Polling with batch service

Onno Boxma[*], Jan van der Wal[†], Uri Yechiali[‡]

July 22, 2008

Abstract

This paper considers a batch service polling system. We first study the case in which the server visits the queues cyclically, considering three different service regimes: gated, exhaustive, and globally gated. We subsequently analyze the case (the so called 'Israeli Queue') in which the server first visits the queue with the 'oldest' customer. In both cases, queue lengths and waiting times are the main performance measures under consideration.

1 Introduction

A polling system is a collection of queues, say $Q_1, \ldots, Q_N$, attended to by a single server. It is usually assumed that the server visits the queues in a cyclic order: $Q_1, \ldots, Q_N, Q_1, \ldots$, employing some service discipline like exhaustive, gated or 1-limited service to serve the customers at the various queues. There exists a remarkably sharp distinction in the complexity of the analysis of polling systems. If the service discipline of the server at each queue satisfies a certain branching property, that allows the joint queue length process to be represented by a multi-type branching process, then a detailed analysis is possible; cf. [7] and [12]. The exhaustive and gated service policies do satisfy this branching property, but the 1-limited service policy does not. Relatively little is known for polling systems in which the branching property is violated, although so-called pseudo-conservation laws have been obtained for a very general class of polling systems [4], which in turn have given rise to rather accurate approximations of, in particular, mean waiting times.

Polling systems find their applications in a wide range of fields, and accordingly there is a huge literature on the performance analysis of polling systems; see several surveys of

[*]Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands. E-mail: o.j.boxma@tue.nl

[†]Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands. E-mail: jan.v.d.wal@tue.nl

[‡]Dept. of Statistics and OR, Tel Aviv University, Israel. E-mail: uriy@post.tau.ac.il

Takagi, like [15]. Remarkably, hardly any papers have been devoted to polling systems in which the customers are not served individually but in batches. That is the topic of the present paper.

In some applications, such 'unlimited' batch service arises quite naturally. Think of a manufacturing system in which the server is an oven in which all available items of a particular type may be heated at the same time, or a paint bath in which all available items of a particular type, say textiles, may be painted simultaneously. Certain road traffic situations (including the transportation of a group of items) and computer-communications protocols may also reasonably accurately be modeled via a polling system with batch service. Below we review a few studies in which polling systems with batch service were studied, motivated by applications in computer-communications.

Unlimited batch service models are considered in the context of teletext, videotex and TDMA systems, as well as for central data-base operations. Ammar and Wong [2] studied a teletext system with N queues, fed by independent Poisson arrival streams. Service times in all queues are deterministic (slotted, unit time each), there are no switch-over times, and the service discipline is locally gated. They showed that the policy which minimizes mean response time is of a cyclic nature, with cycle length $L \geq N$ slots, in which queue i is visited k_i times, where $\sum_{i=1}^{N} k_i = L$. Liu and Nain [10] examined a TDMA model with both the locally gated and exhaustive regimes for the case of zero switching times and homogeneous arrival process to all queues. Dykeman et al. [6] used Howard's policy-iteration algorithm to control a videotex system. They indicated that, even with equal and deterministic service requirements, and with no switching times, the structure of the optimal policy could be very complicated. Van Oyen and Teneketzis [11] formulated both a central data base system and an Automated Guided Vehicle in a manufacturing system as a polling system with an infinite-capacity batch service and zero switching times, where the controller observes only the length of the queue at which the server is located. Van der Wal and Yechiali [19] explored dynamic server's visit-order policies in non-symmetric polling systems with switch-in and switch-out times, where service is in batches of unlimited size. They concentrated on so-called 'Hamiltonian tour' policies in which - in order to give a fair treatment to the various queues - the server attends every non-empty queue exactly once during each cycle. The server then dynamically generates a new visit schedule at the start of each round, depending on the current state of the system and on the various non-homogeneous system parameters.

Model description

We shall study the following polling model. A single server S visits N queues $Q_1, \ldots, Q_N$. Customers arrive at these queues according to independent Poisson processes, with rate λ_i at Q_i, $i = 1, \ldots, N$. S serves customers at Q_i in a batch. The service time of this batch is a random variable, that we shall generically denote by B_i, with Laplace-Stieltjes Transform (LST) $\tilde{B}_i(\cdot)$. We consider several service disciplines. The gated discipline operates as follows. If, upon the arrival of S at Q_i, there are $X_i^i > 0$ customers present at Q_i, then S serves exactly those customers, in one batch, and then moves to the next queue. The

exhaustive service discipline operates as follows. If, after the batch service of X_i^i, Q_i is still empty, then S moves to the next queue. Otherwise, S serves all new waiting customers at Q_i in one batch, requiring another independent service time with the same distribution as B_i. S continues serving such batches until Q_i has become empty. Under the globally gated discipline, S starts the cycle at Q_1, recording X_1^j customers present at that moment in Q_j, $j = 1, \ldots, N$. Then, when visiting Q_j thereafter, only those X_1^j customers are served in one batch. All jobs that arrive during the cycle will be served in the next cycle. In all cyclic disciplines, when S leaves Q_i, he switches to Q_{i+1}. The switch-over time of S from Q_i to Q_{i+1} is a random variable that we shall generically denote by D_i, with LST $\tilde{D}_i(\cdot)$. We shall furthermore make all the usual independence assumptions regarding the involved inter-arrival intervals, service times and switch-over times.

Outline of the paper
In sections 2, 3 and 4 we study three variants in which the queues are polled cyclically. Section 2 deals with the case in which all queues are served according to the gated discipline. We refer to the corresponding model as the *locally gated* model. In section 3 the case is treated in which all queues are served *exhaustively*. In both sections we derive the Probability Generating Function (PGF) of the joint queue length distribution as well as the LST of the waiting time distribution at each queue. Section 4 considers the *globally gated* case. We derive cycle time and waiting time distributions. Section 5 deals with the non-cyclical variant in which after the visit of a queue the next queue to be visited is the one with the most senior job, thus a First-Come-First-Served (FCFS) polling variant.

2 Locally gated batch service

2.1 Preliminaries

The single server S cyclically visits N queues $Q_1, \ldots, Q_N$. When S visits Q_i, he serves all customers present in one batch, and then moves to Q_{i+1}. For this locally gated batch-service polling model, we determine the PGF of the joint steady-state queue length distribution, as well as the LST of the waiting time distribution of a class-i customer, $i = 1, \ldots, N$. Let us now introduce some further notation. In the sequel, $I_{[\cdot]}$ shall denote an indicator function. Furthermore, for $i = 1, \ldots, N$:

$$\begin{aligned} A_i(t) &= \text{number of arrivals to } Q_i \text{ during a time interval of length } t. \\ X_i^j &= \text{number of jobs in queue } Q_j \text{ when } Q_i \text{ is polled.} \\ V_i &= V_i(X_i^i) = B_i I_{[X_i^i > 0]} = \text{the visit time of } S \text{ to } Q_i. \\ G_i(z_1, \ldots, z_N) &= \mathbb{E}\,[\prod_{j=1}^{N} z_j^{X_i^j}]. \end{aligned}$$

It is easily seen that the following "laws of motion" hold for the X_i^j:

$$\begin{aligned} X_{i+1}^j &= X_i^j + A_j(V_i(X_i^i)) + A_j(D_i), \quad j \neq i, \\ X_{i+1}^j &= A_j(V_i(X_i^i)) + A_j(D_i), \quad j = i. \end{aligned} \tag{1}$$

While we present these laws of motion in terms of steady-state quantities, in reality we are expressing the number of jobs in Q_j at the nth visit of S to Q_{i+1} into that at Q_j at the nth visit of S to Q_i. So we look one queue ahead. By doing this N successive times, we can express the number of jobs in Q_j at the $(n+1)$th visit of S to Q_i into those at the nth visit of S to Q_i.

