Markovian polling systems with an application to wireless random-access networks

Jan-Pieter Dorsman, S. Borst, O. Boxma, Maria Vlasiou

Markovian polling systems with an application to wireless random-access networks

Jan-Pieter L. Dorsman [*][†]   Sem C. Borst [*][‡]   Onno J. Boxma [*]   Maria Vlasiou [*][†]

j.l.dorsman@tue.nl   s.c.borst@tue.nl   o.j.boxma@tue.nl   m.vlasiou@tue.nl

January 16, 2014

Abstract

Motivated by an application in wireless random-access networks, we study a class of polling systems with Markovian routing, in which the server visits the queues in an order governed by a discrete-time Markov chain. Assuming that the service disciplines at each of the queues fall in the class of branching-type service disciplines, we derive a functional equation for (the probability generating function of) the joint queue length distribution conditioned on a point in time when the server visits a certain queue. From this functional equation, expressions for the (cross-)moments of the queue lengths follow. We also derive a pseudo-conservation law for this class of polling systems. Using these results, we compute expressions for certain system parameters that minimise the total expected amount of work in systems that arise from the wireless random-access network setting. In addition, we derive approximations for those same parameters that minimise a weighted sum of mean waiting times in these systems. Based on these expressions, we also present an adaptive control algorithm for finding the optimal parameter values in a distributed fashion, which is particularly relevant in the context of wireless random-access networks.

Keywords: queue lengths, binomial service disciplines, Markovian routing, random routing, wireless random-access networks

1   Introduction

In this paper, we study a class of queueing systems consisting of multiple queues attended by a single server. The server visits queues in order to provide service there, and incurs stochastic switch-over times when it moves from one queue to another. These queueing systems are commonly called *polling systems*, and find their origin in many real-life applications, such as manufacturing environments and computer-communication systems. For an overview of the literature on polling systems and a more exhaustive overview of their applications, we refer to surveys such as [2, 15, 18, 19].

Many studies on polling systems require the server to visit the queues in a fixed, cyclic order. In some cases, however, this is not a realistic assumption. Instead, we study polling systems where the server operates under the so-called Markovian routing mechanism. This implies that the order in which the server visits the queues is governed by a discrete-time Markov chain with transition probabilities $p_{i,j}$. Although there is a huge body of literature on polling systems, models with Markovian routing have received remarkably little attention. This is perhaps because the analysis of these systems is of a complex nature. More specifically, it is shown in [16] that there is a striking dichotomy in the complexity of the analysis of polling systems. Polling systems of which the joint queue length process observed at time points where the server starts a visit (also referred to as polling epochs) constitutes a multi-type branching process (MTBP) with immigration (see e.g. [1] for a definition), are more tractable than polling systems which do not satisfy this so-called branching property. Due to the stochastic nature of the server routing, polling systems with a Markovian routing mechanism do not satisfy the branching

property. The few publications that can be found on Markovian polling systems include [7], in which an expression for the expected amount of work in the system at an arbitrary moment is derived for a few service disciplines. This work is extended in [20], where it is shown how to derive expressions for the moments of the (joint) queue lengths for the same service disciplines. Similar results for a slightly more general form of Markovian routing, where the routing probabilities may depend on the event whether a queue is empty or not, are derived in [17]. Observe that the Markovian routing mechanism is very general and captures many variations of polling models studied in the literature. For instance, the cyclic polling model falls in this framework. Another example is the random routing discipline, where after any visit period, the server visits queue j with probability p_j, irrespective of the queue the server just visited (cf. [12]).

The random routing discipline is particularly relevant in the context of wireless random-access networks, which in fact motivated the present study. The various queues in the polling system correspond in that case to the packet buffers at several wireless transmitters, which need to share the medium in a mutually exclusive way because of interference. So-called Carrier-Sense Multiple-Access Collision-Avoidance (CSMA-CA) algorithms provide a common mechanism for governing the use of such a shared wireless medium in a distributed fashion. In CSMA-CA algorithms the various transmitters obey random back-off times between activity periods during which they sense the medium to avoid collisions and provide other nodes an opportunity to activate. In the case of exponentially distributed back-off durations, the alternating use of the medium by the nodes is probabilistically equivalent to random routing in a polling system, as will be described in greater detail in Section 2. The relative values of the back-off rates induce relative priorities among the nodes, and hence a crucial question is how the back-off rates should be selected in order to minimise the overall average packet delay, which corresponds to the optimal selection of the routing probabilities p_j in the polling system.

Almost all studies on polling systems focus on a limited number of service disciplines, such as the exhaustive discipline, where the server only switches to the next queue when the current one is empty, or the gated discipline, where during a visit the server serves only those customers that were already present at the start of it. However, with our motivation by wireless random-access networks in mind, we initially study the complete class of so-called branching-type service disciplines; i.e., service disciplines that would allow the system to satisfy the branching property in case the server were to visit the queues in a cyclic order. Subsequently, we give special attention to the so-called binomial gated discipline introduced in [14], and its exhaustive counterpart as defined in [3] known as the binomial exhaustive discipline. Under the binomial gated discipline, when the server finds n customers present at the start of a visit period at queue j, he will serve a binomial (n, r_j) number of these customers before switching, $0 < r_j \leq 1$. Under the binomial exhaustive discipline, the server not only serves the binomial (n, r_j) number of the customers present at the start of the visit period, but subsequently also the type-j customers arriving during the service of these customers ('the children'), the type-j customers arriving during the service of the children ('the grandchildren'), and so on. As a result, the expected number of type-j customers that are left behind by the server at the end of the visit period equals $n(1 - r_j)$.

The main contributions of this paper can be summarised as follows. First, we analyse Markovian polling systems with branching-type service disciplines in their most general form. We derive a functional equation for (the probability generating function of) the joint queue length distributions at polling epochs, from which expressions for the (cross-)moments of the joint queue-length distribution at an arbitrary point in time follow. We also obtain an explicit expression for the expected amount of waiting work in the system, based on the concept of the so-called pseudo-conservation law obtained in [4]. Although we believe these results to be of independent interest, we subsequently focus on their application to wireless random-access networks. That is, for random polling systems with queues served according to the binomial gated or the binomial exhaustive service discipline, we study the question of how to choose several model parameters such as the routing probabilities p_j so that a weighted sum of mean waiting times of all customer types or a weighted sum of mean queue lengths pertaining to all queues is minimised. We obtain accurate approximations for these optimal parameters that are expressed in closed form and are even exact when the weights are chosen such that the weighted sum represents the mean amount of work in the system. However, these (near-)optimal expressions cannot be used directly to obtain optimal back-off rates in the wireless random-access network setting, since these expressions involve e.g. the arrival rates of all queues, which in practice is not known to a transmitter. Therefore, we propose a distributed algorithm that makes each node choose its back-off rate dynamically based on the durations of previous packet inter-transmissions without requiring information concerning other nodes in the network. When all nodes adhere to this algorithm, the back-off rates converge in some sense to their optimal values over time.

The rest of the paper is structured as follows. In Section 2, we give a formal description of the polling model that we study and we introduce the necessary notation. Section 3 provides the analysis of the queue length distribution of the model in its full generality. Based on this, we derive model parameters in Section 4

that optimise the expected amount of work in the system, and nearly optimise any weighted sum of mean waiting times or mean queue lengths as confirmed by extensive numerical experiments. Based on the resulting expressions for the (near-)optimal routing probabilities, we describe the algorithm for obtaining (near-)optimal back-off rates in the wireless network setting in Section 5. We elaborate on the convergence properties of this algorithm, and provide several numerical examples.

2   Model description and notation

In this section, we give a description of the polling system under consideration, we introduce the notation required and we discuss the equivalence between the polling model and a wireless random-access network. Throughout the paper, matrices and vectors are printed in bold face. The vectors $\mathbf{0}$ and $\mathbf{1}$ represent a vector of appropriate size of which each element equals zero and one, respectively.

We study a queueing system that consists of $N \geq 2$ infinite-buffer queues, $Q_1, \ldots, Q_N$ and a single server. Customers arriving at Q_i, also referred to as type-i customers, do so according to a Poisson process with intensity λ_i. The generic service requirement of a type-i customer is represented by the random variable B_i, of which the Laplace-Stieltjes transform (LST) is given by $\widetilde{B}_i(s) = \mathbb{E}[e^{-sB_i}]$. The workload that Q_i brings to the system is denoted by $\rho_i = \lambda_i \mathbb{E}[B_i]$. We assume throughout that the aggregate workload $\rho = \sum_{i=1}^{N} \rho_i$ is less than one.

All the queues share a single server. However, this server can only serve customers of one queue at a time. Hence, after serving a given number of customers at one queue (a visit period), the server will switch over to another queue to start service there. We refrain from the usual assumption that the server visits the queues in a fixed cyclic order. Instead, we assume that the server adheres to the following scheme, known as Markovian routing. The position of the server is governed by the irreducible discrete-time Markov chain $\{Z_m, m \geq 0\}$ on the state space $\mathcal{S} = \{1, \ldots, N\}$. The one-step transition probability matrix corresponding to this Markov chain is given by $\boldsymbol{P} = (p_{i,j})_{i,j \in \mathcal{S}}$, and its unique invariant probability measure denoted by $\boldsymbol{q} = (q_i)_{i \in \mathcal{S}}$ satisfies the conditions $\boldsymbol{qP} = \boldsymbol{q}$ and $\sum_{j=1}^{N} q_j = 1$. Hence, the queue being served during the m-th visit period is Q_{Z_m}. In short, after completing a visit period to Q_i, the server will switch over to Q_j with probability $p_{i,j}$. Such a switch-over from Q_i to Q_j takes a continuously distributed random amount of time $S_{i,j}$ (also referred to as the switch-over time), of which the LST is given by $\widetilde{S}_{i,j}(s) = \mathbb{E}[e^{-sS_{i,j}}]$. We assume all interarrival times, service times and switch-over times in the model to be independent. In the sequel, we also consider the so-called random routing mechanism, which is a special case of the Markovian routing mechanism that does not allow the routing probabilities to depend on the departure queue Q_i. In other words, when the server adheres to the random routing mechanism, we have that $p_{i,j} = p_j$ for all $i \in \mathcal{S}$, which consequently leads to $q_j = p_j$ for all $j \in \mathcal{S}$.

