Analysis of the adaptive MMAP[K]/PH[K]/1 queue: a multi-type queue with adaptive arrivals and general impatience

Benny Van Houdt

Department of Mathematics and Computer Science

University of Antwerp - IBBT

email: benny.vanhoudt@ua.ac.be

Abstract

In this paper we introduce the adaptive MMAP[K] arrival process and analyze the adaptive MMAP[K]/PH[K]/1 queue. In such a queueing system, customers of K different types with Markovian inter-arrival times and possibly correlated customer types, are fed to a single server queue that makes use of r thresholds. Service times are phase-type and depend on the type of customer in service. Type k customers are accepted with some probability $a_{i,k}$ if the current workload is between threshold $i-1$ and i. The manner in which the arrival process changes its state after generating a type k customer also depends on whether the customer is accepted or rejected.

The solution method exists in reducing the joint workload and arrival process to a fluid queue with r thresholds, the steady state of which is expressed using matrix analytic methods. The time and memory complexity of this approach is also shown to be linear in the number of thresholds, allowing us to study systems with thousands of thresholds.

Markovian multi-type queues with customer impatience form a subclass of the queues considered in this paper. A numerical method to determine the probability of abandonment and the waiting time distribution is provided if the patience distributions have finite support, while for general customer impatience numerical examples show that accurate approximate results can be obtained using a step-function approach. Numerical examples with adaptive sources that model certain types of admission and congestion control are also included.

1 Introduction

Traditional queueing systems have been widely used in the analysis of communication systems and a substantial part of the literature has focused on queues without admission or congestion control. That is, queues typically accept customers as long as there is *room* available and arrival processes typically do not adapt their arrival rate based on feedback provided by the queue. Exceptions include queues with customer impatience, as well as queues with workload dependent arrivals.

In this paper, we develop a fast numerical procedure to analyze a broad class of queueing systems that support both admission and congestion control (though for some very particular cases explicit results could be obtained as well). To this end, we introduce the adaptive MMAP[K] arrival process, which generalizes the MMAP[K] process introduced in [14]. An MMAP[K] process is driven by an underlying Markov chain that generates customers of K types, thus, inter-arrival times are correlated as are the types of consecutive customers. Matching and fitting algorithms to represent workloads as MMAP[K] processes were proposed in [15, 3]. The adaptive MMAP[K] process will differ from an ordinary MMAP[K] as the evolution of the state of the underlying Markov chain depends on whether a (type k) customer is accepted or rejected.

We introduce and analyze the adaptive MMAP[K]/PH[K]/1 queue, where the service times follow a phase-type distribution that depends on the customer type (i.e., we have correlated service and inter-arrival times). The queue also makes use of a set of r thresholds d_1 to d_r such that $0 = d_0 < d_1 < \ldots < d_r < d_{r+1} = \infty$. If the workload belongs to the interval $(d_{i-1}, d_i]$ upon arrival of a type k customer, it is accepted with some probability $a_{i,k}$, for $i = 1, \ldots, r+1$ (and with probability $a_{0,k}$ if the workload is zero).

If the probabilities $a_{i,k}$ do not increase in i for any choice of k and the arrival process is not adaptive, this queue is equivalent to an MMAP[K]/PH[K]/1 queue with customer impatience, where the impatience distribution is a general distribution with finite support. Discrete-time queues with Markovian arrivals and general customer impatience have been analyzed in [20, 21]. In continuous time, the problem is considerably harder and results have only appeared for an exponential amount of patience [2, 5] or a deterministic amount [4]. We note that the arrival process in [2] is an adaptive Poisson process, which

is a special case of the adaptive MMAP[K] process (see Section 2).

The per-type accept and reject rates as well as the waiting time distribution is obtained via the steady state of the joint workload and arrival process. This process is a jump process as in [8], except that the occurrence and jump sizes now also depend on the current workload. The steady state distribution of this jump process is obtained by constructing a fluid queue with r thresholds and by applying a censoring argument. The idea behind this construction was also applied in a discrete-time setting in [19] and in continuous time in [8]. To obtain the steady state distribution of the constructed fluid queue, we can directly apply the theorems established in [7] for a general fluid queue with r thresholds (except for some minor adaptations regarding the boundary behavior). The main advantage of this approach, compared to existing eigenvalue approaches (e.g., [16]), is that the stationary distribution is expressed in terms of some boundary vectors and a set of exponentials of stable matrices (i.e., all their eigenvalues have non positive real parts), which avoids numerical instability.

An important observation made in this paper is that the time and memory complexity of the linear system that one needs to solve to obtain the boundary vectors of [7] can be made linear in the number of thresholds r. This allows us to solve systems with thousands of thresholds. These latter systems are for instance useful whenever we upper or lower bound a continuous impatience distribution by means of a step function, in order to approximate the probability of abandonment or the waiting time distribution.

It should be clear that the class of queues considered in this paper has many applications. Section 5 discusses some examples in a communication network setting, other examples can be found in health care. For instance, consider an emergency unit where patients are partitioned into K different classes based on their condition. A patient waiting in the waiting room may decide to forgo the service because he/she does not wish to wait any longer, which may result in a revenue loss. Further, the amount of patience that a patient has may depend on his/her condition and may change over time. For instance, given that a patient already waited a long time, he/she might be more willing to wait another 30 minutes. As such the hazard rate of the patience distribution should increase over time, while the hazard rate of the exponential distribution remains fixed over time. In such a system it is natural to provide priority to the patients with the most urgent condition [18, 22], but it is equally important to have a means to

estimate the impact of such a priority policy. After all, the lower priority customers may suffer because of it. The queues presented in this paper can be used to assess the performance in case no priority rule is installed and can therefore be used as a point of comparison when quantifying the impact of installing a priority policy.

The paper is structured as follows. Sections 2 and 3 introduce the adaptive MMAP[K] arrival process and its corresponding adaptive MMAP[K]/PH[K]/1 queue, respectively. Section 4 discusses the steady state of the workload process, some of its main performance characteristics and the associated computational complexity of the solution method. Finally, in Section 5 some numerical examples are provided.

2 The adaptive MMAP[K] arrival process

A Markovian arrival process with marked customers (MMAP[K]), introduced in [14], is characterized by a set of $K+1$ square matrices $D_0, \ldots, D_K$ of order m_a. The matrices D_1 to D_K are nonnegative, D_0 has nonnegative off-diagonal entries, while the diagonal entries of D_0 are negative and such that the row sums of $D = \sum_{k=0}^{K} D_k$ are zero. Entry (i,j) of the matrix D_k, for $k > 0$, holds the rate at which type k customers arrive, while the underlying Markov process (characterized by D) makes a transition from state i to j. Entry (i,j) of D_0, with $i \neq j$, holds the rate of having a state change from i to j without an arrival occurring. The type k arrival rate λ_k clearly equals $\lambda_k = \theta D_k e$, where θ is the stochastic invariant vector of D (i.e., $\theta D = 0$).

An adaptive MMAP[K] process is characterized by D_0, a set of K diagonal matrices $\Delta^{(k)}$, for $k = 1, \ldots, K$, and two sets of stochastic matrices $P_k^{(r)}$ and $P_k^{(a)}$, for $k = 1, \ldots, K$. D_0 plays the same role as before, while entry (i,i) of $\Delta^{(k)}$ is the rate at which a type k arrival occurs when the underlying chain is in state i. When a type k arrival occurs, the new state of the underlying chain will depend on whether the arrival is accepted into the system or rejected. If it is rejected the state becomes j with probability $(P_k^{(r)})_{i,j}$, otherwise the state equals j with probability $(P_k^{(a)})_{i,j}$. Notice, if $P_k^{(r)} = P_k^{(a)}$ for all k, we have an ordinary MMAP[K] process with $D_k = \Delta^{(k)} P_k^{(a)}$ and we can define the type k arrival rate λ_k as before. Otherwise, we cannot define the arrival rate as this depends on whether arrivals are accepted

or not.

Example 1: The MMAP[K] process. As indicated above, if $P_k^{(r)} = P_k^{(a)}$ for all k, we have an ordinary MMAP[K] process with $D_k = \Delta^{(k)} P_k^{(a)}$. Any MMAP[K] arrival process is clearly also an adaptive MMAP[K] process by setting $\Delta^{(k)} = D_k e$ and $P_k^{(a)} = P_k^{(r)} = \Delta(D_k e)^{-1} D_k$, where $\Delta(x)$ is a diagonal matrix with x appearing on the main diagonal and e is a column vector of ones. In the above definition, the i-th row of $P_k^{(a)}$ is not properly defined if the i-th entry of $D_k e$ is equal to zero. In such case any stochastic vector may be used instead as the i-th row of $P_k^{(a)}$.

Example 2: The adaptive Poisson process. The adaptive Poisson process of [2] is characterized by L arrival rates $\lambda_1, \ldots, \lambda_L$ and two stochastic matrices P and P^*. These matrices are used in the following manner. While the arrival process is in state i, the process behaves as a Poisson process with rate λ_i, for $i = 1, \ldots, L$. The state of the arrival process jumps from i to j with probability $P^*_{i,j}$ if a customer is rejected, and with probability $P_{i,j}$ if a customer is accepted. Thus, the intensity of the arrival process potentially changes with each arrival to the queue. We can model this arrival process with our framework by setting $m_a = K = L$, $(D_0)_{i,i} = -\lambda_i$, $\Delta^{(k)}_{k,k} = \lambda_k$, $P_k^{(a)} = P$ and $P_k^{(r)} = P^*$.