Introducing $\sigma(z_1, \ldots, z_N) = \sum_{j=1}^N \lambda_j(1-z_j)$, it follows that, for $i = 1, \ldots, N$ (with $G_{N+1} = G_1$):

$$\begin{aligned} G_{i+1}(z_1, \ldots, z_N) &= \mathbb{E}[z_1^{X_i^1} \ldots z_{i-1}^{X_i^{i-1}} z_{i+1}^{X_i^{i+1}} \ldots z_N^{X_i^N} I_{[X_i^i > 0]}] \tilde{B}_i(\sigma(z_1, \ldots, z_N)) \tilde{D}_i(\sigma(z_1, \ldots, z_N)) \\ &+ \mathbb{E}[z_1^{X_i^1} \ldots z_{i-1}^{X_i^{i-1}} z_{i+1}^{X_i^{i+1}} \ldots z_N^{X_i^N} I_{[X_i^i = 0]}] \tilde{D}_i(\sigma(z_1, \ldots, z_N)) \\ &= G_i(z_1, \ldots, z_{i-1}, 1, z_{i+1}, \ldots, z_N) \tilde{B}_i(\sigma(z_1, \ldots, z_N)) \tilde{D}_i(\sigma(z_1, \ldots, z_N)) \\ &+ G_i(z_1, \ldots, z_{i-1}, 0, z_{i+1}, \ldots, z_N)[1 - \tilde{B}_i(\sigma(z_1, \ldots, z_N))] \tilde{D}_i(\sigma(z_1, \ldots, z_N)). \end{aligned} \tag{2}$$

To develop insight into the structure of the solution of this recursion, we first consider the special case of $N = 2$ queues in Subsection 2.2; the general case will subsequently be solved in Subsection 2.3.

2.2 The two-queue case

For $N = 2$, Formula (2) becomes:

$$\begin{aligned} G_1(z_1, z_2) &= G_2(z_1, 1) \tilde{B}_2(\sigma(z_1, z_2)) \tilde{D}_2(\sigma(z_1, z_2)) \\ &+ G_2(z_1, 0)[1 - \tilde{B}_2(\sigma(z_1, z_2))] \tilde{D}_2(\sigma(z_1, z_2)), \end{aligned} \tag{3}$$

$$\begin{aligned} G_2(z_1, z_2) &= G_1(1, z_2) \tilde{B}_1(\sigma(z_1, z_2)) \tilde{D}_1(\sigma(z_1, z_2)) \\ &+ G_1(0, z_2)[1 - \tilde{B}_1(\sigma(z_1, z_2))] \tilde{D}_1(\sigma(z_1, z_2)). \end{aligned} \tag{4}$$

It follows from (4) that $G_2(z_1, 1)$ is expressed in $G_1(1, 1)$ and $G_1(0, 1)$; similarly, $G_2(z_1, 0)$ is expressed in $G_1(1, 0)$ and $G_1(0, 0)$. By substituting (4) with $z_2 = 1$ (respectively, $z_2 = 0$) into (3), we are able to express $G_1(z_1, z_2)$ into known terms plus the four unknown constants $G_1(1, 1)$ (which actually equals 1), $G_1(0, 1)$, $G_1(1, 0)$ and $G_1(0, 0)$:

$$\begin{aligned} G_1(z_1, z_2) &= \{G_1(1, 1) \tilde{B}_1(\sigma(z_1, 1)) \tilde{D}_1(\sigma(z_1, 1)) \\ &+ G_1(0, 1)[1 - \tilde{B}_1(\sigma(z_1, 1))] \tilde{D}_1(\sigma(z_1, 1))\} \\ &\times \tilde{B}_2(\sigma(z_1, z_2)) \tilde{D}_2(\sigma(z_1, z_2)) \\ &+ \{G_1(1, 0) \tilde{B}_1(\sigma(z_1, 0)) \tilde{D}_1(\sigma(z_1, 0)) \\ &+ G_1(0, 0)[1 - \tilde{B}_1(\sigma(z_1, 0))] \tilde{D}_1(\sigma(z_1, 0))\} \\ &\times [1 - \tilde{B}_2(\sigma(z_1, z_2))] \tilde{D}_2(\sigma(z_1, z_2)). \end{aligned} \tag{5}$$

It remains to determine $G_1(0,1)$, $G_1(1,0)$ and $G_1(0,0)$. Those three unknown constants may be found by the substitutions $\{z_1 = 0, z_2 = 1\}$, $\{z_1 = 1, z_2 = 0\}$ and $\{z_1 = 0, z_2 = 0\}$ into (5), resulting in three linear equations with three unknowns.

The above yields the following insight. To determine the PGF $G_i(z_1, z_2)$, what really matters is whether a queue is empty or not when server S visits it. If it is non-empty, the actual queue size does not have an effect on the visit time. Hence the joint queue length distribution at a visit epoch of S at, say, Q_1 is determined by the four possible events *both Q_1 and Q_2 non-empty at the last previous visit of S to Q_1*, ..., *both Q_1 and Q_2 empty at the last previous visit of S to Q_1*. Q_1 being non-empty at the previous visit has probability $\mathbb{P}(X_1^1 > 0) = G_1(1,1) - G_1(0,1) = 1 - G_1(0,1)$, etc.

It should be noticed that the process $\{(U_1^{(n)}, U_2^{(n)}),\ n = 1, 2, \dots\}$, with $U_i^{(n)} = 1$ (0) denoting that Q_i is non-empty (resp., empty) at the nth visit of S to Q_1 is a two-dimensional Markov chain. This Markov chain is irreducible, aperiodic and positive-recurrent, and hence has a unique non-negative steady-state solution. With an obvious notation, we have: $\mathbb{P}(U_1 = 1, U_2 = 1) = 1 - G_1(1,0) - G_1(0,1) + G_1(0,0)$, ..., $\mathbb{P}(U_1 = 0, U_2 = 0) = G_1(0,0)$.

2.3 The N-queue case

The insight obtained in the previous subsection for the case of 2 queues readily allows us to obtain the structure of the solution of the case of an arbitrary number of queues. N successive substitutions of (2) result in an expression of $G_1(z_1, \dots, z_N)$ into the 2^N unknown constants $G_1(1, 1, \dots, 1)$, ..., $G_1(0, 0, \dots, 0)$. These 2^N constants (of which the first actually equals 1) can be obtained by determining the unique steady-state solution of an N-dimensional irreducible, aperiodic and positive-recurrent Markov chain $\{(U_1^{(n)}, \dots, U_N^{(n)}),\ n = 1, 2, \dots\}$, with $U_i^{(n)} = 1$ (0) denoting that Q_i is non-empty (resp. empty) at the nth polling instant of S to Q_1.

