Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

POLLING SYSTEMS WITH SERVER TIMEOUTS

E. de Souza e Silva
H. Richard Gail
R. Muntz

August 1993
CSD-930026

Polling Systems
with Server Timeouts

Edmundo de Souza e Silva[1]
Federal University of Rio de Janeiro, NCE
Cx.P. 2324, CEP 20001, Rio de Janeiro, Brazil

H. Richard Gail
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598

Richard R. Muntz[1]
University of California, Los Angeles
Los Angeles, CA 90024

Abstract

Polling systems have long been the subject of study and are of particular interest in the analysis of high speed communications networks. There are many options for the scheduling policies that can be used at each polling station (gated, exhaustive, customer limited, etc.). In addition, one can impose an upper bound on the total service time delivered to customers at a station per server visit. In the most common case the upper bound is a constant for each polling station, and the resulting system model is not Markovian even when service times and interarrival times are exponential. In this paper, a comprehensive solution is developed for the major scheduling policies with time limits for each polling station. The basic approach is based on studying the embedded Markov chain defined at the sequence of instants when the server arrives at each polling station. The computation of transition probabilities requires transient analysis of the Markov process describing the system evolution between epochs of the embedded chain. Uniformization methods are used to develop efficient algorithms for the transition probabilities and for system performance measures. Example problems are solved using the techniques developed to illustrate the utility of the results.

[1] The work of E. de Souza e Silva and R. R. Muntz was partially supported by grants from National Science Foundation CCR-9215064 and from CNPq(Brazil).

1 Introduction

A polling system consists of multiple queues that are visited by a single server in a fixed order. A variety of multiaccess schemes in computer communications can be modeled using polling systems, such as token ring and token bus local area networks, and some multiplexing techniques (e.g. [1, 24]).

There are various service disciplines for polling systems that have been considered in the literature. Among them, we mention the exhaustive, gated and customer-limited policies. For the exhaustive discipline, the server serves all customers in the visited queue until it empties (including arrivals that occur after the server arrives) before moving to another queue. For the gated discipline, the server takes only those customers that are present when it arrives at the queue. The arrivals that join the queue during service are not served until the next server visit. For the customer-limited discipline, a limit is imposed on the maximum number of customers that can be served during a server visit (both gated and exhaustive customer-limited disciplines have been studied). There is an extensive body of literature on the analysis of polling systems. In these analyses, the above disciplines (and others) have been considered for exponential or general service time distributions and switchover times (the time that the server takes to move from one queue to another) with deterministic, exponential or general distributions. Special cases such as single-buffer systems and symmetric systems (for which all queues have identical characteristics) have also been studied. In particular, a broad class of such systems was considered in [13]. Important performance measures of these systems include joint queue length distributions at server arrival points and server departure points of a particular queue, and joint queue length distributions at customer arrival points and customer departure points [11]. Other measures of interest include throughput, marginal queue length distributions, mean queue lengths, mean waiting times and mean cycle times [27]. A comprehensive survey of the state of the art in the analysis of polling systems is presented in [24, 25].

Not all polling disciplines are amenable to an exact analytic solution. For instance, as pointed out in [14, 25], the customer-limited service disciplines are very difficult to analyze, and only approximate results are available except for special cases (e.g. [3, 5]). Time-limited systems are a variation on the customer-limited policies, in which customers are served from a queue until a limit on the time spent at that queue is exhausted (server timeout). Note that if service times are constant and server timeouts are fixed multiples of the constant service time which are independent of cycle times, this discipline reduces to the customer-limited case. Since the server immediately switches to another queue as soon as the queue being served becomes empty, bandwidth is dynamically allocated to highly loaded queues if there are lightly loaded queues in the system. Establishing a server timeout upon each visit to a queue in the polling cycle is important in order to guarantee a minimum bandwidth to different types of traffic to the system.

Recently, polling systems with server timeouts (time-limited polling systems) have received significant attention from researchers, not only because of intrinsic mathematical interest, but also because such systems provide a natural model for new network protocols and multiplexor schemes (timed-token disciplines). For example, in [2] (page 3) it is stated that "timer-limited policies have received less attention than their practical interest would justify," while in [26] (page 205) it is stated that "there are no results available for a polling model with time-limited service."

A server timeout scheme, called the (T_1-T_2) scheme, has been recently proposed in [23] for allocating bandwidth using multiplexing in wideband packet technology. Voice and data packets are queued separately, and time-limits T_1, T_2 are used to guarantee a minimum bandwidth for voice and data, proportional to $T_1/(T_1+T_2)$ and $T_2/(T_1+T_2)$, respectively. The (T_1-T_2) scheme will be considered in a later section and used to numerically illustrate the results of this paper.

Server timeouts are also used in the IEEE 802.4 standard proposed for the token bus (e.g. see [28]) and the access algorithm for the fiber distributed data interface (FDDI) [22]. Basically, there are two types of stations: high priority stations and ordinary stations. High priority stations are assigned a fixed time limit called the token holding time, which controls the maximum amount of time a station can transmit. If this time expires, the token is passed on to the next station as soon as the transmission of the current packet being sent finishes. The time allocated to an ordinary station is not fixed and depends on the elapsed time from the last visit of the server to the current visit, i.e, the last cycle time as seen by the station. If this cycle time is less than a threshold, called the token target rotation time (TTRT), then the station can transmit until it empties its buffer or until the cycle time plus the current transmission time reaches the threshold. Otherwise, it passes the token to the next station without transmitting any packet. For this last case, any latency is accumulated from cycle to cycle.

There have been several recent papers that are concerned with the analysis of time-limited systems. As mentioned above, Sriram [23] proposed the (T_1-T_2) multiplexing scheme and used simulation to study its performance. Coffman, Fayolle and Mitrani [4] analyzed the (T_1-T_2) multiplexing scheme under the assumption that the times T_1, T_2 are exponentially distributed random variables and there are no switchover times. They show that obtaining the generating functions for the joint steady state distributions of the number of customers in each queue can be reduced to solving a boundary value problem, for which a numerical solution is presented. The papers [20, 21] of Leung and Eisenberg were also motivated by studying the (T_1-T_2) scheme. They analyzed a single M/G/1 queue with server vacations, where the server limits the amount of time spent serving customers between vacation periods. In [20] the gated discipline is considered, and it is assumed that the vacation length is a random variable which is independent of the amount of work in queue when a vacation starts. A functional equation for the probability density function of the amount of work at polling instants is derived. The equation is solved by approximating the complementary distribution function by a weighted sum of Laguerre functions and transforming the functional equation into a set of linear equations. The average waiting time is obtained from the average amount of work at polling instants and at times when a vacation starts. Extensions to the nongated discipline appear in [21]. Note that a more accurate model for the (T_1-T_2) scheme would be obtained if vacations were allowed to depend on the work at the queue, since clearly they are dependent on the time the server spends at each queue in the (T_1-T_2) scheme.

In this paper we propose a methodology to analyze time-limited systems, which enables the incorporation of many important modeling details. In our analysis, we consider queues which can be served up to a fixed amount of time independent of other system parameters, similar to those of the (T_1-T_2) scheme or to the high priority stations of the FDDI protocol. Both preemptive and nonpreemptive timeouts can be taken into account. The exhaustive and gated disciplines are considered in detail, and we also indicate how additional disciplines can be studied using the same method. Among the measures obtained we mention joint queue length distributions at arrival and departure points of the server to a specific queue, joint queue length distributions at customer arrivals and departures, the limiting probability for the length of a particular queue in the system, the mean cycle time, and mean waiting times.

The remainder of the paper is organized as follows. In Section 2 we describe the model and present necessary background material. Sections 3 and 4 describe the technique used to determine joint queue length distributions at server arrival points and server departure points. Section 5 is concerned with the calculation of time average measures. In Section 6 we extend the basic model to analyze other cases. The computational complexity of the solution method is described in Section 7. In Section 8 we present numerical examples to illustrate the approach. Section 9 concludes the paper.

2 Model Description and Background Material

A polling system with M queues (stations) is considered. Customers (messages, jobs) arrive to queue i according to a Poisson process with rate λ_i, and service times of type-i customers are exponentially distributed with mean $1/\mu_i$. All arrival and service processes are independent. The buffer size at each queue (waiting customers plus the customer in service, if any) is either unlimited or equal to a finite value B_i. The server moves in a cyclic fashion from queue i to queue $i+1$ (mod M), and the time required by the server to switch from queue i to the next queue is assumed to be constant and denoted σ_i. Extensions which include a general switching policy (general polling table) are considered later in the paper.

An important feature of the polling systems that are studied in this paper is the existence of a timer at each queue which limits the amount of service a queue can continuously receive during any visit of the server. This length of time, called a server timeout, is assumed constant and is denoted by T_i for queue i. Note that when $T_i \to \infty$, we obtain the usual polling systems considered in previous studies. In the systems considered here, although the server is limited as to the amount of time it can spend during any one visit to a queue, the server immediately leaves if that queue becomes empty. This differs from STDM (synchronous time division multiplexing) type models [26], for which the server is required to stay a fixed time at each queue whether or not there is work to be done (such an assumption simplifies the analysis considerably).

The existence of server timeouts raises the issue of how to deal with the customer in service when the timeout expires. One possibility is to return the customer to the line of waiting jobs and resume serving it during the next visit of the server. Recall that the service time distributions for all types of customers are assumed to be exponentially distributed. Furthermore, we assume that the service time of a preempted customer is resampled, and so the service requirement of the customer preempted when the server leaves the queue is statistically identical to that of a customer that has received no service. Another alternative is to allow the customer to complete service, i.e. extend the allowed time just enough to finish the customer (for example, the overruns implemented in FDDI). Both the first case of preemptive timeouts and the second case of nonpreemptive timeouts are studied in this paper. We also consider cases for which a new customer is not taken into service if the timeout T_i has almost expired. That is, a constant ω_i associated with queue i is given such that if a service completion occurs within ω_i of the end of the timeout, the server does not remain at queue i for the full timeout period, but instead immediately leaves queue i. In this more general situation ($\omega_i = 0$ reduces to the previous case) the customer in service is returned to the waiting line when the timeout expires in the preemptive timeout case or is allowed to finish service in the case of nonpreemptive timeouts. It will be shown that all of these situations can be handled using the results presented in this paper.

The service discipline may differ from queue to queue. The main disciplines considered are the exhaustive and gated policies. However, we indicate how other disciplines, such as exhaustive customer-limited (E-limited) and gated customer-limited (G-limited), may also be analyzed using the same basic approach. At each queue, customers may be served in any order as long as the scheduling discipline is independent of job service times.

Our interest is in calculating various steady state performance measures for polling systems with server timeouts. Important measures are joint queue length distributions at server arrivals to a particular queue, at server departures from a particular queue, at customer arrivals, and at customer departures. Other measures include marginal queue length distributions, mean waiting times for customers, and loss probabilities in the finite buffer case. Before proceeding with the analysis in subsequent sections, the remainder of this section is devoted to developing the basic notation that is needed and to a brief review of the relevant background material.

Consider the vector process $\mathcal{X} = \{X(t) : t \geq 0\}$, where $X(t) = \langle x_1(t), \ldots, x_M(t) \rangle$ and $x_j(t)$ is the number in system at queue j at time t. The state space of $\mathcal{X}$ is the set of M-tuples of integers $\mathcal{S} = \{\langle q_1, \ldots, q_M \rangle : q_j = 0, \ldots, B_j\}$. Although $X(t)$ is regenerative with (for example) regeneration points given by time instants when the server visits a particular queue, say queue i, to find that all M queues are empty, in general $X(t)$ is not a Markov process for a variety of reasons (e.g. we have constant timeout intervals). However, consider the successive visits of the server to a particular queue, say queue i, and let $\eta_1^{(i)}, \eta_2^{(i)}, \ldots$ be the times when the server arrives at this queue. Although these times are not regeneration points for $\mathcal{X}$, the values of $\mathcal{X}$ at these points yield an embedded discrete time vector Markov chain $\mathcal{Y}^{(i)} = \{Y_k^{(i)} : k = 1, 2, \ldots\}$ given by $Y_k^{(i)} = X(\eta_k^{(i)})$. Similarly, the points $\xi_1^{(i)}, \xi_2^{(i)}, \ldots$ when the server departs from queue i, i.e. begins to switch from queue i to queue $i+1$ (mod M), yield an embedded Markov chain $\mathcal{Z}^{(i)} = \{Z_k^{(i)} : k = 1, 2, \ldots\}$ given by $Z_k^{(i)} = X(\xi_k^{(i)})$. We let $\mathbf{H}^{(i)}$ be the transition matrix for the chain $\mathcal{Y}^{(i)}$ and $\mathbf{G}^{(i)}$ be the transition matrix for the chain $\mathcal{Z}^{(i)}$. The steady state probability vector $\beta^{(i)}$ for $\mathcal{Y}^{(i)}$ satisfies $\beta^{(i)} = \beta^{(i)}\mathbf{H}^{(i)}$, while the corresponding vector $\alpha^{(i)}$ for $\mathcal{Z}^{(i)}$ satisfies $\alpha^{(i)} = \alpha^{(i)}\mathbf{G}^{(i)}$. We will first find equilibrium joint queue length distributions $\beta^{(i)}$ (at server arrival points) and $\alpha^{(i)}$ (at server departure points). We then define various reward functions to obtain time average measures of interest using results from Markov chains with rewards.

To calculate the $\alpha^{(i)}$, $\beta^{(i)}$ we will use uniformization to find the transition matrix at the embedded points. Such an approach was used in [8] to analyze scheduled maintenance policies of repairable computer systems. As time evolves, the server in the polling system alternates between switchover intervals and service to queues. Although the process $X(t)$ is not Markovian, if we consider an interval that starts with the arrival of the server to a queue, say queue j, until the server departs (either due to a timeout or to the queue becoming empty) the process is Markovian for that interval of time. This is clear, since the service times are exponential and the arrival processes are Poisson. The non-Markovian nature of the timeout interval requires us to use transient analysis to determine the state probabilities at the end of the interval. The technique we use for the transient analysis is uniformization, which is briefly reviewed below. The details of the calculations will be described in subsequent sections.

The underlying solution method that is used in the analysis of all the various types of polling systems with server timeouts is based on uniformization or randomization [10, 17, 18]. This technique involves discretizing a continuous-time Markov chain (CTMC) in the following way. Let $\mathcal{U} = \{U(t) : t \geq 0\}$ be a CTMC with generator $\mathbf{Q}$ and state space $\mathcal{S}$. For $s \in \mathcal{S}$, let r_s be the cumulative rate of leaving state s, and assume these rates are uniformly bounded. That is, assume there is a finite rate Λ satisfying $\Lambda \geq r_s$ for all s (this clearly holds if the state space is finite, but it also holds for certain chains with countably many states). For each state s, add a fictitious self-transition back to the state with rate $\Lambda - r_s$. This creates a process equivalent to $\mathcal{U}$ for which the rates out of the states are identical and equal to Λ. We may view $U(t) = V_{N(t)}$, where $\mathcal{V} = \{V_n : n = 0, 1, \ldots\}$ is a discrete-time Markov chain and $\mathcal{N} = \{N(t) : t \geq 0\}$ is a Poisson process which is independent of $\mathcal{V}$ and is of rate Λ. The transition matrix of $\mathcal{V}$ is $\mathbf{P} = \mathbf{Q}/\Lambda + \mathbf{I}$.

3 Distributions at Server Arrivals and Departures

As discussed in the previous section, the first step in our analysis involves studying the behavior of embedded Markov chains defined at times when the server arrives to a particular queue and those defined at times when the server departs from a particular queue. In this section we will derive results for the basic polling system with server timeouts, without including many of the extensions that are possible. Specifically, we will consider a system where the server cycles from queue to queue, i.e. in the order $1 \Rightarrow 2 \Rightarrow \cdots \Rightarrow M-1 \Rightarrow M \Rightarrow 1$. The timeout period for queue i is constant and denoted T_i, while the switchover time from queue i to queue $i+1$ (mod M) is constant and denoted σ_i. Furthermore, we assume one of the two basic service disciplines at each queue (exhaustive or gated). We first assume infinite buffer space at each queue, i.e. $B_j = \infty$, and preemptive timeouts (no overruns), but later in the section we also consider systems with finite buffers and the case of nonpreemptive timeouts. Systems with additional and/or alternative assumptions (e.g. general polling tables, other scheduling disciplines) will be discussed in Section 6. In this section we concentrate on deriving the basic results as clearly as possible without considering computational efficiency, in order not to complicate the derivations. In Section 4 we examine how to organize the computations in an efficient manner.

Consider the sequence of time instants $\eta_1^{(i)}, \eta_2^{(i)}, \ldots$ when the server arrives at queue i (the server begins serving customers at queue i at these time instants) for the basic polling system with server timeouts. Recall that for any specific value of i, the regenerative process $X(t) = \langle x_1(t), \ldots, x_M(t)\rangle$, for which $x_j(t)$ is the number in system at queue j at time t, yields an embedded Markov chain $\mathcal{Y}^{(i)}$ when considered at the time instants $\eta_k^{(i)}$. Similarly, the sequence of time instants $\xi_1^{(i)}, \xi_2^{(i)}, \ldots$ when the server leaves queue i (either the queue empties or the timeout expires) yields an embedded Markov chain $\mathcal{Z}^{(i)}$. Our first task is to determine the steady state distributions $\beta^{(i)}$ of $\mathcal{Y}^{(i)}$ and $\alpha^{(i)}$ of $\mathcal{Z}^{(i)}$ over the state space $\mathcal{S} = \{\langle q_1, \ldots, q_M\rangle : q_j = 0, 1, \ldots\}$. These distributions will then be used to find time average measures of interest later in the paper. To obtain $\beta_s^{(i)}$ and $\alpha_s^{(i)}$, $s \in \mathcal{S}$, the entries of the transition matrices $\mathbf{H}^{(i)}$ of $\mathcal{Y}^{(i)}$ and $\mathbf{G}^{(i)}$ of $\mathcal{Z}^{(i)}$ are first determined. The matrix $\mathbf{H}^{(i)}$ is found by examining the behavior of the system over a polling cycle beginning when the server visits queue i and ending when the server returns the next time to queue i. Similarly, the matrix $\mathbf{G}^{(i)}$ records the system behavior over a polling cycle defined by successive departures from queue i. The cyclic switching policy guarantees that each queue is visited once per cycle, with the server alternating between periods of switching and service periods. To determine $\mathbf{H}^{(i)}$ and $\mathbf{G}^{(i)}$, it is convenient to view a polling cycle as consisting of M "mini-cycles" (during each of which the server continuously visits a particular queue) and M switchover intervals (during each of which the server moves from one queue to the next). We refer to a mini-cycle during which the jth queue is served as a j-mini-cycle and a switchover interval when the server switches from queue j to queue $j+1$ (mod M) as a j-switchover interval.