In the context of wireless random-access networks as described in the introduction, each queue Q_i has a back-off rate ν_i which governs the exponentially distributed amount of time the node remains in back-off before starting the next transmission. Because of the memoryless property of the exponential distribution, this is equivalent to a polling system with switch-over times between any pair of queues that are exponentially distributed with parameter $\nu_0 = \sum_{i=1}^{N} \nu_i$ and a random routing policy with routing probabilities $p_j = \nu_j/\nu_0$, $j = 1, \ldots, N$. Yet another equivalent interpretation is that each queue has the same back-off rate ν_0, but only activates at the end of a back-off period with probability p_j. The latter representation will be particularly convenient when we develop an adaptive algorithm for finding the optimal routing probabilities in a distributed fashion in Section 5.

The number of customers that are served during a visit period V_i at Q_i is governed by the service discipline at Q_i. We do not limit our analysis to a single service discipline, but we assume that the service discipline at each of the queues belongs to the class of service disciplines that satisfy the following property.

Property 2.1. If the server arrives at Q_i to find l_i customers there, then during the course of the server's visit, each of these l_i customers will effectively be replaced in an i.i.d. manner by a random population having a probability generating function (PGF) $\widetilde{H}_i(\boldsymbol{z}) = \widetilde{H}_i(z_1, \ldots, z_N)$, which is called the *offspring function* and can be any N-dimensional PGF.

This property is necessary for the joint queue length process at polling epochs to constitute an MTBP (cf. [16]) and allows for a large class of service disciplines. Two service disciplines satisfying this property that will receive specific attention in this paper are the binomial gated and the binomial exhaustive service discipline. Under the binomial gated discipline, the number of type-i customers that are served during a visit period, at the start of which m_i type-i customers are present in the system, is binomially distributed with parameters m_i and r_i, $r_i \in (0, 1]$. Thus, a type-i customer present at the start of a non-empty visit period is still present at the end

of this period with probability $1 - r_i$, or is served during this period with probability r_i. Since during a type-i service time, new customers will arrive at each of the queues, the offspring function is in this case given by $\widetilde{H}_i(\boldsymbol{z}) = (1 - r_i)z_i + r_i\widetilde{B}_i(\sum_{j \in \mathcal{S}} \lambda_j(1 - z_j))$. The binomial exhaustive discipline has many similarities with the binomial gated discipline. Again, a type-i customer present at the start of a visit period remains in the system with probability $1 - r_i$. However, with probability r_i, not only the customer itself will be served during the visit period, but also all of its type-i offspring (thus, the type-i 'children' that arrive during this service time, the type-i 'grandchildren' that arrive during the service times of the children, and so on). Therefore, the visit period now consists of a number of type-i busy periods (time to serve a type-i customer and all of its type-i offspring) that is binomially distributed with parameters m_i and r_i. When denoting the duration of such a busy period generated by a type-i customer by Γ_i, and its corresponding LST by $\widetilde{\Gamma}_i(s) = \mathbb{E}[e^{-s\Gamma_i}]$, the offspring function of a queue adhering to the binomial exhaustive service discipline is thus given by $\widetilde{H}_i(\boldsymbol{z}) = (1 - r_i)z_i + r_i\widetilde{\Gamma}_i(\sum_{j \in \mathcal{S}\setminus\{i\}} \lambda_j(1 - z_j))$. In both of these service disciplines, r_i is a measure of the service exhaustiveness; the higher r_i, the more customers the server will serve on average at Q_i over the course of a non-empty visit period. As such, we will also refer to r_i as the exhaustiveness probability. Observe that for $r_i = 1$ the binomial gated and binomial exhaustive service disciplines as described above reduce to the classical gated and exhaustive service disciplines.

We denote by C_i the time between two consecutive epochs where the server *polls* Q_i. A server is said to poll a queue when it starts a visit period at that queue. The time C_i consists of an average of $1/q_i$ visit periods and subsequent switch-over periods by virtue of the Markovian routing dynamics. Furthermore, any arbitrary visit period and subsequent switch-over period in steady state corresponds to a visit to Q_i with probability q_i. Thus, there are on average q_j/q_i visit periods and switch-overs to Q_j between two polling epochs of Q_i. The expected time the server takes for a visit to Q_j and the subsequent switch-over equals $\mathbb{E}[V_j] + \sum_{k \in \mathcal{S}} p_{j,k}\mathbb{E}[S_{j,k}]$. As a consequence,

$$\mathbb{E}[C_i] = \frac{1}{q_i}\sum_{j \in \mathcal{S}} q_j(\mathbb{E}[V_j] + \sum_{k \in \mathcal{S}} p_{j,k}\mathbb{E}[S_{j,k}]). \tag{1}$$

It follows from balance arguments that $\mathbb{E}[V_i] = \rho_i\mathbb{E}[C_i]$ and $\frac{q_i\mathbb{E}[V_i]}{q_j\mathbb{E}[V_j]} = \frac{\rho_i}{\rho_j}$. As such, we have by (1) that, for every $i \in \mathcal{S}$,

$$\mathbb{E}[C_i] = \frac{\sigma}{q_i(1 - \rho)}, \tag{2}$$

where $\sigma = \sum_{j \in \mathcal{S}} q_j \sum_{k \in \mathcal{S}} p_{j,k}\mathbb{E}[S_{j,k}]$ (see also [7]). Note that σ represents the overall mean of the switch-over times incurred by the server. We denote by ζ_i the reciprocal of the expected number of customers served by the server during a visit period V_i. We thus have that $\zeta_i = \frac{\mathbb{E}[B_i]}{\mathbb{E}[V_i]} = \frac{1}{\lambda_i\mathbb{E}[C_i]}$.

In the remainder of this article, we are interested in the joint queue length distributions (including any customer in service) at several time epochs. To this end, we denote by $\boldsymbol{F}_i = (F_{i,1}, \ldots, F_{i,N})$ the joint stationary queue length conditioned on the event that the server currently polls Q_i. The vectors $\boldsymbol{G}_i$, $\boldsymbol{M}_i$, $\boldsymbol{N}_i$, $\boldsymbol{X}_i$ and $\boldsymbol{Y}_{i,j}$ similarly represent the joint stationary queue length conditioned on a point in time where the server ends a visit period at Q_i, the server starts serving a type-i customer, the server completes service of a type-i customer, the server is serving customers at Q_i and the server is currently switching from Q_i to Q_j, respectively. The unconditional stationary joint queue length of the queues in the system is given by $\boldsymbol{L}$. For an arbitrary N-dimensional random variable $\boldsymbol{R} = (R_1, \ldots, R_N)$, we denote its N-dimensional PGF by $\widetilde{R}(\boldsymbol{z}) = \widetilde{R}(z_1, \ldots, z_N) = \mathbb{E}[\prod_{k \in \mathcal{S}} z_k^{R_k}]$. Furthermore, we define $\widetilde{R}^{(k)}(\boldsymbol{z}) = \frac{\partial}{\partial z_k}\widetilde{R}(\boldsymbol{z})$, $\widetilde{R}^{(k,l)}(\boldsymbol{z}) = \frac{\partial}{\partial z_l}\frac{\partial}{\partial z_k}\widetilde{R}(\boldsymbol{z})$, $r(k) = \widetilde{R}^{(k)}(\boldsymbol{z})|_{\boldsymbol{z}=1}$ and $r(k,l) = \widetilde{R}^{(k,l)}(\boldsymbol{z})|_{\boldsymbol{z}=1}$. Thus, we use lower cases to refer to derivatives of PGFs evaluated in $\boldsymbol{z} = \boldsymbol{1}$. It holds that $r(k) = \mathbb{E}[R_k]$, $r(k,k) = \mathbb{E}[R_k^2] - \mathbb{E}[R_k]$ and $r(k,l) = \mathbb{E}[R_kR_l]$ if $k \neq l$. So, for example, $f_i(k)$ denotes the mean queue length of Q_k when the server polls Q_i. Likewise, $f_i(k,l)$ refers to the second-order cross moment pertaining to the queue lengths of Q_k and Q_l when the server polls Q_i and $k \neq l$. Besides the short-hand notation $\boldsymbol{z} = (z_1, \ldots, z_N)$ that we used above, we will also use $\boldsymbol{z}_i^H = (z_1, \ldots, z_{i-1}, H_i(\boldsymbol{z}), z_{i+1}, \ldots, z_N)$ and $\Sigma(\boldsymbol{z}) = \sum_{k \in \mathcal{S}} \lambda_k(1 - z_k)$. Furthermore, we denote by $\mathbb{1}_{\{A\}}$ the indicator function of the event A. Observe that the distribution of the waiting time W_i for type-i customers with LST $\widetilde{W}_i(s) = \mathbb{E}[e^{-sW_i}]$ is related to the queue length L_i through the distributional form of Little's law $\widetilde{W}_i(s) = \frac{\widetilde{L}_i(1 - s/\lambda_i)}{\widetilde{B}_i(s)}$, as shown in [10].