The rationale behind this solution structure is that, for determining the steady-state joint queue length distribution at a visit of S to Q_1, what really matters is whether $Q_1, \dots, Q_N$ were empty or not at the last previous visit of S to Q_1; not what their actual queue lengths were. The probabilities of those events are obtained by solving an N-dimensional Markov chain with 2^N states.

Remark 2.1
It easily follows from (1) that the mean number of customers in Q_j when S polls Q_i, $f_i^j := \mathbb{E}X_i^j$, satisfies (with $\mathbb{E}V_i$ the mean visit period of S at Q_i):

$$\begin{aligned} f_{i+1}^j &= f_i^j + \lambda_j \mathbb{E}V_i + \lambda_j \mathbb{E}D_i, \quad j \neq i, \\ f_{i+1}^j &= \lambda_j \mathbb{E}V_i + \lambda_j \mathbb{E}D_i, \quad j = i. \end{aligned} \tag{6}$$

Summing (6) over all i yields:

$$f_j^j = \lambda_j \sum_{i=1}^{N} (\mathbb{E}V_i + \mathbb{E}D_i), \tag{7}$$

where $\mathbb{E}V_i = \mathbb{P}(X_i^i > 0)\mathbb{E}B_i = [G_i(1,\ldots,1,\ldots,1) - G_i(1,\ldots,0,\ldots,1)]\mathbb{E}B_i$, the 1 (resp. 0) appearing at the ith position. Notice that those $G_i(\ldots)$ have to be determined via the solution of a Markov chain, as discussed above. Also notice that f_j^j equals the mean number of arrivals at Q_j during one cycle time and that, via (6), f_i^j is readily expressed in f_j^j and the mean visit periods at $Q_j, \ldots, Q_{i-1}$. In particular, focussing on the number of customers in Q_1, f_1^1 is given by (7) while

$$f_i^1 = \lambda_1 \sum_{k=1}^{i-1} (\mathbb{E}V_k + \mathbb{E}D_k), \quad i = 2, \ldots, N. \tag{8}$$

Remark 2.2
Once the PGF $G_i(z_1, \ldots, z_N)$ of the joint queue length distribution when S polls Q_i has been determined for $i = 1, \ldots, N$, it is straightforward to derive the PGF of the joint queue length distribution at the instant at which S begins a switch-over time from Q_i to the next queue, $i = 1, \ldots, N$. Subsequently, it is not hard to determine the PGF of the joint queue length distribution *during* a visit to Q_i (respectively, the PGF of the joint queue length distribution *during* a switch from Q_i to Q_{i+1}). Taking an appropriate weighted average, one finally obtains the PGF of the joint steady-state queue length distribution, and hence also the mean steady-state queue length at any queue Q_i. Application of Little's formula yields the mean time a type-i customer spends in the system (waiting plus in service). One can use the above-discussed queue-length PGF's to also determine the (LST of the) waiting time *distribution*. We refer to [5] for a sketch of how this may be done.

3 Exhaustive batch service

The model studied in this section differs from the batch-service gated polling model of the previous section in only one respect: When S visits a non-empty queue, and the queue has not become empty at the end of a batch service, then S performs yet another batch service for those customers who have arrived at that queue during the previous batch service, and so on, until the queue has become empty. It follows from [19] that the LST $\phi_i(s)$ of a non-zero visit period of Q_i is now given by

$$\phi_i(s) = \frac{\tilde{B}_i(s+\lambda_i)}{1-\tilde{B}_i(s)+\tilde{B}_i(s+\lambda_i)}. \tag{9}$$

The "laws of motion" for the numbers of customers X_i^j at Q_j when S visits Q_i are now given by:

$$\begin{aligned} X_{i+1}^j &= X_i^j + A_j(V_i(X_i^i)) + A_j(D_i), \quad j \neq i, \\ X_{i+1}^j &= A_j(D_i), \quad j = i. \end{aligned} \tag{10}$$

This leads to the following recursion for the PGF's $G_i(z_1,\ldots,z_N)$ of the numbers of customers at the various queues when S arrives at Q_i: With $\sigma_i(z_1,\ldots,z_N) := \sum_{j\neq i}\lambda_j(1-z_j)$,

$$\begin{aligned} G_{i+1}(z_1,\ldots,z_N) &= \mathbb{E}[z_1^{X_i^1}\ldots z_{i-1}^{X_i^{i-1}} z_{i+1}^{X_i^{i+1}}\ldots z_N^{X_i^N} I_{[X_i^i>0]}]\phi_i(\sigma_i(z_1,\ldots,z_N))\tilde{D}_i(\sigma(z_1,\ldots,z_N)) \\ &+ \mathbb{E}[z_1^{X_i^1}\ldots z_{i-1}^{X_i^{i-1}} z_{i+1}^{X_i^{i+1}}\ldots z_N^{X_i^N} I_{[X_i^i=0]}]\tilde{D}_i(\sigma(z_1,\ldots,z_N)) \\ &= G_i(z_1,\ldots,z_{i-1},1,z_{i+1},\ldots,z_N)\phi_i(\sigma_i(z_1,\ldots,z_N))\tilde{D}_i(\sigma(z_1,\ldots,z_N)) \\ &+ G_i(z_1,\ldots,z_{i-1},0,z_{i+1},\ldots,z_N)[1-\phi_i(\sigma_i(z_1,\ldots,z_N))]\tilde{D}_i(\sigma(z_1,\ldots,z_N)). \end{aligned} \tag{11}$$

The PGF $G_i(z_1,\ldots,z_N)$ can be solved in exactly the same way as for the gated case, expressing it into the 2^N constants $G_1(1,1,\ldots,1)$, $\ldots$, $G_1(0,0,\ldots,0)$.

Remark 3.1
It follows from (11) that the mean numbers of customers $f_i^j := \mathbb{E}X_i^j$ satisfy (with $\mathbb{E}V_i$ the mean visit period of S at Q_i):

$$\begin{aligned} f_{i+1}^j &= f_i^j + \lambda_j\mathbb{E}V_i + \lambda_j\mathbb{E}D_i, \quad j \neq i, \\ f_{i+1}^j &= \lambda_j\mathbb{E}D_i, \quad j = i. \end{aligned} \tag{12}$$

Summing (12) over all i yields:

$$f_j^j = \lambda_j\sum_{i\neq j}(\mathbb{E}V_i + \mathbb{E}D_i) + \lambda_j\mathbb{E}D_j, \tag{13}$$

where $\mathbb{E}V_i = \mathbb{E}B_i\frac{\mathbb{P}(X_i^i>0)}{\tilde{B}_i(\lambda_i)} = \mathbb{E}B_i\frac{G_i(1,\ldots,1,\ldots,1)-G_i(1,\ldots,0,\ldots,1)}{\tilde{B}_i(\lambda_i)}$, with a 1 (resp., 0) in G_i appearing at the ith position. Again, those $G_i(\ldots)$ have to be determined via the solution of a

Markov chain.