Example 3: The adaptive Poisson process with background traffic. Consider the same arrival process as in the previous example, but assume we also have a background process modeled as a MAP process characterized by the size m_b matrices (C_0, C_1) (a MAP is an MMAP[K] process with $K = 1$). The background process in this example is not adaptive. Customers generated by the Poisson process in state k are defined as type k customers, while the background customers are identified as type $L+1$; hence, we have $K = L+1$ types of customers. The superposed process is an adaptive MMAP[K] process with $m_a = m_b L$ as the states are of the form (k, j) with $k \in \{1, \ldots, L\}$ and $j \in \{1, \ldots, m_b\}$, where k is the state of the adaptive Poisson source and j of the background process. The type k arrival rate in state (k, j) is λ_k for $k = 1, \ldots, L$. Thus, the diagonal entries of $\Delta^{(k)}$ are given by $\lambda_k e_k \otimes e$ for $k \leq L$ (where e_k is the k-th column of the size L identity matrix and e has size m_b). The state changes from (k, j) to (k', j) with probability $P_{k,k'}$ ($P^*_{k,k'}$) when a type k customer is accepted (rejected), for $k \leq L$.

As such $P_k^{(a)} = P \otimes I$ and $P_k^{(r)} = P^* \otimes I$, for $k = 1, \ldots, L$. The matrix

$$D_0 = \begin{bmatrix} -\lambda_1 & & 0 \\ & \ddots & \\ 0 & & -\lambda_L \end{bmatrix} \otimes I + I \otimes C_0,$$

as the state jumps from (k, j) to (k, j') at rate $(C_0)_{j,j'}$ for $j \neq j'$. Finally, as the background MAP is not adaptive (cf. Example 1), $\Delta^{(L+1)}$ has $e \otimes C_1 e$ on its main diagonal, while $P_{L+1}^{(a)} = P_{L+1}^{(r)} = I \otimes \Delta(C_1 e)^{-1} C_1$, where $\Delta(x)$ is a diagonal matrix with x appearing on the main diagonal.

3 The adaptive MMAP[K]/PH[K]/1 queue

To determine whether a customer is accepted, several possibilities exist. We will focus on the case where the current workload will decide whether a customer is accepted. We assume that the queue makes use of $r \geq 0$ thresholds $d_1, \ldots, d_r$ such that $0 = d_0 < d_1 < \ldots < d_r < d_{r+1} = \infty$. Whenever the workload x upon arrival belongs to the interval $(d_{i-1}, d_i]$, a type k customer is accepted with probability $a_{i,k}$, for $i = 1, \ldots, r+1$. When $x = 0$, a type k arrival is accepted with probability $a_{0,k}$. Whenever $a_{i,k} = 1$ for all i, the matrix $P_k^{(r)}$ is irrelevant and for convenience we will define it equal to $P_k^{(a)}$.

The server will serve all the accepted customers in a first-come-first-served (FCFS) order, meaning the workload observed upon arrival equals the waiting time. The amount of work required to serve a type k customer follows a phase-type distribution with an order m_k representation (α_k, S_k). Hence, the mean service time of an accepted type k customer equals $\alpha_k(-S_k)^{-1}e$, which we denote as $1/\mu_k$.

Example 1: MMAP[K]/PH[K]/1 queue. Setting $a_{i,k} = 1$ for all i and k, implies that all the customers are accepted and we end up with an ordinary MMAP[K]/PH[K]/1 queue as in [14].

Example 2: MMAP[K]/PH[K]/1+D[K] queue. Assume we wish to associate a deadline $\bar{d}_k$ to the type k customers and without loss of generality let $\bar{d}_1 \leq \ldots \leq \bar{d}_K$. Setting $d_k = \bar{d}_k$ and $a_{i,k} = 1$ for $i \leq k$, then implies that a customer is accepted if his waiting time is at most $\bar{d}_k$. If we set $\bar{d}_k = B$ for all k, we are basically considering a queue that accepts new work as long as the workload is at most B.

Example 3: A running example. In this example we consider a specific MMAP[K]/PH[K]/1+D[K] queue that will be used as a running example in Section 4 to clarify matters. Furthermore, in Section 5.1 we will revisit this example, but replace the deterministic amount of patience of the type 1 customers by a Weibull distribution to demonstrate the full potential of the solution method. Consider a 2-state MMAP[2] process that generates arrivals at rate 0.3 in state one and 0.1 in state 2. 90% of the jobs generated in state one are of type 1, while in state 2 both types occur with equal probability, i.e.,

$$\Delta^{(1)} = \begin{bmatrix} 27/100 & 0 \\ 0 & 1/20 \end{bmatrix}, \quad \Delta^{(2)} = \begin{bmatrix} 3/100 & 0 \\ 0 & 1/20 \end{bmatrix}.$$

We will refer to the type 1 jobs as *short* as their service time is exponential with mean 1 and to the type 2 jobs as *long* as their service is also exponential, but with a mean of 10. This implies that $m_1 = m_2 = 1$ and $\alpha_1 = \alpha_2 = 1$, while $S_1 = -1$ and $S_2 = -1/10$. The mean sojourn time in state 1 and 2 is 1000 and 100, respectively. Hence,

$$D_0 = \begin{bmatrix} -3/10 - 1/1000 & 1/1000 \\ 1/100 & -1/10 - 1/100 \end{bmatrix}$$

and $P_k^{(a)} = P_k^{(r)} = I$ as the source is not adaptive and the arrivals do not affect the underlying state of the arrival process. The long jobs (i.e., type 2) are assumed to be patient, i.e., they never leave the waiting room. Short jobs have a deterministic amount of patience equal to 50 and abandon the system as soon as their waiting time exceeds 50. Thus, we have a system with a single threshold $d_1 = 50$ and $a_{i,k} = 1$ for $i = 0, 1, 2$ and $k = 1, 2$, except for $a_{2,1}$ which equals 0 (as type 1 customers are rejected if the workload exceeds d_1).

Example 4: MMAP[K]/PH[K]/1+G[K] queue. Assume the type k customers are impatient and their patience distribution has a finite support $\mathcal{C}_k$. We can model this system by defining the set of thresholds $d_1, \ldots, d_r$ as the union of the supports $\mathcal{C}_k$ for all k and $a_{i,k}$ as the probability that the patience of a type k customer is at least d_i. When the patience distribution is continuous, we can still generate approximate results by replacing the continuous distribution with a step function. In Section 5.1 we will demonstrate this approach by approximating the Weibull distribution with a step-function consisting of as many as 2^{14} steps. Prior work on continuous time queues with Markovian arrivals and

customer impatience was mostly limited to the case where the amount of patience is exponential [5, 2].

Example 5: beyond customer impatience. As long as the probabilities $a_{i,k}$ are non-increasing as a function of i, for all k, the system can be regarded as a queue with customer impatience, otherwise the queue no longer belongs to the class of MMAP[K]/PH[K]/1+G[K] queues. For instance, when $r = 1$, $d_1 = 100$, $a_{1,k} = 0$ and $a_{2,k} = 1$, for some k, we only accept a type k customer when the workload is larger than 100.

4 Workload process and steady-state solution

We define a Markov process $(V_t, Z_t)_{t\geq 0}$ by observing the workload V_t, called the level, and the state of the arrival process Z_t at time t. As usual, the workload is defined as the remaining duration of the busy period, provided that no new arrivals occur. The state space of the process $(V_t, Z_t)_{t\geq 0}$ is $\mathcal{V} = \{(x, j)|x \geq 0, j = 1, \ldots, m_a\}$, meaning the level V_t is a continuous variable and Z_t has a finite range. As the level V_t represents the workload, it decreases linearly at rate 1 as long as no new customers arrive, while it makes upward jumps at the arrival epochs. More specifically, as type k arrivals in state (x, j) occur at rate $\Delta^{(k)}_{j,j}$ and $\alpha_k \exp(S_k u)s_k$, with $s_k = (-S_k)e$, represents the phase-type density that the amount of work required by a type k arrival equals u, we find that the jump rate from state (x, j) to (x', j') with $x' \in (x + u, x + u + du)$ is given by

$$\sum_{k=1}^{K} \Delta^{(k)}_{j,j} a_{i,k} (P^{(a)}_k)_{j,j'} (\alpha_k \exp(S_k u)s_k)du + o(du), \tag{1}$$

when $x \in (d_{i-1}, d_i]$, for $i = 1, \ldots, r + 1$ (and $i = 0$ if $x = 0$). Indeed, $a_{i,k}(P^{(a)}_k)_{j,j'}$ represents the probability that the type k customer is accepted and the state of the arrival process changes from j to j' as a result. The process also jumps from state (x, j) to (x, j') when the arrival process jumps from state j to j' without generating an arrival or while generating an arrival that is rejected. Hence, the jump rate from state (x, j) to (x, j'), with $j' \neq j$, equals

$$(D_0)_{j,j'} + \sum_{k=1}^{K} \Delta^{(k)}_{j,j} (1 - a_{i,k})(P^{(r)}_k)_{j,j'}. \tag{2}$$

The process $(V_t, Z_t)_{t\geq 0}$ is a jump process as in [8], however, in our case the rate at which the jumps

occur as well as the jump sizes are not independent of the current level x of the process. In order to obtain the steady state density of the process $(V_t, Z_t)_{t\geq 0}$, we construct a fluid queue from this jump process by replacing the immediate upward jumps of size h by intervals of length h during which the level increases linearly at rate 1. The resulting fluid queue is a fluid queue with thresholds as discussed in [7]. As a result, an expression for the steady state of the Markov process $(V_t, Z_t)_{t\geq 0}$ can be obtained from the steady-state of the fluid queue with thresholds using a censoring argument (that eliminates the periods during which the level increases).

