [12],

$$\hat{S}_{i,m}(s) = \tilde{p}_{i,m}(\hat{B}_i(s)) . \tag{2.10}$$

Let $\hat{C}_{i,m}(s)$ represent the LST of a cycle time that ends just before a visit to queue i and after m queue visits overall $(m > N)$. Based on (5.39a) of [12],

$$\hat{C}_{i,m}(s) = \tilde{p}_{i,m}(1 - s/\lambda_i) . \tag{2.11}$$

2.2 Steady-state Analysis

The transient transforms of Section 2.1 converge to proper steady-state values as $m \to \infty$ if the stability condition for this model is satisfied. The stability condition is well known to be $\sum_{j=0}^{N-1} \rho_j < 1$, where $\rho_j \equiv \lambda_j b_j$ is the offered load at queue j and b_j is the mean service time. The steady-state limit, if it exists, has to be independent of the transform of the initial distribution $\hat{p}_{w(i-m),0}(\mathbf{z}^{(m)})$. This is possible only if $\mathbf{z}^{(m)}$ approaches a vector of all 1's. We can apply Proposition 2.2 to obtain an explicit representation for the steady-state transform $\hat{p}_i(\mathbf{z}) \equiv \lim_{m\to\infty} \hat{p}_{i,m}(\mathbf{z})$.

Proposition 2.3. *If $L_{i,m}$ converges to a proper limit L_i as $m \to \infty$, then its probability generating function is*

$$\hat{p}_i(\mathbf{z}) = E \prod_{j=0}^{N-1} z_j^{L_{i,j}} = \prod_{k=1}^{\infty} \hat{R}_{w(i-k)}(y^{(k-1)}) , \tag{2.12}$$

$$z_l^{(k)} = \begin{cases} z_l^{(k-1)} & \text{for } l \neq w(i-k) \\ \hat{B}_l(y^{(k-1)}) & \text{for } l = w(i-k) , \end{cases} \tag{2.13}$$

$$y^{(k)} = y^{(k-1)} + \lambda_{w(i-k)} (z_{w(i-k)}^{(k-1)} - z_{w(i-k)}^{(k)}) .$$

with initial conditions $\mathbf{z}^{(0)} = \mathbf{z}$ *and* $y^{(0)} = \sum_{j=0}^{N-1} (\lambda_j - \lambda_j z_j)$.

Furthermore,

$$\lim_{k\to\infty} z^{(k)} = (1, 1, \ldots, 1) \tag{2.14}$$

$$\lim_{k\to\infty} y^{(k)} = 0 \tag{2.15}$$

$$\lim_{k\to\infty} \hat{R}_{w(i-k)}(y^{(k-1)}) = 1 . \tag{2.16}$$

To numerically compute $\hat{p}_i(\mathbf{z})$, we truncate the infinite product in (2.12) at $k = I$ using the stopping criterion

$$|\hat{R}_{w(i-I)}(y^{(I-1)}) - 1| < \varepsilon \tag{2.17}$$

for some suitably small ε. Other stopping criteria, based on (2.14) or (2.15) may also be used. Note that in general I will depend on the particular $\mathbf{z}$ value at which the transform is needed. As in the transient case, the computational complexity in computing each term in the product on the right hand side of (2.12) is $O(1)$ and the overall computational complexity in computing one transform value is $O(I)$.

In this paper we do all computations using double precision arithmetic and typically aim for about 8-to-10 place accuracy in the final answer. For this purpose, setting $\varepsilon = 10^{-12}$ or 10^{-13} in (2.17) is usually adequate.

Under the stability condition, the other transient transforms in (2.10) and (2.11) also approach their steady-state values as given below:

$$\hat{S}_i(s) = \tilde{p}_i(\hat{B}_i(s)) \tag{2.18}$$

and

$$\hat{C}_i(s) = \tilde{p}_i(1 - s/\lambda_i) , \tag{2.19}$$

where as before $\tilde{p}_i(z) = \hat{p}_i(\tilde{\mathbf{z}})$ with $\tilde{z}_k = 1$ for $k \neq i$ and $\tilde{z}_i = z$.