Remark 3.2
The cyclic polling model with exhaustive batch service is closely related to the cyclic polling model with a single buffer at each station. In the latter model, there can be at most one customer in each station, and customers finding a full buffer are rejected. The similarity becomes clear by identifying the visit time at Q_i in our exhaustive batch service model with the service time at Q_i in the single buffer model. In the analysis of the single buffer model, the Markov process of numbers of customers (0 or 1) at particular embedded epochs like server visit epochs or service completion epochs plays a key role. We refer to [14, 17] (non-zero switchover times) and [18] (zero switchover times). See also Takine et al. [16] who analyse the departure process of a symmetric polling system with a single buffer at each station, and Lee & Sunjaya [9] who study a random polling system with single buffers and correlated inputs.

Remark 3.3
Resing [12] has pointed out that there exists a close connection between multi-class branching processes and polling models with the following service discipline: When S visits a queue, it treats all customers initially present at that queue in stochastically the same way. E.g., it serves exactly those customers and nobody else (gated service), or it serves those customers, plus all those arriving during their service, and those arriving during the services of those, etc. (exhaustive service). Binomial gated also falls in this "multitype branching" class: Each customer is included in the batch with the same probability p. If we allow binomial gated in our batch service case, then we loose the nice feature that it is sufficient to know the probability that a queue is empty upon its visit. We do retain that feature if all those present upon the visit of S are served in one batch service, while each of those who have arrived during that batch service are served in a second batch service with the same probability p. More generally: If we are able to determine the PGF of the number of customers who are present in Q_i at the end of a non-zero visit of S to Q_i, and that number does not depend on the number of customers present at the beginning of that visit, then the approach of this and the previous section can be applied. In particular, our approach can in principle be applied when S serves a batch of all customers present at the beginning of his visit, and subsequently applies some - any - rule to proceed after that batch service, as long as this rule does not depend on the number of customers in that first batch. Formula (2) then is still valid, when we replace $\tilde{B}_i(\sigma(z_1, \ldots, z_N))$ by $\tilde{V}_i(\tilde{\sigma}(z_1, \ldots, z_N))$, where $\tilde{V}_i$ denotes the LST of the length of a non-empty visit period and $\tilde{\sigma}$ is a function that has to be specified. We'll briefly outline the approach for the case in which those who have arrived during the first batch service in Q_i are, with a fixed

probability p_i, served during a second batch service. Then

$$\begin{aligned}\tilde{V}_i(\tilde{\sigma}(z_1,\ldots,z_N)) &= \tilde{B}_i(\sum_{j\neq i}\lambda_j(1-z_j)+\lambda_i p_i+\lambda_i(1-p_i)(1-z_i))\\ &+ [\tilde{B}_i(\sum_{j\neq i}\lambda_j(1-z_j)+\lambda_i(1-p_i)(1-z_i)) \qquad (14)\\ &- \tilde{B}_i(\sum_{j\neq i}\lambda_j(1-z_j)+\lambda_i p_i+\lambda_i(1-p_i)(1-z_i))]\tilde{B}_i(\sum_{j=1}^{N}\lambda_j(1-z_j)).\end{aligned}$$

The first term in the RHS corresponds to the case in which the first batch service is not followed by a second batch service: Either there was no arrival at Q_i, or if there were arrivals, they were not chosen for service. Notice that the customers who arrive at Q_i during a first batch service and are *not* chosen for service form a Poisson process with rate $\lambda_i(1-p_i)$. The second term in the RHS corresponds to the case in which the first batch service *is* followed by a second batch service. Its final factor, $\tilde{B}_i(\sum_{j=1}^{N}\lambda_j(1-z_j))$, denotes the generating function of the joint distribution of numbers of arrivals at all N queues during that second batch service.

We end this section with the observation that it should be apparent from the foregoing that our approach also allows one to analyse a cyclic batch-service polling model in which some queues are served according to the gated discipline and others according to the exhaustive discipline.

4 Globally gated; cycle time and waiting time

In this section we consider the Globally Gated regime. Again, the queues are visited in cyclic order: $Q_1, Q_2, \ldots, Q_N, Q_1$, etc. Whenever the server visits or passes Q_i he always needs a switch-over time D_i, so the total switch-over time in a cycle is the sum over all D_i, even if one or more of the queues are empty and will not be served. The sum of the switch-over times is denoted by D, $D = \sum_{i=1}^{N} D_i$, with $D(t)$ as its distribution and $\tilde{D}(\cdot)$ as its LST. In each cycle S serves only those queues that are non-empty at the start of the cycle. So the cycle duration is D plus the sum of the B_i over the queues that are non-empty. We shall obtain the cycle and waiting time distributions.

4.1 Queues visited in a cycle

Let A be the set of indices of the queues visited in the present cycle. So A can be any, possibly empty, subset of $\{1, 2, \ldots, N\}$. Define B_A to be the total visit time to the queues with indices in A, with $B_A(t)$ as its distribution and $\tilde{B}_A(\cdot)$ as its LST.

Define $p_{AA'}$ to be the probability that in the next cycle exactly the queues belonging (with a slight abuse of notation) to A' will be visited, so none of the queues not in A', given that in the present cycle the queues in A are visited.

Then

$$p_{AA'} = \int_{x=0}^{\infty} \int_{y=0}^{\infty} \prod_{i \in A'} (1 - e^{-\lambda_i (x+y)}) \prod_{j \notin A'} e^{-\lambda_j (x+y)} dD(x) dB_A(y).$$

4.2 Cycle time distribution

Given the total duration, t say, of cycle n, the duration of cycle $n+1$ is just the sum of the switch-over times and of the service times of the non-empty queues. Let the random variable C_n denote the n-th cycle duration, $C_n(t)$ be its distribution and $\tilde{C}_n(\cdot)$ its LST. Then

$$\begin{aligned} E[e^{-\omega C_{n+1}} | C_n = t] &= \tilde{D}(\omega) \prod_{i=1}^{N} \left[(1 - e^{-\lambda_i t}) \tilde{B}_i(\omega) + e^{-\lambda_i t} \right] \\ &= \tilde{D}(\omega) \prod_{i=1}^{N} \left[\tilde{B}_i(\omega) + e^{-\lambda_i t} (1 - \tilde{B}_i(\omega)) \right] . \end{aligned} \tag{15}$$

And without the conditioning

$$\begin{aligned} \tilde{C}_{n+1}(\omega) = E[e^{-\omega C_{n+1}}] &= \tilde{D}(\omega) \int_0^{\infty} \prod_{i=1}^{N} \left[\tilde{B}_i(\omega) + e^{-\lambda_i t} (1 - \tilde{B}_i(\omega)) \right] dC_n(t) \\ &= \tilde{D}(\omega) \sum_{A \subset \{1,\ldots,N\}} \int_0^{\infty} \prod_{i \in A} \tilde{B}_i(\omega) \prod_{j \notin A} (1 - \tilde{B}_j(\omega)) \, e^{-\sum_{l \notin A} \lambda_l t} dC_n(t) \\ &= \tilde{D}(\omega) \sum_{A \subset \{1,\ldots,N\}} \prod_{i \in A} \tilde{B}_i(\omega) \prod_{j \notin A} (1 - \tilde{B}_j(\omega)) \, \tilde{C}_n(\Sigma_{l \notin A} \lambda_l) . \end{aligned} \tag{16}$$

In steady state, denote the LST of the cycle time distribution by $\tilde{C}(\cdot)$. It satisfies (16) with $\tilde{C}_n(\cdot)$ replaced by $\tilde{C}(\cdot)$. As $\tilde{B}(\omega)$ is known, $\tilde{C}(\omega)$ is known once the coefficients $\tilde{C}(\Sigma_{i \notin A)} \lambda_i)$ are known. To obtain these coefficients one may substitute $\omega = \Sigma_{i \notin A} \lambda_i$ for all A which results in a linear system of 2^N equations from which (in principle) the $\tilde{C}(\Sigma_{i \notin A} \lambda_i)$ can be computed.