Let $s = \langle q_1, \ldots, q_M \rangle \in \mathcal{S}$, where q_j represents the number of customers in queue j, $j = 1, \ldots, M$, and similarly let $s' = \langle q'_1, \ldots, q'_M \rangle \in \mathcal{S}$. Define $\mathbf{D}^{(j)}$ to be the transition matrix for the j-mini-cycle. That is, its (s, s') element $d^{(j)}_{s,s'}$ is the probability that the queue lengths are s' at the end of a j-mini-cycle given that at the start of the j-mini-cycle the queue lengths were s. Similarly define $\mathbf{C}^{(j)}$ to be the transition matrix for the vector of queue lengths for a j-switchover interval. Then it is clear that

$$\begin{aligned} \mathbf{H}^{(i)} &= \mathbf{D}^{(i)}\mathbf{C}^{(i)} \ldots \mathbf{C}^{(M)}\mathbf{D}^{(1)} \ldots \mathbf{D}^{(i-1)}\mathbf{C}^{(i-1)} \\ \mathbf{G}^{(i)} &= \mathbf{C}^{(i)}\mathbf{D}^{(i+1)} \ldots \mathbf{C}^{(M)}\mathbf{D}^{(1)} \ldots \mathbf{C}^{(i-1)}\mathbf{D}^{(i)}. \end{aligned} \tag{1}$$

Using (1) it is easy to see that the vectors $\beta^{(i)} = \beta^{(i)}\mathbf{H}^{(i)}$ and $\alpha^{(i)} = \alpha^{(i)}\mathbf{G}^{(i)}$ satisfy the equations

$$\begin{array}{ll} \beta^{(1)} = \alpha^{(M)}\mathbf{C}^{(M)} & \\ \beta^{(i)} = \alpha^{(i-1)}\mathbf{C}^{(i-1)} & i = 2, \ldots, M \\ \alpha^{(i)} = \beta^{(i)}\mathbf{D}^{(i)} & i = 1, \ldots, M. \end{array} \tag{2}$$

Once the matrices $\mathbf{D}^{(j)}$, $\mathbf{C}^{(j)}$, $j = 1, \ldots, M$, and one of the $\beta^{(i)}$ or $\alpha^{(i)}$ have been computed, all the joint equilibrium probability vectors at server arrival points and server departure points can be calculated. However, we caution the reader that a direct application of equation (1) to determine one of the $\mathbf{H}^{(i)}$ or $\mathbf{G}^{(i)}$ and thus one of the $\beta^{(i)}$ or $\alpha^{(i)}$ is expensive and not recommended, since it does not take advantage of the special structure possessed by the matrices $\mathbf{D}^{(j)}$, $\mathbf{C}^{(j)}$. In Section 4 we address the issue of calculating $\beta^{(i)}$ and $\alpha^{(i)}$ in an efficient manner without explicitly computing $\mathbf{H}^{(i)}$ and $\mathbf{G}^{(i)}$. In the remainder of this section we concentrate on finding expressions for the entries of $\mathbf{D}^{(j)}$, $\mathbf{C}^{(j)}$ under various scheduling disciplines and buffer size assumptions.

The transition matrices $\mathbf{C}^{(j)}$, $j = 1, \ldots, M$, are simple to compute. The transition probability $c^{(j)}_{s,s'}$ from $s = \langle q_1, \ldots, q_M \rangle$ to $s' = \langle q'_1, \ldots, q'_M \rangle$ over the j-switchover interval of constant length σ_j is

$$c^{(j)}_{s,s'} = \begin{cases} 0 & \text{if } q'_i < q_i \text{ for any } i,\ i = 1, \ldots, M \\ \prod_{i=1}^{M} e^{-\lambda_i \sigma_j} \frac{(\lambda_i \sigma_j)^{k_i}}{k_i!} & \text{otherwise} \end{cases} \tag{3}$$

where $k_i = q'_i - q_i$.

The major problem is to determine the transition matrices $\mathbf{D}^{(j)}$. Recall that $d^{(j)}_{s,s'}$, the (s, s') element of $\mathbf{D}^{(j)}$, is the probability that the queue lengths are $s' = \langle q'_1, \ldots, q'_M \rangle$ at the end of the mini-cycle given that the vector of queue lengths is $s = \langle q_1, \ldots, q_M \rangle$ at the start of the mini-cycle. Although the determination of these transition probabilities requires a slightly different procedure for each of the two main types of scheduling disciplines (exhaustive and gated), the same general approach is followed. In each case the calculation of the transition probabilities is based on the transient analysis of a Markov chain that describes the behavior of the jth queue during a j-mini-cycle in which the jth queue is served. Transient analysis is required, since the j-mini-cycle can end if the jth queue empties or if the timeout interval is exhausted. We will use the uniformization technique that was briefly described in the previous section for the transient analysis. We first discuss in detail the computation of the transition matrix $\mathbf{D}^{(j)}$ for the case of exhaustive service, and then we describe the modifications required for the gated discipline. Both the infinite buffer case and the finite buffer case are considered. We close the section by discussing the additional steps needed to handle nonpreemptive timeouts.

3.1 Exhaustive Service and Infinite Buffers

Consider a j-mini-cycle that starts with the queue lengths equal to $s = \langle q_1, \ldots, q_M \rangle$. The state of the jth queue at the end of the mini-cycle will either be $q'_j = 0$, if the queue empties prior to the completion of the timeout period, or $q'_j > 0$ if the timeout occurs before the queue empties. Therefore, we have the following mutually exclusive and exhaustive outcomes for the j-mini-cycle, where τ_j is the length of the j-mini-cycle:

(a) $q'_j = 0$ and $0 < \tau_j < T_j$;

(b) $q'_j > 0$ and $\tau_j = T_j$.

Note that the first set of outcomes is a continuum over the range $(0, T_j)$ for τ_j and that the second set of outcomes is discrete. We also note that the transition probabilities for the other queues are determined by the length of the j-mini-cycle, since they are simply incremented by the number of arrivals during the mini-cycle.

Given that queue j is served under the exhaustive policy, the continuous-time Markov chain $\mathcal{W}^{(j)} = \{W^{(j)}(t) : t \geq 0\}$ that gives the distribution for the number at queue j during a j-mini-cycle is simply a one-dimensional birth-death process with parameters λ_j and μ_j, but for which state 0 is an absorbing state (the server leaves queue j if it becomes empty). The state transition rate diagram for $\mathcal{W}^{(j)}$ is illustrated in Figure 1. The state probabilities for queue j at the end of the j-mini-cycle can be found by calculating the transient behavior of the chain $\mathcal{W}^{(j)}$ over the constant interval $(0, T_j)$, since no state changes occur once the chain reaches the absorbing state 0 (the server leaves before the timeout

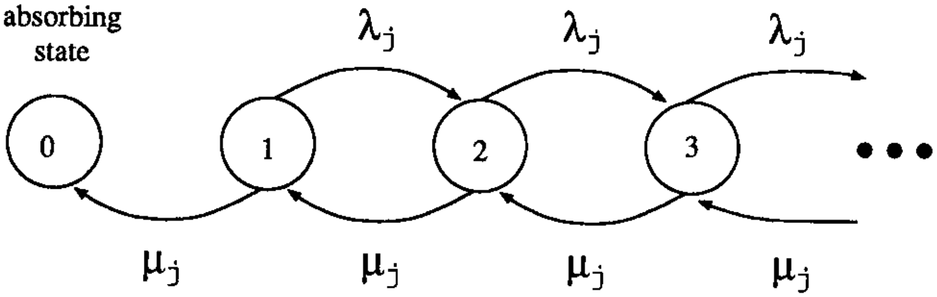

Figure 1: Exhaustive service and infinite buffers.

expires). These calculations can be easily done using uniformization. Specifically, given an initial probability vector ν for number in system at queue j at the beginning of the mini-cycle, the state distribution at the end of the mini-cycle for exhaustive service is

$$\mathbf{p}^{(j)}(\nu) = \sum_{n=0}^{\infty} e^{-\Lambda_j T_j} \frac{(\Lambda_j T_j)^n}{n!} \boldsymbol{\pi}(n,\nu), \tag{4}$$

where $\Lambda_j = \lambda_j + \mu_j$, $\boldsymbol{\pi}(0,\nu) = \nu$, and $\boldsymbol{\pi}(n,\nu) = \boldsymbol{\pi}(n-1,\nu)\mathbf{W}^{(j)}$, with $\mathbf{W}^{(j)}$ the transition matrix of the uniformized chain corresponding to $\mathcal{W}^{(j)}$. Note that $\mathrm{p}_0^{(j)}(\nu)$, the 0th element of $\mathbf{p}^{(j)}(\nu)$, is the probability that the queue emptied prior to T_j and thus that the j-mini-cycle ended prior to T_j (given the initial distribution ν). Since 0 is an absorbing state in $\mathcal{W}^{(j)}$, this entry includes the probability ν_0 that the server arrived to an empty queue j. Choosing the initial probability distribution $\nu = \mathbf{e}_{q_j}$, where $\mathbf{e}_k$ is the vector with kth entry equal to 1 and all other entries equal to 0, corresponds to assuming that the length of queue j is q_j at the start of the mini-cycle. In this case, we will use the simplified notation, $\boldsymbol{\pi}(n, q_j)$, for convenience.

The distribution for τ_j, the length of this j-mini-cycle, given a distribution ν for number initially at queue j can be calculated as follows. Let

$$F(t,\nu) \stackrel{\text{def}}{=} P[\tau_j \le t|\nu], \qquad 0 \le t < T_j.$$

Equivalently, $F(t,\nu)$ is the probability of being in the absorbing state 0 at time t. Thus

$$F(0,\nu) = P[\tau_j = 0|\nu] = \nu_0 = \pi_0(0,\nu)$$

represents the probability that the server visits an empty queue j, and hence the beginning and ending state are identical, i.e. $s = s'$. For $0 < t < T_j$, we have

$$F(t,\nu) = \sum_{n=0}^{\infty} e^{-\Lambda_j t} \frac{(\Lambda_j t)^n}{n!} \pi_0(n,\nu). \tag{5}$$

In this case $\pi_0(n,\nu)$ is the probability of being in the absorbing state 0 at step n of the uniformized chain corresponding to $\mathcal{W}^{(j)}$. The density $F'(t,\nu)$ for $0 < t < T_j$ is given by

$$F'(t,\nu) = \sum_{n=1}^{\infty} e^{-\Lambda_j t} \frac{(\Lambda_j t)^{n-1}}{(n-1)!} \Lambda_j \pi_0(n,\nu) + \sum_{n=0}^{\infty} e^{-\Lambda_j t} \frac{(\Lambda_j t)^n}{n!} (-\Lambda_j) \pi_0(n,\nu),$$

or

$$F'(t,\nu) = \sum_{n=1}^{\infty} e^{-\Lambda_j t} \frac{(\Lambda_j t)^{n-1}}{(n-1)!} \Lambda_j \left\{ \pi_0(n,\nu) - \pi_0(n-1,\nu) \right\}. \tag{6}$$

Note that for $n > 0$ the quantity

$$\phi_0(n,\nu) \stackrel{\text{def}}{=} \pi_0(n,\nu) - \pi_0(n-1,\nu) \tag{7}$$

is the probability of being absorbed (entering state 0) at exactly the nth step of the uniformized Markov chain. For $0 < t < T_j$ it is clear that $F'(t,\nu)$ is the density function for the outcomes $(q'_j = 0, \tau_j)$ with $0 < \tau_j < T_j$. It is also clear that the probability of the outcome $(q'_j > 0, T_j)$ is equal to the probability that the Markov chain $\mathcal{W}^{(j)}$ is in state q'_j at time T_j, and this probability is $\mathbf{p}^{(j)}_{q'_j}(\nu)$, the q'_jth element of $\mathbf{p}^{(j)}(\nu)$.

We now calculate the entries of $\mathbf{D}^{(j)}$. Note that (similar to the switchover interval case) $d^{(j)}_{s,s'} = 0$ when $q'_i < q_i$ for some $i \neq j$. Otherwise there must be $k_i = q'_i - q_i$ arrivals at queue i during the j-mini-cycle. First suppose that the j-mini-cycle ends with $q'_j > 0$ (and therefore $\tau_j = T_j$). For queue j the probability that the j-mini-cycle lasts until T_j and the final queue length is q'_j is the probability that the uniformized Markov chain corresponding to $\mathcal{W}^{(j)}$ is in state q'_j at time T_j when the initial state is q_j. Therefore,

$$\begin{aligned} d^{(j)}_{s,s'} &= P[W^{(j)}(T_j) = q'_j \mid W^{(j)}(0) = q_j] \prod_{i \neq j} e^{-\lambda_i T_j} \frac{(\lambda_i T_j)^{k_i}}{k_i!} \\ &= \mathbf{p}^{(j)}_{q'_j}(q_j) \prod_{i \neq j} e^{-\lambda_i T_j} \frac{(\lambda_i T_j)^{k_i}}{k_i!}. \end{aligned}$$

Substituting for $\mathbf{p}^{(j)}_{q'_j}(q_j)$ from equation (4) yields

$$d^{(j)}_{s,s'} = \left\{ \sum_{n=0}^{\infty} e^{-\Lambda_j T_j} \frac{(\Lambda_j T_j)^n}{n!} \pi_{q'_j}(n, q_j) \right\} \prod_{i \neq j} e^{-\lambda_i T_j} \frac{(\lambda_i T_j)^{k_i}}{k_i!}, \tag{8}$$

where $\boldsymbol{\pi}(0, q_j) = \mathbf{e}_{q_j}$. Setting $\gamma_j = \sum_{i \neq j} \lambda_i + \Lambda_j = \sum_{i=1}^{M} \lambda_i + \mu_j$ and $\kappa_j = \sum_{i \neq j} k_i$, this can be easily rewritten as

$$\begin{aligned} d^{(j)}_{s,s'} &= \sum_{n=\kappa_j}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \prod_{i \neq j} \left(\frac{\lambda_i}{\gamma_j} \right)^{k_i} \left(\frac{\lambda_j + \mu_j}{\gamma_j} \right)^{n-\kappa_j} \\ &\quad \times \frac{n!}{(n-\kappa_j)! \prod_{i \neq j} k_i!} \pi_{q'_j}(n - \kappa_j, q_j). \end{aligned} \tag{9}$$

This equation may also be interpreted in the following manner. Consider the superposition of M Poisson processes of rates λ_i, $i \neq j$, and $\Lambda_j = \lambda_j + \mu_j$, respectively. For an interval of length T_j, the nth term of the series on the right-hand side of (9) gives the probability that n events in the composite process occurred, k_i transitions of process i, $i \neq j$, occurred during the interval, and the uniformized Markov chain associated with the jth process was in state q_j' after $n - \kappa_j$ transitions.

To find $d^{(j)}_{s,s'}$ when $q_j' = 0$, we use the previously derived density $F'(t, q_j)$. The idea is to condition on the length of the j-mini-cycle, $\tau_j = t$, and proceed as above. We have

$$d^{(j)}_{s,s'} = \int_0^{T_j} \prod_{i \neq j} e^{-\lambda_i t} \frac{(\lambda_i t)^{k_i}}{k_i!} F'(t, q_j)\, dt.$$

Substituting for $F'(t, q_j)$ from (6) and using the definition (7) yields

$$d^{(j)}_{s,s'} = \int_0^{T_j} \prod_{i \neq j} e^{-\lambda_i t} \frac{(\lambda_i t)^{k_i}}{k_i!} \left\{ \sum_{n=1}^{\infty} e^{-\Lambda_j t} \Lambda_j \frac{(\Lambda_j t)^{n-1}}{(n-1)!} \phi_0(n, q_j) \right\} dt,$$

where $\boldsymbol{\pi}(0, q_j) = \mathbf{e}_{q_j}$. Setting $m = n + \kappa_j = n + \sum_{i \neq j} k_i$ and interchanging the order of integration and summation, we obtain

$$\begin{aligned} d^{(j)}_{s,s'} &= \sum_{m=\kappa_j+1}^{\infty} \prod_{i \neq j} \left(\frac{\lambda_i}{\gamma_j} \right)^{k_i} \left(\frac{\lambda_j + \mu_j}{\gamma_j} \right)^{m-\kappa_j} \frac{(m-1)!}{(m-1-\kappa_j)! \prod_{i \neq j} k_i!} \phi_0(m - \kappa_j, q_j) \\ &\quad \times \int_0^{T_j} \gamma_j e^{-\gamma_j t} \frac{(\gamma_j t)^{m-1}}{(m-1)!}\, dt. \end{aligned}$$

Recognizing the m-stage Erlangian distribution, we have

$$\begin{aligned} d^{(j)}_{s,s'} &= \sum_{m=\kappa_j+1}^{\infty} \prod_{i \neq j} \left(\frac{\lambda_i}{\gamma_j} \right)^{k_i} \left(\frac{\lambda_j + \mu_j}{\gamma_j} \right)^{m-\kappa_j} \frac{(m-1)!}{(m-1-\kappa_j)! \prod_{i \neq j} k_i!} \phi_0(m - \kappa_j, q_j) \\ &\quad \times \sum_{n=m}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \end{aligned}$$

or, upon reversing the order of summation,

$$\begin{aligned} d^{(j)}_{s,s'} &= \sum_{n=\kappa_j+1}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \sum_{m=\kappa_j+1}^{n} \prod_{i \neq j} \left(\frac{\lambda_i}{\gamma_j} \right)^{k_i} \left(\frac{\lambda_j + \mu_j}{\gamma_j} \right)^{m-\kappa_j} \\ &\quad \times \frac{(m-1)!}{(m-1-\kappa_j)! \prod_{i \neq j} k_i!} \phi_0(m - \kappa_j, q_j). \end{aligned} \qquad (10)$$

The above expression represents the probability that k_i arrivals occur at queue i, $i \neq j$, and the server leaves queue j before the timeout T_j expires. Similar to the previous case for which $\tau_j = T_j$, this result may be interpreted in terms of the superposition of M Poisson processes representing arrivals to queue i, $i \neq j$, and transitions of the uniformized chain for queue j corresponding to $\mathcal{W}^{(j)}$. The index n represents the number of transitions of this aggregate process during an interval of length T_j, while m represents the transition at which absorption occurred. For $i \neq j$, k_i transitions of type i occurred (arrivals to queue i), $m - \kappa_j$ transitions of type j occurred, and the final transition was of type j and corresponded to queue j becoming empty. In the event that $m < n$, additional transitions of the composite process would occur during an interval T_j, but these are simply ignored since the j-mini-cycle ends if queue j empties.

3.2 Exhaustive Service and Finite Buffers

When the buffer sizes at the M queues are finite, the above arguments must be modified to find the elements of the transition matrix $\mathbf{D}^{(j)}$. The procedure that was followed in the infinite buffer case involved first determining the distribution of τ_j, the length of a j-mini-cycle, and then using it to find the entries $d^{(j)}_{s,s'}$. If queue j has a finite buffer size of B_j, the birth-death chain of Figure 1 must be truncated at state B_j as in Figure 2 below. Equations (5) and (6) again give the distribution of τ_j, but the vectors $\boldsymbol{\pi}(n)$ are now obtained from the transition matrix $\mathbf{W}^{(j)}$ of the uniformized chain corresponding to the truncated birth-death process.

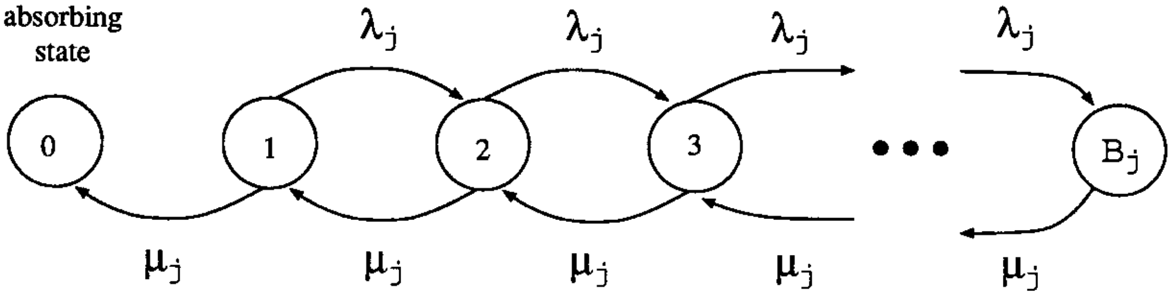

Figure 2: Exhaustive service and finite buffers.