We also have some steady-state transforms which do not have transient counterparts. As on p. 109 of Takagi [12], the steady-state number of customers at queue i at an arbitrary time is

$$\hat{Q}_i(\mathbf{z}) = [(1 - \sum_{k=0}^{N-1} \rho_k)/\lambda_i \sum_{k=0}^{N-1} r_k] \times \frac{\hat{B}_i(\lambda_i - \lambda_i z)}{z - \hat{B}_i(\lambda_i - \lambda_i z)} [\tilde{p}_i(z) - \tilde{p}_i(\hat{B}_i(\lambda_i - \lambda_i z))] , \tag{2.20}$$

where r_i is the mean switchover time from queue i to queue $(i+1)$ mod N. As on p. 120 of [12], the LST of the waiting time of an arbitrary customer at queue i is

$$\hat{W}_i(s) = \frac{(1 - \sum_{k=0}^{N-1} \rho_k)}{\sum_{k=0}^{N-1} r_k} \frac{[\tilde{p}_i(\hat{B}_i(s)) - \tilde{p}_i(1 - s/\lambda_i)]}{s - \lambda_i + \lambda_i \hat{B}_i(s)} \tag{2.21}$$

2.3 Computational Complexity and Comparison With Other Algorithms

Our inversion algorithm for the computation of distributions and moments requires computation of the transform at a number of complex values of $\mathbf{z}$. In general, computation of the k^{th} moment requires computation of the transform at $(kl+1)$ values, while computing one point of the distribution requires computation of the transform at about $40l$ values, where l is a parameter for roundoff error control. For 8-to-10-place accuracy we need $l = 2$ for the moment computation and $l = 1$ for the distribution computation.

In Sections 2.1 and 2.2 we have shown that the computational complexity for computing one transform value is $O(m)$ in the transient case and $O(I)$ for the steady-state case, where m is the number of queue visits and I is the number of iterations for convergence of the transform value, determined by the stopping criterion (2.17). Hence, for all our moment and distribution computations the complexity is $O(m)$ in the transient case and $O(I)$ in the steady-state case. Henceforth we only discuss the computational complexity of the steady-state case since the transient computation is faster.

A key measure of the computational complexity is the dependence of I upon N, the number of queues. In Section 6 we show that $I = O(N^{\alpha})$ where α is in the range 0.6 to 0.8. So for all our computations also the complexity is $O(N^{\alpha})$. This is for computing performance measures at one queue. For computation at all queues the complexity is $O(N^{1+\alpha})$. For the computation of mean waiting times at gated or exhaustive systems the algorithm by Sarkar and Zangwill [10] has been widely reported to be the fastest. The computational complexity of that algorithm is $O(N^3)$ (same complexity for one queue and all queues) and hence is considerably slower than ours for large N.

Recently Konheim, Levy and Srinivasan [7] developed an algorithm for computing the mean and second moment of the waiting time with complexity $O(N)$ for one queue and $O(N^2)$ for all queues. Since $\alpha < 1$ we are somewhat faster than this algorithm as well. The algorithm in [7] may be extended to higher moments as well but would involve pretty cumbersome expressions. Of course, the main focus of our paper is on computing distributions, for which no previous work has been reported.

It is to be noted that both our method and the method in [7] are iterative, so that the convergence becomes slow as the total server utilization ρ approaches 1. Dependence of I on ρ is discussed in Section 6. By contrast, the computational complexity in [10] does not critically depend on ρ. Hence, for ρ sufficiently close to 1, the algorithm in [10] would be faster. However, our algorithm does work adequately even at $\rho = 0.95$.

3. Zero Switchover Times

As the switchover times approach zero, the average cycle time approaches zero and many transient and steady-state quantities defined as averages at polling instants (such as number in system, server visit time, etc.) also approach zero since there are infinitely many polling instants in finite time whenever the system is empty. However, the steady-state queue length and waiting time, whose transforms are given by (2.20) and (2.21), remain non-trivial and well defined. We show that both these quantities can readily be computed by our iterative-transform calculation and transform-inversion method. We explicitly do it only for the waiting time but a very similar treatment is possible for the queue length. There has been a considerable literature focusing on the case of zero switchover times and the relationship between zero and nonzero switchover times, e.g., [2], [9], [11], but there has not been any previous attempt at transform inversion.