3 Analysis

In this section, we analyse the queue lengths of the various queues in the Markovian polling model with branching-type service disciplines. In particular, we derive in Section 3.1, for all $i \in \mathcal{S}$, a functional equation for the PGF $\widetilde{F}_i(z)$, which represents the queue lengths at a moment the server starts a visit period at Q_i. Based on this, we show how to compute the first- and second-order (cross-)moments of $\boldsymbol{F}_i$. These results lead to an expression for $\mathbb{E}[L_i]$, the mean queue length of Q_i at an arbitrary point in time, which we present in Section 3.2. Finally, for our class of polling systems, we derive in Section 3.3 an explicit expression for the expected amount $\sum_{i \in \mathcal{S}} \rho_i \mathbb{E}[W_i]$ of waiting work in the system waiting to be processed on the basis of the pseudo-conservation law.

3.1 Joint queue length at polling epochs

We now derive a functional equation for the PGF $\widetilde{F}_i(z)$ of the queue length distribution conditioned on the event that the server polls $Q_i, i \in \mathcal{S}$. Based on this functional equation, all moments of the joint queue length distribution at a polling epoch of Q_i can be derived. In particular, we show how to derive solvable sets of equations for $f_i(k)$ and $f_i(k,l), k, l \in \mathcal{S}$. From these sets we obtain expressions for the first and second-order (cross-)moments of the joint queue length distribution at polling epochs. We note that by using the same methodology, expressions for higher-order moments can be derived.

3.1.1 Functional equation for $\widetilde{F}_i(z)$

To obtain a functional equation for $\widetilde{F}_i(z)$, we first relate the queue length distribution at a polling epoch of Q_i to the queue length distribution at the preceding polling instant at any queue. To this end, recall that Z_m refers to the index of the queue that the server visits at the m-th polling instant. Furthermore, let $\boldsymbol{J}_m = (J_{m,1}, \ldots, J_{m,N})$ and $\boldsymbol{K}_m = (K_{m,1}, \ldots, K_{m,N})$ be the joint queue length at the start of the m-th visit period (to any queue) since the start-up of the system, and its end, respectively. By conditioning on Z_m and Z_{m+1}, we have that

$$
\mathbb{E}[\mathbb{1}_{\{Z_{m+1}=j\}} \prod_{k \in \mathcal{S}} z_k^{J_{m+1,k}}]
$$
$$
= \sum_{i \in \mathcal{S}} \mathbb{P}(Z_{m+1} = j \mid Z_m = i)\mathbb{P}(Z_m = i)\mathbb{E}[\prod_{k \in \mathcal{S}} z_k^{J_{m+1,k}} \mid Z_{m+1} = j, Z_m = i]. \tag{3}
$$

Observe that, as per Property 2.1, the total population in the system during the m-th visit period only changes through the i.i.d. replacement of every type-Z_m customer by a population with PGF $\widetilde{H}_{Z_m}(z)$. More colloquially speaking, the type-Z_m customers that get served during the m-th visit period allow new customers of any type to arrive to the system over the course of this visit period. As the number of arriving customers of any type is independent of $J_{m,i}, i \in \mathcal{S}\backslash\{Z_m\}$, we have that

$$
\mathbb{E}[\prod_{k \in \mathcal{S}} z_k^{K_{m,k}} \mid Z_m = i] = \mathbb{E}[\prod_{k \in \mathcal{S}\backslash\{i\}} z_k^{J_{m,k}} \mid Z_m = i] \sum_{n=0}^{\infty} (\widetilde{H}_i(z))^n \mathbb{P}(J_{m,i} = n)
$$
$$
= \mathbb{E}[(\widetilde{H}_i(z))^{J_{m,i}} \prod_{k \in \mathcal{S}\backslash\{i\}} z_k^{J_{m,k}} \mid Z_m = i]. \tag{4}
$$

Furthermore, the population at the start of the $(m+1)$-st visit period consists of the customers already there at the end of the m-th visit period, and the customers that arrive during the subsequent switch-over period according to type-specific Poisson processes. As these two subpopulations are independent, we obtain

$$
\mathbb{E}[\prod_{k \in \mathcal{S}} z_k^{J_{m+1,k}} \mid Z_{m+1} = j, Z_m = i]
$$
$$
= \mathbb{E}[\prod_{k \in \mathcal{S}} z_k^{K_{m,k}} \mid Z_m = i] \int_{t=0}^{\infty} \sum_{n_1=0}^{\infty} \cdots \sum_{n_N=0}^{\infty} \prod_{k \in \mathcal{S}} (z_k^{n_k} e^{-\lambda_k} \frac{\lambda_k^{n_k}}{n_k!}) d\mathbb{P}(S_{i,j} < t)
$$
$$
= \mathbb{E}[\prod_{k \in \mathcal{S}} z_k^{K_{m,k}} \mid Z_m = i] \widetilde{S}_{i,j}(\Sigma(z)). \tag{5}
$$

5

Combining (3)–(5) with the fact that $\mathbb{P}(Z_{m+1} = j \mid Z_m = i) = p_{i,j}$ now gives

$$\mathbb{E}[\mathbb{1}_{\{Z_{m+1}=j\}} \prod_{k\in\mathcal{S}} z_k^{J_{m+1,k}}]$$
$$= \sum_{i\in\mathcal{S}} p_{i,j}\mathbb{P}(Z_m = i)\mathbb{E}[(\widetilde{H}_i(\boldsymbol{z}))^{J_{m,i}} \prod_{k\in\mathcal{S}\setminus\{i\}} z_k^{J_{m,k}} \mid Z_m = i]\widetilde{S}_{i,j}(\Sigma(\boldsymbol{z})). \tag{6}$$

Next, observe that we can rewrite the left-hand side of (6) as $\mathbb{E}[\mathbb{1}_{\{Z_{m+1}=j\}} \prod_{k\in\mathcal{S}} z_k^{J_{m+1,k}}] = \mathbb{P}(Z_{m+1} = j)\mathbb{E}[\prod_{k\in\mathcal{S}} z_k^{J_{m+1,k}} \mid Z_{m+1} = j]$. Furthermore, we have by definition that $\lim_{m\to\infty}\mathbb{P}(Z_m = i) = q_i$ and $\lim_{m\to\infty}\mathbb{E}[\prod_{k\in\mathcal{S}} z_k^{J_{m,k}} \mid Z_m = i] = \widetilde{F}_i(\boldsymbol{z})$. Hence, by letting $m \to \infty$, (6) implies the following functional equation for all $i, j \in \mathcal{S}$:

$$q_j\widetilde{F}_j(\boldsymbol{z}) = \sum_{i\in\mathcal{S}} p_{i,j}q_i\widetilde{F}_i(\boldsymbol{z}_i^H)\widetilde{S}_{i,j}(\Sigma(\boldsymbol{z})). \tag{7}$$

3.1.2 Queue length moments at polling epochs

Now that we have derived the functional equation (7), we can compute all (cross-)moments of the queue lengths. We now show how to compute the first and second-order (cross-)moments of the marginal queue lengths found in the system at polling instants. Higher-order (cross-)moments can be computed through the same methodology, at the cost of a larger computational complexity.

First, recall that $\mathbb{E}[L_l \mid \text{server just polled } Q_j] = \mathbb{E}[F_{j,l}] = f_j(l)$, $\mathbb{E}[L_l^2 \mid \text{server just polled } Q_j] = \mathbb{E}[F_{j,l}^2] = f_j(l) + f_j(l,l)$ and that $\mathbb{E}[L_lL_m \mid \text{server just polled } Q_j] = \mathbb{E}[F_{j,l}F_{j,m}] = f_j(l,m)$ for any $j, l, m \in \mathcal{S}, l \neq m$. To obtain these numbers, we first take the derivative with respect to z_l in both sides of (7). This leads, for $j, l \in \mathcal{S}$, to

$$q_j\widetilde{F}_j^{(l)}(\boldsymbol{z}) = \sum_{i\in\mathcal{S}} p_{i,j}q_i\lambda_l\mathbb{E}[S_{i,j}e^{-(\Sigma(\boldsymbol{z}))S_{i,j}}]\widetilde{F}_i(\boldsymbol{z}_i^H) + \sum_{i\in\mathcal{S}\setminus\{l\}} p_{i,j}q_i\widetilde{S}_{i,j}(\Sigma(\boldsymbol{z}))\widetilde{F}_i^{(l)}(\boldsymbol{z}_i^H)$$
$$+ \sum_{i\in\mathcal{S}} p_{i,j}q_i\widetilde{S}_{i,j}(\Sigma(\boldsymbol{z}))\widetilde{H}_i^{(l)}(\boldsymbol{z})\widetilde{F}_i^{(i)}(\boldsymbol{z}_i^H). \tag{8}$$

Evaluating this equation in the point $\boldsymbol{z} = \mathbf{1}$ subsequently leads to

$$q_jf_j(l) = \sum_{i\in\mathcal{S}} p_{i,j}q_i\lambda_l\mathbb{E}[S_{i,j}] + \sum_{i\in\mathcal{S}\setminus\{l\}} p_{i,j}q_if_i(l) + \sum_{i\in\mathcal{S}} p_{i,j}q_ih_i(l)f_i(i). \tag{9}$$

This set of N^2 equations leads to expressions for $f_j(l), j, l \in \mathcal{S}$. If one is only interested in the values of $f_j(l)$ for a specific value of l, the complexity of these computations can be reduced to only solving a set of N equations, since an explicit expression for $f_i(i)$ is available; see Remark 3.1.