The fully symmetric case

In the fully symmetric case, i.e., with equal λ_i and equal $B_i(\cdot)$, (16) simplifies to

$$\tilde{C}_{n+1}(\omega) = \tilde{D}(\omega) \sum_{l=0}^{N} \binom{N}{l} \tilde{B}(\omega)^{N-l} (1 - \tilde{B}(\omega))^l \tilde{C}_n(\lambda l) .$$

In this case, instead of having to solve a system with 2^N equations, we only get N equations for the N unknowns $\tilde{C}(\lambda l)$ for $l = 1, \ldots, N$.

Empty cycles
So far we assumed that the server continues his trip along the queues even if it is known that the next cycle will be 'empty', i.e., all queues are empty at the beginning of the cycle. An alternative assumption would be that, if at the end of a cycle all queues are empty, the server waits for the first arrival and only then starts the next cycle (in which thus exactly one non-empty queue will be visited).

In this case the expressions of the previous section have to be modified a bit. The term $\tilde{D}(\omega) \prod_{i=1}^{N} e^{-\lambda_i t}$ in (15) that corresponds to the next cycle starting with all queues empty has to be replaced by

$$\tilde{D}(\omega) e^{-\sum_{i=1}^{N} \lambda_i t} \sum_{j=1}^{N} \frac{\lambda_j}{\sum_{l=1}^{N} \lambda_l} \tilde{B}_j(\omega).$$

Of course, this modification also changes Equation (16).

4.3 Waiting times

Next let us consider the waiting time of an arbitrary customer. Recall that the queues are always visited in the order $Q_1, Q_2, \ldots, Q_N$, skipping empty queues. Consider a Q_m job. Now condition on the duration t of the cycle in which the job arrives. For an arbitrary arrival the density of its arrival cycle duration is $\frac{tdC(t)}{E(C)}$. Further, given the present cycle duration t, the probability that queue $l < m$ will be visited in the next cycle is $1 - e^{-\lambda_l t}$.

The waiting time W_m of a job of class m consists of two parts: the residual duration of the arrival cycle and the waiting time in the next cycle. By (our) definition, a cycle starts with a service time for Q_1 or, if Q_1 is empty, with the switch-over time D_1 from Q_1 to Q_2. Conditioning on the cycle duration t, the first part is $U[0, t]$ distributed (uniform on $[0, t]$). The second part consists of the switch-over times needed to reach Q_m and the $m - 1$ visit times (with possibly 0 duration) to the queues 1 up to $m - 1$.

This results in:

$$
\begin{aligned}
Ee^{-\omega W_m} &= \prod_{i=1}^{m-1} \tilde{D}_i(\omega) \int_0^\infty \left(\frac{1-e^{-\omega t}}{t\omega}\right) \prod_{l=1}^{m-1} \left((1-e^{-\lambda_l t})\tilde{B}_l(\omega) + e^{-\lambda_l t}\right) \frac{t}{E(C)} dC(t) \\
&= \prod_{i=1}^{m-1} \tilde{D}_i(\omega) \int_0^\infty \left(\frac{1-e^{-\omega t}}{t\omega}\right) \prod_{l=1}^{m-1} \left(\tilde{B}_l(\omega) + (1-\tilde{B}_l(\omega))e^{-\lambda_l t}\right) \frac{t}{E(C)} dC(t) \\
&= \prod_{i=1}^{m-1} \tilde{D}_i(\omega) \frac{1}{\omega E(C)} \int_0^\infty (1-e^{-\omega t}) \sum_{A \subset \{1,\ldots,m-1\}} \prod_{l \in A} \tilde{B}_l(\omega) \prod_{k \notin A} (1-\tilde{B}_k(\omega)) e^{-\sum_{l \notin A} \lambda_l t} dC(t) \\
&= \prod_{i=1}^{m-1} \tilde{D}_i(\omega) \frac{1}{\omega E(C)} \sum_{A \subset \{1,\ldots,m-1\}} \prod_{l \in A} \tilde{B}_l(\omega) \prod_{k \notin A} (1-\tilde{B}_k(\omega)) \left(\tilde{C}(\Sigma_{l \notin A}\lambda_l) - \tilde{C}(\omega + \Sigma_{l \notin A}\lambda_l)\right) .
\end{aligned}
$$

In case all λ_l and all $B_l(\cdot)$ are equal, this simplifies to

$$
Ee^{-\omega W_m} = \prod_{i=1}^{m-1} \tilde{D}_i(\omega) \frac{1}{\omega E(C)} \sum_{l=0}^{m-1} \binom{m-1}{l} \tilde{B}(\omega))^{m-1-l} (1-\tilde{B}(\omega))^l \left(\tilde{C}(\lambda l) - \tilde{C}(\omega + \lambda l)\right) .
$$

4.4 Mean waiting times and elevator polling

The (mean) waiting time of a class m job consists of three parts. The first part is the residual duration of the cycle in which the job arrives. The second part is the sum of the switch-over times in the next cycle before Q_m is reached. The third part is the sum of the visiting times to the non-empty queues among Q_1 up to Q_{m-1}. This results in:

$$
\begin{aligned}
EW_m &= EC^{res} + \sum_{i=1}^{m-1} ED_i + \sum_{i=1}^{m-1} \int_0^\infty (1-e^{-\lambda_i t}) EB_i \frac{t}{EC} dC(t) \qquad (17) \\
&= EC^{res} + \sum_{i=1}^{m-1} ED_i + \sum_{i=1}^{m-1} EB_i (1 + \frac{d}{d\lambda_i} \frac{\tilde{C}(\lambda_i)}{EC})
\end{aligned}
$$

From this we immediately see that, as expected, the mean waiting time for jobs of class m is increasing in m.

Elevator polling

To increase fairness one might use elevator polling [1]. In elevator polling the queues are visited in the order Q_1 up to Q_N in the odd cycles and in the order Q_N down to Q_1 in the even cycles. So the visit order is $Q_1, Q_2, \ldots, Q_N, Q_N, Q_{N-1}, \ldots, Q_1, Q_1, Q_2$, etc. It is assumed that the switch-over times from Q_i to Q_{i+1} and from Q_{i+1} to Q_i are equal and both equal to D_i. When changing directions, the switch-over times from Q_N to Q_N

and from Q_1 to Q_1 are zero. Then for the globally gated case the characteristics of 'up' cycles $(Q_1, Q_2, \ldots, Q_N)$ and 'down' cycles $(Q_N, Q_{N-1}, \ldots, Q_1)$ are identical; the order has no effect on the duration, hence no effect on the next cycle. To simplify the notation let us write $\tilde{C}'(\lambda_i) = \frac{d}{d\lambda_i}\tilde{C}(\lambda_i)$. Then, using the expression (17) for both service orders, we get the following mean waiting times for elevator polling :

$$\begin{aligned} EW_m &= EC^{res} + \frac{1}{2}\left[\sum_{i=1}^{m-1} ED_i + \sum_{i=1}^{m-1} EB_i(1 + \frac{\tilde{C}'(\lambda_i)}{EC}) + \sum_{i=m}^{N-1} ED_i + \sum_{i=m+1}^{N} EB_i(1 + \frac{\tilde{C}'(\lambda_i)}{EC})\right] \\ &= EC^{res} + \frac{1}{2}\left[\sum_{i=1}^{N-1} ED_i + \sum_{i=1}^{N} EB_i(1 + \frac{\tilde{C}'(\lambda_i)}{EC}) - EB_m(1 + \frac{\tilde{C}'(\lambda_m)}{EC})\right] . \end{aligned}$$

So, for elevator polling the only difference between the mean waiting times of the queues is in the term $EB_m(1 + \frac{\tilde{C}'(\lambda_m)}{EC})$. From this we see that if all EB_m and all λ_m are equal, then so are the mean waiting times. But even if the EB_m and λ_m are different then the differences between the mean waiting times of the various queues are much smaller than in (17).