We assume that the mean switchover time $r_k = r$ and that $r \to 0$. In this section we consider strictly cyclic polling and so the steady-state waiting times and queue lengths in the limiting system as r_k approaches zero are well defined. However, for non-strictly-cyclic polling (such as polling tables) the limiting system is not well defined unless something more is specified. In [2] the extra specification involves the fixed quantities $p_k \equiv \lim_{R \to 0} r_k / R$, where R is the total switchover time and r_k is the switchover time between pseudostations k and $k+1$. (Pseudostation k is the k^{th} station polled in the polling table. The same station may appear more than once in the polling table, each time as a different pseudostation). Note that the quantities p_k add up to one and essentially p_k represents the probability that at the instant of a new message arrival an an empty system the server is in transit between pseudostations k and $k+1$.

For the cyclic polling system considered here, note that (2.21) may be written as

$$\hat{W}_i(s) = \frac{(1 - \sum_{k=0}^{N-1} \rho_k)}{s - \lambda_i + \lambda_i \hat{B}_i(s)} \; \frac{\tilde{p}_i(\hat{B}_i(s)) - \tilde{p}_i(1 - s/\lambda_i)}{Nr} . \quad (3.1)$$

As $r \to 0$, $\hat{R}_i(s) \to 1$ for all i and s. Therefore, from (2.12), $\hat{p}_i(\mathbf{z}) \to 1$ for all i and $\mathbf{z}$. So both the numerator and denominator of (3.1) approach zero. Applying L'Hospital's rule to (3.1), we get

$$\hat{W}_i(s) = [(1 - \sum_{k=0}^{N-1} \rho_k)/(s - \lambda_i + \lambda_i \hat{B}_i(s))]$$

$$\times \frac{\frac{\partial \tilde{p}_i(B_i(s))}{\partial r}\Big|_{r=0} - \frac{\partial \tilde{p}_i(1 - s/\lambda_i)}{\partial r}\Big|_{r=0}}{N} . \quad (3.2)$$

Taking logarithms on both sides of (2.12) yields

$$\log \hat{p}_i(\mathbf{z}) = \sum_{k=1}^{\infty} \log \hat{R}_{w(i-k)}(y^{(k-1)}) .$$

Taking derivatives on both sides with respect to r we get,

$$\frac{\frac{\partial \hat{p}_i(\mathbf{z})}{\partial r}}{\hat{p}_i(\mathbf{z})} = \sum_{k=1}^{\infty} \frac{\frac{\partial \hat{R}_{w(i-k)}(y^{(k-1)})}{\partial r}}{\hat{R}_{w(i-k)}(y^{(k-1)})} . \quad (3.3)$$

As $r \to 0$, $\hat{p}_i(\mathbf{z}) \to 1$ and $\hat{R}_{w(i-k)}(y^{(k-1)}) \to 1$. Hence, we get

$$\frac{\partial \hat{p}_i(\mathbf{z})}{\partial r}\Big|_{r=0} = \sum_{k=1}^{\infty} \frac{\partial \hat{R}_{w(i-k)}(y^{(k-1)})}{\partial r}\Big|_{r=0} . \quad (3.4)$$

Since we are going to let $r \to 0$, we specify how the switchover times depend on r. Let the switchover-times be i.i.d. random variables, each distributed as r times a fixed random variable with mean 1, i.e., be $X_r \equiv rX_1$ where X_1 is a nonnegative random variable with mean 1 and finite higher-order moments. Since

$$\frac{\partial Ee^{-srX_1}}{\partial r}\Big|_{r=0} = E(-sX_1 e^{-srX_1})\Big|_{r=0} = -s , \quad (3.5)$$

$$\frac{\partial \hat{R}_{w(i-k)}(s)}{\partial r}\Big|_{r=0} = -s . \quad (3.6)$$