To find a similar set of equations for $f_j(l,m), j, l, m \in \mathcal{S}$, we first derive a functional equation for $\widetilde{F}_j^{(l,m)}(\boldsymbol{z})$. By differentiating both sides of (8) with respect to $z_m, m \in \mathcal{S}$, we obtain

$$q_j\widetilde{F}_j^{(l,m)}(\boldsymbol{z}) = \sum_{i\in\mathcal{S}} p_{i,j}q_i\lambda_l\lambda_m\mathbb{E}[S_{i,j}^2e^{-\Sigma(\boldsymbol{z})S_{i,j}}]\widetilde{F}_i(\boldsymbol{z}_i^H) + \sum_{i\in\mathcal{S}\setminus\{m\}} p_{i,j}q_i\lambda_l\mathbb{E}[S_{i,j}e^{-\Sigma(\boldsymbol{z})S_{i,j}}]\widetilde{F}_i^{(m)}(\boldsymbol{z}_i^H)$$
$$+ \sum_{i\in\mathcal{S}} p_{i,j}q_i\lambda_l\mathbb{E}[S_{i,j}e^{-\Sigma(\boldsymbol{z})S_{i,j}}]\widetilde{H}_i^{(m)}(\boldsymbol{z})\widetilde{F}_i^{(i)}(\boldsymbol{z}_i^H) + \sum_{i\in\mathcal{S}\setminus\{l\}} p_{i,j}q_i\lambda_m\mathbb{E}[S_{i,j}e^{-\Sigma(\boldsymbol{z})S_{i,j}}]\widetilde{F}_i^{(l)}(\boldsymbol{z}_i^H)$$
$$+ \sum_{i\in\mathcal{S}\setminus\{l,m\}} p_{i,j}q_i\widetilde{S}_{i,j}(\Sigma(\boldsymbol{z}))\widetilde{F}_i^{(l,m)}(\boldsymbol{z}_i^H) + \sum_{i\in\mathcal{S}\setminus\{l\}} p_{i,j}q_i\widetilde{S}_{i,j}(\Sigma(\boldsymbol{z}))\widetilde{H}_i^{(m)}(\boldsymbol{z})\widetilde{F}_i^{(i,l)}(\boldsymbol{z}_i^H)$$
$$+ \sum_{i\in\mathcal{S}} p_{i,j}q_i\lambda_m\mathbb{E}[S_{i,j}e^{-\Sigma(\boldsymbol{z})S_{i,j}}]\widetilde{H}_i^{(l)}(\boldsymbol{z})\widetilde{F}_i^{(i)}(\boldsymbol{z}_i^H) + \sum_{i\in\mathcal{S}} p_{i,j}q_i\widetilde{S}_{i,j}(\Sigma(\boldsymbol{z}))\widetilde{H}_i^{(l,m)}(\boldsymbol{z})\widetilde{F}_i^{(i)}(\boldsymbol{z}_i^H)$$
$$+ \sum_{i\in\mathcal{S}} p_{i,j}q_i\widetilde{S}_{i,j}(\Sigma(\boldsymbol{z}))\widetilde{H}_i^{(l)}(\boldsymbol{z})\widetilde{H}_i^{(m)}(\boldsymbol{z})\widetilde{F}_i^{(i,i)}(\boldsymbol{z}_i^H) + \sum_{i\in\mathcal{S}\setminus\{l,m\}} p_{i,j}q_i\widetilde{S}_{i,j}(\Sigma(\boldsymbol{z}))\widetilde{H}_i^{(l)}(\boldsymbol{z})\widetilde{F}_i^{(i,m)}(\boldsymbol{z}_i^H).$$

Similar to the computations above, evaluating this equation in the point $\boldsymbol{z} = \mathbf{1}$ leads to

$$
\begin{aligned}
q_j f_j(l, m) = &\sum_{i \in \mathcal{S}} p_{i,j} q_i \lambda_l \lambda_m \mathbb{E}[S_{i,j}^2] + \sum_{i \in \mathcal{S} \setminus \{m\}} p_{i,j} q_i \lambda_l \mathbb{E}[S_{i,j}] f_i(m) + \sum_{i \in \mathcal{S}} p_{i,j} q_i \lambda_l \mathbb{E}[S_{i,j}] h_i(m) f_i(i) \\
&+ \sum_{i \in \mathcal{S} \setminus \{l\}} p_{i,j} q_i \lambda_m \mathbb{E}[S_{i,j}] f_i(l) + \sum_{i \in \mathcal{S} \setminus \{l,m\}} p_{i,j} q_i f_i(l, m) + \sum_{i \in \mathcal{S} \setminus \{l\}} p_{i,j} q_i h_i(m) f_i(i, l) \\
&+ \sum_{i \in \mathcal{S}} p_{i,j} q_i \lambda_m \mathbb{E}[S_{i,j}] h_i(l) f_i(i) + \sum_{i \in \mathcal{S}} p_{i,j} q_i h_i(l, m) f_i(i) \\
&+ \sum_{i \in \mathcal{S}} p_{i,j} q_i h_i(l) h_i(m) f_i(i, i) + \sum_{i \in \mathcal{S} \setminus \{m\}} p_{i,j} q_i h_i(l) f_i(i, m).
\end{aligned}
\tag{10}
$$

For any $j, l, m \in \mathcal{S}$, this constitutes a set of N^3 equations for $f_j(l, m)$. Since expressions for $f_i(j)$, $i, j \in \mathcal{S}$, are already known by earlier computations, expressions for $f_j(l, m)$ can now be calculated for all $j, l, m \in \mathcal{S}$. As mentioned above, the expressions for $f_j(l)$ and $f_j(l, m)$ then subsequently lead to expressions for the first- and second order (cross-)moments of the queue lengths when the server polls Q_j. Although these moments may be of separate interest, we also use the derived expressions for $f_j(l)$ and $f_j(l, m)$ in Section 3.2 to obtain moments for L_j, the queue length of Q_j at an arbitrary point in time. We finish this section with the observation that the sets of equations expressed in (9) and (10) are uniquely solvable, provided that the aggregate workload ρ is smaller than one. One can confirm this by reducing (9) and (10) to equation sets of the form $A\boldsymbol{x} = \boldsymbol{b}$, and showing in a tedious, but straightforward way that the coefficient matrix A is invertible in that case.

Remark 3.1. For any $i \in \mathcal{S}$, the term $f_i(i)$ can be computed explicitly. To do this, we make use of an observation, most notably made by [9], that each time a visit beginning or a service completion occurs, this coincides with either a service beginning or a visit completion. All service beginning epochs in a visit period to Q_i are also service completion epochs at Q_i, except the first service beginning epoch, because it is actually a visit beginning epoch. Likewise, all service completion epochs at Q_i are also service beginning epochs at that queue, except the last service completion epoch, because it is actually a visit completion epoch. Since ζ_i denotes the fraction of service beginning (completion) epochs that also count as a visit beginning (completion) epoch, this observation leads to

$$
\zeta_i \widetilde{F}_i(\boldsymbol{z}) + \widetilde{M}_i(\boldsymbol{z}) = \zeta_i \widetilde{G}_i(\boldsymbol{z}) + \widetilde{N}_i(\boldsymbol{z}),
\tag{11}
$$

or more specifically for the means,

$$
\zeta_i f_i(i) + m_i(i) = \zeta_i g_i(i) + n_i(i).
\tag{12}
$$

Over the course of a service time at Q_i, on average ρ_i type-i customers arrive, after which one type-i customer leaves the system because its service is completed. Therefore, $m_i(i) - n_i(i) = \rho_i - 1$. Furthermore, we have by Property 2.1 that $g_i(i) = h_i(i) f_i(i)$. Relation (12) therefore reduces to an explicit expression for $f_i(i)$:

$$
f_i(i) = \frac{1 - \rho_i}{\zeta_i(1 - h_i(i))} = \frac{\lambda_i \sigma(1 - \rho_i)}{q_i(1 - \rho)(1 - h_i(i))},
\tag{13}
$$

where the second equality follows from the fact that $\zeta_i = 1/(\lambda_i \mathbb{E}[C_i])$ combined with (2).

3.2 Joint queue length at an arbitrary point in time

In Section 3.1, we have studied the PGF and moments of the joint queue length distribution at polling epochs. We now extend these results to obtain results for the queue lengths *at an arbitrary point in time*. We largely follow the approach of [5, Theorem 1] to express $\widetilde{L}(\boldsymbol{z})$, the PGF of the stationary joint queue length at an arbitrary point in time, in the conditional queue length PGFs studied above. Expressions for all (cross-)moments of the unconditional queue lengths in the moments of the queue lengths found at polling epochs subsequently follow from this relation.