5 FCFS among queues (the Israeli queue)

In this section it is assumed that the order in which the queues are served is not cyclic but $FCFS$, i.e., based upon the first arrival in each non-empty queue. In other words, once the server concludes a visit to a queue, the next queue to visit is the one with the most senior job, i.e., the job that arrived first among all jobs present. As before, the server provides batch service, and the number of jobs in a queue is not relevant for the batch service time. Switch-over (set-up) times are assumed to be 0. Alternatively, if there is a set-up time when a non-empty queue is polled, this set-up time is assumed to be part of the batch-service time. We restrict ourselves to the case that all batch service times B_i are i.i.d. like B, exponentially distributed with mean $1/\mu$, and that all classes have the same arrival rate λ (see also Remarks 5.1 and 5.2). Some thought will reveal (see below) that the model under consideration has the same structure as the classical 'machine-repairman' model, a relation that has also been observed and used by Takagi [14] in a related model of single-message buffers.

We focus our attention on the distribution of the waiting time of an arbitrary customer. In Subsection 5.1 the case of gated service, where gating is done at the *end* of the service time, will be studied. Subsections 5.2 and 5.3 respectively consider the cases of exhaustive service and of gated service, with gating at the *beginning* of service times.

5.1 Gated service with gating at the end of service

For the regime of gated service with gating at the end of service, when the server visits Q_i, all class-i jobs that arrive during the service at Q_i are served in the same batch. So, at the end of the service time, Q_i is empty again. Note that gating at the end of a service represents a very efficient service operation in which a queue is in fact exhaustively served in one batch service.
This variant describes a typical Israeli queue-discipline in a line for buying tickets for a movie or show, operating as follows: A new arrival who finds a non-empty queue, first looks for a 'friend' standing in the line. If he finds such a friend, he kindly asks her to buy an extra (one or more) ticket(s) when she reaches the cashier. The service by the cashier is done in batches, i.e. it takes the same time to sell one or several tickets.

In the analysis it is useful to distinguish between the jobs arriving in an empty queue, called *first-arrivals*, and the jobs that arrive when the queue is non-empty. Note that one may associate with each job that is a first-arrival a 'family' of jobs that arrive during this first-arrival's response time. Conditioning on the first-arrival's response time, x say, the size of its family is Poisson distributed with mean λx and each job in its family will have a uniformly $U[0, x]$ distributed response time.
Let us first consider only the first-arrivals within a batch and ignore the other arrivals. Then this system can be seen as a *machine-repairman* system. Each class alternates between two situations: 1) there is a job waiting to be served or in service, and 2) there is no job to be served. If the server is seen as the 'repairman' and each class as a machine (if the queue is empty the machine is 'up', and if the queue is non-empty, the machine is 'down'), then the similarity with the machine-repairman model is clear.
As all service times are exponential and all arrival processes are independent Poisson processes, the equilibrium distribution for the number of non-empty queues is easily obtained from the balance equations. With $p_k^{(N)}$ denoting the probability that k queues are non-empty (k machines down) and $N-k$ queues are empty ($N-k$ machines up), one has – as in the machine repairman model:

$$\lambda(N-k)p_k^{(N)} = \mu p_{k+1}^{(N)} \ , \quad k = 0, \ldots, N-1.$$

Hence, with $\rho = \lambda/\mu$, the equilibrium state probabilities are

$$p_k^{(N)} = K^{(N)} \frac{\rho^k N!}{(N-k)!} \ , \ k = 0, \ldots, N,$$

where $K^{(N)} = \left(\sum_{k=0}^{N} \frac{\rho^k N!}{(N-k)!} \right)^{-1}$.

The first-arrival's waiting time

By the arrival theorem [13] that holds for the closed product-form network underlying the machine-repairman model, each first-arrival sees the repairman queue in equilibrium as if

its own class has never been active – so as if there are only $N-1$ classes. Let W_{first} denote the waiting time (not including service time) for a first-arrival. Then, with probability $p_k^{(N-1)}$ the first-arrival of a queue has to wait an Erlang(k,μ) time. Thus (cf. also [8] or [3] for this result for the machine-repairman model):

$$E[e^{-\omega W_{first}}] = K^{(N-1)} \sum_{k=0}^{N-1} \frac{\rho^k (N-1)!}{(N-1-k)!} \left(\frac{\mu}{\mu+\omega}\right)^k \ .$$

Hence,

$$P[W_{first} = 0] = p_0^{(N-1)} = K^{(N-1)} \ ,$$

and the density $f_{W_{first}}(t),\ t>0$, of W_{first} satisfies

$$f_{W_{first}}(t) = K^{(N-1)} \sum_{k=1}^{N-1} \frac{\rho^k (N-1)!}{(N-1-k)!} \frac{\mu(\mu t)^{k-1} e^{-\mu t}}{(k-1)!} \ , \quad t>0 \ .$$

Further,

$$\begin{aligned} EW_{first} &= K^{(N-1)} \sum_{k=0}^{N-1} \frac{\rho^k (N-1)!}{(N-1-k)!} \frac{k}{\mu} \\ &= \frac{K^{(N-1)}}{\mu} \sum_{k=0}^{N-1} \frac{\rho^k (N-1)!}{(N-1-k)!} \left[(N-1)-(N-1-k)\right] \qquad (14) \\ &= \frac{N-1}{\mu} - \frac{K^{(N-1)}}{\mu} \sum_{k=0}^{N-2} \frac{\rho^k (N-2)!}{(N-2-k)!} (N-1) \\ &= \frac{N-1}{\mu} \left(1 - \frac{K^{(N-1)}}{K^{(N-2)}}\right) \ . \end{aligned}$$

The family member's response time

For family members, the 'non-first-arrivals', it is more convenient to talk about response times. The first-arrival's response time S_{first} is the (independent) sum of its waiting time and its (exponential) service time. That is, $S_{first} = W_{first} + B$. All other family members arrive during a first-arrival's response time. As the arrival processes are Poisson, the unconditional family member's response time is a residual first-arrival's response time, with density $f_{S_{first}}(t) = P(S_{first} > t)/E(S_{first})$.