Running example continued: For Example 3 introduced in Section 3 the state space $\mathcal{V}$ of the jump process $(V_t, Z_t)_{t\geq 0}$ is given by $\{(x, j)|x \geq 0, j = 1, 2\}$. A jump from state (x, j) to (x, j'), with $j' \neq j$, can only occur when the arrival process jumps from state 1 to 2 (at rate 1/1000) or vice versa (at rate 1/100). This is in agreement with (2) as $P_k^{(r)} = I$ for $k = 1, 2$. Upward jumps correspond to arrivals that are accepted and these add an exponential amount of work with $\mu_1 = 1$ and $\mu_2 = 1/10$. If $x \leq 50$, both types are accepted, meaning the rate of jumps from (x, j) to (x', j), with $x' \in (x + u, x + u + du)$, equals $0.27 \exp(-\mu_1 u)du + 0.03\mu_2 \exp(-\mu_2 u)du + o(du)$ for $j = 1$ and $0.05(\exp(-\mu_1 u) + \mu_2 \exp(-\mu_2 u))du + o(du)$ for $j = 2$. For $x > 50$, only type 2 customers are accepted and the rate therefore reduces to $0.03\mu_2 \exp(-\mu_2 u)du + o(du)$ for $j = 1$ and to $0.05\mu_2 \exp(-\mu_2 u)du + o(du)$ for $j = 2$.

4.1 Construction of the fluid queue with thresholds

A fluid queue with r thresholds $0 = d_0 < d_1 < d_2 < \ldots < d_r < d_{r+1} = \infty$ and a set of phases $\mathcal{S} = \mathcal{S}^+ \cup \mathcal{S}^-$, is fully characterized by $F_{--}^{(0)}$, $F_{-+}^{(0)}$ and the $r + 1$ matrices $F^{(1)}$ to $F^{(r+1)}$, where $F^{(i)}$ is partitioned as

$$F^{(i)} = \begin{bmatrix} F_{++}^{(i)} & F_{+-}^{(i)} \\ F_{-+}^{(i)} & F_{--}^{(i)} \end{bmatrix},$$

$F_{ab}^{(i)}$ describes the rate of change between the phases in $\mathcal{S}^a$ and $\mathcal{S}^b$ when the level $x \in (d_{i-1}, d_i]$, for $a, b \in \{+, -\}$ and $F_{-b}^{(0)}$ describes the rate of change between the phases in $\mathcal{S}^-$ and $\mathcal{S}^b$ when the level x equals zero. We denote m^- and m^+ as the cardinality of the set $\mathcal{S}^-$ and $\mathcal{S}^+$, respectively. Whenever the phase of the queue is part of $\mathcal{S}^+$ ($\mathcal{S}^-$), the level x increases (decreases) at rate one (while it remains

zero if $x = 0$ and the phase is in $\mathcal{S}^-$).

Next, we introduce the fluid queue with r thresholds obtained by replacing the upward jumps of $(V_t, Z_t)_{t\geq 0}$ by intervals of the appropriate length during which the level increases linearly at rate one. When the level decreases the fluid queue behaves as $(V_t, Z_t)_{t\geq 0}$, as such the set $\mathcal{S}^-$ of phases during which the fluid decreases at rate 1 is of size $m^- = m_a$, the number of states of the arrival process. Periods during which the level increases correspond to the upward jumps of $(V_t, Z_t)_{t\geq 0}$. The amount of work added during these jumps follows some phase-type distribution characterized by an order m_k representation (α_k, S_k), for some $k \in \{1, \ldots, K\}$. Therefore, we keep track of the type k of the customer, the work of which we are adding, as well as the phase v of its phase-type distribution. Furthermore, while the fluid increases we also keep track of the state j of the adaptive MMAP[K] arrival process. Thus, $\mathcal{S}^+ = \{(k, v, j) | k = 1, \ldots, K, v \in \{1, \ldots, m_k\}, j = 1, \ldots, m_a\}$ is the set of phases during which the fluid increases at rate 1 and its cardinality m^+ equals $m_a \sum_{k=1}^{K} m_k$.

As the fluid queue evolves identical to the jump process when level of the fluid queue decreases, Equation (2) yields

$$F_{--}^{(i)} = D_0 + \sum_{k=1}^{K} \Delta^{(k)} P_k^{(r)} (1 - a_{i,k}),$$

where the dependency on $i \in \{0, \ldots, r+1\}$ is caused by the presence of the probabilities $a_{i,k}$. When a type k arrival is accepted, the jump process $(V_t, Z_t)_{t\geq 0}$ jumps from some state (x, j) to some state (x', j'). The fluid queue however will jump from phase $j \in \mathcal{S}^-$ to phase $(k, v, j') \in \mathcal{S}^+$, where v will be determined by the vector α_k. Due to Equation (1), we can write this in matrix form as

$$F_{-+}^{(i)} = \sum_{k=1}^{K} ((0, \ldots, 0, \alpha_k, 0, \ldots, 0) \otimes \Delta^{(k)} P_k^{(a)}) a_{i,k},$$

where the number of zeros appearing before and after α_k equals $\sum_{j=1}^{k-1} m_j$ and $\sum_{j=k+1}^{K} m_j$, respectively.

Next assume we are adding the work generated by a type k arrival, meaning we are in some state $(k, v, j') \in \mathcal{S}^+$. The state j' will remain frozen as long as the fluid increases, because the upward jumps of $(V_t, Z_t)_{t\geq 0}$ are instantaneous. The threshold values $d_1, \ldots, d_r$ do not affect the evolution of the fluid queue either, only the matrix S_k does as $(S_k)_{v,v'}$ reflects the jump rate from (k, v, j') to (k, v', j'). Hence,

$F^{(i)}_{++}$ does not depend on i and equals

$$F^{(i)}_{++} = \begin{bmatrix} S_1 & & 0 \\ & \ddots & \\ 0 & & S_K \end{bmatrix} \otimes I,$$

where I is a size m_a identity matrix that reflects the fact that the state of the arrival process remains frozen. Similarly, a jump from phase $(k, v, j') \in \mathcal{S}^+$ to phase $j' \in \mathcal{S}^-$ occurs with rate $(s_k)_v$, yielding

$$F^{(i)}_{+-} = \begin{bmatrix} s_1 \\ \vdots \\ s_K \end{bmatrix} \otimes I,$$

where I is the size m_a identity matrix.

Running example continued: For Example 3 introduced in Section 3 the set $\mathcal{S}^-$ contains only two phases, while $\mathcal{S}^+$ contains 4 elements being $(1,1,1), (1,1,2), (2,1,1)$ and $(2,1,2)$ as $m_1 = m_2 = 1$. The matrices $F^{(i)}_{--}$ and $F^{(i)}_{-+}$ are given by

$$F^{(0)}_{--} = F^{(1)}_{--} = D_0, \quad F^{(2)}_{--} = \begin{bmatrix} -3/100 - 1/1000 & 1/1000 \\ 1/100 & -1/20 - 1/100 \end{bmatrix}$$

and

$$F^{(0)}_{-+} = F^{(1)}_{-+} = \begin{bmatrix} 27/100 & 0 & 3/100 & 0 \\ 0 & 1/20 & 0 & 1/20 \end{bmatrix}, \quad F^{(2)}_{-+} = \begin{bmatrix} 0 & 0 & 3/100 & 0 \\ 0 & 0 & 0 & 1/20 \end{bmatrix}$$

as type 1 customers are not accepted when $x > d_1 = 50$. Due to the exponential service requirements we see that

$$F^{(i)}_{++} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1/10 & 0 \\ 0 & 0 & 0 & -1/10 \end{bmatrix}, \quad F^{(i)}_{+-} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1/10 & 0 \\ 0 & 1/10 \end{bmatrix},$$

for $i = 0, 1, 2$.

4.2 Steady-state distribution of the workload process

For reasons of completeness, we start by restating a special case of the main theorem in [7]. Let $\pi(x)$, for $x > 0$, be the steady state density vector of level x of the fluid queue and write $\pi(x)$ as $(\pi_-(x), \pi_+(x))$ for $x > 0$. For $x = 0$, denote $p_-(0)$ as the steady state probability mass vector of the fluid queue corresponding to level 0 with a phase in $\mathcal{S}^-$ and $\pi_+(0)$ as the steady state density of level 0 with a phase in $\mathcal{S}^+$. Further, let $\xi^{(i)} = (\xi_-^{(i)}, \xi_+^{(i)})$ be the unique stochastic invariant vector of $F^{(i)}$, then the following theorem is due to [7]. In fact, $F_{--}^{(0)} = F_{--}^{(1)}$ and $F_{-+}^{(0)} = F_{-+}^{(1)}$ in [7], but it is not hard to see that the result remains valid in case these equalities no longer hold.