To relate the unconditional queue length to the various conditional queue lengths studied before, we first observe that the server serves Q_i a fraction ρ_i of the time. At an arbitrary epoch during the remaining fraction

$(1-\rho_i)$ of time, the server is in a switch-over process, which with probability $\frac{q_i p_{i,k} \mathbb{E}[S_{i,k}]}{\sigma}$ happens to be a switch-over from Q_i to Q_k. As a result, the unconditional PGF $\widetilde{L}(\boldsymbol{z})$ satisfies

$$\widetilde{L}(\boldsymbol{z}) = \sum_{i \in \mathcal{S}} \left(\rho_i \widetilde{X}_i(\boldsymbol{z}) + \frac{(1-\rho)q_i}{\sigma} \sum_{k \in \mathcal{S}} p_{i,k} \mathbb{E}[S_{i,k}] \widetilde{Y}_{i,k}(\boldsymbol{z}) \right), \tag{14}$$

where the PGFs $\widetilde{X}_i(\boldsymbol{z})$ and $Y_{i,k}(\boldsymbol{z})$ represent the joint queue lengths at arbitrary points during a visit period at Q_i and a switch-over period from Q_i to Q_k, respectively. The customer population present in the system at an arbitrary point in a visit period to Q_i is comprised of the population already there at the start of the current type-i service, and the customers that have arrived during the past part of the current service period. As these two components are independent, we have that $\widetilde{X}_i(\boldsymbol{z}) = \widetilde{M}_i(\boldsymbol{z}) \frac{1 - \widetilde{B}_i(\Sigma(\boldsymbol{z}))}{\Sigma(\boldsymbol{z}) \mathbb{E}[B_i]}$. Furthermore, it is easy to see that $\widetilde{N}_i(\boldsymbol{z}) = z_i^{-1} \widetilde{B}_i(\Sigma(\boldsymbol{z})) \widetilde{M}_i(\boldsymbol{z})$. Combining these two relations with (11) leads to

$$\widetilde{X}_i(\boldsymbol{z}) = \frac{\zeta_i}{\mathbb{E}[B_i]} \frac{z_i(\widetilde{F}_i(\boldsymbol{z}) - \widetilde{G}_i(\boldsymbol{z}))}{z_i - \widetilde{B}_i(\Sigma(\boldsymbol{z}))} \frac{1 - \widetilde{B}_i(\Sigma(\boldsymbol{z}))}{\Sigma(\boldsymbol{z})}, \tag{15}$$

where, due to the fact that the service disciplines satisfy Property 2.1,

$$\widetilde{G}_i(\boldsymbol{z}) = \widetilde{F}_i(\boldsymbol{z}_i^H). \tag{16}$$

Similarly, as the customer population at an arbitrary point in a switch-over period from Q_i to Q_k is comprised of the population at the end of the past visit period to Q_i and the subsequent customer arrivals in the past part of the switch-over time, we have that

$$\widetilde{Y}_{i,k}(\boldsymbol{z}) = \widetilde{G}_i(\boldsymbol{z}) \frac{1 - \widetilde{S}_{i,k}(\Sigma(\boldsymbol{z}))}{\Sigma(\boldsymbol{z}) \mathbb{E}[S_{i,k}]}. \tag{17}$$

A combination of the equations (14)–(17) leads to the unconditional PGF $\widetilde{L}$ of the joint queue length expressed in the PGFs $\widetilde{F}_i$ that represent the joint queue length at the moment the server polls Q_i.

We now show how one can use this relation to derive expressions for the unconditional mean marginal queue lengths $\mathbb{E}[L_i] = l(i)$. The same method can be used to obtain expressions for higher (cross-)moments, although the computations become lengthier. By using (15), we first obtain the first moment of $X_{i,i}$ as follows:

$$
\begin{aligned}
x_i(i) &= \lim_{z_i \uparrow 1} \frac{d}{dz_i} (\widetilde{X}_i(\boldsymbol{z})|_{z_k=1 \ \forall k \neq i}) = \lim_{z_i \uparrow 1} \frac{d}{dz_i} \left(\frac{\zeta_i}{\mathbb{E}[B_i]} \frac{z_i(\mathbb{E}[z_i^{F_{i,i}}] - \mathbb{E}[z_i^{G_{i,i}}])}{z_i - \widetilde{B}_i(\lambda_i(1-z_i))} \frac{1 - \widetilde{B}_i(\lambda_i(1-z_i))}{\lambda_i(1-z_i)} \right) \\
&= \frac{\zeta_i(f_i(i) - g_i(i))}{1 - \rho_i} + \frac{\lambda_i^2 \zeta_i \mathbb{E}[B_i^2](f_i(i) - g_i(i))}{2(1-\rho_i)^2} + \frac{\zeta_i(f_i(i,i) - g_i(i,i))}{2(1-\rho_i)} + \frac{\lambda_i \mathbb{E}[B_i^2]}{2\mathbb{E}[B_i]} \\
&= 1 + \frac{\lambda_i^2 \mathbb{E}[B_i^2] + \zeta_i(f_i(i,i)(1 - h_i(i)^2) - f_i(i)h_i(i,i))}{2(1-\rho_i)} + \frac{\lambda_i \mathbb{E}[B_i^2]}{2\mathbb{E}[B_i]}, \tag{18}
\end{aligned}
$$

where we used in the fourth equality that $\zeta_i(f_i(i) - g_i(i)) = 1 - \rho_i$ (cf. Remark 3.1), and the fact that $g_i(i,i) = f_i(i,i)(h_i(i))^2 + f_i(i)h_i(i,i)$ due to (16). A similar, but slightly shorter computation yields that the first moment of $X_{i,j}$, $i \neq j$ is given by

$$
\begin{aligned}
x_i(j) &= \lim_{z_j \uparrow 1} \frac{d}{dz_j} (\widetilde{X}_i(\boldsymbol{z})|_{z_k=1 \ \forall k \neq j}) = \lim_{z_j \uparrow 1} \frac{d}{dz_j} \left(\frac{\zeta_i}{\mathbb{E}[B_i]} \frac{\mathbb{E}[z_j^{F_{i,j}}] - \mathbb{E}[z_j^{G_{i,j}}]}{\lambda_j(1-z_j)} \right) \\
&= \frac{\zeta_i(g_i(j,j) - f_i(j,j))}{2\lambda_j \mathbb{E}[B_i]} = \frac{\zeta_i(2f_i(i,j)h_i(j) + f_i(i,i)(h_i(j))^2 + f_i(i)h_i(j,j))}{2\lambda_j \mathbb{E}[B_i]}, \tag{19}
\end{aligned}
$$

where the fourth equality again follows from (16), which implies that $g_i(j,j) = f_i(j,j) + 2f_i(i,j)h_i(j) + f_i(i,i)(h_i(j))^2 + f_i(i)h_i(j,j)$.

As for the mean queue length $y_{i,k}(j)$ during a switch-over period from Q_i to Q_k, we have by (17) that, for all $i, j, k \in \mathcal{S}$,

$$
\begin{aligned}
y_{i,k}(j) &= \lim_{z_j \uparrow 1} \frac{d}{dz_j} (\widetilde{Y}_i(\boldsymbol{z})|_{z_l=1 \ \forall l \neq j}) = \lim_{z_j \uparrow 1} \frac{d}{dz_j} \left(\mathbb{E}[z_j^{G_{i,j}}] \frac{1 - \widetilde{S}_{i,k}(\lambda_j(1-z_j))}{(\lambda_j(1-z_j))\mathbb{E}[S_{i,k}]} \right) = g_i(j) + \lambda_j \frac{\mathbb{E}[S_{i,k}^2]}{2\mathbb{E}[S_{i,k}]} \\
&= \mathbb{1}_{\{j \neq i\}} f_i(j) + f_i(i)h_i(j) + \lambda_j \frac{\mathbb{E}[S_{i,k}^2]}{2\mathbb{E}[S_{i,k}]}, \tag{20}
\end{aligned}
$$

where the last equality follows from $g_i(j) = \mathbb{1}_{\{j \neq i\}} f_i(j) + f_i(i) h_i(j)$ (cf. (16)).

We can now derive an expression for the unconditional mean queue length $\mathbb{E}[L_j]$ in terms of the f-terms computed in the previous section. After differentiating both sides of (14) and evaluating the result in $z = 1$, we obtain

$$\mathbb{E}[L_j] = \sum_{i \in \mathcal{S}} \left(\rho_i x_i(j) + \frac{(1-\rho) q_i}{\sigma} \sum_{k \in \mathcal{S}} p_{i,k} \mathbb{E}[S_{i,k}] y_{i,k}(j) \right). \tag{21}$$

Since we already found expressions for $x_i(i)$, $x_i(j)$ and $y_{i,k}(j)$ in (18), (19) and (20), respectively, $\mathbb{E}[L_j]$ is now obtained in terms of moments of the queue lengths at polling instants, which we have already considered in Section 3.1.2. Note that this expression for the mean queue length also easily allows for expressions for the mean waiting time and the mean amount of work. Regarding the latter, we provide an expression for the expected total amount of work in the system in the next section.

3.3 Pseudo-conservation law

For polling systems with a server that visits the queues in a cyclic fashion, a stochastic decomposition for the total amount of work in the system has been derived in [4]. In particular, the total amount of work in the polling system at an arbitrary epoch can be decomposed into the amount of work in a corresponding M/G/1 system at an arbitrary epoch and the amount of work in the polling system in a switching period of the server. This decomposition allows for the derivation of a strikingly simple expression for the weighted sum $\sum_{i \in \mathcal{S}} \rho_i \mathbb{E}[W_i]$. This result is known as the pseudo-conservation law (PCL). Following [4], the PCL has been extended to allow for polling systems with Markovian routing in [7], but this extension only allows the server to serve the queues in exclusively an exhaustive, gated or one-limited manner. In this section, we further extend the PCL to allow for any branching-type service discipline.