The response time, overall

In order to get the response time of an arbitrary job, one only needs the probability that an arrival is a first-arrival. Since per first-arrival there are on the average $\lambda E(S_{first})$ non-first-arrivals, the probability p_{first} that a job is a first-arrival is $1/(1+\lambda E(S_{first}))$. So, with

probability p_{first} a job has a first-arrival's response time and with probability $1 - p_{first}$ it has a residual first-arrival's response time. Denoting by S_{arb} the sojourn time of an arbitrary job, we have

$$\begin{aligned} E(S_{arb}) &= \frac{1}{1+\lambda E(S_{first})} E(S_{first}) + \frac{\lambda E(S_{first})}{1+\lambda E(S_{first})} \frac{E(S_{first}^2)}{2E(S_{first})} \\ &= \frac{E(S_{first}) + \lambda E(S_{first}^2)/2}{1+\lambda E(S_{first})}. \end{aligned}$$

A comparison between the Israeli queue and a regular queue

It turns out that the Israeli queue is very efficient. To demonstrate this we calculate the ratio $R(a)$ between the mean sojourn time of an arbitrary customer in a regular $M/M/1$ queue with arrival rate $N\lambda$ and service rate μ, and the corresponding mean sojourn time $E(S_{arb})$ in the 'FCFS among queues' discipline with N families, where the traffic intensity is $a = N\lambda/\mu < 1$.

Let $N = 2$. Then

$$E\big(S_{sojourn}(M(N\lambda)/M(\mu)/1)\big) = \frac{1}{\mu}\left(\frac{1}{1-a}\right) = \frac{1}{\mu}\left(\frac{1}{1-2\rho}\right),$$

while

$$E(S_{arb}) = \frac{1}{\mu}\left(\frac{1+3\rho+3\rho^2}{1+2\rho+2\rho^2}\right).$$

This follows since

$$\begin{aligned} E(W_{first}) &= \frac{N-1}{\mu}\left(1 - \frac{K^{(N-1)}}{K^{(N-2)}}\right) = \frac{1}{\mu}(1-K^{(1)}) = \frac{1}{\mu}\frac{\rho}{1+\rho} \\ E(S_{first}) &= E(W_{first}) + \frac{1}{\mu} = \frac{1}{\mu}\left(\frac{1+2\rho}{1+\rho}\right) \\ E(S_{first}^2) &= E(W_{first}^2) + 2E(W_{first})\frac{1}{\mu} + \frac{2}{\mu^2} \\ &= \frac{1}{\mu^2}\frac{2\rho}{1+\rho} + \frac{1}{\mu^2}\frac{2\rho}{1+\rho} + \frac{2}{\mu^2} = \frac{2}{\mu^2}\cdot\frac{1+3\rho}{1+\rho}. \end{aligned}$$

We thus have

$$R(a=2\rho) = \left.\frac{E(S_{sojourn})}{E(S_{arb})}\right|_{N=2} = \left(\frac{1}{1-2\rho}\right)\cdot\left(\frac{1+2\rho+2\rho^2}{1+3\rho+3\rho^2}\right) > 1.$$

The above ratio $R(a = 2\rho)$ is an increasing convex function running from $R(0) = 1$ to $R(1) = \infty$.

The following table gives some values of $R(\,\cdot\,)$ for $0 < \rho < 0.5$:

ρ	0.1	0.2	0.3	0.4	0.45
$R(a=2\rho)$	1.147	1.434	2.051	3.955	7.794

5.2 Exhaustive service, gating at the beginning of service

Now consider the case of exhaustive service, with gating at the *beginning* of service. Then, after the first arrival there is a stream of arrivals of the same class during the first-arrival's waiting time which are served together with it. However, customers of the same class arriving during the batch service are *not* included in the current batch and form a new batch. When a Q_i service time is completed, it is checked whether there have been arrivals to Q_i during this service time. Such an event occurs with probability $\lambda/(\lambda+\mu)$. If so, a new batch-service at Q_i is started in which all jobs that arrived during the previous service time are served in one batch. If at the end of this (second) service time again new arrivals are found in Q_i, then Q_i gets yet another service time, etc. If we now consider the sum of all these service times as the new *generalized* service time of Q_i, then this generalized service time is again exponential, but with mean

$$\frac{1}{1-\frac{\lambda}{\lambda+\mu}}\frac{1}{\mu}=\frac{\lambda+\mu}{\mu^2}.$$

This follows since a geometric (with probability of success $\mu/(\lambda+\mu)$) sum of i.i.d. Exponential(μ) random variables is Exponential with parameter $\mu^2/(\lambda+\mu)$. We can now use the results of the analysis of the previous subsection to determine the waiting time distribution of a first-arrival: Just replace the $\exp(\mu)$ service times by $\exp(\mu^2/(\lambda+\mu))$ service times. The ones that arrive during the first-arrival's waiting time get a response time equal to a residual first-arrival's waiting time plus an ordinary service time. All other jobs arrive during a service time, so their response time is a residual service time. The occurrence fractions of these three types of arrivals are determined by observing the following. Per first-arrival there are on average $\lambda E(W_{first})$ arrivals during the waiting time. The mean number of arrivals during the generalized service time is $\lambda\ (\lambda+\mu)/\mu^2$.

Remark 5.1
When the arrival rates of the various classes are *not* the same, one still has a machine-repairman model and it is possible to derive the waiting time distribution of a customer of class i, $i=1,\ldots,N$. However, the analysis becomes more intricate and less elegant, and it is omitted.

5.3 Gated service, gating at the beginning of service

There is an essential difference between the cases of Sections 5.1 and 5.2 and the case of gated service with gating at the *beginning* of service. In these previous models, if the (generalized) service time is completed, the queue is empty. For gating at the beginning of service it is possible that during this service time a job corresponding to the *same* queue arrives and initiates a new batch for the same queue as the one in service. Then this new job (batch) has to join the tail of the repairman queue.

When describing the state of the system let us again look at the number of 'batch jobs' at the server and not at individual jobs.

Detailed state description
A detailed state description would contain the batch jobs at the server as well as their order. In that set of jobs at most one class can be present twice; once in service and once waiting. The one waiting can be anywhere in the line. We will not pursue this any further. As we will see below, a simpler state description suffices.

Aggregated state description
A more aggregated state description is obtained using as states the pairs (k, s) and (k, d), with k the number of different classes present at the server (including the one in service) and s and d indicating whether these classes all have a *single* queue or that there is one class having a *double* queue. So, in this state description, the order of the jobs (queues) at the repairman is ignored. State 0 is the situation that the server is idle. With this state description only two types of events are possible: a service completion or an arrival. Immediately after the service completion each Q has at most one job at the repairman and by symmetry all orders are 'the same'.