Once the distribution of τ_j has been determined, it remains to find analogues of equations (9) and (10) for $d^{(j)}_{s,s'}$, where $s' = \langle q'_1, \ldots, q'_M \rangle$ and $s = \langle q_1, \ldots, q_M \rangle$. As before, we need only consider the case $q'_i - q_i \geq 0$, $i \neq j$, since otherwise $d^{(j)}_{s,s'} = 0$. Recall that for infinite buffers the transition probabilities $d^{(j)}_{s,s'}$ are a function of the length of queue i ($i \neq j$) only through the difference $q'_i - q_i$. However, this is not true in the finite buffer case, since this difference does not necessarily represent the number of arrivals to queue i during the mini-cycle. For example, if a transition from s to s' has its ith entry at the buffer limit, i.e. $q'_i = B_i$, then this transition represents the infinite set of cases for which there are *at least* $B_i - q_i$ arrivals at queue i.

If $q_i' < B_i$ for all $i \neq j$, then the equations for the infinite buffer case may be used, but if $q_i' = B_i$ for some $i \neq j$, then we need to sum infinitely many of those equations to account for all relevant arrival patterns. We will organize the terms of the sum in a way that leads to a recursion in Section 4 for calculating the entries $d^{(j)}_{s,s'}$. In particular, for states s and s', let $k_i^* = q_i' - q_i$ be the difference in the state of queue i at the beginning and the end of the mini-cycle, and let $\kappa_j^* = \sum_{i\neq j} k_i^*$. Also let k_i represent the number of arrivals to queue i, $i \neq j$, and let $a = \sum_{i \neq j} k_i$ represent the total number of arrivals to the nonserved queues. Define the set of vectors

$$\mathbf{K}^{(j)}_{s,s'}(a) = \{\langle k_1, \dots, k_{j-1}, k_{j+1}, \dots, k_M \rangle : \sum_{i\neq j} k_i = a, k_i = k_i^* \text{ if } q_i' < B_i, k_i \geq k_i^* \text{ if } q_i' = B_i\},$$

which corresponds to all possible arrivals that give a state change from s to s'. Let n represent the number of transitions of the superposition of the uniformized chain for queue j and the arrival processes to queues $i \neq j$. Then from (9), for $q_j' > 0$ $(\tau_j = T_j)$ we have

$$\begin{aligned} d^{(j)}_{s,s'} &= \sum_{n=\kappa_j^*}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \sum_{a=\kappa_j^*}^{n} \sum_{\mathbf{k}\in\mathbf{K}^{(j)}_{s,s'}(a)} \prod_{i\neq j} \left(\frac{\lambda_i}{\gamma_j}\right)^{k_i} \left(\frac{\lambda_j+\mu_j}{\gamma_j}\right)^{n-a} \\ &\quad \times \frac{n!}{(n-a)!\prod_{i\neq j} k_i!} \pi_{q_j'}(n-a, q_j). \end{aligned} \tag{11}$$

This equation is obtained by conditioning on the number of events n during an interval of length T_j and grouping the conditional probabilities according to the total number of arrivals a to the nonserved queues. Similarly, from (10), for $q_j' = 0$ $(\tau_j < T_j)$ we have

$$\begin{aligned} d^{(j)}_{s,s'} &= \sum_{n=\kappa_j^*+1}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \sum_{a=\kappa_j^*}^{n-1} \sum_{\mathbf{k}\in\mathbf{K}^{(j)}_{s,s'}(a)} \sum_{m=a+1}^{n} \prod_{i\neq j} \left(\frac{\lambda_i}{\gamma_j}\right)^{k_i} \left(\frac{\lambda_j+\mu_j}{\gamma_j}\right)^{m-a} \\ &\quad \times \frac{(m-1)!}{(m-1-a)!\prod_{i\neq j} k_i!} \phi_0(m-a, q_j). \end{aligned} \tag{12}$$

By interchanging the order of summation, we may also write these equations in the form (for $q_j' > 0$)

$$\begin{aligned} d^{(j)}_{s,s'} &= \sum_{a=\kappa_j^*}^{\infty} \sum_{n=a}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \sum_{\mathbf{k}\in\mathbf{K}^{(j)}_{s,s'}(a)} \prod_{i\neq j} \left(\frac{\lambda_i}{\gamma_j}\right)^{k_i} \left(\frac{\lambda_j+\mu_j}{\gamma_j}\right)^{n-a} \\ &\quad \times \frac{n!}{(n-a)!\prod_{i\neq j} k_i!} \pi_{q_j'}(n-a, q_j), \end{aligned} \tag{13}$$

and (for $q'_j = 0$)

$$\begin{aligned} d^{(j)}_{s,s'} &= \sum_{a=\kappa^*_j}^{\infty} \sum_{n=a+1}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \sum_{\mathbf{k} \in \mathbf{K}^{(j)}_{s,s'}(a)} \sum_{m=a+1}^{n} \prod_{i \neq j} \left(\frac{\lambda_i}{\gamma_j}\right)^{k_i} \left(\frac{\lambda_j + \mu_j}{\gamma_j}\right)^{m-a} \\ &\times \frac{(m-1)!}{(m-1-a)! \prod_{i \neq j} k_i!} \phi_0(m-a, q_j). \end{aligned} \tag{14}$$

The recursions developed in Section 4 are based on equations (13) and (14).

3.3 Gated Service and Infinite Buffers

When service at queue j is given in a gated fashion, one must modify the Markov chain $\mathcal{W}^{(j)}$ that is used in the uniformization procedure to obtain the entries of $\mathbf{D}^{(j)}$. One possible approach is to consider a pure death process, with parameter μ_j and with 0 as an absorbing state, that keeps track of departures from queue j during the j-mini-cycle. Although such a chain yields the distribution of the mini-cycle length, arrivals must be added to obtain the $\mathbf{D}^{(j)}$ matrices and thus the queue length distributions $\beta^{(i)}$ and $\alpha^{(i)}$, i.e. one must perform a convolution. Furthermore, using a pure death process is not sufficient to calculate time average measures, since the length of time in a mini-cycle during which the system contains a particular number of customers is needed in these calculations. Thus instead, we use a two-dimensional chain for $\mathcal{W}^{(j)}$ with states of the form (u, v), where $0 \leq v \leq u$. Here u represents the actual number of customers in the served queue j, and v is the number of original customers in that queue which are still in the system. That is, v represents the number of customers that were present when the server arrived which have not yet departed, and u is the sum of v and the number of arrivals that have occurred. The state transition rate diagram is illustrated in Figure 3. States of the form $(u, 0)$ are absorbing in $\mathcal{W}^{(j)}$, because they represent cases when the server leaves queue j before the timeout expires. Since the number of original customers and the total number of customers in the system are the same at the beginning of the mini-cycle, an initial state of the chain has the form $u = v \geq 1$.

As in the exhaustive service case, we can apply uniformization (with $\Lambda_j = \lambda_j + \mu_j$) to construct the corresponding discrete time Markov chain and then use it to find the distribution of the mini-cycle length τ_j and the entries $d^{(j)}_{s,s'}$. Analogues of equations (5) and (6) hold in the gated case, except that the $\boldsymbol{\pi}$ vectors are obtained from the two-dimensional chain of Figure 3 and the absorbing state 0 in the exhaustive case is replaced by states of the form $(u, 0)$, $u \geq 0$. That is,

$$F(t, \nu) = \sum_{n=0}^{\infty} e^{-\Lambda_j t} \frac{(\Lambda_j t)^n}{n!} \sum_{u=0}^{\infty} \pi_{(u,0)}(n, \nu), \tag{15}$$

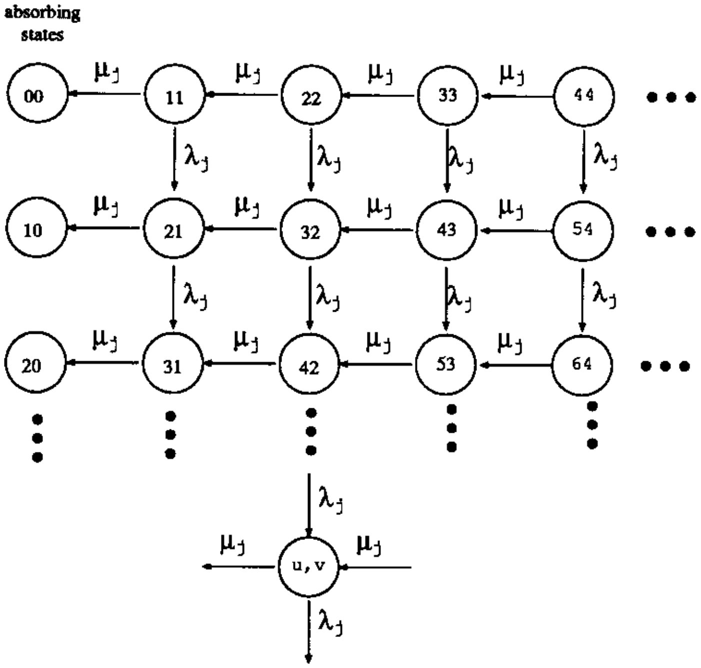

Figure 3: Gated service and infinite buffers.

and

$$F'(t,\nu) = \sum_{n=1}^{\infty} e^{-\Lambda_j t}\frac{(\Lambda_j t)^{n-1}}{(n-1)!}\Lambda_j \sum_{u=0}^{\infty} \phi_{(u,0)}(n,\nu) \tag{16}$$

where

$$\phi_{(u,0)}(n,\nu) \stackrel{\text{def}}{=} \pi_{(u,0)}(n,\nu) - \pi_{(u,0)}(n-1,\nu).$$

Also note that a vector $\mathbf{p}^{(j)}$ giving the distribution of states (u,v) at the end of the mini-cycle in the gated case can be obtained from equation (4), but using the two-dimensional chain of Figure 3. To obtain the distribution of total number of customers present when the server leaves, states with the same value of u must be aggregated.

To determine $d^{(j)}_{s,s'}$, equations (9) and (10) of the exhaustive case can be used with certain modifications. As noted above, states of the two-dimensional chain $\mathcal{W}^{(j)}$ must be aggregated to obtain queue length information. Furthermore, there are possible outcomes for the gated case in addition to those listed for the exhaustive case. For example, there may be customers present at queue j at the end of a j-mini-cycle even when the server leaves before the timeout period expires, i.e. the case $q'_j > 0$ and $0 < \tau_j < T_j$ may occur. The equations corresponding to (9) and (10) are given below for gated service.

For $q'_j > 0$, both $0 < \tau_j < T_j$ (the timeout does not expire) and $\tau_j = T_j$ (the timeout expires) are possible. Let $d^{(j)}_{s,s',0}$ and $d^{(j)}_{s,s',1}$ be the transition probabilities corresponding to these two cases, respectively. In the first case ($0 < \tau_j < T_j$) the state of the chain $\mathcal{W}^{(j)}$ at time T_j is the absorbing state $(q'_j, 0)$, while in the second case ($\tau_j = T_j$) $\mathcal{W}^{(j)}$ at time T_j is in one of the states (q'_j, v), $v \geq 1$. Therefore, we have

$$\begin{aligned} d^{(j)}_{s,s',0} &= \sum_{n=\kappa_j+1}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \sum_{m=\kappa_j+1}^{n} \prod_{i \neq j} \left(\frac{\lambda_i}{\gamma_j}\right)^{k_i} \left(\frac{\lambda_j + \mu_j}{\gamma_j}\right)^{m-\kappa_j} \\ &\quad \times \frac{(m-1)!}{(m-1-\kappa_j)! \prod_{i \neq j} k_i!} \phi_{(q'_j,0)}(m - \kappa_j, (q_j, q_j)) \end{aligned} \qquad (17)$$

and

$$\begin{aligned} d^{(j)}_{s,s',1} &= \sum_{n=\kappa_j}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \prod_{i \neq j} \left(\frac{\lambda_i}{\gamma_j}\right)^{k_i} \left(\frac{\lambda_j + \mu_j}{\gamma_j}\right)^{n-\kappa_j} \\ &\quad \times \frac{n!}{(n-\kappa_j)! \prod_{i \neq j} k_i!} \sum_{v=1}^{q'_j} \pi_{(q'_j,v)}(n - \kappa_j, (q_j, q_j)). \end{aligned} \qquad (18)$$

Here $\gamma_j = \sum_{i=1}^{M} \lambda_i + \mu_j$ and $\kappa_j = \sum_{i \neq j} k_i$, as in the exhaustive case. Finally, we have

$$d^{(j)}_{s,s'} = d^{(j)}_{s,s',0} + d^{(j)}_{s,s',1}. \qquad (19)$$

For $q'_j = 0$, we must have $0 < \tau_j < T_j$, and the state of the chain $\mathcal{W}^{(j)}$ at T_j must be the absorbing state $(0, 0)$, i.e. all original customers were served to completion during the mini-cycle and no arrivals occurred. Thus

$$\begin{aligned} d^{(j)}_{s,s'} &= \sum_{n=\kappa_j+1}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \sum_{m=\kappa_j+1}^{n} \prod_{i \neq j} \left(\frac{\lambda_i}{\gamma_j}\right)^{k_i} \left(\frac{\lambda_j + \mu_j}{\gamma_j}\right)^{m-\kappa_j} \\ &\quad \times \frac{(m-1)!}{(m-1-\kappa_j)! \prod_{i \neq j} k_i!} \phi_{(0,0)}(m - \kappa_j, (q_j, q_j)), \end{aligned} \qquad (20)$$

which is identical to (17) except that the final state of interest is $(0, 0)$ instead of $(q'_j, 0)$ where $q'_j > 0$. We emphasize that the transition matrix $\mathbf{W}^{(j)}$ used to calculate ϕ and π in the above equations differs from that for the corresponding equations in the exhaustive case. Namely, $\mathbf{W}^{(j)}$ corresponds to the two-dimensional chain of Figure 3 instead of to the birth-death chain of Figure 1 for the exhaustive case.

3.4 Gated Service and Finite Buffers

The calculations for gated service with finite buffers are similar to those for exhaustive service in section 3.2. We recall that, when the buffer is infinite, equation (18) for gated service is similar to (9) for exhaustive service and equations (17) and (20) are similar to (10). When the buffer is finite, we obtain equations similar to (11) and (12) using the same arguments, i.e., we define the same set of vectors $\mathbf{K}_{s,s'}^{(j)}(a)$ and sum the equations over this set to account for all the possibilities when the state of any queue i, $i \neq j$, at the end of the mini-cycle is at the buffer size limit.

3.5 Nonpreemptive Timeouts

For the case of preemptive timeouts considered in Sections 3.1-3.4, recall that the customer in service at the end of a timeout is returned to the set of waiting customers by assumption. If the customer is not returned when the timeout expires but instead is allowed to complete service (thus extending the visit of the server for more than T_j time units), then the following modification is required. Uniformization is used on the chain $\mathcal{W}^{(j)}$ for an interval of length T_j. If the queue empties before the server timeout T_j expires, then one proceeds as before. However, if the server timeout does expire, then arrivals to all queues will continue while the customer in service is permitted to finish, and the number of additional arrivals is independent of the system state at time T_j. If q_j' is the state of the served queue j at T_j, then the state at the end of the j-mini-cycle will be q_j'' if there are $q_j'' - q_j' + 1$ customers that arrive to the jth queue after T_j. Also, for the nonserved queues $i \neq j$, if q_i' is the state of the queue at T_j, the state at the end of the j-mini-cycle is q_i'' if there are $q_i'' - q_i'$ arrivals to queue i during the overrun.

Since exponential service time distributions have been assumed, the residual service time (overrun length) is exponential with mean $1/\mu_j$. Furthermore, the random variables ζ_i, $i = 1, \ldots, M$, which represent the additional arrivals at queue i, are independent given the length of the overrun. These properties make it easy to calculate the distribution of the random vector $\zeta = \langle \zeta_1, \ldots, \zeta_M \rangle$. In fact, one immediately sees that

$$P[\zeta_1 = v_1, \ldots, \zeta_M = v_M] = \left(\frac{\mu_j}{\gamma_j}\right) \prod_{i=1}^{M} \left(\frac{\lambda_i}{\gamma_j}\right)^{v_i} \frac{(v_1 + \cdots + v_M)!}{v_1! \cdots v_M!},$$

where recall that $\gamma_j = \sum_{i=1}^{M} \lambda_i + \mu_j$. Let $\mathbf{E}^{(j)} = [e_{s',s''}^{(j)}]$ be the matrix which gives the transition probabilities from state s' at time T_j to state s'' at the end of the j-mini-cycle. Then $e_{s',s''}^{(j)} = 0$ unless $q_i'' - q_i' \geq 0$ for $i \neq j$ and $q_j'' - q_j' + 1 \geq 0$. In this case

$$e_{s',s''}^{(j)} = P[\zeta_1 = q_1'' - q_1', \ldots, \zeta_j = q_j'' - q_j' + 1, \ldots, \zeta_M = q_M'' - q_M'].$$

The transition probabilities for the j-mini-cycle may now be found using the above information. Let $f_{s,s''}^{(j)}$ be the transition probabilities from the beginning of the mini-cycle to its end (including overruns). First consider the infinite buffer case, and assume that the service discipline at queue j is exhaustive. Note that if $q''_j > 0$, then the queue could not have emptied during the mini-cycle, and so also $q'_j > 0$. Thus

$$f^{(j)}_{s,s''} = \sum_{s':0<q'_j\leq q''_j+1} d^{(j)}_{s,s'} e^{(j)}_{s',s''},$$

where the entries $d^{(j)}_{s,s'}$ are given by equation (9). However, when $q''_j = 0$, then either the queue emptied before T_j or the previous case occurred with $q'_j = 1$ and no arrivals to queue j during the overrun. In this case

$$f^{(j)}_{s,s''} = d^{(j)}_{s,s''} + \sum_{s':q'_j=1} d^{(j)}_{s,s'} e^{(j)}_{s',s''}.$$

In the infinite buffer case, when service at queue j is given in a gated fashion with nonpreemptive timeouts, a similar procedure is used to calculate the transition probabilities for the j-mini-cycle. However, unlike the exhaustive discipline, the timeout may not expire $(0 < \tau_j < T_j)$ but $q''_j > 0$, and in this case $q''_j = q'_j$. If $q''_j > 0$ and the timeout did expire, then there must have been $q''_j - q'_j + 1$ arrivals during the overrun. Thus when $q''_j > 0$

$$f^{(j)}_{s,s''} = d^{(j)}_{s,s'',0} + \sum_{s':0<q'_j\leq q''_j+1} d^{(j)}_{s,s',1} e^{(j)}_{s',s''},$$

where $d^{(j)}_{s,s'',0}$ and $d^{(j)}_{s,s'',1}$ are calculated using the gated preemptive equations (17) and (18). A similar argument applies when $q''_j = 0$, and we have

$$f^{(j)}_{s,s''} = d^{(j)}_{s,s''} + \sum_{s':q'_j=1} d^{(j)}_{s,s',1} e^{(j)}_{s',s''},$$

where $d^{(j)}_{s,s''}$ is given by equation (20).

4 Evaluation of $\mathbf{D}^{(j)}$, $\beta^{(i)}$, $\alpha^{(i)}$

In this section we show how to evaluate the matrices $\mathbf{D}^{(j)}$ and the joint probability vectors $\beta^{(i)}$ and $\alpha^{(i)}$ in an efficient and numerically stable manner. We will examine in detail a system with M queues served in a cyclic fashion with exhaustive service at each queue, preemptive timeouts, and infinite buffer sizes. We will then relax this latter assumption and study the finite buffer case. Recursions for the gated discipline are similar to those for the exhaustive discipline, and they are not given here.