In particular, it is shown in [7] that the total amount of waiting work in polling systems with Markovian routing is given by

$$\sum_{i \in \mathcal{S}} \rho_i \mathbb{E}[W_i] = \rho \frac{\sum_{i \in \mathcal{S}} \lambda_i \mathbb{E}[B_i^2]}{2(1-\rho)} + \frac{1}{\sigma} \sum_{i \in \mathcal{S}} q_i \sum_{k \in \mathcal{S}} p_{i,k} \mathbb{E}[S_{i,k}] \mathbb{E}[\Psi_{i,k}], \tag{22}$$

where the latter term represents the total expected amount of work in the system during a switch-over period, and where $\mathbb{E}[\Psi_{i,k}]$ is the total expected amount of work in the system when the server is in the process of switching from Q_i to Q_k. The authors in [7] then determine $\mathbb{E}[\Psi_{i,k}]$ for the exhaustive, gated, and one-limited service discipline. Observe, however, that the amount of work in the system equals the sum of the (remaining) service requirements of all the customers present in the system. As a result, we have for a switch-over period from Q_i to Q_k that

$$\mathbb{E}[\Psi_{i,k}] = \sum_{j \in \mathcal{S}} y_{i,k}(j) \mathbb{E}[B_j]. \tag{23}$$

By combining (22) and (23) with (20) and (13), respectively, we thus have for the general case that

$$\sum_{i \in \mathcal{S}} \rho_i \mathbb{E}[W_i] = \rho \frac{\sum_{i \in \mathcal{S}} \lambda_i \mathbb{E}[B_i^2]}{2(1-\rho)} + \frac{1}{\sigma} \sum_{i \in \mathcal{S}} q_i \sum_{k \in \mathcal{S}} p_{i,k} \mathbb{E}[S_{i,k}] \sum_{j \in \mathcal{S} \setminus \{i\}} f_i(j) \mathbb{E}[B_j]$$

$$+ \frac{1}{1-\rho} \sum_{i \in \mathcal{S}} \lambda_i \frac{1-\rho_i}{1-h_i(i)} \sum_{k \in \mathcal{S}} p_{i,k} \mathbb{E}[S_{i,k}] \sum_{j \in \mathcal{S}} \mathbb{E}[B_j] h_i(j) + \frac{\rho}{2\sigma} \sum_{i \in \mathcal{S}} q_i \sum_{k \in \mathcal{S}} p_{i,k} \mathbb{E}[S_{i,k}^2], \tag{24}$$

provided that the service discipline pertaining to each queue satisfies Property 2.1. This expression uses the $f_i(j)$-terms that we computed in Section 3.1.2. In Section 4, we will use this newly derived PCL for optimisation purposes.

4 Optimisation of routing probabilities and exhaustiveness probabilities

In this section, we use results obtained in Section 3 to study the optimisation problem of how to choose the routing probabilities in the polling model so as to minimise a weighted sum $\sum_{i \in \mathcal{S}} c_i \mathbb{E}[W_i]$ of the mean waiting times (or

equivalently through Little's law, the mean queue lengths). Given that each of the queues adheres to a binomial exhaustive or a binomial gated service discipline, we also study the question of how to choose the exhaustiveness probabilities r_i with the same objective in mind.

Of course, one can optimise these numbers by implementing (21) including the set of equations (10) to compute $\sum_{i \in \mathcal{S}} c_i \mathbb{E}[W_i] = \sum_{i \in \mathcal{S}} \frac{c_i}{\lambda_i}(\mathbb{E}[L_i] - \rho_i)$, and searching through the complete parameter set using numerical optimisation methods. However, this method lacks transparency and provides little insight into the effects of the model parameters. Moreover, its computation time becomes prohibitively long as the number of queues increases. Therefore, there is a need for symbolic and transparent (near-)optimal expressions which are easy to implement and are suitable for optimisation purposes.

To allow for such expressions, suppose that the first two moments of the switch-over time distributions between each pair of queues are the same; i.e. $\mathbb{E}[S_{i,j}] = \mathbb{E}[S]$ and $\mathbb{E}[S_{i,j}^2] = \mathbb{E}[S^2]$ for all $i, j \in \mathcal{S}$. Furthermore, suppose that the server visits the queues according to a random routing scheme; i.e., the routing probability does not depend on the queue the server visited just before the switch-over time ($p_{i,j} = p_j$ for all $i \in \mathcal{S}$ and $q_j = p_j$ for all $j \in \mathcal{S}$). Under these assumptions, we derive simple and closed-form expressions for the routing probabilities and exhaustiveness probabilities that optimise the mean waiting or total amount of work in the system ($c_i = \rho_i$ for all $i \in \mathcal{S}$) in Section 4.1. For general weighted sums of the expected waiting times (i.e., for general c_i), we derive expressions of similar complexity in Section 4.2 that are nearly optimal, and we show them to be accurate. Contrary to the numerical method described above, the expressions that we derive provide insight into the effects of the model parameters on the waiting times and their computation times are negligible.

4.1 Minimising the mean total amount of work in the system

We start with optimising the mean total amount of work in the system, which is the sum of the mean amount $\sum_{i \in \mathcal{S}} \rho_i \mathbb{E}[W_i]$ of waiting work in the system and the mean amount $\sum_{i \in \mathcal{S}} \rho_i \frac{\mathbb{E}[B_i^2]}{\mathbb{E}[B_i]}$ of remaining work to be processed of any customer that is currently being served. Since the latter expression is insensitive to the routing probabilities and the exhaustiveness probabilities, the probabilities that minimise the mean total amount of work in the system also minimise $\sum_{i \in \mathcal{S}} \rho_i \mathbb{E}[W_i]$. We therefore focus on this expression in the remainder of this section.

Recall that the server now initiates a setup to Q_j with probability p_j after a visit period, regardless of which queue it actually visited. While doing so, it incurs a switch-over time with first two moments $\mathbb{E}[S]$ and $\mathbb{E}[S^2]$ for all $j \in \mathcal{S}$. Following the analysis of [7, Remark 5.4] based on results of [12], one can show that under these assumptions the PCL in (24) reduces to

$$\sum_{i \in \mathcal{S}} \rho_i \mathbb{E}[W_i] = \rho \frac{\sum_{i \in \mathcal{S}} \lambda_i \mathbb{E}[B_i^2]}{2(1 - \rho)} + \frac{\mathbb{E}[S]}{1 - \rho} \sum_{i \in \mathcal{S}} \frac{\rho_i(1 - \rho_i)}{p_i} + \rho \left(\frac{\mathbb{E}[S^2]}{2\mathbb{E}[S]} - \mathbb{E}[S] \right) + \frac{\mathbb{E}[S]}{1 - \rho} \sum_{i \in \mathcal{S}} \frac{\rho_i(1 - \rho_i)h_i(i)}{p_i(1 - h_i(i))}. \quad (25)$$

The last term is the only term in this expression that depends on the service disciplines of the queues. Furthermore, the summands $\frac{\mathbb{E}[S]}{1 - \rho} \frac{\rho_i(1 - \rho_i)h_i(i)}{p_i(1 - h_i(i))} = f_i(i)h_i(i)\mathbb{E}[B_i]$ of the last term equal the expected amount of work the server leaves behind at Q_i when completing a visit period there.

4.1.1 Routing probabilities

Considering (25), it is obvious that for any branching-type service discipline at any queue, the problem of finding the routing probabilities p_i^{opt} that minimise the mean total amount of work in the system is equivalent to the problem of finding the variable $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_N)$ that

$$\text{minimises} \quad f(\boldsymbol{\tau}) = \sum_{i \in \mathcal{S}} \frac{\rho_i(1 - \rho_i)}{\tau_i} \left(1 + \frac{h_i(i)}{1 - h_i(i)} \right) = \sum_{i \in \mathcal{S}} \frac{\rho_i(1 - \rho_i)}{\tau_i(1 - h_i(i))} \quad (26)$$

$$\text{subject to} \quad u(\boldsymbol{\tau}) = \sum_{i \in \mathcal{S}} \tau_i = 1, \quad v_{1,j}(\boldsymbol{\tau}) = -\tau_j \leq 0 \quad \text{and} \quad v_{2,j}(\boldsymbol{\tau}) = \tau_j - 1 \leq 0 \quad \text{for all } j \in \mathcal{S}.$$

This non-linear optimisation problem with equality and inequality constraints can be solved using a standard application of the Karush-Kuhn-Tucker (KKT) conditions (see e.g. [8, Section 5.5.3]). Let $\boldsymbol{\tau}^* = (\tau_1^*, \ldots, \tau_N^*)$ be given by $\tau_i^* = \frac{\sqrt{\rho_i(1 - \rho_i)/(1 - h_i(i))}}{\sum_{j \in \mathcal{S}} \sqrt{\rho_j(1 - \rho_j)/(1 - h_j(j))}}$. Define $\mathcal{L}$ as the number for which $\nabla f(\boldsymbol{\tau}^*) + \mathcal{L} \nabla u(\boldsymbol{\tau}^*)$ equals $\mathbf{0}$, i.e.,

$\mathcal{L} = \left(\sum_{j \in \mathcal{S}} \sqrt{\rho_j (1 - \rho_j)/(1 - h_j(j))} \right)^2$. Furthermore, let the N-dimensional vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ be equal to $\boldsymbol{0}$. It is then easily verified that $\boldsymbol{\tau}^*$, $\mathcal{L}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ satisfy the KKT conditions

$$\nabla f(\boldsymbol{\tau}^*) + \mathcal{L} \nabla u(\boldsymbol{\tau}^*) + \boldsymbol{a} \nabla \boldsymbol{v}_1(\boldsymbol{\tau}^*) + \boldsymbol{b} \nabla \boldsymbol{v}_2(\boldsymbol{\tau}^*) = \boldsymbol{0}, \qquad \text{(stationarity)}$$
$$u(\boldsymbol{\tau}^*) = 0, \boldsymbol{v}_1(\boldsymbol{\tau}^*) \le \boldsymbol{0}, \boldsymbol{v}_2(\boldsymbol{\tau}^*) \le \boldsymbol{0}, \qquad \text{(primal feasibility)}$$
$$\boldsymbol{a} \boldsymbol{v}_1(\boldsymbol{\tau}^*) = \boldsymbol{0}, \boldsymbol{b} \boldsymbol{v}_2(\boldsymbol{\tau}^*) = \boldsymbol{0}, \qquad \text{(complementary slackness)}$$
$$\boldsymbol{a} \ge \boldsymbol{0} \text{ and } \boldsymbol{b} \ge \boldsymbol{0}. \qquad \text{(non-negativity)}$$