Thus one gets the following (finite) set of balance equations for the steady-state probabilities $p(0)$, $p(k, s)$ and $p(k, d)$ for the states 0, (k, s) and (k, d); $k = 1, \ldots, N$:

$$\begin{aligned}
p(k,s)[\,\lambda(N-k+1)+\mu\,] &= p(k,d)\mu + p(k+1,s)\mu + p(k-1,s)\lambda(N-k+1),\ 0<k<N,\\
p(N,s)[\,\lambda+\mu\,] &= p(N,d)\mu + p(N-1,s)\lambda,\\
p(k,d)[\,\lambda(N-k)+\mu\,] &= p(k,s)\lambda + p(k-1,d)\lambda(N-k+1),\ k=2,\ldots,N,\\
p(1,d)[\,\lambda(N-1)+\mu\,] &= p(1,s)\lambda,\\
p(N,d)\mu &= p(N,s)\lambda + p(N-1,d)\lambda,\\
p(0)\lambda N &= p(1,s)\mu\ .
\end{aligned}$$

These, together with the normalizing condition, easily yield the steady-state probabilities. Computing the waiting time distribution from this equilibrium distribution is not straightforward. The distribution of the position of the second queue of the class in service is needed. Consider an arbitrary arrival. With probability $p(0)$ the arrival finds the server idle, thus has a waiting time 0. For arrivals that find the server busy, consider the number of classes at the end of the last service. With probability q_k this number is k. We distinguish three cases.
(i) This number is 0 as the result of a service completion in $(1, s)$. Then it takes an $\exp(N\lambda)$ delay to generate a new arrival in state 0.
(ii) It is k $(1 \leq k < N)$ as the result of a service completion in either $(k+1, s)$ or (k, d).
(iii) It is N as the result of a service completion in (N, d).
Since all service times have the same $\exp(\mu)$ distribution, the steady-state probabilities are equal to the probabilities just before a service completion.

So we have for $k = 1, \ldots, N-1$:

$$q_0 : q_k : q_N = p(1,s)\ :\ [p(k+1,s)+p(k,d)]\ :\ p(N,d)\ .$$

So,

$$\begin{aligned} q_0 &= [1-p(0)]^{-1}p(1,s), \\ q_k &= [1-p(0)]^{-1}[p(k+1,s)+p(k,d)]\ , \quad k=1,\ldots,N-1, \\ q_N &= [1-p(0)]^{-1}p(N,d). \end{aligned}$$

Consider an arrival that finds the server busy. The probability that it arrives in a service interval that started with k jobs left behind is q_k. As (it?) is well-known, the 'age' of this service time and its residual are independent and both exponential with rate μ. Let $q_{k,l}$ be the probability that, if the interval started with k batch jobs waiting, just before the arrival there are exactly l classes not present in the repairman queue. Integration with respect to the age distribution gives, for $l=0,1,\ldots,N-k$, $k=1,\ldots,N$:

$$\begin{aligned} q_{k,l} &= \int_0^\infty \binom{N-k}{l}(e^{-\lambda x})^l(1-e^{-\lambda x})^{N-k-l}\mu e^{-\mu x}dx \\ &= \binom{N-k}{l}\sum_{j=0}^{N-k-l}(-1)^{N-k-l-j}\binom{N-k-l}{j}\frac{\mu}{\mu+\lambda l+(N-k-l-j)\lambda}. \end{aligned}$$

It should be noted that $q_{0,l}=q_{1,l}$: When the previous service completion left the system empty, a new arrival after $\exp(N\lambda)$ returns the system to state $(1,s)$.

Next, given that l classes are absent just before the arrival, the probability that the new arrival belongs to an absent class is l/N in which case it joins the tail of the line and gets a waiting time that is $Erlang(N-l+1,\mu)$. However, with probability $(N-l)/N$ its class is already present, in which case the arrival belongs to each class in the line with equal probability: The position in the line of its class is uniform on $[1,\ldots,N-l]$. Thus, given the number of absent classes l, its waiting time density $f_{W|l}$ will be

$$f_{W|l}(t)=\frac{l}{N}\frac{\mu(\mu t)^{N-l}}{(N-l)!}e^{-\mu t}+\sum_{m=1}^{N-l}\frac{1}{N}\frac{\mu(\mu t)^m}{m!}e^{-\mu t}\ .$$

So the waiting time of an arbitrary job satisfies

$$P[W=0]=p(0)\ ,$$

and has density

$$f_W(t)=\sum_{k=0}^{N}q_k\sum_{l=0}^{N-k}q_{k,l}f_{W|l}(t)\ , \quad t>0.$$

Remark 5.2
For the asymmetric case, with different arrival rates but the same service rate, the situation is far more complicated. It seems that the states have to describe the order in the queue, so that a detailed state description is needed.

References

[1] E. Altman, A. Khamisy and U. Yechiali (1992). On elevator polling with globally gated regime. *Queueing Systems* 11, 85–90.

[2] M.H. Ammar and J.W. Wong (1987). On the optimality of cyclic transmission in teletext systems. *IEEE Transactions on Communications* COM-35, 68–73.

[3] O.J. Boxma, D. Denteneer and J.A.C. Resing (2003). Delay models for contention trees in closed populations. *Performance Evaluation* 53, 169–185.

[4] O.J. Boxma and W.P. Groenendijk (1987). Pseudo-conservation laws in cyclic-service systems. *J. Appl. Probab.* 24, 949–964.

[5] O.J. Boxma, J. van der Wal and U. Yechiali (2007). Polling with batch service. EURANDOM Report 2007-058.

[6] H.D. Dykeman, M.H. Ammar and J.W. Wong (1986). Scheduling algorithms for videotex systems under broadcast delivery. In: Proceedings of the International Conference on Communications (ICC'86), pp. 1847–1851.

[7] S.W. Fuhrmann (1981). Performance analysis of a class of cyclic schedules. Bell Laboratories Technical Memorandum 81-59531-1.

[8] H. Kobayashi (1978). *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology.* Addison-Wesley, Reading, MA.

[9] T.Y.S. Lee and J. Sunjaya (1996). Exact analysis of asymmetric random polling systems with single buffers and correlated input process. *Queueing Systems* 23, 131–156.

[10] Z. Liu and P. Nain (1992). Optimal scheduling in some multiqueue single-server systems. *IEEE Transactions on Automatic Control* 37, 247–252.

[11] M.P. Van Oyen and D. Teneketzis (1996). Optimal batch service of a polling system under partial information. *Methods and Models in OR* 44, 401–419.

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

[13] K.C. Sevcik and I. Mitrani (1979). The distribution of queueing network states at input and output instants. In: M. Arato et al. (eds.), *Performance '79* (North-Holland, Amsterdam), pp. 319–335.

[14] H. Takagi (1985) On the analysis of a symmetric polling system with single-message buffers. *Performance Evaluation* 5, 149–157.

[15] H. Takagi (2000). Analysis and application of polling models. In: *Performance Evaluation: Origins and Directions*, eds. G. Haring, Chr. Lindemann and M. Reiser, Lecture Notes in Computer Science vol. 1769 (Springer, Berlin), pp. 423–442.

[16] T. Takine, Y. Takahashi and T. Hasegawa (1986). Performance analysis of a polling system with single buffers and its application to interconnected networks. *IEEE J. Sel. Areas in Commun.* SAC-4, 802–812.

[17] T. Takine, Y. Takahashi and T. Hasegawa (1987). Analysis of an asymmetric polling system with single buffers. In: P.-J. Courtois and G. Latouche (eds.), *Performance '87* (Elsevier Science Publishers BV, Amsterdam), pp. 241–251.

[18] T. Takine, H. Takagi, Y. Takahashi and T. Hasegawa (1990). Analysis of asymmetric single-buffer polling and priority systems without switchover times. *Performance Evaluation* 11, 253–264.

[19] J. van der Wal and U. Yechiali (2003). Dynamic visit-order rules for batch-service polling. *Probability in the Engineering and Informational Sciences* 17, 351–367.