4.1 $\mathbf{D}^{(j)}$ for Exhaustive Service and Infinite Buffers

We first consider the evaluation of the transition probability matrix $\mathbf{D}^{(j)}$ for a mini-cycle when queue j is being served. The entries $d^{(j)}_{s,s'}$ of this matrix are given by equations (9) and (10) for the basic system with exhaustive service at the jth queue. The infinite series in these equations have terms involving both the Poisson distribution and the multinomial distribution. Given a specified error tolerance ϵ, the terms of the infinite series are bounded by the corresponding terms of the Poisson distribution, and thus an upper limit $N = N(\epsilon)$ can be determined in advance to ensure that the calculated value will be within ϵ of the actual value [16]. For example, truncating the infinite series in these equations at the integer N introduces an error

$$e(N) \leq \sum_{n=N+1}^{\infty} e^{-\gamma_j T_j} \frac{(\gamma_j T_j)^n}{n!} \leq \epsilon, \tag{21}$$

by choice of N. Note that only entries in equations (9) and (10) with $\kappa_j \leq N$ will be calculated.

For notational convenience assume that $j = 1$. Our interest is in calculating the finite sum $\Upsilon[k_2, \ldots, k_M; N, q_1, q_1']$ corresponding to equation (9), where (for $r \geq \kappa_1 = \sum_{i=2}^{M} k_i$)

$$\Upsilon[k_2, \ldots, k_M; r, q_1, q_1'] = \sum_{n=\kappa_1}^{r} \psi_1(n) \Omega[k_2, \ldots, k_M; n] \pi_{q_1'}(n - \kappa_1, q_1). \tag{22}$$

Here $\psi_1(n) = e^{-\gamma_1 T_1}(\gamma_1 T_1)^n / n!$ is the nth Poisson term. Also in the infinite buffer case

$$\Omega[k_2, \ldots, k_M; n] = \left(\frac{\lambda_2}{\gamma_1}\right)^{k_2} \cdots \left(\frac{\lambda_M}{\gamma_1}\right)^{k_M} \left(\frac{\lambda_1 + \mu_1}{\gamma_1}\right)^{n-\kappa_1} \binom{n}{k_2 \cdots k_M}, \tag{23}$$

where the multinomial coefficient is given by (for $n \geq \kappa_1$)

$$\binom{n}{k_2 \cdots k_M} = \frac{n!}{k_2! \cdots k_M! (n - \kappa_1)!}.$$

Recall that $\Omega[k_2, \ldots, k_M; n]$ is the probability of k_i events of type i, $i = 2, \ldots, M$, given n total events. Then (for $n \geq \kappa_1$)

$$\begin{aligned} \Omega[k_2, \ldots, k_M; n] \;=\; & \sum_{i=2}^{M} \left(\frac{\lambda_i}{\gamma_1}\right) \Omega[k_2, \ldots, k_i - 1, \ldots, k_M; n-1] \mathcal{I}(k_i \geq 1) \\ & + \left(\frac{\lambda_1 + \mu_1}{\gamma_1}\right) \Omega[k_2, \ldots, k_M; n-1] \mathcal{I}(n - 1 \geq \kappa_1), \end{aligned} \tag{24}$$

where $\mathcal{I}(E)$ is the indicator function

$$\mathcal{I}(E) = \begin{cases} 1 & \text{if } E \text{ holds} \\ 0 & \text{otherwise.} \end{cases}$$

Finally we have (for $r \geq \kappa_1$)

$$\begin{aligned}\Upsilon[k_2,\ldots,k_M;r,q_1,q_1'] &= \Upsilon[k_2,\ldots,k_M;r-1,q_1,q_1']\mathcal{I}(r-1 \geq \kappa_1) \\ &\quad +\psi_1(r)\Omega[k_2,\ldots,k_M;r]\pi_{q_1'}(r-\kappa_1,q_1). \qquad (25)\end{aligned}$$

The functions, $\psi_1(r)$ and $\pi_{q_1'}(r-\kappa_1,q_1)$ are obtained recursively from their corresponding expressions for $r-1$. We mention that care must be taken when computing $\psi_1(r)$. A numerically stable recursion for its calculation is presented in [9].

Note that the main computational step in this procedure is the evaluation of the probability $\pi_{q_1'}(r-\kappa_1,q_1)$. But $\pi_{q_1'}$ can be obtained from a vector matrix multiplication, where the matrix represents the uniformized chain corresponding to $\mathcal{W}^{(1)}$. This chain is small, and it is either one-dimensional or two-dimensional for the cases that we consider in this paper. Note also that these recursions are numerically stable, since the calculations involve only adding and multiplying probabilities.

The calculation of quantities obtained by truncating equation (10) is similar. We wish to calculate the finite sum $\Theta[k_2,\ldots,k_M;N,q_1]$, where (for $r \geq \kappa_1+1$)

$$\Theta[k_2,\ldots,k_M;r,q_1] = \sum_{n=\kappa_1+1}^{r} \psi_1(n)\Gamma[k_2,\ldots,k_M;n,q_1]. \qquad (26)$$

For the infinite buffer case, we have (for $n \geq \kappa_1+1$)

$$\begin{aligned}&\Gamma[k_2,\ldots,k_M;n,q_1] = \\ &\quad\sum_{m=\kappa_1+1}^{n}\left(\frac{\lambda_2}{\gamma_1}\right)^{k_2}\cdots\left(\frac{\lambda_M}{\gamma_1}\right)^{k_M}\left(\frac{\lambda_1+\mu_1}{\gamma_1}\right)^{m-\kappa_1}\binom{m-1}{k_2\cdots k_M}\phi_0(m-\kappa_1,q_1). \qquad (27)\end{aligned}$$

Recall that $\Gamma[k_2,\ldots,k_M;n,q_1]$ is the probability of k_i events of type i, $i=2,\ldots,M$, and absorption occurred (queue 1 emptied) at the mth event given there were n total events. Then $(n \geq \kappa_1+1)$

$$\begin{aligned}\Gamma[k_2,\ldots,k_M;n,q_1] &= \Gamma[k_2,\ldots,k_M;n-1,q_1]\mathcal{I}(n-1 \geq \kappa_1+1) \\ &\quad + \left(\frac{\lambda_1+\mu_1}{\gamma_1}\right)\Omega[k_2,\ldots,k_M;n-1]\phi_0(n-\kappa_1,q_1). \qquad (28)\end{aligned}$$

Finally we have $(r \geq \kappa_1+1)$

$$\Theta[k_2,\ldots,k_M;r,q_1] = \Theta[k_2,\ldots,k_M;r-1,q_1]\mathcal{I}(r-1 \geq \kappa_1+1)+\psi_1(r)\Gamma[k_2,\ldots,k_M;r,q_1]. \quad (29)$$

The quantity $\phi_0(n-\kappa_1,q_1) = \pi_0(n-\kappa_1,q_1) - \pi_0(n-1-\kappa_1,q_1)$ can be easily obtained recursively in a numerically stable manner without subtractions as follows. Instead of calculating $\boldsymbol{\pi}(n,q_1)$ directly using the transition matrix $\mathbf{W}^{(j)}$ of the uniformized discrete time

Markov chain, we partition these quantities with respect to the absorbing state 0 as

$$\mathbf{W}^{(j)} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{u} & \mathbf{U}^{(j)} \end{bmatrix},$$

and $\boldsymbol{\pi}(n-1, q_1) = \langle \pi_0(n-1, q_1), \theta(n-1, q_1) \rangle$. Then we calculate $\theta(n, q_1) = \theta(n-1, q_1)\mathbf{U}^{(j)}$, $\phi_0(n, q_1) = \theta(n-1, q_1)\mathbf{u}$, and $\pi_0(n, q_1) = \pi_0(n-1, q_1) + \phi_0(n, q_1)$.

The transition probabilities $d^{(1)}_{s,s'}$ can be easily obtained from the quantities Υ and Θ in the following manner. For $s' = \langle q'_1, \ldots, q'_M \rangle$, $s = \langle q_1, \ldots, q_M \rangle$, define $k_i = q'_i - q_i \geq 0$, $i = 2, \ldots, M$. Then, choosing $N = N(\epsilon)$ as in (21), we have within a prescribed error tolerance ϵ,

$$d^{(1)}_{s,s'} = \begin{cases} \Upsilon[k_2, \ldots, k_M; N, q_1, q'_1] & \text{if } q'_1 > 0 \\ \Theta[k_2, \ldots, k_M; N, q_1] & \text{if } q'_1 = 0. \end{cases} \tag{30}$$

Of course, $d^{(1)}_{s,s'} = 0$ if $q'_i < q_i$ for some $i = 2, \ldots, M$.

4.2 $\mathbf{D}^{(j)}$ for Exhaustive Service and Finite Buffers

There are additional complications when a finite limit is imposed on one or more of the queue buffers. Unlike the infinite buffer case, arrivals to queue i, $i \neq j$, are either accepted or rejected depending on whether or not the corresponding buffer is full. In this case there are no simple expressions corresponding to those of (23) and (27). However, recursions similar to those of (24) and (28) are presented which enable the relevant probabilities to be calculated in an efficient manner. The total number of arrivals to all of the queues $i \neq j$ is explicitly represented in the recursions for calculating $\mathbf{D}^{(j)}$. As before, assume that $j = 1$ to simplify the notation.

We assume that there is a limit B_i on the size of the buffer at queue i, $i = 2, \ldots, M$. This places a limit on the number of arrivals that can be accepted at queue i during the mini-cycle, and that limit depends on the initial state of the queue. That is, if there are q_i at queue i at the beginning of the mini-cycle, only the first $B_i - q_i$ arrivals will be accepted. However, any arrivals after the first B_i will clearly not change the number in queue i, since they will certainly be rejected regardless of the initial queue length. For example, all vectors $\langle k_2, \ldots, k_M \rangle$ representing the number of arrivals at the nonserved queues that satisfy $k_i \geq B_i$ yield the same vector $\langle B_2, \ldots, B_M \rangle$ for number in queue, independent of the starting state. It is convenient to aggregate states together based on this observation. We now develop direct recursions based on (13) and (14) which exploit such an aggregation of states. We remark that similar recursions can be given based on (11) and (12), but they are slightly more complicated to implement than the ones we present.

Let $\mathbf{l} = \langle l_2, \ldots, l_M \rangle$ be a vector such that $l_i \leq B_i$ for all i, where the integers l_i represent the minimum of the buffer limit and the number of arrivals at queue i, and let $\delta_1 = \sum_{i=2}^{M} l_i$. Let $a \geq \delta_1$ represent the total number of arrivals to the nonserved queues. Define the set of vectors

$$\mathbf{K}(\mathbf{l}, a) = \{\langle k_2, \ldots, k_M \rangle : \sum_{i=2}^{M} k_i = a, k_i = l_i \text{ if } l_i < B_i, k_i \geq l_i \text{ if } l_i = B_i\},$$

which corresponds to all patterns of arrivals that yield the vector $\mathbf{l}$.

We consider the case when the server timeout expires (absorption does not occur), and develop recursions similar to those of (24) and (25). Let (for $n \geq a \geq \delta_1$)

$$\Phi[l_2, \ldots, l_M, a; n] = \sum_{\mathbf{k} \in \mathbf{K}(\mathbf{l}, a)} \left(\frac{\lambda_2}{\gamma_1}\right)^{k_2} \cdots \left(\frac{\lambda_M}{\gamma_1}\right)^{k_M} \left(\frac{\lambda_1 + \mu_1}{\gamma_1}\right)^{n-a} \binom{n}{k_2 \cdots k_M}.$$

Then $\Phi[l_2, \ldots, l_M, a; n]$ gives the probability of having a total arrivals at queues $2, \ldots, M$, with l_i arrivals at queue i if $l_i < B_i$ and at least l_i arrivals at queue i if $l_i = B_i$, given n total events (arrivals at the nonserved queues and transitions of the uniformized chain for the served queue). Also let (for $r \geq a \geq \delta_1$)

$$\Psi[l_2, \ldots, l_M, a; r, q_1, q_1'] = \sum_{n=a}^{r} \psi_1(n) \Phi[l_2, \ldots, l_M, a; n] \pi_{q_1'}(n - a, q_1).$$

Then $\Psi[l_2, \ldots, l_M, a; r, q_1, q_1']$ gives the probability of having at most r events in an interval of length T_j of which a are arrivals at the nonserved queues with exactly l_i arrivals if $l_i < B_i$ and at least l_i arrivals if $l_i = B_i$ at queue i, and the state at the served queue is q_1' given that the initial state was q_1.

We now show that these quantities can be calculated recursively without generating the complete set of states of the infinite buffer case and aggregating them. For example, the equation corresponding to (24) for Φ is simply

$$\begin{aligned} \Phi[l_2, \ldots, l_M, a; n] &= \sum_{i=2}^{M} \left(\frac{\lambda_i}{\gamma_1}\right) \Phi[l_2, \ldots, l_i - 1, \ldots, l_M, a - 1; n - 1] \mathcal{I}(0 \leq l_i - 1 < B_i) \\ &\quad + \sum_{i=2}^{M} \left(\frac{\lambda_i}{\gamma_1}\right) \Phi[l_2, \ldots, l_i, \ldots, l_M, a - 1; n - 1] \mathcal{I}(l_i = B_i) \\ &\quad + \left(\frac{\lambda_1 + \mu_1}{\gamma_1}\right) \Phi[l_2, \ldots, l_M, a; n - 1] \mathcal{I}(n - 1 \geq a). \end{aligned} \tag{31}$$

The first sum represents the case of an arrival at queue i that possibly may be accepted depending on the initial state, while the second sum represents an arrival that will surely be rejected since at least B_i arrivals occurred previously. The third case represents a transition of the uniformized chain for queue 1. Also, the equation corresponding to (25) for Ψ is

$$\begin{aligned}\Psi[l_2,\ldots,l_M,a;r,q_1,q_1'] &= \Psi[l_2,\ldots,l_M,a;r-1,q_1,q_1']\mathcal{I}(r-1\geq a)\\ &\quad +\psi_1(r)\Phi[l_2,\ldots,l_M,a;r]\pi_{q_1'}(r-a,q_1). \qquad (32)\end{aligned}$$

Finally

$$\begin{aligned}\Psi^*[l_2,\ldots,l_M;N,q_1,q_1'] &= \sum_{a=\delta_1}^{r}\Psi[l_2,\ldots,l_M,a;N,q_1,q_1'] \qquad (33)\\ &= \Psi^*[l_2,\ldots,l_M,a-1;N,q_1,q_1']+\Psi[l_2,\ldots,l_M,a;N,q_1,q_1'] \qquad (34)\end{aligned}$$

represents the probability of having at most r events during T_j with l_i arrivals if $l_i < B_i$ and at least l_i arrivals if $l_i = B_i$ for the nonserved queues, and the state at the served queue at the end of the mini-cycle is q_1' given the state at the beginning of the mini-cycle was q_1.

The recursions for the case when absorption occurs (the timeout does not expire) are similar. The quantities corresponding to Γ and Θ are (for $n-1\geq a\geq\delta_1$)

$$\Delta[l_2,\ldots,l_M,a;n,q_1] = \sum_{\mathbf{k}\in\mathbf{K}(\mathbf{l},a)}\sum_{m=a+1}^{n}\left(\frac{\lambda_2}{\gamma_1}\right)^{k_2}\cdots\left(\frac{\lambda_M}{\gamma_1}\right)^{k_M}\left(\frac{\lambda_1+\mu_1}{\gamma_1}\right)^{m-a}\binom{m-1}{k_2\cdots k_M}\phi(m-a,q_1), \qquad (35)$$

and (for $r-1\geq a\geq\delta_1$)

$$\Xi[l_2,\ldots,l_M,a;r,q_1] = \sum_{n=a+1}^{r}\psi_1(n)\Delta[l_2,\ldots,l_M,a;n,q_1]. \qquad (36)$$

The recursion for Δ is

$$\begin{aligned}\Delta[l_2,\ldots,l_M,a;n,q_1] &= \Delta[l_2,\ldots,l_M,a;n-1,q_1]\mathcal{I}(n-1\geq a+1)\\ &\quad +\left(\frac{\lambda_1+\mu_1}{\gamma_1}\right)\Phi[l_2,\ldots,l_M,a;n-1]\phi(n-a,q_1), \qquad (37)\end{aligned}$$

while that for Ξ is

$$\begin{aligned}\Xi[l_2,\ldots,l_M,a;r,q_1] &= \Xi[l_2,\ldots,l_M,a;r-1,q_1]\mathcal{I}(r-1\geq a+1)\\ &\quad +\psi_1(r)\Delta[l_2,\ldots,l_M,a;r,q_1]. \qquad (38)\end{aligned}$$

Finally let

$$\begin{aligned}\Xi^*[l_2,\ldots,l_M;N,q_1] &= \sum_{a=\delta_1}^{r-1}\Xi[l_2,\ldots,l_M,a;N,q_1] \qquad (39)\\ &= \Xi^*[l_2,\ldots,l_M,a-1;N,q_1]+\Xi[l_2,\ldots,l_M,a;N,q_1]. \qquad (40)\end{aligned}$$

We now use these quantities to develop a recursion to calculate the entries of $\mathbf{D}^{(1)}$ in the finite buffer case. Consider a pair of states s, s', and the calculation of $d^{(1)}_{s,s'}$. First note that if, for all $i = 2, \ldots, M$, either $q_i = 0$ or $q'_i < B_i$, then the above recursions calculate this entry. In this case, using equation (33) to within ϵ:

$$d^{(1)}_{s,s'} = \begin{cases} \Psi^*[l_2, \ldots, l_M; N, q_1, q'_1] & q'_1 > 0 \\ \Xi^*[l_2, \ldots, l_M; N, q_1] & q'_1 = 0. \end{cases} \tag{41}$$

Thus we need only determine how to calculate $d^{(1)}_{s,s'}$ when $q_i > 0$ and $q'_i = B_i$ for at least one nonserved queue. In this case, several terms calculated from the above recursions must be added to obtain the correct value. For example, if $q'_i = B_i$, then there are at least $l_i = B_i - q_i$ arrivals to queue i during the mini-cycle. Therefore, in order to obtain the transition probability $d^{(1)}_{s,s'}$, one must consider all cases which represent at least l_i arrivals. Recall that the vectors Ψ^*, Ξ^* with ith entry B_i include all situations with at least B_i arrivals. Therefore, we need only add up the cases for which there are $l_i, l_i + 1, \ldots, B_i$ arrivals.

We now consider the case when there is a nonserved queue with $q_i > 0$ and $q'_i = B_i$. We claim that these entries are given by the recursion

$$d^{(1)}_{s,s'} = d^{(1)}_{s-e_i,s'} + d^{(1)}_{s-e_i,s'-e_i}, \tag{42}$$

where we use the notation $s - e_i = \langle q_1, \ldots, q_i - 1, \ldots, q_M \rangle$. To see this, note that only the state of queue i changes in the equation. The first term of the right-hand sum represents the case of at least $l_i + 1$ arrivals to queue i, while the second term represents the case of exactly l_i arrivals to this queue. Since the left-hand term represents the case of at least l_i arrivals to queue i, we see that (42) holds. The initial conditions are given by equation (41).

4.3 $\mathbf{D}^{(j)}$ for Gated Service

For the infinite case with exhaustive service we obtained equations (9) and (10) and found recursions to calculate the values of $d^{(j)}_{s,s'}$. In the finite buffer case equations (11) and (12) are used instead and the recursions are similar, though slightly more complex.