The existence of values of $\mathcal{L}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ that satisfy the KKT conditions is required for $\boldsymbol{\tau}^*$ to be the solution to the optimisation problem, but it does in general not imply that $\boldsymbol{\tau}^*$ is indeed optimal. However, since the objective function $f(\boldsymbol{\tau})$ is convex in $\tau_1, \dots, \tau_N$, these conditions are sufficient for $\boldsymbol{\tau}^*$ to be the solution to (26). As such, the optimal routing probabilities p_i that minimise the mean total amount of work in the system are given by

$$p_i^{opt} = \frac{\sqrt{\rho_i (1 - \rho_i)/(1 - h_i(i))}}{\sum_{j \in \mathcal{S}} \sqrt{\rho_j (1 - \rho_j)/(1 - h_j(j))}}. \tag{27}$$

Remark 4.1. The optimal routing probabilities given in (27) generalise results found in [6, Section 4]. In that paper, the authors derive optimal routing probabilities for the special cases of exhaustive and gated service; i.e., $h_i(i) = 0$ and $h_i(i) = \rho_i$, respectively.

4.1.2 Exhaustiveness probabilities

We now assume that each of the queues adheres to either a binomial exhaustive or a binomial gated service discipline. We therefore partition the set $\mathcal{S}$ of queue indices in a set $\mathcal{I}_{BE}$ of indices corresponding to queues served according to the binomial exhaustive service discipline and a set $\mathcal{I}_{BG}$ of indices referring to queues with the binomial gated discipline. Recall that the last term in (25) is the only term in that expression that is sensitive to the service discipline, and thus also to the exhaustiveness probabilities r_i. As we now have that $h_i(i) = 1 - (1 - \rho_i \mathbb{1}_{\{i \in \mathcal{I}_{BG}\}}) r_i$, the last term of (25) can be simplified to $\frac{\mathbb{E}[S]}{1-\rho} \sum_{i \in \mathcal{S}} k_i (\frac{1}{r_i} - (1 - \rho_i \mathbb{1}_{\{i \in \mathcal{I}_{BG}\}}))$, where

$$k_i = \frac{\rho_i (1 - \rho_i)}{p_i (1 - \rho_i \mathbb{1}_{\{i \in \mathcal{I}_{BG}\}})}. \tag{28}$$

We aim to find the exhaustiveness probabilities that minimise $\sum_{i \in \mathcal{S}} \rho_i \mathbb{E}[W_i]$. Of course, when there are no restrictions on r_i, all exhaustiveness probabilities should be chosen equal to one in order to prevent unnecessary setups, so that the amount of work in the system is minimised. However, as a result of this choice, the waiting times of the various customers may be heavily influenced by their type in combination with their time of arrival depending on the position of the server. This introduces a source of customer unfairness, especially in case of the binomial exhaustive service discipline. Furthermore, in practice, there may be costs involved with having demanding customer types, or equivalently, high exhaustiveness probabilities. In the wireless random-access network application, a high exhaustiveness of one node may heavily delay the transmission of packets by other nodes. Therefore, we add the constraint $\sum_{i \in \mathcal{S}} d_i r_i \le 1$ to the problem, where the parameters $d_i > 0$ can be interpreted as cost parameters.

Taking everything into account, the problem of finding the optimal exhaustiveness probabilities reduces to the problem of finding the vector $\boldsymbol{\tau} = (\tau_1, \dots, \tau_N)$ that

$$\text{minimises} \quad f(\boldsymbol{\tau}) = \sum_{i \in \mathcal{S}} \frac{k_i}{\tau_i} \tag{29}$$

$$\text{subject to} \quad v_{1,i}(\boldsymbol{\tau}) = -\tau_i \le 0, \quad v_{2,i}(\boldsymbol{\tau}) = \tau_i - 1 \le 0 \quad \text{and} \quad v_3(\boldsymbol{\tau}) = \sum_{j \in \mathcal{S}} d_j \tau_j \le 1, \quad \text{for all } i \in \mathcal{S}.$$

Note that $f(\boldsymbol{\tau})$ is a decreasing function in $\tau_1, \dots, \tau_N$. Thus, if $\sum_{i \in \mathcal{S}} d_i < 1$, the constraint $v_3(\boldsymbol{\tau}) \le 0$ cannot be binding, as the constraint $v_{2,i}(\boldsymbol{\tau}) \le \boldsymbol{0}$ will prohibit that. As such, for this case, the solution to this problem is given by $\tau_i^* = 1$ for all $i \in \mathcal{S}$. Therefore, we proceed with considering the case $\sum_{i \in \mathcal{S}} d_i \ge 1$. If the constraints $v_{1,i}(\boldsymbol{\tau}) \le 0$ and $v_{2,i}(\boldsymbol{\tau}) \le 0$ did not exist, one could show that (29) is minimised by the vector $\boldsymbol{\tau}(0)$ with elements

$$\tau_i(0) = \frac{\sqrt{k_i/d_i}}{\sum_{j \in \mathcal{S}} \sqrt{k_j d_j}} \tag{30}$$

for each $i \in \mathcal{S}$. However, this vector does not necessarily satisfy the constraint $v_{2,i}(\boldsymbol{\tau}(0)) \leq 0$. It is reasonable to conjecture that if $\tau_i(0) \geq 1$, the optimal vector $\boldsymbol{\tau}^*$ satisfies $\tau_i^* = 1$. In such a case, the optimal solution may be found by truncating any values in (30) at one as needed, and, given that these values equal one, re-evaluating the problem to solve for the remaining values. As any of the remaining values may become greater than one after re-evaluation, this needs to be iterated until all values are not larger than one. At most N of these iterations are needed to achieve this.

To summarise all of the above, it is reasonable to conjecture that the solution $\boldsymbol{\tau}^* = \boldsymbol{\tau}(N)$ to the problem specified in (29) has elements that are defined through the recursion

$$\tau_i(j) = \mathbb{1}_{\{\tau_i(j-1) \geq 1 \vee \sum_{k \in \mathcal{S}} d_k < 1\}} + \mathbb{1}_{\{\tau_i(j-1) < 1 \wedge \sum_{k \in \mathcal{S}} d_k \geq 1\}} \frac{\left(1 - \sum_{k \in \mathcal{S}} d_k \mathbb{1}_{\{\tau_k(j-1) < 1\}}\right)\sqrt{k_i/d_i}}{\sum_{k \in \mathcal{S}} \mathbb{1}_{\{\tau_k(j-1) < 1\}} \sqrt{k_j d_j}} \quad (31)$$

for $j = 1, \ldots, N$. The number j corresponds to the j-th step of the recursion. We now show that $\boldsymbol{\tau}^* = \boldsymbol{\tau}(N)$ is indeed a solution to this problem including all mentioned constraints. To this end, we introduce $\mathcal{E} = \mathbb{1}_{\{\sum_{j \in \mathcal{S}} d_j \tau_j^* = 1\}} \left(\sum_{j \in \mathcal{S}} \mathbb{1}_{\{\tau_j^* < 1\}} \sqrt{k_j d_j}\right)^2$. Furthermore, let the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ be given by $a_i = 0$ and $b_i = \mathbb{1}_{\{\tau_i^* = 1\}}(k_i - d_i \mathcal{E})$ respectively. Through some straightforward computations, it can be shown that these particular choices for $\boldsymbol{a}, \boldsymbol{b}$ and $\mathcal{E}$ satisfy the following KKT conditions for the problem in (29):

$$\nabla f(\boldsymbol{\tau}^*) + \boldsymbol{a}\nabla v_1(\boldsymbol{\tau}^*) + \boldsymbol{b}\nabla v_2(\boldsymbol{\tau}^*) + \mathcal{E}\nabla v_3(\boldsymbol{\tau}^*) = \boldsymbol{0}, \qquad \text{(stationarity)}$$
$$\boldsymbol{v}_1(\boldsymbol{\tau}^*) \leq \boldsymbol{0}, \boldsymbol{v}_2(\boldsymbol{\tau}^*) \leq \boldsymbol{0}, v_3(\boldsymbol{\tau}^*) \leq 0, \qquad \text{(primal feasibility)}$$
$$\boldsymbol{a}\boldsymbol{v}_1(\boldsymbol{\tau}^*) = \boldsymbol{0}, \boldsymbol{b}\boldsymbol{v}_2(\boldsymbol{\tau}^*) = \boldsymbol{0}, \mathcal{E}v_3(\boldsymbol{\tau}^*) = 0, \qquad \text{(complementary slackness)}$$
$$\boldsymbol{a} \geq \boldsymbol{0}, \boldsymbol{b} \geq \boldsymbol{0} \text{ and } \mathcal{E} \geq 0. \qquad \text{(non-negativity)}$$

Since the KKT conditions are satisfied and $f(\boldsymbol{\tau})$ is a convex function in $\tau_1, \ldots, \tau_N$, $\boldsymbol{\tau}^*$ is indeed optimal for this problem.

Going back to the original problem of finding the routing probabilities that minimise the mean total amount of work in the system under the restriction that $\sum_{i \in \mathcal{S}} d_i r_i \leq 1$, we thus have that the optimal exhaustiveness probabilities r_i^{opt} are given by

$$r_i^{opt} = \tau_i(N), \qquad (32)$$

where $\tau_i(N)$ is defined through the recursion (31) together with the initial value (30), and k_i as defined in (28).