Theorem 1 *The fluid queue with thresholds $0 = d_0 < d_1 < \ldots < d_r < d_{r+1} = \infty$ is positive recurrent if $\xi_+^{(r+1)} e < \xi_-^{(r+1)} e$. Moreover, if $\xi_-^{(i)} e \neq \xi_+^{(i)} e$ for $i = 1, \ldots, r$, its steady state density $\pi(x)$ can be expressed as*

$$(\pi_+(x), \pi_-(x)) = \pi_+(d_r) e^{K^{(r+1)}(x-d_r)} [I, \Psi^{(r+1)}],$$

for $x > d_r$ and

$$\begin{aligned}(\pi_+(x), \pi_-(x)) \quad &= \quad (\pi_+(d_{i-1}) N_1^{(i)} + \pi_-(d_i) N_3^{(i)}) e^{K^{(i)}(x-d_{i-1})} [I, \Psi^{(i)}] + \\ &\qquad (\pi_+(d_{i-1}) N_2^{(i)} + \pi_-(d_i) N_4^{(i)}) e^{\hat{K}^{(i)}(d_i - x)} [\hat{\Psi}^{(i)}, I],\end{aligned}$$

for $x \in (d_{i-1}, d_i)$ and $i = 1, \ldots, r$, where

$$N^{(i)} = \begin{bmatrix} N_1^{(i)} & N_2^{(i)} \\ N_3^{(i)} & N_4^{(i)} \end{bmatrix} = \begin{bmatrix} I & e^{K^{(i)} b_i} \Psi^{(i)} \\ e^{\hat{K}^{(i)} b_i} \hat{\Psi}^{(i)} & I \end{bmatrix}^{-1},$$

and $b_i = d_i - d_{i-1}$.

The matrices $K^{(i)}$, $\Psi^{(i)}$ and $U^{(i)}$, for $i = 1, \ldots, r+1$, appearing in the above theorem can be computed as follows. $\Psi^{(i)}$ is the smallest non-negative solution to the algebraic Riccati equation

$$F_{+-}^{(i)} + \Psi^{(i)} F_{--}^{(i)} + F_{++}^{(i)} \Psi^{(i)} + \Psi^{(i)} F_{-+}^{(i)} \Psi^{(i)} = 0.$$

The matrices $K^{(i)}$ and $U^{(i)}$ are then readily obtained from $\Psi^{(i)}$ as

$$K^{(i)} = F_{++}^{(i)} + \Psi^{(i)} F_{-+}^{(i)}, \qquad U^{(i)} = F_{--}^{(i)} + F_{-+}^{(i)} \Psi^{(i)}.$$

Additionally, one also needs these matrices for the level reversed process (for $i = 1, \ldots, r$), denoted as $\hat{K}^{(i)}$, $\hat{\Psi}^{(i)}$ and $\hat{U}^{(i)}$. Hence,

$$F_{-+}^{(i)} + \hat{\Psi}^{(i)} F_{++}^{(i)} + F_{--}^{(i)} \hat{\Psi}^{(i)} + \hat{\Psi}^{(i)} F_{+-}^{(i)} \hat{\Psi}^{(i)} = 0,$$

and

$$\hat{K}^{(i)} = F_{--}^{(i)} + \hat{\Psi}^{(i)} F_{+-}^{(i)}, \quad \hat{U}^{(i)} = F_{++}^{(i)} + F_{+-}^{(i)} \hat{\Psi}^{(i)}.$$

Efficient algorithms to solve algebraic Riccati equations are briefly discussed in Appendix A.

Let $\theta^{(i)}$, for $i = 1, \ldots, r+1$, be the stochastic invariant vector of

$$D_0 + \sum_{k=1}^{K} \Delta^{(k)} (P_k^{(a)} a_{i,k} + P_k^{(r)} (1 - a_{i,k}))$$

and define $\rho^{(i)} = \sum_{k=1}^{K} \frac{a_{i,k}}{\mu_k} \theta^{(i)} \Delta^{(k)} e$ as the load in the i-th region (d_{i-1}, d_i).

Theorem 2 *The Markov process $(V_t, Z_t)_{t \geq 0}$ is positive recurrent if $\rho^{(r+1)} < 1$.*

Proof: Due to Theorem 1, the fluid queue that was constructed from (V_t, Z_t) is positive recurrent if $\xi_+^{(r+1)} e < \xi_-^{(r+1)} e$, where $\xi^{(r+1)} = (\xi_+^{(r+1)}, \xi_-^{(r+1)})$ is the invariant vector of $F^{(r+1)}$. It is not hard to show that $\xi^{(i)}$, the invariant vector of $F^{(i)}$, is proportional to

$$\left(\sum_{k=1}^{K} \frac{a_{i,k}}{\mu_k} ((0, \ldots, 0, \beta_k, 0, \ldots, 0) \otimes \theta^{(i)} \Delta^{(k)} P_a^{(k)}), \theta^{(i)} \right),$$

where β_k is the stochastic invariant vector of $S_k - S_k e \alpha_k$. Hence, $\xi_+^{(i)} e < \xi_-^{(i)} e$ is equivalent to $\rho^{(i)} < 1$, for $i = 1, \ldots, r+1$. □

Let $\bar{\pi}(x)$, for $x > 0$, be the steady state density at level x of the Markov process $(V_t, Z_t)_{t \geq 0}$. For $x = 0$, denote $\bar{p}_-(0)$ as the steady state probability mass vector corresponding to level 0 with a phase in $\mathcal{S}^-$ and $\bar{\pi}_+(0)$ the steady state density of level 0 with a phase in $\mathcal{S}^+$.

Theorem 3 *If $\rho^{(i)} \neq 1$ for $i = 1, \ldots, r$, and $\rho^{(r+1)} < 1$, the steady state density $\bar{\pi}(x)$ of the Markov process $(V(t), Z(t))_{t \geq 0}$ can be expressed as*

$$\bar{c} \bar{\pi}(x) / c = \pi_+(d_r) e^{K^{(r+1)}(x - d_r)} \Psi^{(r+1)}, \tag{3}$$

for $x > d_r$ and

$$\bar{c}\bar{\pi}(x)/c = (\pi_+(d_{i-1})N_1^{(i)} + \pi_-(d_i)N_3^{(i)})e^{K^{(i)}(x-d_{i-1})}\Psi^{(i)} + (\pi_+(d_{i-1})N_2^{(i)} + \pi_-(d_i)N_4^{(i)})e^{\hat{K}^{(i)}(d_i-x)}, \quad (4)$$

for $x \in (d_{i-1}, d_i)$ and $i = 1, \ldots, r$, where $N_j^{(i)}$, for $j = 1, \ldots, 4$, is defined as in Theorem 1.

Finally, the density vectors obey $\bar{c}\bar{\pi}(d_i) = c\pi_+(d_i)$, for $i = 1, \ldots, r$, $\bar{c}\bar{\pi}_+(0) = c\pi_+(0)$ and the probability mass $\bar{p}_-(0)$ is found as $\bar{c}\bar{p}_-(0) = cp_-(0)$, where $c, \bar{c} \geq 1$ are normalizing constants.

Proof: If we censor the fluid queue on the time epochs during which the level is not increasing, we end up with a new stochastic process that has the same stationary distribution as the Markov process $(V_t, Z_t)_{t\geq 0}$, as a result the theorem is immediate from Theorem 1. □

Remark 1: The use of two normalizing constants c and $\bar{c}$ may appear redundant, in fact only $\bar{c}$ needs to be computed. This can be understood by noting that the computational method of [7] to determine the steady state of a fluid queue with thresholds, computes the densities $c\pi(x)$ and normalizes these by c. As we are interested in the steady state of the jump process $(V_t, Z_t)_{t\geq 0}$, which is obtained by censoring the fluid queue, the normalization constant c is of no use to us. Instead we normalize by another constant denoted as $\bar{c}$.

Remark 2: Whenever $\rho^{(i)}$ equals one for some i, the matrices $N_j^{(i)}$, for $j = 1, \ldots, 4$, are not properly defined as $N^{(i)}$ is singular in this case (see [6, Section 4]).

4.3 Computational complexity

Apart from computing the matrices $\Psi^{(i)}$, $K^{(i)}$, $\hat{\Psi}^{(j)}$ and $\hat{K}^{(j)}$, for $i = 1, \ldots, r+1$ and $j = 1, \ldots, r$, as discussed in Appendix A, we also need to determine the probability vector $cp_-(0)$, the density $c\pi_+(0)$, the densities $c\pi(d_i) = c(\pi_-(d_i), \pi_+(d_i))$, for $i = 1, \ldots, r$, and the normalizing factor $\bar{c}$. For the computation of $cp_-(0)$, $c\pi_+(0)$ and $\bar{c}$ we refer to Appendix B. The density vectors $c\pi(d_i) = c(\pi_-(d_i), \pi_+(d_i))$, for

$i = 1, \ldots, r$ are the unique solution to the following set of linear equations [7][Theorem 3.4]:

$$\begin{aligned} c\pi_-(d_r) &= c\pi_+(d_r)\Psi^{(r+1)} \\ c\pi_+(d_i) &= c\pi_+(d_{i-1})\Lambda_{++}^{(i)} + c\pi_-(d_i)\hat{\Psi}_{-+}^{(i)} \\ c\pi_-(d_i) &= c\pi_+(d_i)\Psi_{+-}^{(i+1)} + c\pi_-(d_{i+1})\hat{\Lambda}_{--}^{(i+1)} \end{aligned}$$

where the second equation is valid for $i = 1, \ldots, r$ and the third for $i = 1, \ldots, r-1$. An expression for the matrices $\Lambda_{++}^{(i)}$, $\hat{\Psi}_{-+}^{(i)}$, $\Psi_{+-}^{(i)}$ and $\hat{\Lambda}_{--}^{(i)}$ is also provided in Appendix B. Solving this system with standard numerical techniques would imply that the time complexity is cubic in rm, where m is the number of phases of the fluid queue. In this section we introduce an algorithm for solving this system such that the time complexity grows as rm^3, while the memory occupancy grows as rm^2.