The entries $d^{(j)}_{s,s'}$ for gated service can be calculated in the same way as for the exhaustive system. First in the infinite buffer case, equations (17), (18) and (20) were obtained. We note that equation (18) is identical to (9) except that $\pi_{q'_j}$ in (9) is replaced by $\sum_v \pi_{(q'_j,v)}$ in (18). As a consequence, we can use a recursion which is identical to that for Υ in equation (25) to calculate a new quantity Υ^G if we replace $\pi_{q'_j}$ by $\sum_v \pi_{(q'_j,v)}$ in (22). Note that the definition of Ω remains the same.

Next we find recursions for equations (17) and (20). Comparing those with equation (10), we see that they are identical except that $\phi_0(\ )$ in (10) is replaced by $\phi_{(q'_j,0)}(\ )$ in (17) and $\phi_{(0,0)}(\ )$ in (20). As a consequence, we can define $\Gamma^{G,a}$ and $\Gamma^{G,b}$ analogous to Γ in (27). Similarly, we define $\Theta^{G,a}$ and $\Theta^{G,b}$ analogous to Θ in (29). Finally, within a prespecified ϵ:

$$d^{(1)}_{s,s'} = \begin{cases} \Upsilon^G[k_2,\ldots,k_M;N,q_1,q_1'] + \Theta^{G,a}[k_2,\ldots,k_M;N,q_1] & \text{if } q_1' > 0 \\ \Theta^{G,b}[k_2,\ldots,k_M;N,q_1] & \text{if } q_1' = 0. \end{cases} \tag{43}$$

4.4 Calculation of the Joint Probability Vectors

Once the entries of $\mathbf{D}^{(j)}$, $j = 1,\ldots,M$, have been evaluated, the steady state vectors $\beta^{(i)}$, $\alpha^{(i)}$, $i = 1,\ldots,M$, can be calculated simultaneously in an iterative fashion as follows. Let $\beta^{(i)}(n)$ and $\alpha^{(i)}(n)$ be the values calculated at iteration n, and without loss of generality we start the iteration with the initial value $\beta^{(1)}(0)$. Then recalling equation (2), we iterate from mini-cycle to mini-cycle using the recursion

$$\begin{array}{ll} \alpha^{(i)}(n) = \beta^{(i)}(n)\mathbf{D}^{(i)} & i = 1,\ldots,M \\ \beta^{(i)}(n) = \alpha^{(i-1)}(n)\mathbf{C}^{(i-1)} & i = 2,\ldots,M \\ \beta^{(1)}(n) = \alpha^{(M)}(n-1)\mathbf{C}^{(M)} & . \end{array}$$

The matrices $\mathbf{D}^{(j)}$ and $\mathbf{C}^{(j)}$ have special structure which can be exploited in the iterations. Note that we have (see equation (1))

$$\begin{array}{ll} \beta^{(i)}(n) = \beta^{(i)}(n-1)\mathbf{H}^{(i)} & i = 1,\ldots,M \\ \alpha^{(i)}(n) = \alpha^{(i)}(n-1)\mathbf{G}^{(i)} & i = 1,\ldots,M. \end{array}$$

The matrices $\mathbf{H}^{(i)}$, $\mathbf{G}^{(i)}$, which do not have such special structure, are not calculated explicitly. We are essentially employing the power method, so that the iteration converges to $\beta^{(i)}$, $\alpha^{(i)}$, respectively.

5 Time Average Measures

In the previous two sections, we studied the embedded Markov chains $\mathcal{Y}^{(i)}$ and $\mathcal{Z}^{(i)}$ defined at time points when the server visits queue i and when the server leaves queue i, respectively. The steady state vectors $\beta^{(i)}$ of $\mathcal{Y}^{(i)}$ and $\alpha^{(i)}$ of $\mathcal{Z}^{(i)}$ were obtained using numerically stable recursions. We now show how $\beta^{(i)}$ and $\alpha^{(i)}$ can be used to calculate various time average measures using results from Markov chains with rewards. We also retain assumptions from the previous sections, namely, infinite buffers, cyclic switching policy, preemptive timeouts, constant switchover times and either exhaustive or gated service at each queue. In the finite buffer case, the procedure is virtually identical to that for infinite buffers, and we omit the details.

Let $P_{\mathcal{R}}$ be the limiting probability as $t \to \infty$ that the process $X(t)$ is in a particular subset $\mathcal{R} \subset \mathcal{S}$ of states, i.e. $P_{\mathcal{R}} = \lim_{t\to\infty} P[X(t) \in \mathcal{R}]$. For example, $\mathcal{R}$ may be the set of states for which the lengths of the M queues have certain values, if joint queue length distributions are desired. To obtain the marginal queue length distribution of queue i, we choose $\mathcal{R}$ to be the set of states for which the length of that queue has a particular value. Such limiting marginal distributions can also be used to obtain other measures, such as expected queue lengths over all time. The average waiting times for each queue can then be calculated using Little's result. Note that the PASTA property and the fact that the process $X(t)$ changes in unit steps imply that distributions at customer arrival points and customer departure points are given by time averages. In the case of finite buffers, blocking probabilities can be easily evaluated by exploiting the Poisson arrival assumption.

We assume that the regenerative process $X(t)$ is stable in the sense that the limiting probabilities exist. Recently, Georgiadis and Szpankowski [15] have shown that the stability condition for the time-limited polling system with preemptive timeouts and exponential service times is $\rho < 1$ and

$$\rho_i \frac{\sigma}{1-\rho} < T_i$$

for all queues i, where $\rho_i = \lambda_i/\mu_i$, $\rho = \sum_{i=1}^{M} \rho_i$ and $\sigma = \sum_{i=1}^{M} \sigma_i$ is the total mean switchover time in a polling cycle. For nonpreemptive timeouts, the condition is simply $\rho < 1$ and

$$\rho_i \frac{\sigma}{1-\rho} < T_i + \frac{1}{\mu_i}$$

for all queues i (see also the work of Fricker and Jaïbi [12]).

Define the random variable $R(t)$ to be the amount of time $X(t)$ spends in $\mathcal{R}$ during $(0,t)$. Then from the ergodic theorem for regenerative processes (see [19]), the limiting probability for set $\mathcal{R}$ satisfies

$$P_{\mathcal{R}} \overset{\text{w.p.1}}{=} \lim_{t\to\infty} \frac{R(t)}{t}. \tag{44}$$

It is well known that $P_{\mathcal{R}}$ is the ratio of the expected time spent in the set $\mathcal{R}$ during a regeneration cycle to the expected length of such a cycle. However, this probability can also be expressed in terms of expectations over a polling cycle (i.e. the time between two consecutive server arrivals or two consecutive server departures at the same queue).

To see this, we use results about Markov chains with rewards. For notational convenience consider a polling cycle defined by times when the server arrives to queue 1. We will see that concentrating on other types of polling cycles yields the same equation for $P_{\mathcal{R}}$. Let R_s be a reward equal to the cumulative time that the process $X(t)$ spends in the set of states $\mathcal{R}$ during a polling cycle, given that the cycle started in state s. When $\mathcal{R}$ is the set of all states $\mathcal{S}$, the cumulative time is simply the length of a polling cycle, and we designate the random variable in this special case as L_s. Recall that the kth polling cycle is given by the interval

(η_{k-1}, η_k), where we have dropped the superscript (1) for convenience. Next let $R_{s'}(k)$ be the total reward (total time in $\mathcal{R}$) during $(0, \eta_k)$ given that $Y_0 = s'$. From Theorem 7.14 of [19], we have for all $s' \in \mathcal{S}$

$$\lim_{k\to\infty} \frac{R_{s'}(k)}{k} \overset{\text{w.p.1}}{=} \sum_{s\in\mathcal{S}} E[R_s]\beta_s, \tag{45}$$

where β_s is the sth element of the probability vector satisfying $\beta = \beta\mathbf{H}$ (i.e. $\beta^{(1)} = \beta^{(1)}\mathbf{H}^{(1)}$).

Let $M(t)$ be the number of transitions of the embedded Markov chain by time t, i.e. $M(t) = \max\{k : \eta_k \leq t\}$. We clearly have

$$\frac{R(\eta_{M(t)})}{t} \leq \frac{R(t)}{t} \leq \frac{R(\eta_{M(t)+1})}{t}. \tag{46}$$

The lefthand side of equation (46) can be written as

$$\frac{R(\eta_{M(t)})}{t} = \frac{\eta_{M(t)}}{t} \cdot \frac{M(t)}{\eta_{M(t)}} \cdot \frac{R(\eta_{M(t)})}{M(t)}.$$

It is easy to see that $\lim_{t\to\infty} \eta_{M(t)}/t \overset{\text{w.p.1}}{=} 1$. Further, using equation (45) with $\mathcal{R}$ we have (independent of the initial state s')

$$\lim_{t\to\infty} \frac{R(\eta_{M(t)})}{M(t)} \overset{\text{w.p.1}}{=} \sum_{s\in\mathcal{S}} E[R_s]\beta_s,$$

while using that equation with $\mathcal{S}$ gives

$$\lim_{t\to\infty} \frac{\eta_{M(t)}}{M(t)} \overset{\text{w.p.1}}{=} \sum_{s\in\mathcal{S}} E[L_s]\beta_s.$$

Recall that R_s is the amount of time spent in the set of states $\mathcal{R}$ during a polling cycle given the initial state s, and L_s is the length of such a polling cycle. Similar results hold for the right-hand side of (46). Therefore, using equation (44) we may express the limiting probability in terms of quantities involving a polling cycle and the embedded Markov chain as

$$P_{\mathcal{R}} = \frac{\sum_{s\in\mathcal{S}} E[R_s]\beta_s}{\sum_{s\in\mathcal{S}} E[L_s]\beta_s}. \tag{47}$$

This is the key equation we use to obtain limiting probabilities for the process $X(t)$. Similar equations may be derived for the other types of polling cycles (i.e. server arrivals to queue $i \neq 1$ or server departures from a queue).

Equation (47) above requires the calculation of the expected value of the time that the process is in $\mathcal{R}$ during a polling cycle that starts when the server visits queue 1 (given the initial state), and also the expected length of such a cycle. However, as indicated in Section

3, it is easier to compute quantities over mini-cycles and switchover intervals and then sum the resulting expectations. To this end, we define $U_s^{(j)}$ to be the time spent in $\mathcal{R}$ during a j-mini-cycle given that s was the state at the start of the mini-cycle, and $V_s^{(j)}$ to be the time spent in $\mathcal{R}$ during a j-switchover interval given the state at the start of the interval was s.

By a simple conditioning argument, we may express $E[R_s]$ in terms of the above quantities as

$$E[R_s] = \sum_{j=1}^{M} \sum_{s' \in \mathcal{S}} E[U_{s'}^{(j)}] p_{s,s'}^{(j)} + \sum_{j=1}^{M} \sum_{s' \in \mathcal{S}} E[V_{s'}^{(j)}] o_{s,s'}^{(j)}.$$

The matrices $\mathbf{P}^{(j)}$, with (s, s') element $p_{s,s'}^{(j)}$, give transition probabilities from the start of the polling cycle to the start of the j-mini-cycle. Similarly, the matrices $\mathbf{O}^{(j)}$, with (s, s') element $o_{s,s'}^{(j)}$, represent transitions from the start of the polling cycle to the start of the j-switchover interval. They satisfy

$$\begin{aligned} &\mathbf{P}^{(1)} = \mathbf{I} \\ &\mathbf{O}^{(j)} = \mathbf{P}^{(j)}\mathbf{D}^{(j)} \qquad j = 1, \ldots, M \\ &\mathbf{P}^{(j)} = \mathbf{O}^{(j-1)}\mathbf{C}^{(j-1)} \quad j = 2, \ldots, M, \end{aligned} \tag{48}$$

where $\mathbf{I}$ is the identity matrix. The numerator of (47) can therefore be written as

$$\sum_{s \in \mathcal{S}} E[R_s]\beta_s = \sum_{j=1}^{M} \sum_{s' \in \mathcal{S}} E[U_{s'}^{(j)}] \sum_{s \in \mathcal{S}} \beta_s p_{s,s'}^{(j)} + \sum_{j=1}^{M} \sum_{s' \in \mathcal{S}} E[V_{s'}^{(j)}] \sum_{s \in \mathcal{S}} \beta_s o_{s,s'}^{(j)}.$$

Now from (2) and (48), the following equalities are easily seen to hold, namely,

$$\begin{aligned} \beta^{(1)}\mathbf{P}^{(j)} &= \beta^{(j)} \quad j = 1, \ldots, M \\ \beta^{(1)}\mathbf{O}^{(j)} &= \alpha^{(j)} \quad j = 1, \ldots, M. \end{aligned}$$

Using these results, the numerator of (47) is

$$\sum_{s \in \mathcal{S}} E[R_s]\beta_s = \sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[U_s^{(j)}]\beta_s^{(j)} + \sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[V_s^{(j)}]\alpha_s^{(j)}.$$

In a similar manner, the denominator of (47) is given by

$$\sum_{s \in \mathcal{S}} E[L_s]\beta_s = \sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[B_s^{(j)}]\beta_s^{(j)} + \sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[A_s^{(j)}]\alpha_s^{(j)},$$

where $B_s^{(j)}$ is the length of a j-mini-cycle given that it starts in state s and $A_s^{(j)}$ is the length of a j-switchover interval given that the initial state is s. Thus (47) may be expressed in terms of quantities involving mini-cycles and switchover intervals as

$$P_{\mathcal{R}} = \frac{\sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[U_s^{(j)}]\beta_s^{(j)} + \sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[V_s^{(j)}]\alpha_s^{(j)}}{\sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[B_s^{(j)}]\beta_s^{(j)} + \sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[A_s^{(j)}]\alpha_s^{(j)}}. \tag{49}$$

The same equation results from considering arrival polling cycles or departure polling cycles for any queue i.

5.1 $P_{\mathcal{R}}$ for Exhaustive Service and Infinite Buffers

The quantities in the denominator of equation (49) are easiest to obtain (and are also independent of $\mathcal{R}$), and we will begin with their calculation. First note that since the switchover intervals are constant, we have for $j = 1, \ldots, M$ that $E[A_s^{(j)}] = \sigma_j$ independent of s. Thus the second term in the denominator of equation (49) is simply

$$\sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[A_s^{(j)}] \alpha_s^{(j)} = \sum_{j=1}^{M} \sigma_j, \tag{50}$$

the expected total switchover time in a polling cycle. Clearly this result is independent of the scheduling disciplines and buffer sizes at the various queues.

We next wish to determine $E[B_s^{(j)}]$. Recall that the length of a j-mini-cycle depends on the number present at queue j when the server arrives, but not on the number at queue i, $i \neq j$. Therefore, we have $E[B_s^{(j)}] = E[B_{s'}^{(j)}]$ when $q_j = q'_j$, where $s = \langle q_1, \ldots, q_M \rangle$ and $s' = \langle q'_1, \ldots, q'_M \rangle$. Let the vector $\mathbf{b}^{(j)}$ be obtained from $\beta^{(j)}$ by aggregating states together with the same jth element. That is, for $q = 0, 1, \ldots$, define the set of states $\mathcal{S}_q^{(j)} = \{\langle q_1, \ldots, q_M \rangle \in \mathcal{S} : q_j = q\}$, and let

$$\mathbf{b}_q^{(j)} = \sum_{s \in \mathcal{S}_q^{(j)}} \beta_s^{(j)}.$$

The first denominator term of equation (49) can be written as

$$\sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[B_s^{(j)}] \beta_s^{(j)} = \sum_{j=1}^{M} \sum_{q=0}^{\infty} E[\tau_j \,|\, q] \mathbf{b}_q^{(j)},$$

where recall that τ_j is the length of a j-mini-cycle. Given that there are q at queue j at the start of a j-mini-cycle, the distribution $F(t, q)$ of the length τ_j of the mini-cycle was calculated in equation (5). The conditional expected length can be obtained using the formula $E[\tau_j \,|\, q] = \int_0^{T_j} [1 - F(t, q)] dt$ and this equation. Proceeding in a manner similar to the derivation of equation (10), we find that

$$E[\tau_j \,|\, q] = T_j - T_j \sum_{n=0}^{\infty} e^{-\Lambda_j T_j} \frac{(\Lambda_j T_j)^n}{n!} \left\{ \frac{\sum_{m=0}^{n} \pi_0(m, q)}{n+1} \right\}. \tag{51}$$

Here $\boldsymbol{\pi}(m, q) = \boldsymbol{\pi}(m-1, q) \mathbf{W}^{(j)}$, where $\mathbf{W}^{(j)}$ is the transition matrix of the uniformized chain corresponding to $\mathcal{W}^{(j)}$, and $\boldsymbol{\pi}(0, q) = \mathbf{e}_q$. This equation may also be derived directly by noting that $E[\tau_j \,|\, q]$ is the expected time during the mini-cycle when the system is not in the absorbing state 0, which is T_j minus the expected time when the system is in state 0. Arguments similar to those used in obtaining equation (55) below yield the result.

Note that a direct application of equation (51) in calculating the first denominator term of (49) would involve computing the conditional expected j-mini-cycle length for each queue size q. However, this is not necessary, since it is easy to show that computations need only be carried out using the uniformized chain with an initial distribution $\mathbf{b}^{(j)}$. To see this, we observe that

$$\sum_{q=0}^{\infty} \mathbf{b}_q^{(j)} \boldsymbol{\pi}(m,q) = \sum_{q=0}^{\infty} \mathbf{b}_q^{(j)} \boldsymbol{\pi}(0,q)\left(\mathbf{W}^{(j)}\right)^m = \sum_{q=0}^{\infty} \mathbf{b}_q^{(j)} \mathbf{e}_q \left(\mathbf{W}^{(j)}\right)^m = \mathbf{b}^{(j)}\left(\mathbf{W}^{(j)}\right)^m,$$

and hence

$$\sum_{q=0}^{\infty} \mathbf{b}_q^{(j)} \boldsymbol{\pi}(m,q) = \boldsymbol{\pi}(0,\mathbf{b}^{(j)})\left(\mathbf{W}^{(j)}\right)^m = \boldsymbol{\pi}(m,\mathbf{b}^{(j)}).$$

It is then apparent from the form of equation (51) that

$$\sum_{q=0}^{\infty} E[\tau_j \,|\, q]\mathbf{b}_q^{(j)} = T_j - T_j \sum_{n=0}^{\infty} e^{-\Lambda_j T_j} \frac{(\Lambda_j T_j)^n}{n!} \left\{ \frac{\sum_{m=0}^{n} \pi_0(m, \mathbf{b}^{(j)})}{n+1} \right\},$$

where $\boldsymbol{\pi}(m,\mathbf{b}^{(j)}) = \boldsymbol{\pi}(m-1,\mathbf{b}^{(j)})\mathbf{W}^{(j)}$, but the initial distribution is $\boldsymbol{\pi}(0,\mathbf{b}^{(j)}) = \mathbf{b}^{(j)}$. Thus using this uniformization procedure, we obtain the expected length of a polling cycle (the first denominator term) as

$$\sum_{j=1}^{M} \sum_{s \in \mathcal{S}} E[B_s^{(j)}]\beta_s^{(j)} = \sum_{j=1}^{M} T_j - \sum_{j=1}^{M} T_j \sum_{n=0}^{\infty} e^{-\Lambda_j T_j} \frac{(\Lambda_j T_j)^n}{n!} \left\{ \frac{\sum_{m=0}^{n} \pi_0(m, \mathbf{b}^{(j)})}{n+1} \right\}. \tag{52}$$

Here $\sum_{j=1}^{M} T_j$ is the sum of all M server timeouts, the maximum amount of time in a polling cycle during which the server can be busy when there are no overruns.