Combining (3.4) and (3.6), we get

$$\frac{\partial \hat{p}_i(\mathbf{z})}{\partial r}\Big|_{r=0} = -\sum_{k=1}^{\infty} y^{(k-1)} . \quad (3.7)$$

Now we can use the same iterations given by (2.12)–(2.16) but replace (2.12) with (3.7) above to compute $\frac{\partial \hat{p}_i(\mathbf{z})}{\partial r}\Big|_{r=0}$, as needed in (3.2). Then the stopping criterion (2.17) has to be replaced by the following stopping criterion derived from (2.15): Stop at $k = I$ whenever

$$|y^{(k)}| < \varepsilon \quad (3.8)$$

for suitably small ε. Noting that $\frac{\partial \tilde{p}_i(z)}{\partial r} = \frac{\partial \hat{p}_i}{\partial r}(1, 1, \ldots, 1, z, 1, \ldots, 1)$ with z appearing in the i^{th} place, we can thus readily compute (3.2) which gives the steady-state waiting time transform.

We believe that the iteration we have just derived for obtaining steady-state transform values with zero switchover times is new. Note that the computational complexity in the zero switchover time case is the same as in the non-zero switchover time case and is faster than alternative algorithms (for mean waiting time and queue length computation).

4. Accuracy Verification

Since we are unaware of any published numerical results on second or higher moments for the models in this paper, an important consideration is the verification of the accuracy of our computations.

We implemented the algorithms in Ferguson and Aminetzah [6] and Sarkar and Zangwill [10] to verify our mean computations. However, these alternatives do not apply to the example in Section 5 with 1000 queues, because these algorithms are too slow. Even for the mean computation we use transform inversion. Hence, the fact that the transform inversion is accurate for means leads us to believe that it is accurate in other cases as well.

For steady-state performance measures the fact that the transforms converge with an error specification of about $10^{-12} - 10^{-13}$ shows that the transforms are being computed accurately. Our experience with the transform inversion algorithms in other contexts tells us that if the transform is being computed accurately, it is most likely inverted accurately as well.

In the special case of $N = 1$ and zero-switchover time, the gated discipline should match exactly an M/G/1 queue with FIFO service for which alternate Pollaczek-Khintchine transform expressions are available. We did observe that our results satisfy this consistency check.

As explained in [4], there is a parameter l for controlling round-off error. Typically $l=1$ is sufficient for the distribution computation and $l=2$ is sufficient for moment computation. Different values of l correspond to different contours on the complex plane. Hence, if two computations are done using different values of l and they agree up to, say 8 decimal places, then it is very likely that they are both accurate up to that many places. We verify all computations in this paper using $l=1$ and 2 for distributions and $l=2$ and 3 for moments. In all cases the agreement is good.

For these reasons, we are confident that the accuracy of all computations to be shown are high (8 or more decimal places), even though we do not show all the numerics up to that many places. All computations were done on a SUN SPARC-2 workstation using double-precision arithmetic.

5. An Example with 1000 Queues

We first consider an asymmetric 1000-queue system with gated service discipline. All queues are assumed to have the same service time distribution, a two-stage Erlang (E_2) distribution with mean 1 and coefficient of variation 0.5. All switchover times are assumed to be zero. The arrival rates at queues $0, 1, \ldots, 999$ are

$$\lambda_i = \begin{cases} 2\times10^{-4} + 2i\times10^{-6} & \text{for } 0 \le i \le 500 \\ 2\times10^{-4} + 2(1000-i)\times10^{-6} & \text{for } 501 \le i \le 999 \ . \end{cases} \tag{5.1}$$

Formula (5.1) makes the smallest arrival rate 2×10^{-4} at queue 0, the largest arrival rate 1.2×10^{-3} at queue 500, and the overall server utilization $\rho = 0.7$.

We show below the first three moments and three points of the steady-state waiting-time distribution seen by an arbitrary customer at queue 0.