If we define the following matrices

$$A_{i,i} = \begin{bmatrix} 0 & \hat{\Psi}_{-+}^{(i)} \\ \Psi_{+-}^{(i+1)} & 0 \end{bmatrix}, \quad A_{i,i+1} = \begin{bmatrix} 0 & 0 \\ 0 & \Lambda_{++}^{(i+1)} \end{bmatrix},$$

for $i = 1, \ldots, r-1$ and

$$A_{i,i-1} = \begin{bmatrix} \hat{\Lambda}_{--}^{(i)} & 0 \\ 0 & 0 \end{bmatrix}, \quad A_{r,r} = \begin{bmatrix} 0 & \hat{\Psi}_{-+}^{(r)} \\ \Psi^{(r+1)} & 0 \end{bmatrix},$$

for $i = 2, \ldots, r$, we find that $\pi = c(\pi(d_1), \ldots, \pi(d_r))$ is the unique solution of the linear system of the form

$$\pi = \pi A + (0, \ldots, 0, \pi_+(0)\Lambda_{++}^{(1)}, 0, \ldots, 0),$$

where we have m^- zeros appearing before $\pi_+(0)\Lambda_{++}^{(1)}$ and A can be written as

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & & 0 \\ A_{2,1} & \ddots & \ddots & \\ & \ddots & \ddots & A_{r-1,r} \\ 0 & & A_{r,r-1} & A_{r,r} \end{bmatrix}.$$

This allows us to make use of the following algorithm to compute the vectors $c\pi(d_i)$, for $i = 1, \ldots, r$:

Theorem 4 *Define the matrices $\tilde{A}_{i,i}$ recursively as $\tilde{A}_{1,1} = A_{1,1}$ and*

$$\tilde{A}_{i,i} = A_{i,i} + A_{i,i-1}(I - \tilde{A}_{i-1,i-1})^{-1}A_{i-1,i},$$

for $i = 2, \ldots, r$. *Further, let* $\tilde{c}_1 = (0, \ldots, 0, \pi_+(0)\Lambda^{(1)}_{++})$ *and for* $i = 2, \ldots, r$ *let*

$$\tilde{c}_i = \tilde{c}_{i-1}(I - \tilde{A}_{i-1,i-1})^{-1}A_{i-1,i}.$$

Then, $c\pi(d_r) = \tilde{c}_r(I - \tilde{A}_{r,r})^{-1}$ *and for* $i = r-1, \ldots, 1$ *we have*

$$c\pi(d_i) = (\tilde{c}_i + c\pi(d_{i+1})A_{i+1,i})\,(I - \tilde{A}_{i,i})^{-1}.$$

Proof: The equations above can be obtained by repeated substitution. A similar approach was also used in [24] for the policy evaluation step of a Markov decision process skip-free in both directions. □

Taking the above algorithm and the computational procedures outlined in Appendix A and B into account, we find that the overall time (memory) complexity is linear in the number of thresholds r and cubic (square) in $m = m_a(1 + \sum_{k=1}^{K} m_k)$, the number of phases of the fluid queue.

4.4 Waiting time distributions and probabilities of abandonment

Define the probability vectors $\bar{p}_-(x) = \bar{p}_-(0) + \int_{y=0}^{x} \bar{\pi}(y)dy$, for $x > 0$. Due to Theorem 3 we find

$$\begin{aligned}\frac{\bar{c}\bar{p}_-(x)}{c} &= \frac{\bar{c}\bar{p}_-(d_{i-1})}{c} + (\pi_+(d_{i-1})N_1^{(i)} + \pi_-(d_i)N_3^{(i)})A^{(i)}(x)\Psi^{(i)} + \\ &\quad (\pi_+(d_{i-1})N_2^{(i)} + \pi_-(d_i)N_4^{(i)})\hat{A}^{(i)}(x) \qquad (5)\end{aligned}$$

for $x \in (d_{i-1}, d_i]$ and $i = 1, \ldots, r$, (where $A^{(i)}(x)$ and $\hat{A}^{(i)}(x)$ are defined in Appendix B), while

$$\frac{\bar{c}\bar{p}_-(x)}{c} = \frac{\bar{c}\bar{p}_-(d_r)}{c} + \pi_+(d_r)(-K^{(r+1)})^{-1}(I - e^{K^{(r+1)}(x-d_r)})\Psi^{(r+1)}, \qquad (6)$$

for $x > d_r$.

Theorem 5 *The probability* $P[W_k \leq x]$ *that an accepted type-k customer has a waiting time of at most* x, *can be computed as*

$$P[W_k \leq x] = \frac{1}{\lambda_k^{(a)}}\left(\bar{p}_-(0)a_{0,k} + \sum_{j=1}^{i-1}(\bar{p}_-(d_j) - \bar{p}_-(d_{j-1}))a_{j,k} + (\bar{p}_-(x) - \bar{p}_-(d_{i-1}))a_{i,k}\right)\Delta^{(k)}e, \qquad (7)$$

for $x \in [d_{i-1}, d_i)$ *(with* $d_0 = 0$ *and* $d_{r+1} = \infty$*) and* $i = 1, \ldots, r+1$, *where the type-k accept rate*

$$\lambda_k^{(a)} = \left(\bar{p}_-(0)a_{0,k} + \sum_{i=1}^{r+1}(\bar{p}_-(d_i) - \bar{p}_-(d_{i-1}))a_{i,k}\right)\Delta^{(k)}e$$

and the type-k arrival rate $\lambda_k = \bar{p}_-(\infty)\Delta^{(k)}e$.

Proof: The probability that an accepted type k customer has a waiting time of at most x can be computed as the accepted arrival rate of the type k customers that find a workload of at most x upon arrival, divided by the total accepted arrival rate of the type k customers. Therefore, the result follows from Theorem 3. □

5 Numerical Examples

5.1 Queues with customer impatience

In the first example, we revisit Example 3 of Section 3, but to make things more challenging, we now assume the patience of a short job follows a Weibull distribution with parameters (k, λ), where λ is set such that its mean is 50. Hence, $k = 1$ results in an exponential amount of patience, when $k > 1$ customers become less patient as time passes (i.e., they have an increasing hazard rate) and $k < 1$ implies that the hazard decreases which results in a heavy tail.

The Weibull distribution is clearly a continuous distribution, i.e., it is not a finite support distribution. To make use of our framework, we will approximate it with a finite support distribution and gradually increase the number of support points to as many as $2^{14} = 16384$. Recall, the time and memory complexity of this method is linear in the number of thresholds, which allows us to solve systems with as many as 2^{14} thresholds in less than 30 seconds on a 3.33 GHz CPU (and 4 Gbyte of RAM), where approximately 8 seconds was required to solve the 2^{15} algebraic Riccati equations (for which $m^- = 2$ and $m^+ = 4$). Denote X as the (Weibull) patience distribution and consider the following two manners to position the thresholds. The *linear* (lin) positioning defines the values of the r thresholds d_i such that $P[X > d_i] = (r + 1 - i)/(r + 1)$, for $i = 1, \ldots, r$ (with $d_0 = 0$ and $d_{r+1} = \infty$). The *quadratic* (qdr) positioning sets d_i such that $P[X > d_i] = (r + 1 - i)^2/(r + 1)^2$, for $i = 1, \ldots, r$.

We also consider two possibilities for the abandonment probabilities $a_{i,1}$ (notice, $a_{i,2} = 1$ as the long jobs are patient). The first, called the *Round up*, sets $a_{i,1} = P[X > d_i]$, for $i = 1, \ldots, r + 1$. In the second mode, termed *Round down*, we set $a_{i,1} = P[X > d_{i-1}]$, for $i = 1, \ldots, r + 1$. In both modes $a_{0,1}$ equals one. Thus, these two approaches approximate the Weibull rejection probabilities by means of a

mode, r	$k = 0.5$		$k = 1$		$k = 2$		$k = 4$		$k = 8$	
	lin	qdr.	lin	qdr.	lin	qdr.	lin	qdr.	lin	qdr.
up, 2^6	.20064	.20257	.10077	.10424	.05153	.05715	.03265	.03968	.02668	.03438
up, 2^{10}	.19738	.19751	.09703	.09724	.04630	.04660	.02611	.02652	.01937	.01984
up, 2^{14}	.19718	.19718	.09681	.09682	.04602	.04604	.02574	.02576	.01892	.01895
down, 2^{14}	.19715	.19714	.09677	.09676	.04599	.04597	.02570	.02569	.01889	.01889
down, 2^{10}	.19694	.19682	.09655	.09635	.04578	.04561	.02558	.02548	.01883	.01879
down, 2^6	.19368	.19165	.09306	.09025	.04338	.04157	.02429	.02346	.01825	.01798

Table 1: Probability of abandonment for type 1 jobs for a Weibull distribution with $k = 0.5$ to 8 and mean 50 for $r = 64, 1024$ and 16384 support points

step function. In the first case the steps are above the Weibull curve, while in the latter approach they are below. We therefore expect that the Round up mode will overestimate the rejection probability, while the Round down mode is expected to result in an underestimation.