The bracketed term in the above expression (52) may be easily calculated in a recursive manner as follows [9]. Setting

$$f(n) = \frac{\sum_{m=0}^{n} \pi_0(m, \mathbf{b}^{(j)})}{n+1},$$

we have the recursion

$$f(n+1) = \frac{n+1}{n+2} f(n) + \frac{\pi_0(n+1, \mathbf{b}^{(j)})}{n+2}.$$

Note that $f(n) \leq 1$ for all n.

We now consider the calculation of the numerator terms of equation (49). Expressions for these terms will depend upon the set $\mathcal{R}$ of interest. For simplicity, we will assume that $\mathcal{R}$ is the subset of states for which a certain queue, say queue i, has h_i customers, which will enable us to obtain marginal queue distributions over all time. However, from the derivations it will be clear how to handle other sets (e.g. to obtain joint queue length distributions).

We first consider the numerator term corresponding to the switchover intervals, specifically the j-switchover interval. Since the length of such an interval is a constant σ_j, determining the time spent with h_i at queue i during the switchover period only depends on the size of queue i at its start and the number of arrivals which occur (at rate λ_i) to that particular queue. Let q_i be the number in queue i at the start of the j-switchover interval. Clearly if $q_i > h_i$, then there cannot be h_i at queue i at any time during the switchover period, since no departures from any queue can occur. Thus we need only concentrate on the case when $q_i \leq h_i$. In this case, there must be at least $h_i - q_i$ arrivals to queue i during the switchover interval for there to be h_i at queue i at some time. If there are n such Poisson arrivals, then the switchover interval is split into $n+1$ subintervals. It is well known that the random variables representing the lengths of these subintervals are exchangeable [6, 7], and thus the expected length of any subinterval is simply $\sigma_j/(n+1)$. It is also clear that if there are $n \geq h_i - q_i$ arrivals, then the amount of time during the switchover period when there are h_i at queue i is given by the length of a single subinterval, since there are no departures from the queue. Using these observations, we may write

$$\sum_{s\in\mathcal{S}} E[V_s^{(j)}]\alpha_s^{(j)} = \sum_{q_i=0}^{h_i} \sum_{n=h_i-q_i}^{\infty} e^{-\lambda_i\sigma_j} \frac{(\lambda_i\sigma_j)^n}{n!} \left(\frac{\sigma_j}{n+1}\right) \mathbf{a}_{q_i}^{(i,j)}, \tag{53}$$

where similar to the definition of $\mathbf{b}^{(j)}$, the vector $\mathbf{a}^{(i,j)}$ is obtained from $\alpha^{(j)}$ by aggregating states together with the same ith entry. That is, for $q = 0, 1, \ldots$,

$$\mathbf{a}_q^{(i,j)} = \sum_{s\in\mathcal{S}_q^{(i)}} \alpha_s^{(j)}.$$

Thus the second numerator term becomes

$$\sum_{j=1}^{M} \sum_{s\in\mathcal{S}} E[V_s^{(j)}]\alpha_s^{(j)} = \sum_{j=1}^{M} \frac{1}{\lambda_i} \sum_{q_i=0}^{h_i} E_{h_i-q_i,\lambda_i}(\sigma_j)\mathbf{a}_{q_i}^{(i,j)}, \tag{54}$$

where $E_{k,\lambda}(t) = 1 - \sum_{n=0}^{k} e^{-\lambda t}(\lambda t)^n/n!$ is the $(k+1)$-stage Erlangian distribution. Note that each of the M terms of the right-hand expression in equation (54) is bounded above by $1/\lambda_i$, the expected length of time between two arrivals at queue i. As was the case for $E[A_s^{(j)}]$, the result for $E[V_s^{(j)}]$ is also independent of the scheduling disciplines at the various queues, since it only involves the switchover periods. We also mention that the Erlangian distribution is easy to calculate in a stable recursive manner (see [16]).

Our final task is to obtain an expression for the first term in the numerator of equation (49), which corresponds to the expected time during a j-mini-cycle when there are h_i at queue i. We split our discussion according to whether $j = i$ or $j \neq i$, and consider the case when $j = i$ first. In this case we wish to find the expected time during an i-mini-cycle when queue i (which is being served) has h_i customers. We can assume that $h_i > 0$, since the server immediately switches when queue i becomes empty, and so $h_i = 0$ does not occur during an i-mini-cycle. Using the uniformized chain corresponding to $\mathcal{W}^{(i)}$ (with absorbing state 0), the expected time can be found by transient analysis over an interval of length T_i. Given n transitions of the uniformized Markov chain, note that there may be several subintervals during which there are h_i customers, and each subinterval contributes an amount of time $T_i/(n+1)$ to the expectation. Thus, given q_i at the start of the mini-cycle, it is easy to show that (see also [9])

$$E[U^{(i)}\,|\,q_i] = \sum_{n=0}^{\infty} e^{-\Lambda_i T_i} \frac{(\Lambda_i T_i)^n}{n!} \left\{ \frac{T_i \sum_{m=0}^{n} \pi_{h_i}(m, q_i)}{n+1} \right\},$$

where $\boldsymbol{\pi}(m, q_i) = \boldsymbol{\pi}(m-1, q_i)\mathbf{W}^{(i)}$ and $\boldsymbol{\pi}(0, q_i) = \mathbf{e}_{q_i}$. Therefore, using the vector $\mathbf{b}^{(i)}$ for the initial distribution of the uniformized chain as was done in equation (52), we obtain

$$\sum_{s \in \mathcal{S}} E[U_s^{(i)}]\beta_s^{(i)} = T_i \sum_{n=0}^{\infty} e^{-\Lambda_i T_i} \frac{(\Lambda_i T_i)^n}{n!} \left\{ \frac{\sum_{m=0}^{n} \pi_{h_i}(m, \mathbf{b}^{(i)})}{n+1} \right\}, \tag{55}$$

where $\boldsymbol{\pi}(m, \mathbf{b}^{(i)}) = \boldsymbol{\pi}(m-1, \mathbf{b}^{(i)})\mathbf{W}^{(i)}$ and $\boldsymbol{\pi}(0, \mathbf{b}^{(i)}) = \mathbf{b}^{(i)}$. Note that equation (51) is essentially the special case $h_i = 0$ of this equation.

The case when $j \neq i$ is similar to the derivation of equation (54), since it also involves arrivals to queue i during a particular interval of time. The amount of time during the j-mini-cycle when there are h_i at queue i depends, of course, on the number q_i at that queue at the start of the mini-cycle. However, it also depends on q_j, the initial number at queue j, since this controls the length of the mini-cycle. To determine $E[U^{(j)}\,|\,q_i, q_j]$, as before the only possibility to consider is when $h_i \geq q_i$, the number at queue i at the start of the j-mini-cycle. Given that the mini-cycle length is t, the expected time when there are h_i at queue i has been determined previously in the derivation of equation (53). Thus further conditioning on the length of the j-mini-cycle, we have

$$\begin{aligned} E[U^{(j)}\,|\,q_i, q_j] \;&=\; \int_0^{T_j} \sum_{n=h_i-q_i}^{\infty} e^{-\lambda_i t} \frac{(\lambda_i t)^n}{n!} \left(\frac{t}{n+1} \right) F'(t, q_j)\, dt \\ &\quad + \sum_{n=h_i-q_i}^{\infty} e^{-\lambda_i T_j} \frac{(\lambda_i T_j)^n}{n!} \left(\frac{T_j}{n+1} \right) [1 - F(T_j, q_j)]. \end{aligned}$$

Substituting the expressions from equations (5) and (6), we follow the steps used to derive equation (10) and obtain

$$E[U^{(j)}\,|\,q_i, q_j] \;=\; \frac{1}{\lambda_i} \sum_{n=h_i-q_i+1}^{\infty} \left\{ \sum_{k=n+1}^{\infty} e^{-(\lambda_i+\Lambda_j)T_j} \frac{[(\lambda_i + \Lambda_j)T_j]^k}{k!} \right.$$

$$
\begin{aligned}
&\times \sum_{m=n+1}^{k} \left(\frac{\lambda_i}{\lambda_i+\Lambda_j}\right)^n \left(\frac{\Lambda_j}{\lambda_i+\Lambda_j}\right)^{m-n} \binom{m-1}{n} \phi_0(m-n, q_j) \\
&+ \sum_{k=n}^{\infty} e^{-(\lambda_i+\Lambda_j)T_j} \frac{[(\lambda_i+\Lambda_j)T_j]^k}{k!} \\
&\times \left(\frac{\lambda_i}{\lambda_i+\Lambda_j}\right)^n \left(\frac{\Lambda_j}{\lambda_i+\Lambda_j}\right)^{k-n} \binom{k}{n} [1-\pi_0(k-n, q_j)] \Bigg\}
\end{aligned}
$$

where $\boldsymbol{\pi}(m, q_j) = \boldsymbol{\pi}(m-1, q_j)\mathbf{W}^{(j)}$ and $\boldsymbol{\pi}(0, q_j) = \mathbf{e}_{q_j}$.

We now uncondition on q_i, q_j and use the same procedure as before to ensure that calculations with the uniformized chain are not required for each of these possible pairs. Specifically, let the vector $\mathbf{b}^{(i,j)}(q')$ be obtained from $\beta^{(j)}$ by aggregating states with the same jth element and with ith element equal to q'. That is, for $q = 0, 1, \ldots$, define the set of states $\mathcal{S}_q^{(i,j)}(q') = \{\langle q_1, \ldots, q_M \rangle \in \mathcal{S} : q_j = q, q_i = q'\}$, and let

$$
\mathbf{b}_q^{(i,j)}(q') = \sum_{s \in \mathcal{S}_q^{(i,j)}(q')} \beta_s^{(j)}.
$$

Then we may write

$$
\begin{aligned}
\sum_{s\in\mathcal{S}} E[U_s^{(j)}]\beta_s^{(j)} \;=\; & \frac{1}{\lambda_i} \sum_{q_i=0}^{h_i} \sum_{n=h_i-q_i+1}^{\infty} \Bigg\{ \sum_{k=n+1}^{\infty} e^{-(\lambda_i+\Lambda_j)T_j} \frac{[(\lambda_i+\Lambda_j)T_j]^k}{k!} \\
&\times \sum_{m=n+1}^{k} \left(\frac{\lambda_i}{\lambda_i+\Lambda_j}\right)^n \left(\frac{\Lambda_j}{\lambda_i+\Lambda_j}\right)^{m-n} \binom{m-1}{n} \phi_0(m-n, \mathbf{b}^{(i,j)}(q_i)) \\
&+ \sum_{k=n}^{\infty} e^{-(\lambda_i+\Lambda_j)T_j} \frac{[(\lambda_i+\Lambda_j)T_j]^k}{k!} \\
&\times \left(\frac{\lambda_i}{\lambda_i+\Lambda_j}\right)^n \left(\frac{\Lambda_j}{\lambda_i+\Lambda_j}\right)^{k-n} \binom{k}{n} \left[1-\pi_0(k-n, \mathbf{b}^{(i,j)}(q_i))\right] \Bigg\} \qquad (56)
\end{aligned}
$$

where $\boldsymbol{\pi}(m, \mathbf{b}^{(i,j)}(q_i)) = \boldsymbol{\pi}(m-1, \mathbf{b}^{(i,j)}(q_i))\mathbf{W}^{(j)}$ and $\boldsymbol{\pi}(0, \mathbf{b}^{(i,j)}(q_i)) = \mathbf{b}_{q_i}^{(i,j)}$. Recall that ϕ_0 may be calculated in a numerically stable manner as illustrated at the end of Section 4.1, while we recommend calculating $1-\pi_0$ as $\sum_{q=1}^{\infty} \pi_q$ to avoid subtractions. Equations (55) and (56) give the first numerator term of (49). It is easy to obtain a recursion for (56) following the same steps as those for $d_{s,s'}$, but which is much simpler.

5.2 $P_{\mathcal{R}}$ for Gated Service and Infinite Buffers

In order to calculate $P_{\mathcal{R}}$ for the case of gated service, we need to determine the various expectations in equation (49). As noted above, the calculation of quantities corresponding to the switchover intervals is the same as before. Namely, equation (50) for $E[A_s^{(j)}]$ and equation (54) for $E[V_s^{(j)}]$ remain valid, independent of the service disciplines at the queues. The quantities $E[U_s^{(j)}]$ and $E[B_s^{(j)}]$ associated with the mini-cycles do depend on the discipline, since the mini-cycle length differs according to the type of service at the queue. However, the formula for $E[B_s^{(j)}]$ is essentially the same as equation (52), except that the uniformized Markov chain used to determine $\boldsymbol{\pi}$ is obtained from the two-dimensional chain of Figure 3, instead of the birth-death chain of Figure 1. The final quantity to calculate for gated service is $E[U_s^{(j)}]$. When $j \neq i$, i.e. the queue length of interest does not correspond to the queue being served in the j-mini-cycle, equation (56) is used as before, except that $\boldsymbol{\pi}$ is obtained as described above from the two-dimensional chain. Thus it only remains to determine $E[U_s^{(j)}]$ for $j = i$. In order to calculate this expectation, it is necessary to determine the proportion of time during a mini-cycle when a particular queue length of the served queue occurred.

To find the expected amount of time that the served queue contains $h_i > 0$ customers during the mini-cycle given an initial state (q_i, q_i), all states of the form (h_i, v), where $v = 1, \ldots, \min(h_i, q_i)$, must be included. The absorbing states $(u, 0)$ represent the only cases for which the mini-cycle length $\tau_i < T_i$, and they do not contribute to this expectation. Thus, as was the case for the one-dimensional chains considered in Sections 5.1-5.2, uniformization for a length T_i can be used to obtain the result of interest. Specifically, proceeding as in the derivation of equation (55), we first have for the initial state (q_i, q_i)

$$E[U^{(i)} \,|\, (q_i, q_i)] = \sum_{n=0}^{\infty} e^{-\Lambda_i T_i} \frac{(\Lambda_i T_i)^n}{n!} \left\{ \frac{T_i \sum_{m=0}^{n} \sum_{v=1}^{\min(h_i, q_i)} \pi_{(h_i, v)}(m, (q_i, q_i))}{n+1} \right\},$$

where $\pi(m, (q_i, q_i)) = \boldsymbol{\pi}(m-1, (q_i, q_i))\mathbf{W}^{(i)}$ and $\boldsymbol{\pi}(0, (q_i, q_i)) = \mathbf{e}_{(q_i, q_i)}$. Starting the uniformization with the equilibrium queue length distribution $\mathbf{b}^{(i)}$, where the qth entry corresponds to the probability that the initial state is (q, q), yields

$$\sum_{s \in \mathcal{S}} E[U_s^{(i)}] \beta_s^{(i)} = T_i \sum_{n=0}^{\infty} e^{-\Lambda_i T_i} \frac{(\Lambda_i T_i)^n}{n!} \left\{ \frac{\sum_{m=0}^{n} \sum_{v=1}^{h_i} \pi_{(h_i, v)}(m, \mathbf{b}^{(i)})}{n+1} \right\}, \qquad (57)$$

where $\boldsymbol{\pi}(m, \mathbf{b}^{(i)}) = \boldsymbol{\pi}(m-1, \mathbf{b}^{(i)})\mathbf{W}^{(i)}$ and $\pi_{(q,q)}(0, \mathbf{b}^{(i)}) = \mathbf{b}_q^{(i)}$ (so that $\pi_{(u,v)}(0, \mathbf{b}^{(i)}) = 0$ for $u \neq v$).

6 Extensions to the Basic Model

In this section we show how different extensions to the basic model considered above can be easily handled using our method, such as general polling tables and additional service disciplines. Another interesting model that can be solved is where a customer is not allowed to start service if the timeout interval is about to expire, since it is very likely that such a customer will be unable to finish service.

For example, a system with a general polling table can be handled as in [11] by relabeling the queues in the order given by the switching policy and proceeding as for the cyclic case. Time average measures for queues that are visited more than once during the polling cycle (i.e. that appear several times in the polling table) can be easily obtained, since these quantities are calculated using expectations of random variables over mini-cycles and switchover intervals.

Additional service disciplines can also be handled using the method developed in this paper. Specifically, we briefly describe the changes necessary to analyze the E-limited and G-limited cases [14]. The chain for the E-limited case (with a limit K_j served during a j-mini-cycle) is more complicated to obtain than simply by truncating the exhaustive chain. The state for $\mathcal{W}^{(j)}$ in this case must count the number in the system *and* the number of departures, since either quantity may cause the mini-cycle to end before the timeout expires. Thus $\mathcal{W}^{(j)}$ is a two-dimensional Markov chain with states (n, m), where n is the number in system for queue j, and m is the number of departures from queue j since the start of the j-mini-cycle. The state transition rate diagram with $K_j = 3$ is given in Figure 4.

States of the form $(0, l)$ correspond to queue j being empty and are absorbing states. However, since the number of departures from queue j is limited to at most K_j, states of the form (l, K_j) are also absorbing states, i.e. the server leaves when such a state is encountered. As before, uniformization for a length T_j can be used to find the distribution of the j-mini-cycle length and, in the case when the timeout expires first, the probability distribution for the number in queue j at time T_j. The uniformization rate is $\Lambda_j = \lambda_j + \mu_j$ as in the exhaustive case.

There is an additional complication that occurs in this case, because the j-mini-cycle can end prior to the timeout without queue j going empty This happens in the E-limited case when the limit K_j on the number of customers served is reached. To account for this possibility, we divide the cases for computing transition probabilities differently than before. We distinguish the following cases, which are disjoint and mutually exclusive.

(a) The timeout expires and causes the mini-cycle to end.

(b) Queue j becomes empty and causes the mini-cycle to end.

(c) The number of departures from queue j reaches K_j and causes the mini-cycle to end.

In case (a) we can use equation (9) with the appropriate uniformized Markov chain and

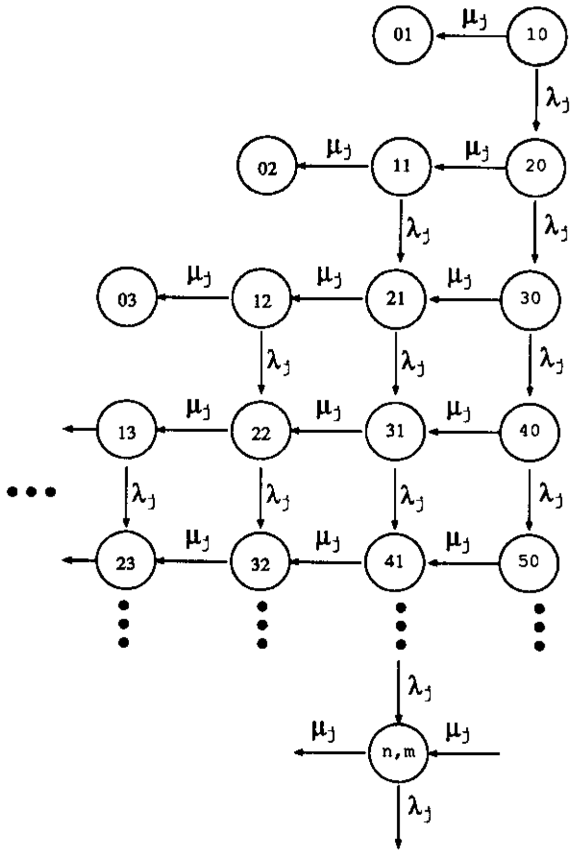

Figure 4: E-limited service.

with $\pi_{q'_j}(n-\sum_{i\neq j} k_i)$ replaced by $\sum_{l=0}^{K_j} \pi_{q'_j,l}(n-\sum_{i\neq j} k_i)$. Here the notation $\pi_{q'_j,l}$ refers to the state (q'_j, l) in the two-dimensional Markov chain.