Performance Measure	1st Moment	2nd Moment	3rd Moment
Value	3.497866	45.64751	1008.372
Performance Measure	$Pr(W > 10)$	$Pr(W > 20)$	$Pr(W > 30)$
Value	0.1023567	0.02536360	0.00706867

Table 1. Numerical results for the steady-state waiting-time distribution in the 1000-queue example.

The mean was computed in less than 5 seconds, all 3 moments were computed in about 30 seconds, and the distribution values were computed in about 5 minutes. By contrast, we estimated that the algorithm in [10] would have needed 3.5 hours to compute just the mean and the algorithm in [6] would have taken substantially longer, again to compute just the mean.

6. Rate of Convergence to Steady State

All our steady-state computations have a complexity $O(I)$ where I is the number of iterations needed to satisfy (2.17) in computing one transform value. In this example we study numerically how I depends on the number of queues, N, server utilization, ρ, and error criterion, ε. The number I also depends on several other parameters, such as the service-time and switchover-time distributions, whether the computation is for mean or for higher moments, and so on. However, those factors have been observed to be less critical than N, ρ and ε.

We consider the average number of iterations needed per transform value in computing the mean steady-state waiting time in a symmetric system with gated service discipline, zero switchover time and a gamma service-time distribution with mean 1 and squared coefficient of variation 2. We investigate how I grows with N for fixed ρ and ε; here $\rho = 0.7$ and $\varepsilon = 10^{-12}$. We found that $I(20) \approx 500$, while $I(100) \approx 2000$. A crude approximation would be $I(N) \approx 25N$, but the growth is actually slower than linear. It appears that $I = O(N^{\alpha})$ where $\alpha \approx 0.77$. We experimented with several other cases and in the range of 1 to 100 queues we have found that consistently $I = O(N^{\alpha})$ for α between 0.6 and 0.8.

Next we fix N at 10, ε at 10^{-12} and show in Table 2 how I grows with ρ.

ρ	0.2	0.4	0.6	0.7	0.8	0.9	0.95
I	73	118	196	270	413	824	1604

Table 2. Number of iterations required as a function of the traffic intensity ρ.

Clearly I approaches ∞ as ρ approaches 1. The growth rate is slower than $-1/\log\rho$ and faster than $1/(1-\rho)$. This appears to be the case in other examples as well.

Finally, we fix N at 10, ρ at 0.8 and show in the table below how I grows with ε. I appears to grow roughly linearly with $-\log\varepsilon$.

ε	10^{-8}	10^{-10}	10^{-12}	10^{-14}
I	172	297	413	530

Table 3. Number of iterations as a function of the specified error ε in the stopping criterion.

7. An Example with Time-Varying Behavior

In this example we consider transient and time-varying behavior of a polling system. This example is motivated by the fact that in many polling systems some overload control action is taken whenever the cycle time (or the weighted sum of several successive cycle times) exceeds a threshold. The threshold is chosen large enough so that it is unlikely to be exceeded under normal traffic conditions. We do not study the effect of any overload-control mechanism, but rather study how the transient cycle times behave under a sudden surge of overload and specifically how the probability of exceeding a large threshold changes during and after the overload. We consider a symmetric 10-queue system with a gated service discipline. The switchover time is a 4-stage Erlang (E_4) distribution with mean 0.1 and squared coefficient of variation 0.25. The service time is a 2-stage Erlang (E_2) distribution with mean 1 and squared coefficient of variation 0.5. The arrival rate is time-varying and it changes from cycle to cycle so that the instantaneous offered load ρ (the product of total arrival rate and mean service time) changes with cycle number. In particular, the instantaneous offered load ρ during cycle k is 1.2 for $5 \leq k \leq 10$ and is 0.8 otherwise.

We let the threshold be 40. When the offered load is stationary with $\rho = 0.8$, the threshold is 8 times the mean cycle time, and the probability of exceeding it would be 8.862472×10^{-4} (this value we obtain from the steady-state analysis). Let C_n represent the length of the n^{th} cycle. In Figure 1 we show how the time-dependent cycle-time tail probabilities $P(C_n > 40)$ evolve with n under the time-varying load. Both during and after the overload, the probability of exceeding the threshold remains much higher than the steady-state value. This demonstrates that the observed cycle time should be a good indicator of an overload.