Table 5.1 shows the resulting probability of abandonment of the short jobs for various k values, where the number of support points considered equaled 64, 1024 and 16384. As expected, the results of the Round up are above those of the Round down and both seem to converge to one another as the number of support points r increases, for both the linear and quadratic positioning of the thresholds. This indicates that a fairly accurate estimate of the probability of abandonment can be obtained if many support points are used, where the linear positioning provides a slightly better accuracy for a fixed r. The abandonment probability decreases with increasing k as the patience distribution becomes more deterministic.

Figure 1 depicts the waiting time distribution of the type 1 customers for $k = 0.5$ and $k = 8$ for various r values (where we zoomed in on the tail for the $k = 8$ case). For $k = 0.5$, the distribution for the type 2 customers is also shown and we find that the round up and down curves are close to each other even for a small number of support points (256 for type 1 and 16 for type 2), using the linear threshold positioning. For $k = 8$, the linear positioning has difficulty matching the tail of the type 1

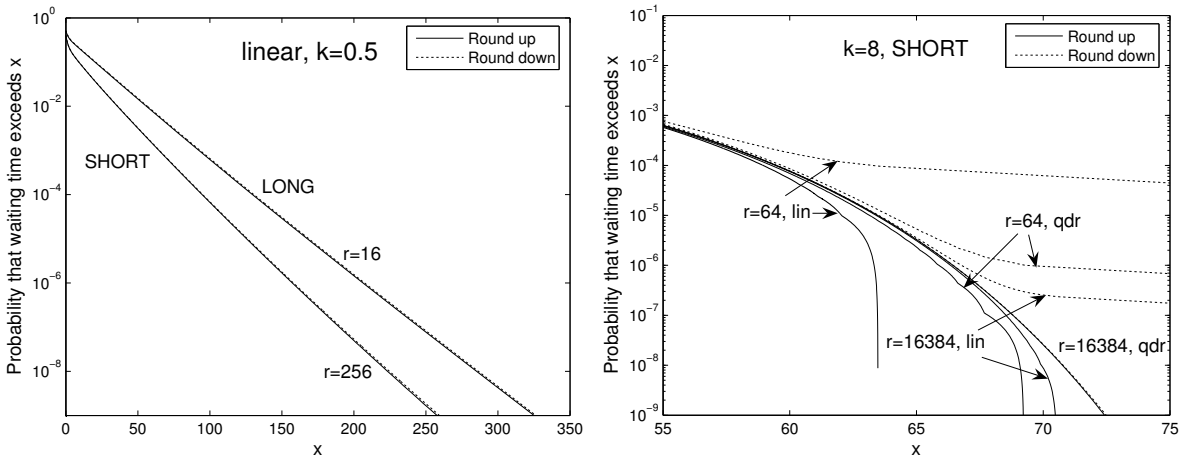

Figure 1: Waiting time distribution for type 1 (and 2) jobs for $k = 0.5$ and $k = 8$ for various r values

waiting time distribution mostly due to the location of the last threshold d_r. The quadratic positioning performs substantially better with respect to capturing the tail behavior as both $r = 16384$ curves nearly coincide.

5.2 Adaptive Poisson source with background traffic

We continue by demonstrating our approach on a buffer fed by an adaptive Poisson process that is multiplexed with a non-adaptive background process as in Example 3 of Section 2. We assume that all the packets have a mean length of 1, meaning the mean arrival rate equals the load. The background process is a 2 state MAP characterized by

$$C_0 = \begin{bmatrix} -0.2 & 0 \\ 0 & -0.5 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 0.2(1-p_c) & 0.2p_c \\ 0.5p_c & 0.5(1-p_c) \end{bmatrix},$$

with $p_c = 1/1000$. In other words, the background process alternates between periods with an arrival rate of 0.2 and 0.5, where the mean duration of a period is 1000 arrivals. Thus, the mean arrival rate of the background source is $2/7$. The adaptive Poisson process has four arrival rates λ_1 to λ_4 and these equal $0.3, 0.45, 0.6$ and 0.75. In other words, when the background source is in state 1, a rate λ of 0.75 seems appropriate to achieve a good utilization with a limited loss, while $\lambda = 0.45$ is more appropriate if the background source is in state 2. The adaptive Poisson source is assumed to decrease its rate immediately when a packet is rejected. When a packet is accepted, the Poisson source will increase its

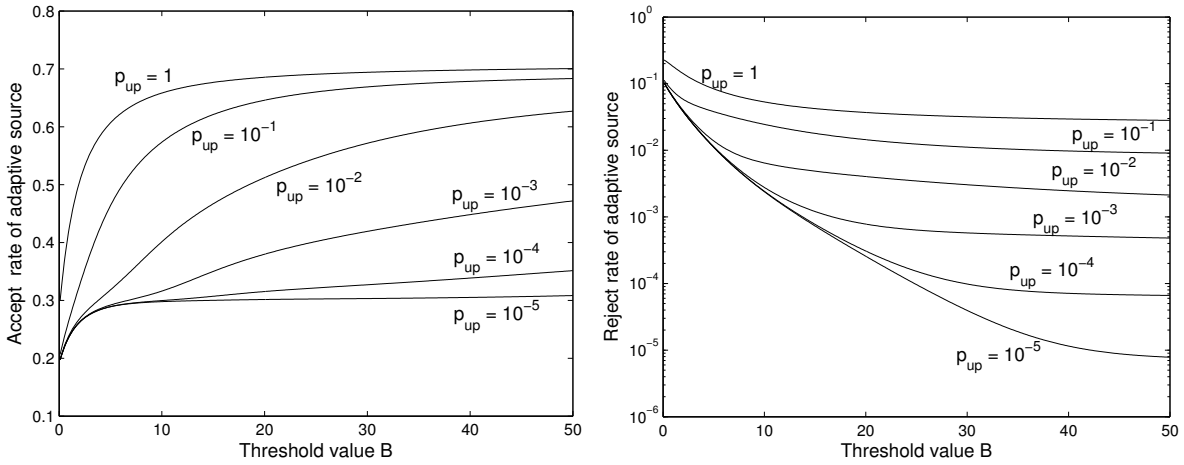

Figure 2: Accept and reject rate of the adaptive Poisson source as a function of the threshold B for $p_{up} = 1$ to $p_{up} = 10^{-5}$ in a queue with a single threshold at B

rate, but only with some probability p_{up}, hence P and P^* are given by

$$P = \begin{bmatrix} 1-p_{up} & p_{up} & 0 & 0 \\ 0 & 1-p_{up} & p_{up} & 0 \\ 0 & 0 & 1-p_{up} & p_{up} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad P^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Notice, the adaptive Poisson source does not alter its rate when a background packet is accepted/rejected. We start by assuming that the buffer uses a single threshold B and that a packet is accepted if and only if the current workload is at most B. In order to focus on the arrival rates, we assume that all packets have an exponential duration (with mean 1), but different phase type distributions could be used as well (we can even make the service time dependent on the rate of the adaptive source as we associated a customer type to each rate).

Figure 2 shows the accept and reject rate of the adaptive Poisson source as a function of B for various p_{up} values. As expected the accept rate, i.e., throughput, increases as B and p_{up} increase, while the reject rate decreases with B and increases with p_{up}. In other words, when p_{up} is large, the adaptive source becomes very aggressive which results in a high accept rate at the expense of many rejects. Lowering p_{up} reduces the number of rejects, but also the throughput. The key value in selecting p_{up} is clearly related to the rate at which the background traffic changes between both states (that is, every 1000 arrivals on

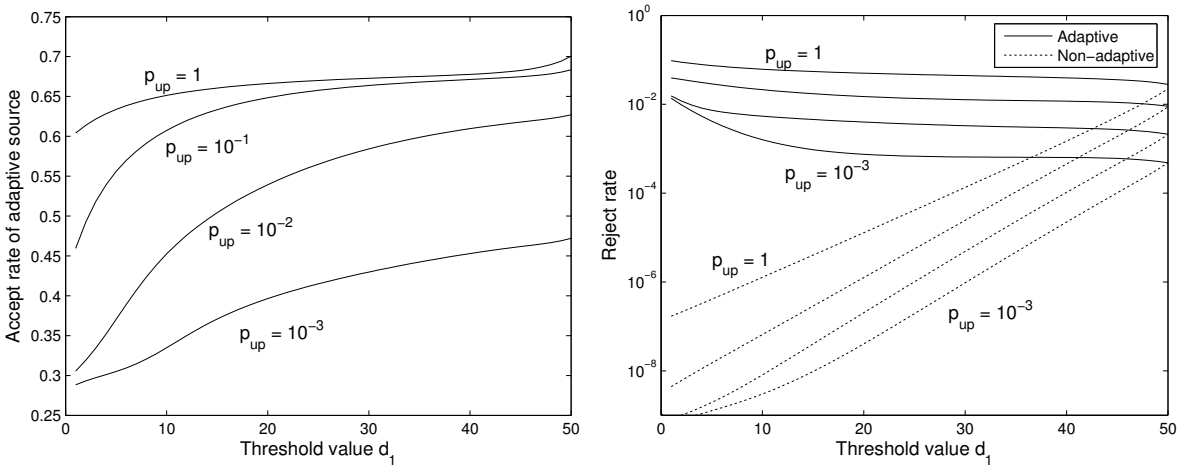

Figure 3: Accept and reject rate of the adaptive Poisson and background source as a function of the threshold d_1 for $p_{up} = 1$ to $p_{up} = 10^{-3}$ in a queue with 10 equidistant thresholds with $d_{10} = 50$

average), as the adaptive source should be able to change its rate sufficiently fast in order to adapt itself to the non-adaptive background source. The loss rates of the background source, not shown here, were quite similar to the reject rates of the adaptive source.