For case (b) we can use equation (10) with $\phi_0(m-\sum_{i\neq j} k_i)$ replaced by the finite sum $\sum_{l=0}^{K_j} \pi_{0,l}(m-\sum_{i\neq j} k_i)$, since there are multiple absorbing states with $q'_j = 0$.

In case (c), K_j customers were served in the j-mini-cycle. For the transition to occur, $q'_j \geq (q_j - K_j)^+$ and the Markov chain describing the j-mini-cycle must reach the absorbing state (q'_j, K_j). To compute the transition probability in this case we use equation (10) with the Markov chain associated with E-limited service and with the reference to the absorbing state 0 taken to refer to state (q'_j, K_j).

This completes the development of the basic equations necessary to compute the transition matrix $\mathbf{D}^{(j)}$ for the E-limited scheduling discipline. As before, the equilibrium vectors $\beta^{(i)}$ and $\alpha^{(i)}$ at server arrivals to queue i and server departures from queue i can now be found similar to the procedure developed for the exhaustive and gated disciplines.

The chain $\mathcal{W}^{(j)}$ for the G-limited case (with a limit of K_j customers served within a single j-mini-cycle) is identical to the two-dimensional chain for the gated case, except that it is truncated at states for which the original customers in the queue satisfy $(q_j - K_j)^+ = \max\{q_j - K_j, 0\}$. These states are made absorbing states.

6.1 Reducing the Timeout Interval

If a departure occurs from the served queue j when the timeout interval has almost expired, then the next customer (if any) will probably not complete service before T_j. That is, this next customer probably either will be returned to the waiting line of customers in the preemptive case or will create an overrun in the nonpreemptive case. Thus it may be advantageous for the server to immediately begin switching to the next queue near the end of the timeout period instead of accepting an additional customer into service. These considerations give rise to the following extension of the basic time-limited polling system. We assume that there is a constant ω_j for queue j, $j = 1, \ldots, M$, such that if the timeout interval has exceeded $T_j - \omega_j$ but has not yet expired by reaching T_j, then no new customer will be allowed to enter service. Thus, if any customer finishes in the interval $(T_j - \omega_j, T_j)$, the server leaves queue j and begins to switch to the next queue. Note that setting the parameter $\omega_j = 0$ reduces to the usual time-limited system studied in previous sections. We now show that this model can be easily analyzed using the techniques developed above.

We first claim that this extension provides no additional generality in the nonpreemptive timeout case. To see this, consider a j-mini-cycle and note that any customer in service at time $T_j - \omega_j$ will be served to completion in this new system. Furthermore, no additional customers will be accepted into service. That is, if the system is not empty at $T_j - \omega_j$, the customer being served at that time either will remain in service at T_j creating an overrun, or will leave in the interval $(T_j - \omega_j, T_j)$. We may thus consider the remaining service time of this customer after $T_j - \omega_j$ simply as an "overrun," during which arrivals to all queues continue to occur. Thus the behavior of the new system is seen to be equivalent to one with nonpreemptive timeouts and with a timeout period of length $T_j - \omega_j$.

We next consider the preemptive timeout case, and indeed obtain a more general system than those studied above. As before, suppose there is a customer in service at time $T_j - \omega_j$ in the served queue j. Then either the customer departs before time T_j and the server begins switching to the next queue, or the customer does not finish service before the timeout expires and is preempted back to the waiting line at time T_j. Of course, arrivals to all of the queues continue during the service time of this customer. Thus the system is similar to one with nonpreemptive timeouts and an "overrun," the length of which is independent of the state at $T_j - \omega_j$ and is given by an exponential random variable of mean μ_j truncated at ω_j. That is, given that the served queue j is not empty at $T_j - \omega_j$ during a j-mini-cycle, the "overrun" length χ_j has distribution

$$P[\chi_j \leq t] = \begin{cases} 1 - e^{-\mu_j t} & t < \omega_j \\ 1 & t = \omega_j. \end{cases}$$

However, this system differs from the nonpreemptive timeout case, since the customer in service at $T_j - \omega_j$ only leaves the system if $\chi_j < \omega_j$ and is returned to the waiting line if $\chi_j = \omega_j$.

To determine the transition probabilities for this new model, we first use uniformization on the chain $\mathcal{W}^{(j)}$ for an interval of length $T_j - \omega_j$. If the served queue j empties before time $T_j - \omega_j$, then we proceed as before to obtain the transition probabilities. Now suppose that queue j is not empty at time $T_j - \omega_j$, and let $q_j^* > 0$ denote the corresponding state. Arrivals to all M queues during the "overrun" must be added as in the nonpreemptive timeout case. However, the final state of the served queue j depends on whether or not the "overrun" was of length ω_j (i.e. whether or not the timeout T_j expired). Let q_i' be the state of queue i at the end of the j-mini-cycle, $i = 1, \ldots, M$. For the nonserved queues $i \neq j$, there must be $q_i' - q_i^*$ arrivals after $T_j - \omega_j$. For the served queue, if $\chi_j < \omega_j$, then $q_j' - q_j^* + 1$ arrivals must occur after $T_j - \omega_j$, while if $\chi_j = \omega_j$, then $q_j' - q_j^*$ customers must arrive during the "overrun."

Define $l_i = q_i' - q_i^*$ for $i = 1, \ldots, M$. Let $e^{(j)}_{s^*,s',0}$ and $e^{(j)}_{s^*,s',1}$ be the probability of a transition from state s^* at time $T_j - \omega_j$ to state s' at the end of the j-mini-cycle when $\chi_j < \omega_j$ and $\chi_j = \omega_j$, respectively. Then we have

$$e^{(j)}_{s^*,s',0} = \left(\frac{\mu_j}{\gamma_j}\right)\left(\frac{\lambda_j}{\gamma_j}\right)^{l_j+1} \prod_{i \neq j} \left(\frac{\lambda_i}{\gamma_j}\right)^{l_i} \frac{(l+1)!}{(l_j+1)! \prod_{i \neq j} l_i!} E_{l+1,\gamma_j}(\omega_j),$$

where $\gamma_j = \sum_{i=1}^{M} \lambda_i + \mu_j$, $l = \sum_{i=1}^{M} l_i$, and $E_{l,\gamma_j}(t)$ is the $(l+1)$-stage Erlangian distribution. We also have

$$e^{(j)}_{s^*,s',1} = e^{-\gamma_j \omega_j} \prod_{i=1}^{M} \frac{(\lambda_i \omega_j)^{l_i}}{l_i!}.$$

We now continue similar to the nonpreemptive timeout case. Consider a system with exhaustive service and infinite buffers. If $q_j' > 0$, then the queue could not have emptied before time $T_j - \omega_j$, i.e. $q_j^* > 0$. Therefore, the transition probability $f^{(j)}_{s,s'}$ from the beginning of the j-mini-cycle to its end is for $q_j' > 0$

$$f^{(j)}_{s,s'} = \sum_{s^*:q_j^*>0} d^{(j)}_{s,s^*} e^{(j)}_{s^*,s'},$$

where $d^{(j)}_{s,s^*}$ is given by equation (9) with T_j replaced by $T_j - \omega_j$ and $e^{(j)}_{s^*,s'} = e^{(j)}_{s^*,s',0} + e^{(j)}_{s^*,s',1}$. When $q_j' = 0$, then either queue j became empty before $T_j - \omega_j$ or $q_j^* = 1$ and the customer in service at $T_j - \omega_j$ finished before T_j with no arrivals during the "overrun." Thus for $q'_j = 0$ we have

$$f^{(j)}_{s,s'} = d^{(j)}_{s,s'} + \sum_{s^*:q^*_j=1} d^{(j)}_{s,s^*} e^{(j)}_{s^*,s',0}.$$

Similar arguments apply to the gated discipline and to the finite buffer case.

7 Computational Requirements

In this section we discuss the major computational costs involved in calculating the expressions for the various measures of interest developed in previous sections. We first consider in detail the calculation of the transition probability matrices $\mathbf{D}^{(j)}$ and the equilibrium probability vectors $\beta^{(i)}$ and $\alpha^{(i)}$ for a system with M queues served in cyclic order with exhaustive service at each individual queue, preemptive timeouts and infinite buffer size. We then consider the similar case with finite buffers. Finally we comment on the cost of calculating the time average probabilities.

Recall that the entries $d^{(j)}_{s,s'}$ of $\mathbf{D}^{(j)}$ are determined in equations (9) and (10). Given $\epsilon > 0$, the infinite series in the equations can be truncated at $N = N(\epsilon)$ to obtain results that are within that given tolerance. The value of N is proportional to $\gamma_j T_j$. Since the service rates and arrival rates are of the same order of magnitude, the problem is not "stiff." Therefore, N is proportional to the number of messages that arrive and are served in a mini-cycle, which is in general not large.

The recursions to calculate the entries of $\mathbf{D}^{(j)}$ are given in equations (24), (25), (28), and (29) (for $j = 1$). The recursion for $\Omega[k_2, \ldots, k_M; n]$ is illustrated in Figure 5. In that figure each cell consists of a vector of values $\Omega[k_2, \ldots, k_M; n]$ such that $\sum_{i=2}^{M} k_i = \kappa_1$. The arrows indicate the previous values needed in the recursion. For example, each cell in column $\kappa_1 = l$ contains $\binom{l+M-2}{M-2}$ elements. Note, however, that only a row (or column) of elements needs to be stored to calculate the values needed for $d^{(1)}_{s,s'}$. Further, there are a total of $\sum_{l=0}^{N} \binom{l+M-2}{M-2} \approx (N+1)^{M-1}$ elements in the Nth row (the largest row). In order to calculate $\Upsilon[k_2, \ldots, k_M; r, q_1, q'_1]$, the values of $\psi_1(r)$ and $\pi_{q'_1}(r - \kappa_1, q_1)$ are required. The cost of the recursion for $\psi_1(r)$ is negligible compared to the recursion to calculate $\Omega[k_2, \ldots, k_M; n]$. The cost to calculate $\pi_{q'_1}(r - \kappa_1, q_1)$ is also negligible, since it only involves the multiplication of a vector of dimension N by $\mathbf{W}^{(1)}$, the uniformized transition rate matrix corresponding to a one-dimensional birth-death chain in the exhaustive case. The recursion to calculate $\Upsilon[k_2, \ldots, k_M; r, q_1, q'_1]$ is simple and should be done in parallel with the recursion to calculate Ω. In more detail, as each value of Ω in a cell is obtained, a value of Υ for that cell can be calculated. Note that an element in a cell of values of Υ needs only the corresponding Υ value in the same column and previous row. These comments also apply to the recursion

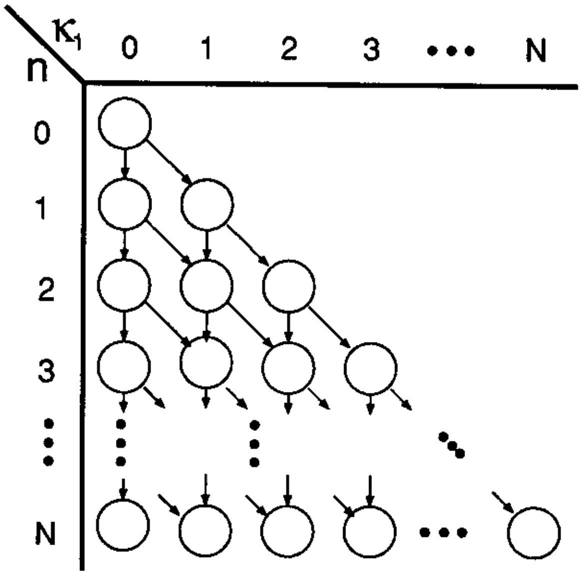

Figure 5: Recursion for Ω.

used to obtain Γ and Θ. In summary, the total storage necessary to implement the recursions needed to obtain all entries of the matrix $\mathbf{D}^{(1)}$ is $O((N+1)^{M+1})$, and therefore a total storage of $O(M(N+1)^{M+1})$ is required for all the M matrices $\mathbf{D}^{(j)}$, assuming that the values of $N(\epsilon)$ for each mini-cycle are approximately equal.

The number of operations (floating point multiplications and divisions) needed to carry out the recursions to obtain the entries in the matrix $\mathbf{D}^{(1)}$ is seen to be $O(M(N+1)^{M+1})$. The calculation of the stationary probability vectors $\beta^{(i)}$ and $\alpha^{(i)}$ is done iteratively. Each iteration requires a vector matrix multiplication for each mini-cycle, where the vector has dimension $(N+1)^M$. Even in the infinite buffer case the state space, and thus the length of $\beta^{(i)}$ and $\alpha^{(i)}$, is truncated for a given error tolerance.

For queues with limited buffer space, the recursions given by equations (31), (32), (37) and (38) are used. These recursions are similar to those for the infinite buffer case, and Figure 5 may also be used to illustrate the recursion for Φ, where κ_1 is replaced by a. When all buffer sizes are finite and equal to B, a cell in column a has at most $(B+1)^{M-1} - B^{M-1}$ elements for $a \geq B(M-1)$, or $\binom{a+M-2}{M-2}$ for small values of a ($a \leq B$). Similar to the infinite buffer case, only one column (or row) of elements needs to be stored. (Note that only a cell of Υ values needs to be stored if recursion by column is used.) A total storage of $O((B+1)^{M+1})$ is necessary to obtain $\mathbf{D}^{(1)}$, while $O(M(B+1)^{M+1})$ is needed for all the M matrices $\mathbf{D}^{(j)}$.

It is important to stress that the storage necessary to carry out the recursions needed for all the entries in the matrices $\mathbf{D}^{(j)}$ is significantly less than the total number $(B+1)^{2M}/2$ of nonzero entries in $\mathbf{D}^{(j)}$, since many of these entries are identical. Furthermore, the procedure used to calculate $\beta^{(i)}$ and $\alpha^{(i)}$ is to iterate from mini-cycle to mini-cycle until convergence is achieved. Note that such an iteration scheme takes advantage of the structure of the matrices $\mathbf{D}^{(j)}$, $\mathbf{C}^{(j)}$ in terms of storage. If $\beta^{(i)} = \beta^{(i)}\mathbf{H}^{(i)}$ or $\alpha^{(i)} = \alpha^{(i)}\mathbf{G}^{(i)}$ is solved directly instead, the total storage needed would be $(B+1)^{2M}$, since $\mathbf{H}^{(i)}$ and $\mathbf{G}^{(i)}$ do not necessarily have special structure. Note also that all recursions have a probabilistic interpretation and involve only additions and multiplications, and so they are numerically stable.

Once the vectors $\beta^{(i)}$ and $\alpha^{(i)}$ have been obtained, the additional computational requirements to calculate the time average probabilities $P_{\mathcal{R}}$ are minor. For simplicity, again consider the case of infinite buffers, preemptive timeouts, and exhaustive service discipline. From equation (52) (and related equations) observe that the expectations needed to compute $P_{\mathcal{R}}$ depend mainly on the vectors $\boldsymbol{\pi}(n)$, and these have already been calculated in the recursions for $\mathbf{D}^{(j)}$ described above. Once the $\boldsymbol{\pi}(n)$ are available, it is easy to see that the computational requirements needed to calculate $P_{\mathcal{R}}$ are $O(MN^2)$.

8 Examples

In this section we present simple examples to illustrate the applicability of the method we developed in previous sections. The first example is the model of the so called (T_1-T_2) scheme for multiplexing voice and data [23]. In this scheme, voice packets are served until their queue is exhausted or until a maximum service timeout of T_1 time units is reached. Data packets are served in a similar manner, with a timeout of T_2 units. Note that the capacity C of the channel is allocated dynamically between the two sources of traffic, but a minimum of $[T_i/(T_1+T_2)]C$ is guaranteed for voice and data ($i = 1,2$), respectively. In this example the capacity of the channel is assumed to be 1.5 Mbps, the voice load is assumed to be 60% of the capacity and the data load is allowed to vary. Voice and data packets have an average size of 600 bits and 400 bits, respectively, In Figure 6 we plot the average delay of data and voice packets when the average data load varies from 10% to 60% of the total channel capacity for three different buffer sizes: 10, 20 and 30 packets. The maximum amount of time allocated for voice and data packets is 8 msec and 2 msec, respectively, and we assume that the switchover times are equal to 0.1 msec. Therefore, voice packets are guaranteed 80% of the bandwidth, while data packets can obtain at most 20% of the channel capacity. As is evident from Figure 6, the increase in data traffic has little effect on the expected delay of voice packets (represented by the three curves on the bottom of the figure), even when the total offered load is above the maximum channel capacity.

Figure 7 is similar to Figure 6, but in this case the maximum amount of time allocated for data packets is increased to 4 msec, and so voice packets have 66% of the capacity while data packets have 33%. In this case, the increase in data load has a bigger effect on the delay

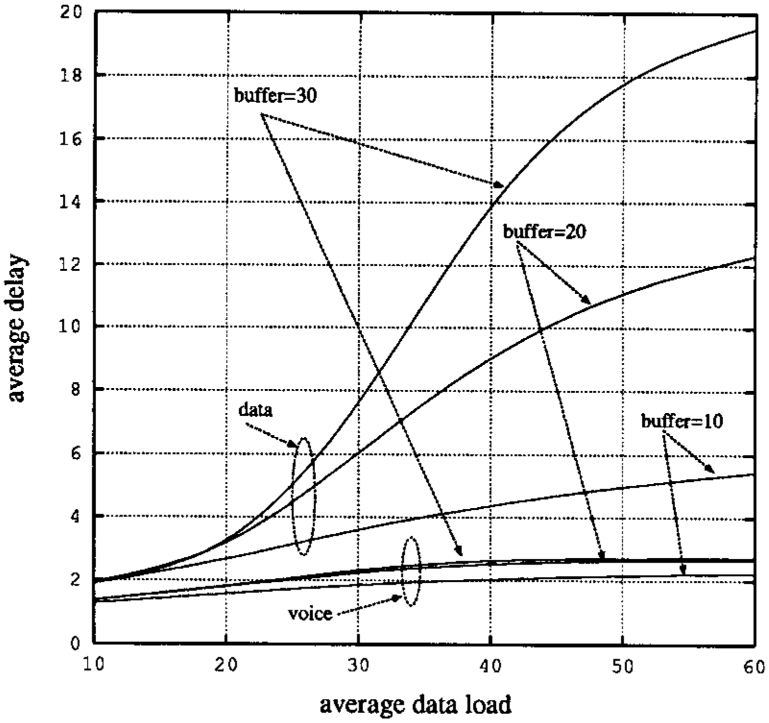

Figure 6: Effect of data load increase: $T_{\text{voice}} = 8$, $T_{\text{data}} = 2$.

of voice packets, because they are allowed to use a higher percentage of the bandwidth. In Figure 8 we plot the average delay for voice and data packets packets both for (T_1-T_2) equal to (8-2) and (8-4), to highlight the effect of bandwidth allocation. The buffer size is assumed to be 30 packets.

Finally, in Figure 9, we plot the blocking probabilities for voice packets versus the load of data packets for buffer sizes of 10, 20, 30 and (T_1-T_2) equal to (8-2) and (8-4) (the average voice load remains at 60% of the capacity). The effect of buffer size increase on the probability of blocking voice packets can be observed in the figure, as well as the effect of increasing the timeout limit for data packets. For instance, when the average data load is at 20% of the channel capacity (the total load is then 80% of the capacity), the blocking probability increases four times (approximately from 4.0810^{-5} to 16.510^{-5}) when the timeout value for data packets increases from 2 to 4. Clearly, as the data load decreases, the timeout value for data packets has a smaller effect on the blocking probability for voice packets, and this effect is nearly negligible when the data load is 10.