Acknowledgement

We would like to thank Uri Yechiali for a discussion about closed polling systems which led us to undertake this work, and Kin Leung and Martin Eisenberg for further stimulating discussions.

References

[1] J. Abate and W. Whitt, "The Fourier-series method for inverting transforms of probability distributions," *Queueing Systems* 10 (1992) 5-88.

[2] G. L. Choudhury, "Polling with a general service order table: gated service," *Proc. IEEE INFOCOM '90*, San Francisco, 1990, 268-276.

[3] G. L. Choudhury and D. M. Lucantoni, "Numerical computation of the moments of a probability distribution from its transform," *Operations Research* to appear.

[4] G. L. Choudhury, D. M. Lucantoni and W. Whitt, "Multidimensional transform inversion with applications to the transient M/G/1 queue," *Ann. Appl. Prob.* 4 (1994) 719-740.

[5] A. Federgruen and Z. Katalan, "Approximating queue size and waiting time distributions in general polling models," *Queueing Systems* 18 (1994) 353-386.

[6] M. J. Ferguson and Y. J. Aminetzah, "Exact results for nonsymmetric token ring systems," *IEEE Trans. on Commun.*, Vol. COM-33 (1985) 223-231.

[7] A. G. Konheim, H. Levy and M. M. Srinivasan, "Descendant set: an efficient approach for the analysis of polling systems," *IEEE Trans. Commun.* 42 (1994) 1245-1253.

[8] H. Levy and L. Kleinrock, "Polling systems with zero switch-over periods: a general method for analyzing expected delay," *Perf. Eval.*, 13 (1991) 97-107.

[9] J. A. C. Resing, "Polling systems and multitype branching processes," *Queueing Systems* 13 (1993) 409-426.

[10] D. Sarkar and W. I. Zangwill, "Expected waiting time for nonsymmetric cyclic queueing systems — exact results and applications," *Management Science* 35 (1989) 1463-1474.

[11] M. M. Srinivasan, S-C. Niu and R. B. Cooper, "Relating polling models with zero and nonzero switchover times," *Queueing Systems*, to appear.

[12] H. Takagi, *Analysis of Polling Systems*, MIT Press, 1986.

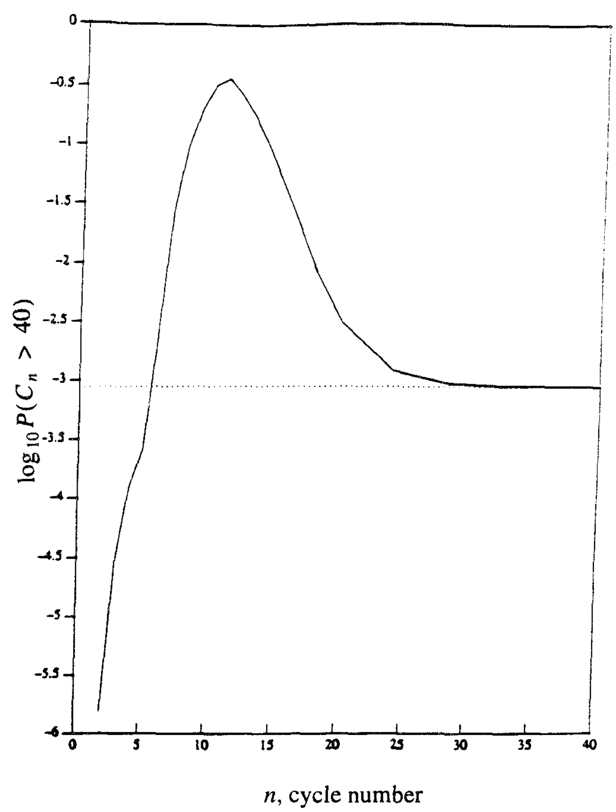

Figure 1. The time-dependent cycle-time tail probability (in log scale) for the example with time-varying load.