Next, we consider a setup with $r = 10$ equidistant thresholds where the first threshold is positioned at d_1 and the last at $d_{10} = 50$. We further assume that background jobs are only rejected when the workload is above d_{10}, while a packet from the adaptive source is rejected with probability $i/10$ when the workload is part of $(d_i, d_{i+1}]$, with $d_0 = 0$ and $d_{11} = \infty$. Hence, the adaptive customers can be regarded as impatient with a uniform discrete distribution for their amount of patience.

Figure 3 shows the success and reject rates for the adaptive source, as well as the reject rate of the non-adaptive traffic, as a function of d_1 for various p_{up} values. For $p_{up} = 0.1$, we see that introducing 10 thresholds with $d_1 = 20$ causes a limited reduction in the accept rate of the adaptive source (and a slightly increased reject rate), while the loss rate of the background source improves dramatically.

5.3 Policing in-profile and out-of-profile traffic

The last example considers a superposition of N sources. Each source behaves as a Poisson source with rate $\lambda_1 = \rho/N$, but occasionally augments its rate by λ_2 for some time. During the periods where the rate equals $\lambda_1 + \lambda_2$, we mark each packet as being out-of-profile with probability $\lambda_2/(\lambda_1 + \lambda_2)$.

The superposition of these N sources is fed to a queue that makes use of r equidistant thresholds and in-profile packets are always accepted unless the workload exceeds d_r. Out-of-profile packets are rejected with probability $i/(5r-4i)$ when the workload is part of $(d_i, d_{i+1}]$ for $i = 0, \ldots, r$ (with $d_0 = 0$ and $d_{r+1} = \infty$). Notice, the reject probability is convex in i (as in a queue that uses random early detection [9]). The sources are not adaptive, i.e., $P_k^{(a)} = P_k^{(r)} = I$. The remaining parameters of the MMAP[K]/PH[K]/1+G[K] queue are as follows: $K = 2$, $m = N + 1$ (as it suffices to keep track of the number of sources generating traffic at rate $\lambda_1 + \lambda_2$) and

$$D_0 = \begin{bmatrix} -\rho & & & \\ & -\rho - \lambda_2 & & \\ & & \ddots & \\ & & & -\rho - N\lambda_2 \end{bmatrix} + \begin{bmatrix} -Nr_i & Nr_i & & \\ r_d & \ddots & \ddots & \\ & \ddots & \ddots & r_i \\ & & Nr_d & -Nr_d \end{bmatrix},$$

where r_i and r_d are the rates at which a source increases or decreases its arrival rate, respectively. The matrix $\Delta^{(1)} = \Delta(\rho, \ldots, \rho)$, while $\Delta^{(2)} = \Delta(0, \lambda_2, \ldots, N\lambda_2)$.

Figure 4 depicts the reject probability for a system with $r = 10$ thresholds, $N = 10$ sources, $\rho = 0.85$, $\lambda_2 = 0.05$, $r_i = 1/300$ and $r_d = 1/1000$. Two types of service are considered: an Erlang-5 distribution (with mean=1 and SCV=1/5) and a Coxian distribution with rates 2 and 1/5 with $p_1 = 1/10$ (i.e., it has a mean=1 and SCV=5). The total accepted rate increased from 0.951 for $d_1 = 1$ to 0.957 for $d_1 = 50$ for an SCV=1/5 and from 0.917 to 0.924 for an SCV=5. Thus, for d_1 small, most of the in-profile packets are accepted, while the throughput exceeds 0.85 by allowing some out-of-profile traffic.

A Appendix

To find the smallest non-negative solution Ψ and $\hat{\Psi}$ of the algebraic Riccati equation

$$F_{+-} + \Psi F_{--} + F_{++}\Psi + \Psi F_{-+}\Psi = 0,$$

and its dual (i.e., for the level-reversed queue)

$$F_{-+} + \hat{\Psi} F_{++} + F_{--}\hat{\Psi} + \hat{\Psi} F_{+-}\hat{\Psi} = 0,$$

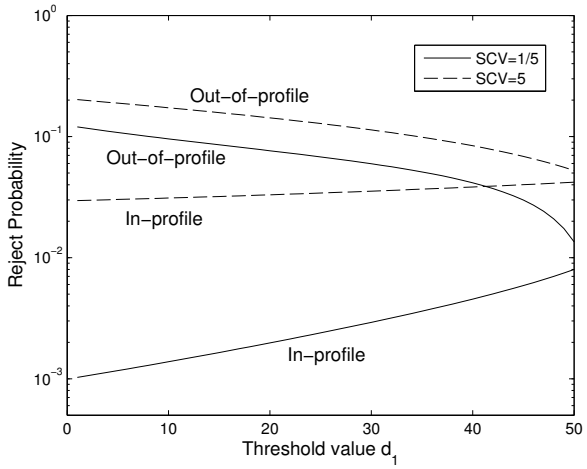

Figure 4: Reject rates of the in-profile and out-of-profile packets as a function of the the threshold d_1 in a queue with 10 equidistant thresholds with $d_{10} = 50$

we make use of the Structure-preserving Doubling Algorithm (SDA) discussed in [13]. The SDA algorithm works as follows. First define $A = -F_{++}, B = F_{+-}, C = F-+$ and $D = -F_{--}$. Next, set $\gamma = \max\{\max_i a_{ii}, \max_i d_{ii}\}$ and let $A_\gamma = A + \gamma I$ and $D_\gamma = D + \gamma I$. Further, let $W_\gamma = A_\gamma - BD_\gamma^{-1}C$ and $V_\gamma = D_\gamma - CA_\gamma^{-1}B$. Next, the SDA algorithm initializes E_0, F_0, G_0 and H_0 as $E_0 = I - 2\gamma V_\gamma^{-1}$, $F_0 = I - 2\gamma W_\gamma^{-1}$, $G_0 = 2\gamma D_\gamma^{-1} C W_\gamma^{-1}$ and $H_0 = 2\gamma W_\gamma^{-1} B D_\gamma^{-1}$. Finally, the iteration

$$\begin{aligned} E_{k+1} &= E_k(I - G_k H_k)^{-1} E_k, \\ F_{k+1} &= F_k(I - H_k G_k)^{-1} F_k, \\ G_{k+1} &= G_k + E_k(I - G_k H_k)^{-1} G_k F_k, \\ H_{k+1} &= H_k + F_k(I - H_k G_k)^{-1} H_k E_k, \end{aligned}$$

guarantees that G_k and H_k converges quadratically[1] to Ψ and $\hat{\Psi}$, respectively. The iteration is repeated until $\min(\|E_k\|_1, \|F_k\|_1) < 10^{-15}$.

The computation time of SDA can be further reduced by means of the ADDA algorithm [23], which uses the same iteration as SDA, but initializes E_0, F_0, G_0 and H_0 using two parameters $\alpha = \max_i a_{ii}$

[1]Except for the *null-recurrent* case, which never occurs in our case as $\rho^{(i)} \neq 1$ for all i.

and $\beta = \max_i d_{ii}$. For the examples presented in Section 5.1 ADDA reduced the computation time of SDA by 5 to 7 percent (where we rescaled E_k and F_k after each iteration to avoid overflows as indicated in Remark 3.1 of [23] and used $\|E_k\|_1 \|F_k\|_1 < 10^{-15}$ as a stopping criteria). The somewhat limited gain of ADDA can be understood by noting that α and β do not differ too much in this numerical example.

Alternatively, an algebraic Riccati equation can be solved using the approach taken in [17], which constructs a Quasi-Birth-Death (QBD) Markov chain such that Ψ can be recovered from the well-known G matrix. To solve this QBD any algorithm with quadratic convergence such as cyclic or logarithmic reduction [1] can be used and the runtime of these algorithms can be further reduced by exploiting the internal structure of the matrices characterizing the QBD (e.g., see [12]). Finally, an algebraic Riccati equation can also be solved with quadratic convergence using the Newton iteration [11], which requires the solution of a Sylvester matrix equation [10] of the form $AX + XB = C$ during each iteration. Even when exploiting the internal structure, both the QBD-based approach and the Newton iteration typically require slightly more than twice as much time as the SDA algorithm.