The timeout parameter for each queue sets a limit on the maximum bandwidth that can be allocated to a traffic source. Queues with larger values for this parameter may obtain a higher percentage of the channel capacity when needed and so have higher “priority,” while “lower priority” traffic is still guaranteed a minimum bandwidth. It is clear that varying the

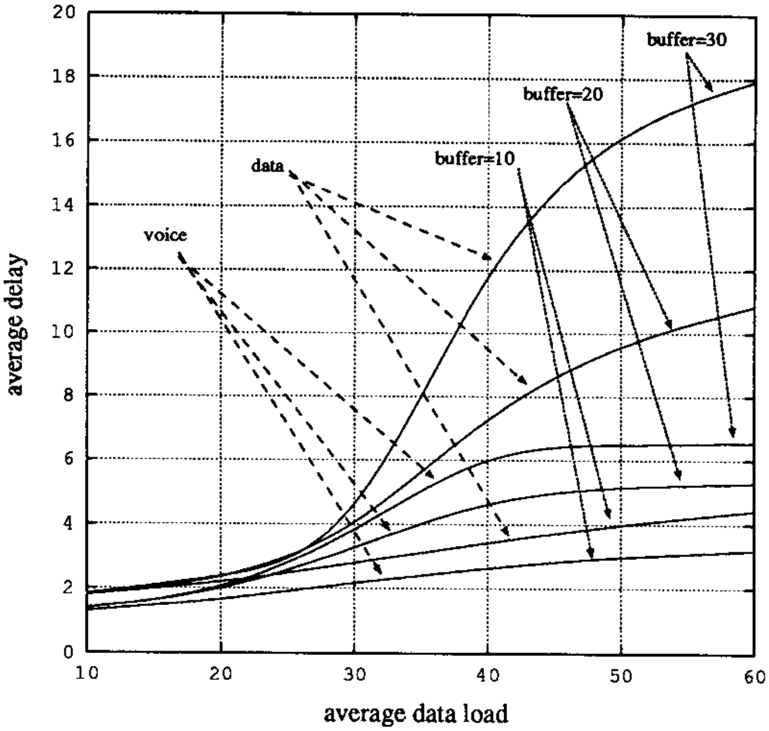

Figure 7: Effect of data load increase: $T_{\text{voice}} = 8$, $T_{\text{data}} = 4$.

timeout values has the effect of varying the relative priorities assigned to each traffic source, and so there is a wide variety of choices for the parameters. In the second example we show the effect of varying the timeout parameters in a system with three queues when the capacity of the channel is 2 Mbps. The three sources are identical, i.e. the average packet lengths are equal to 1000 bits and the load is the same. In Figure 10 we plot the average waiting time for packets of each of the three sources when the combined load of all sources vary from 30% to 90% of the total channel capacity for three sets of timeout parameters: 6-3-2, 7-2-2 and 4-4-3. The buffer size is assumed to be 10, and the switchover times are equal to 0.01. Comparing the first and third sets of curves, we see that the the expected delays of the third set are less spread than those of the first one. The second queue is practically not affected when we change the "priority" from 4-4-3 to 6-3-2, while the first queue is given a higher fraction of the bandwidth at the expense of the third queue.

Finally, an interesting question to ask is what effect the values for the timeout parameters have on performance measures when the percentage of the channel capacity allocated to each source is maintained constant. Figures 11.a, 11.b and 11.c show this effect. The parameters for this system are the same as for the first example (with buffer size equal to 20), but the timeout values vary according to $4t$-$2t$ for $t = 1, \ldots, 4$. We first note that in Figure 11.a (for which the data load is 30% of the capacity), increasing the timeout values favors the

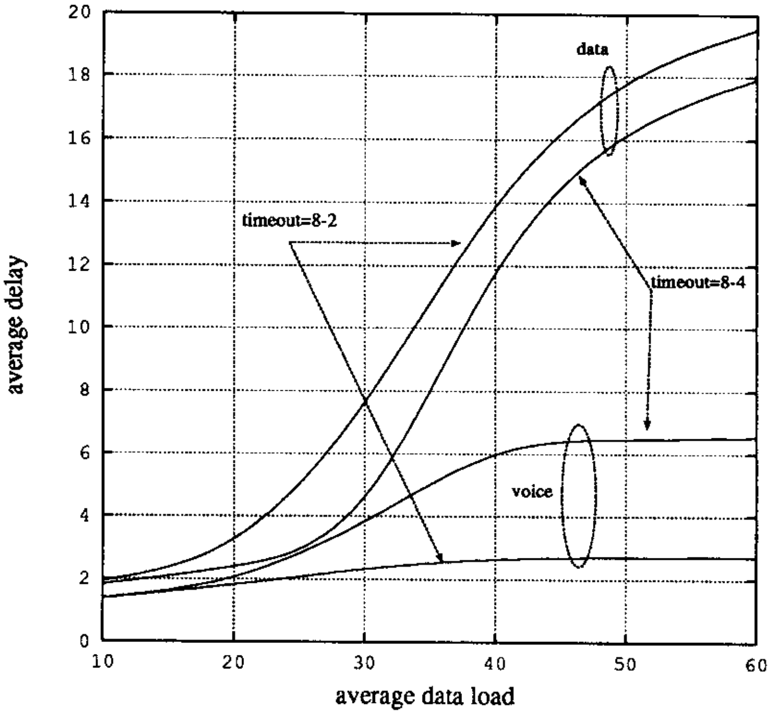

Figure 8: Effect of varying the maximum time allocated to data packets.

voice packets more than the data packets. This is somehow expected, since the voice load is 60% of the channel capacity. What it is not clear is whether the delay of data packets may improve as well. It is interesting to note that the delay of data packets has a minimum around timeout values of 8-4.

As the load of data packets increases, the proportional increase in the timeout values begins to favor the data packets more than the voice packets. This is shown in Figure 11.b. Finally, in Figure 11.c, we see that the load of data packets is sufficiently high so that an increase in the timeout values has a negative effect on the voice packets, even if the maximum fraction of capacity allocated to the sources remains constant. Note that the voice packet delay curve of Figure 11.c has a minimum.

Two observations should be made to help in understanding Figures 11.a through 11.c. First, note that the influence of the switchover time increases as the timeout values decrease. Second, as $t \rightarrow \infty$ the system tends to an exhaustive service discipline.

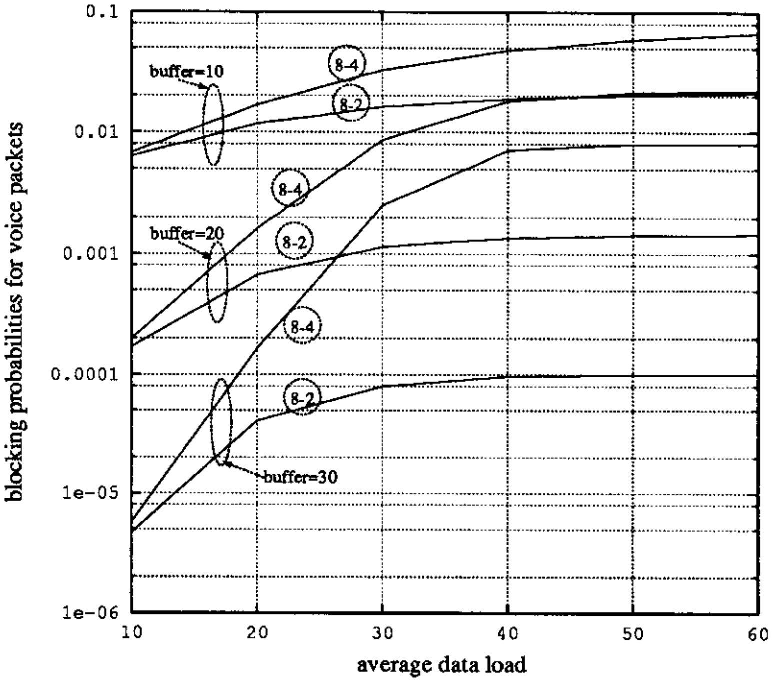

Figure 9: Blocking probabilities.

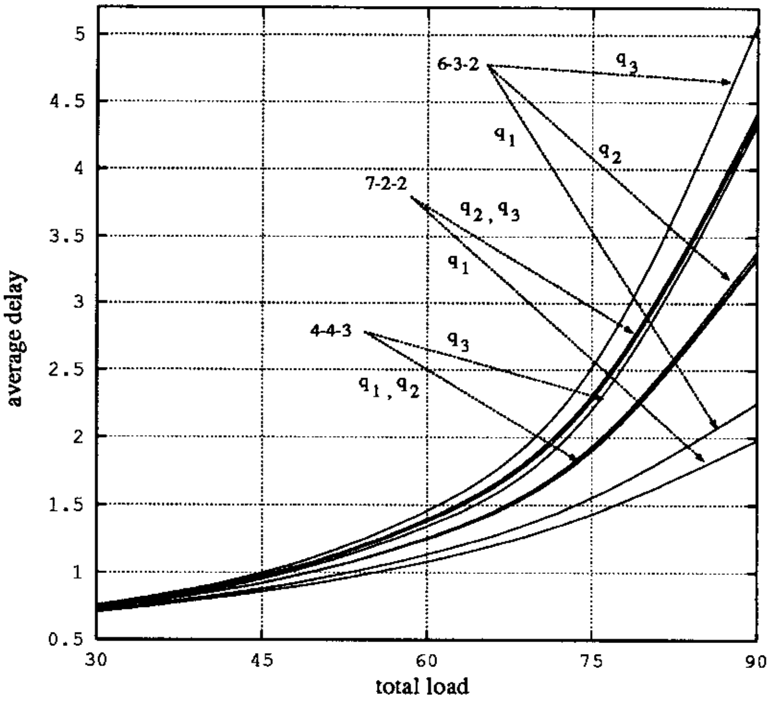

Figure 10: Setting "priorities" from the timeout parameters.

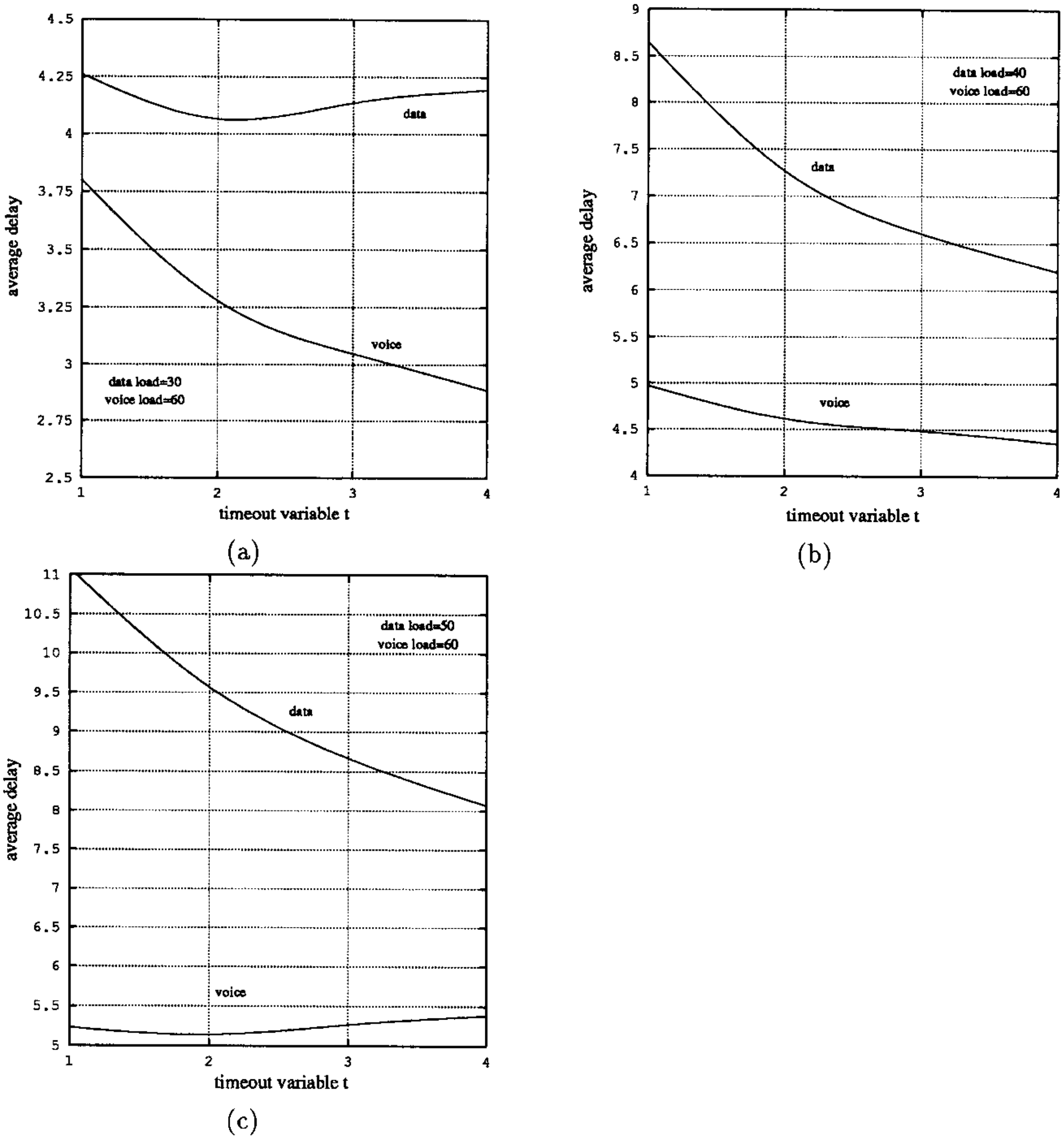

Figure 11: Effect of varying the timeout values while preserving the percentage of allocated channel capacity.

9 Conclusions

Time-limited polling systems have become increasingly important, especially in the area of high speed networking. We have presented a solution procedure for obtaining queue length distributions and related performance measures for a general class of time-limited polling systems. When the time limit is not exponential, the state evolution is non-Markovian, and these models have resisted a closed form solution. In this paper we have developed efficient numerical algorithms for solving these models based on the method of embedded Markov chains.

The uniformization technique is used to perform transient analysis of the evolution of the system between consecutive epochs of the embedded chain. Uniformization is used in this manner to calculate transition probabilities, from which the stationary state probabilities for the embedded Markov chain are obtained. Computational procedures are also developed to calculate the time in a set of states between epochs, conditioned on the starting state. Combined with the stationary state probabilities of the embedded Markov chain, overall performance measures can be computed. In the case of the polling models treated in this paper, it is also shown that additional computational savings can be obtained by taking advantage of the special structure of the model. Examples are given which illustrate the application of these methods to high speed communication switches and bandwidth allocation strategies.

The results here can also be viewed as illustrating a general technique for computing performance measures for models that can be solved via the embedded Markov chain method, but for which the transition probabilities and performance measures between embedded points are difficult to obtain in closed form.

Acknowledgments

We wish to thank O.J. Boxma, K. C. Chang, S. W. Fuhrmann and L. Georgiadis for useful discussions. We also thank Paulo Alcantara Saraiva Leão for his comments and for implementing the algorithm.

References

[1] D. Bertsekas and R. Gallager. *Data Networks.* Prentice Hall, New Jersey, 2nd edition, 1992.

[2] O.J. Boxma. Analysis and optimization of polling systems. Technical report, CWI, report BS-R9102, 1991.

[3] O.J. Boxma and W.P. Groenendijk. Two queues with alternating service and switching times. In *Queueing Theory and its Applications - Liber Amicorum for J.W. Cohen*, pages 261–282. North-Holland, 1988.

[4] E.G. Coffman, Jr., G. Fayolle, and I. Mitrani. Two queues with alternating service periods. In *Proc. Performance '87*, pages 227–239, 1987.

[5] J.W. Cohen and O.J. Boxma. The M/G/1 queue with alternating service formulated as a Riemann-Hilbert problem. In *Proc. Performance '81*, pages 181–199, 1981.

[6] H.A. David. *Order Statistics, 2nd Ed.* John Wiley & Sons, 1981.

[7] E. de Souza e Silva and H.R. Gail. Calculating availability and performability measures of repairable computer systems using randomization. *Journal of the ACM*, 36(1):171–193, 1989.

[8] E. de Souza e Silva and H.R. Gail. Analyzing scheduled maintenance policies for repairable computer systems. *IEEE Trans. on Communications*, 39(11):1309–1324, 1990.

[9] E. de Souza e Silva and H.R. Gail. Performability analysis of computer systems: from model specification to solution. *Performance Evaluation*, 14:157–196, 1992.

[10] E. de Souza e Silva and H.R. Gail. The uniformization method in performability analysis. In *Proceedings of the 2nd International Workshop on Performability Analysis*, 1993.

[11] M. Eisenberg. Queues with periodic service and changeover time. *Operations Research*, 20(2):440–451, 1972.

[12] C. Fricker and M.R. Jaïbi. Monotonicity and stability of periodic polling models. Technical report, Tilburg University, report FEW 559, 1992.

[13] S.W. Fuhrmann. Performance analysis of a class of cyclic schedules. Technical report, Bell Laboratories, report TM 81-59531-1, 1981.

[14] S.W. Fuhrmann and Y.T. Wang. Analysis of cyclic service systems with limited service: Bounds and approximations. *Performance Evaluation*, 9(1):35–54, 1988.

[15] L. Georgiadis and W. Szpankowski. Stability criteria for yet another multidimensional distributed system. Technical report, Purdue University, report CSD-TR-91-071, 1991.

[16] W.K. Grassmann. Means and variances of time averages in Markovian environments. *European Journal of Operational Research*, 31:132–139, 1987.

[17] W.K. Grassmann. Finding transient solutions in Markovian event systems through randomization. In *Numerical Solution of Markov Chains*, pages 357–371. Marcel Dekker, Inc., 1991.

[18] D. Gross and D.R. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32(2):343–361, 1984.

[19] D.P. Heyman and M.J. Sobel. *Stochastic Models in Operations Research, Volume I.* McGraw-Hall, 1982.

[20] K.K. Leung and M. Eisenberg. A single-server queue with vacations and gated time-limited service. *IEEE Trans. on Communications*, 38(9):1454–1462, 1990.

[21] K.K. Leung and M. Eisenberg. A single-server queue with vacations and non-gated time-limited service. *Performance Evaluation*, 12(2):115–125, 1991.

[22] F.E. Ross. FDDI - a tutorial. *IEEE Communications Magazine*, 24(5):10–17, 1986.

[23] K. Sriram. Dynamic bandwidth allocation and congestion control schemes for voice and data multiplexing in wideband and packet technology. In *ICC-90*, pages 1003–1009, 1990.

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

[25] H. Takagi. Queueing analysis of polling models: an update. In *Stochastic Analysis of Computer and Communication Systems*, pages 267–318. North-Holland, 1990.

[26] H. Takagi. Application of polling models to computer networks. *Computer Networks and ISDN Systems*, pages 193–211, 1991.

[27] K.S. Watson. Performance evaluation of cyclic service strategies - a survey. In *Proc. Performance '84 and 1984 ACM SIGMETRICS Conf.*, pages 521–533, 1984.

[28] O.-C. Yue and C.A. Brooks. Performance of the timed token scheme in MAP. *IEEE Trans. on Communications*, 38(7):1006–1012, 1990.