B Appendix

To compute the densities $c\pi(0)$ and $c\pi(d_i)$, for $i = 1, \ldots, r$, one first computes the first passage probability matrices

$$\Psi_{+-}^{(i)} = (\Psi^{(i)} - e^{\hat{U}^{(i)} b_i} \Psi^{(i)} e^{U^{(i)} b_i})(I - \hat{\Psi}^{(i)} e^{\hat{U}^{(i)} b_i} \Psi e^{U^{(i)} b_i})^{-1} \tag{8}$$

$$\hat{\Psi}_{-+}^{(i)} = (\hat{\Psi}^{(i)} - e^{U^{(i)} b_i} \hat{\Psi}^{(i)} e^{\hat{U}^{(i)} b_i})(I - \Psi^{(i)} e^{U^{(i)} b_i} \hat{\Psi}^{(i)} e^{\hat{U}^{(i)} b_i})^{-1} \tag{9}$$

$$\Lambda_{++}^{(i)} = (I - \Psi^{(i)} \hat{\Psi}^{(i)}) e^{\hat{U}^{(i)} b_i} (I - \Psi^{(i)} e^{U^{(i)} b_i} \hat{\Psi} e^{\hat{U}^{(i)} b_i})^{-1} \tag{10}$$

$$\hat{\Lambda}_{--}^{(i)} = (I - \hat{\Psi}^{(i)} \Psi^{(i)}) e^{U^{(i)} b_i} (I - \hat{\Psi}^{(i)} e^{\hat{U}^{(i)} b_i} \Psi e^{U^{(i)} b_i})^{-1}. \tag{11}$$

Additionally, one needs to compute $\Pi_{+-}^{(1)}$ using $\Pi_{+-}^{(r+1)} = \Psi^{(r+1)}$ and

$$\Pi_{+-}^{(i)} = \Psi_{+-}^{(i)} + \Lambda_{++}^{(i)} \Pi_{+-}^{(i+1)} (I - \hat{\Psi}_{-+}^{(i)} \Pi_{+-}^{(i+1)})^{-1} \hat{\Lambda}_{--}^{(i)},$$

for $i = r, \ldots, 1$. Analogue to [7][Theorem 3.4], the vector $cp_-(0)$ is given by

$$cp_-(0)(F_{--}^{(0)} + F_{-+}^{(0)} \Pi_{+-}^{(1)}) = 0$$

with $cp_-(0)e = 1$ and the density $c\pi_+(0)$ as $cp_-(0)F_{-+}^{(0)}$. Finally, having obtained the vectors $c(\pi_-(d_i), \pi_+(d_i))$ as indicated in Section 4.3, for $i = 1, \ldots, r$, the normalizing constant $\bar{c}$ can be computed as

$$\begin{aligned}\bar{c} &= 1 + c\pi_+(d_r)(-K^{(r+1)})^{-1}e + c\sum_{i=1}^{r}(\pi_+(d_{i-1})N_1^{(i)} + \pi_-(d_i)N_3^{(i)})A^{(i)}(d_i)\Psi^{(i)}e + \\ & c\sum_{i=1}^{r}(\pi_+(d_{i-1})N_2^{(i)} + \pi_-(d_i)N_4^{(i)})\hat{A}^{(i)}(d_i)e, \end{aligned} \tag{12}$$

where the matrices $A^{(i)}(x)$ and $\hat{A}^{(i)}(x)$, for $i = 1, \ldots, r$ can be expressed as

$$\begin{aligned}A^{(i)}(x) &= \int_{y=d_{i-1}}^{x} e^{K^{(i)}(y-d_{i-1})}dy = 1_{\{\rho^{(i)}>1\}}(x - d_{i-1})v^{(i)}u^{(i)} + \\ & (1_{\{\rho^{(i)}<1\}}(-K^{(i)})^{-1} + 1_{\{\rho^{(i)}>1\}}(-K^{(i)})^{\#})(I - e^{K^{(i)}(x-d_{i-1})}), \end{aligned} \tag{13}$$

for $x \in (d_{i-1}, d_i]$, where $u^{(i)}$ and $v^{(i)}$ are the left and right eigenvector corresponding to the eigenvalue 0 of the singular matrix $K^{(i)}$ (if $\rho^{(i)} > 1$), such that $u^{(i)}e = 1$ and $u^{(i)}v^{(i)} = 1$ and $M^{\#}$ denotes the group inverse of the matrix M. Similarly,

$$\begin{aligned}\hat{A}^{(i)}(x) &= \int_{y=d_{i-1}}^{x} e^{\hat{K}^{(i)}(d_i-y)}dy = 1_{\{\rho^{(i)}<1\}}(x - d_{i-1})\hat{v}^{(i)}\hat{u}^{(i)} + \\ & (1_{\{\rho^{(i)}>1\}}(-\hat{K}^{(i)})^{-1} + 1_{\{\rho^{(i)}<1\}}(-\hat{K}^{(i)})^{\#})(e^{\hat{K}^{(i)}(d_i-x)} - e^{\hat{K}^{(i)}b_i}), \end{aligned} \tag{14}$$

for $x \in (d_{i-1}, d_i]$, where $\hat{u}^{(i)}$ and $\hat{v}^{(i)}$ are the left and right eigenvector corresponding to the eigenvalue 0 of the singular matrix $\hat{K}^{(i)}$ (if $\rho^{(i)} < 1$), such that $\hat{u}^{(i)}e = 1$ and $\hat{u}^{(i)}\hat{v}^{(i)} = 1$.

References

[1] D. A. Bini, B. Meini, S. Steffé, and B. Van Houdt. Structured Markov chains solver: algorithms. In *SMCtools Workshop*, Pisa, Italy, 2006. ACM Press.

[2] O.J. Boxma and B.J. Prabhu. Analysis of an M/G/1 queue with customer impatience and an adaptive arrival process. Technical Report 2009-028, EURANDOM, Eindhoven, 2009.

[3] P. Buchholz, P. Kemper, and J. Kriege. Multi-class markovian arrival processes and their parameter fitting. *Performance Evaluation*, 67:1092–1106, 2010.

[4] B. D. Choi, B. Kim, and D. Zhu. MAP/M/c queue with constant impatient time. *Math. Oper. Res.*, 29:309–325, May 2004.

[5] M. Combé. Impatient customers in the MAP/G/1 queue. Technical Report BS-R9413, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1994.

[6] A. da Silva Soares and G. Latouche. Matrix-analytic methods for fluid queues with finite buffers. *Perform. Eval.*, 63:295–314, May 2006.

[7] A. da Silva Soares and G. Latouche. Fluid queues with level dependent evolution. *European Journal of Operational Research*, 196:1041–1048, 2009.

[8] T. Dzial, L. Breuer, A. da Silva Soares, G. Latouche, and M. Remiche. Fluid queues to solve jump processes. *Perform. Eval.*, 62:132–146, October 2005.

[9] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.*, 1:397–413, August 1993.

[10] G. H. Golub, S. Nash, and C. Van Loan. A Hessenberg-Schur method for the problem AX+XB=C. *IEEE Transactions on Automatic Control*, 24:909–913, 1979.

[11] C.-H. Guo. Nonsymmetric algebraic Riccati equations and Wiener–Hopf factorization for M-matrices. *SIAM J. Matrix Anal. Appl.*, 23:225–242, January 2001.

[12] C.-H. Guo. Efficient methods for solving a nonsymmetric algebraic Riccati equation arising in stochastic fluid models. *J. Comput. Appl. Math.*, 192:353–373, August 2006.

[13] C.-H. Guo, B. Iannazzo, and B. Meini. On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equation. *SIAM J. Matrix Anal. Appl.*, 29:1083–1100, 2007.

[14] Q. He. Queues with marked customers. *Adv. in Appl. Probab.*, 28:567–587, 1996.

[15] András Horváth, Gábor Horváth, and Miklós Telek. A traffic based decomposition of two-class queueing network with priority service. *Computer Networks*, 53:1235–1248, 2009.

[16] M. Mandjes, D. Mitra, and W. Scheinhardt. Models of network access using feedback fluid queues. *Queueing Syst. Theory Appl.*, 44:365–398, August 2003.

[17] V. Ramaswami. Matrix analytic methods for stochastic fluid flows. In *Teletraffic Engineering in a Competitive World - Proc. of the 16th International Teletraffic Congress (ITC 16)*, pages 1019–1030. Elsevier Science B.V., 1999.

[18] J. Herrmann S. Fomundam. A survey of queuing theory applications in healthcare. Technical Report ISR Technical Report 2007-24, University of Maryland, 2007.

[19] B. Van Houdt and C. Blondia. The waiting time distribution of a type k customer in a MMAP[K]/PH[K]/c (c=1,2) queue using QBDs. *Stochastic Models*, 20(1):55–69, 2004.

[20] J. Van Velthoven, B. Van Houdt, and C. Blondia. Response time distribution in a D-MAP/PH/1 queue with general customer impatience. *Stochastic Models*, 21:745–765, 2005.

[21] J. Van Velthoven, B. Van Houdt, and C. Blondia. On the probability of abandonment in queues with limited sojourn and waiting times. *Operations and Research Letters*, 34:333–338, 2006.

[22] Q. Wang. Modeling and analysis of high risk patient queues. *European Journal of Operational Research*, 155(2):502–515, 2004.

[23] W.-G. Wang, W.-C. Wang, and R.-C. Li. ADDA: Alternating-Directional Doubling Algorithm for M-matrix algebraic Riccati equations. Technical Report 2011-04, The University of Texas Arlington, 2011.

[24] L.B. White. A new policy evaluation algorithm for Markov decision processes with quasi birth-death structure. *Stochastic Models*, 21:785–797, 2005.