Remark 4.2. In Sections 4.1.1 and 4.1.2, we have derived separate expressions for the optimal routing probabilities and exhaustiveness probabilities. Note that the found expressions for p_i^{opt} (r_i^{opt}) involve the parameters r_i (p_i), so that there is an interaction between the optimal routing probabilities and the optimal exhaustiveness probabilities. Joint optimisation of both the routing probabilities and exhaustiveness probabilities seems to be a hard problem. One may, however, obtain optimal values for both the routing probabilities and the exhaustiveness probabilities by using an alternating approach that first finds the optimal routing probabilities given an arbitrary set of exhaustiveness probabilities, then determines new optimal exhaustiveness probabilities based on the newly chosen routing probabilities, and so on. Numerical experiments show that only a few of these iterations are already enough to obtain virtually optimal values for these parameters.

4.2 Minimising a weighted sum of the mean waiting times

Now that we have found the routing probabilities and the exhaustiveness probabilities that minimise the expected amount of work in the system, the question arises which routing probabilities and exhaustiveness probabilities minimise the weighted sum $\sum_{i \in \mathcal{S}} c_i \mathbb{E}[W_i] = \sum_{i \in \mathcal{S}} \frac{c_i}{\lambda_i}(\mathbb{E}[L_i] - \rho_i)$ with arbitrary, positive weights c_i that are not necessarily equal to ρ_i. By (21), this sum depends on $f_i(j)$ and $f_i(i, j)$ for each $i, j \in \mathcal{S}$ and thus constitutes an intricate function of the model parameters. Optimisation of this function is hard and does not lead to simple expressions for optimal model parameters. Therefore, we instead aim to find simple expressions that lead to a near-optimal value of $\sum_{i \in \mathcal{S}} c_i \mathbb{E}[W_i]$, or equivalently, a near-optimal value of $\sum_{i \in \mathcal{S}} \frac{c_i}{\lambda_i} \mathbb{E}[L_i]$. To this end, we initially consider a more tractable problem, namely the optimisation of the weighted sum $\sum_{i \in \mathcal{S}} \frac{c_i}{\lambda_i} f_i(i)$. We thus replace $\mathbb{E}[L_i]$, the mean queue length of Q_i at any point in time, by $f_i(i)$, which refers to the mean queue length of Q_i when it is polled by the server. In Section 4.2.1, we derive expressions for routing probabilities and exhaustiveness probabilities that minimise $\sum_{i \in \mathcal{S}} \frac{c_i}{\lambda_i} f_i(i)$. Using numerical results, we will see in Section 4.2.1 that these expressions also represent probabilities that nearly optimise $\sum_{i \in \mathcal{S}} c_i \mathbb{E}[W_i]$.

4.2.1 Near-optimal expressions

We initially study the adapted problem of minimising $\sum_{i \in \mathcal{S}} \frac{c_i}{\lambda_i} f_i(i)$. Due to (13), we thus wish to minimise

$$\sum_{i \in \mathcal{S}} \frac{c_i}{\lambda_i} f_i(i) = \sum_{i \in \mathcal{S}} \frac{c_i \mathbb{E}[S](1 - \rho_i)}{p_i(1 - \rho)(1 - h_i(i))}. \tag{33}$$

To find expressions for the routing probabilities $p_i^{n\text{-}opt}$ that minimise this sum, observe that this adapted problem is equivalent to problem (26) but with $\rho_i(1 - \rho_i)$ in the numerator replaced by $c_i(1 - \rho_i)$. By following the analysis of Section 4.1.1, one finds that the $p_i^{n\text{-}opt}$ are given by

$$p_i^{n\text{-}opt} = \frac{\sqrt{c_i(1 - \rho_i)/(1 - h_i(i))}}{\sum_{j \in \mathcal{S}} \sqrt{c_j(1 - \rho_j)/(1 - h_j(j))}} \tag{34}$$

for all $i \in \mathcal{S}$.

We now consider the exhaustiveness probabilities, where we again take the constraint $\sum_{i \in \mathcal{S}} d_i r_i \leq 1$ into account. Recall that $h_i(i) = 1 - (1 - \rho_i \mathbb{1}_{\{i \in \mathcal{I}_{BG}\}}) r_i$ when the server serves each of the queues according to the binomial exhaustive or the binomial gated service discipline. As such, we observe using (33) that minimising $\sum_{i \in \mathcal{S}} \frac{c_i}{\lambda_i} f_i(i)$ is equivalent to the minimisation of $\sum_{i \in \mathcal{S}} \frac{\kappa_i}{r_i}$, where $\kappa_i = \frac{c_i(1 - \rho_i)}{p_i(1 - \rho_i \mathbb{1}_{\{i \in \mathcal{I}_{BG}\}})}$. By performing similar calculations to those in Section 4.1.2, we now have that the exhaustiveness probabilities $r_i^{n\text{-}opt}$ that minimise (33) are given by

$$r_i^{n\text{-}opt} = r_i(N), \tag{35}$$

where $r_i(j)$ is for all $i, j \in \mathcal{S}$ recursively defined through

$$r_i(j) = \mathbb{1}_{\{r_i(j-1) \geq 1 \vee \sum_{k \in \mathcal{S}} d_k < 1\}} + \mathbb{1}_{\{r_i(j-1) < 1 \wedge \sum_{k \in \mathcal{S}} d_k \geq 1\}} \frac{(1 - \sum_{k \in \mathcal{S}} d_k \mathbb{1}_{\{r_i(j-1) < 1\}}) \sqrt{\kappa_i/d_i}}{\sum_{k \in \mathcal{S}} \mathbb{1}_{\{r_i(j-1) < 1\}} \sqrt{\kappa_j d_j}}$$

with

$$r_i(0) = \frac{\sqrt{\kappa_i/d_i}}{\sum_{j \in \mathcal{S}} \sqrt{\kappa_j d_j}}.$$

We have now found the routing probabilities $p_i^{n\text{-}opt}$ and exhaustiveness probabilities $r_i^{n\text{-}opt}$ that minimise the weighted sum $\sum_{i \in \mathcal{S}} \frac{c_i}{\lambda_i} f_i(i)$. Observe however, that in case $c_i = \rho_i$ for all $i \in \mathcal{S}$, the expressions in (34) and (35) coincide with (27) and (32). As such, these expressions also represent the probabilities that minimise $\sum_{i \in \mathcal{S}} c_i \mathbb{E}[W_i]$ when $c_i = \rho_i$. It is therefore reasonable to assume that for general c_i, the $p_i^{n\text{-}opt}$ and $r_i^{n\text{-}opt}$ nearly optimise the weighted sum of mean waiting times. In the next section, we conclude on the basis of numerical results that this is indeed the case, so that (34) and (35) can be used for the optimisation of waiting times.

4.2.2 Numerical validation

In this section, we numerically study the accuracy of the near-optimal values $p_i^{n\text{-}opt}$ and $r_i^{n\text{-}opt}$ as computed in (34) and (35). To this end, we consider a collection of 1152 model instances corresponding to all possible combinations of the parameter settings given in Table 1. For each of these systems, we compute the smallest possible value β^{opt} of the weighted sum of mean waiting times, by determining the optimal routing and exhaustiveness probabilities using numerical optimisation methods in combination with the results found in Section 3. We also compute the routing and exhaustiveness probabilities derived in Section 4.2.1 that should nearly optimise the weighted sum $\sum_{i \in \mathcal{S}} c_i \mathbb{E}[W_i]$ by iteratively calculating (34) and (35) using an alternating approach as sketched in Remark 4.2. We denote the value of the weighted sum that corresponds to these probabilities as $\beta^{n\text{-}opt}$.

Based on these numbers, we calculate the accuracy error $\Delta^{n\text{-}opt}$ of the near-optimal probabilities for each system:

$$\Delta^{n\text{-}opt} = 100\% \times \frac{\beta^{n\text{-}opt} - \beta^{opt}}{\beta^{opt}}.$$

Parameter	Considered parameter settings
N	$2, 3, 4, 5$
Service policy	Binomial exhaustive, binomial gated
ρ	$0.1, 0.5, 0.99$
$(B_i)_{i \in \mathcal{S}}$	$\left(\text{Exponential}\left(\frac{1}{2i}\right)\right)_{i \in \mathcal{S}}, (\text{Deterministic}\,(i))_{i \in \mathcal{S}}$
$(S_i)_{i \in \mathcal{S}}$	$(\text{Uniform}\,(0,1))_{i \in \mathcal{S}}, (\text{Uniform}\,(0,100))_{i \in \mathcal{S}}$
$(\lambda_i)_{i \in \mathcal{S}}$	$\left(\frac{\rho}{N\mathbb{E}[B_i]}\right)_{i \in \mathcal{S}}, \left(\frac{2i\rho}{N(N+1)\mathbb{E}[B_i]}\right)_{i \in \mathcal{S}}, \left(\frac{2(N+1-i)\rho}{N(N+1)\mathbb{E}[B_i]}\right)_{i \in \mathcal{S}}$
$(c_i)_{i \in \mathcal{S}}$	$\left(\rho_i^2\right)_{i \in \mathcal{S}}, \left(e^{N+1-i}\right)_{i \in \mathcal{S}}$
$(d_i)_{i \in \mathcal{S}}$	$\left((N+1-i)^{-1}\right)_{i \in \mathcal{S}}, \left(\sqrt{N+1-i}\right)_{i \in \mathcal{S}}$

Table 1: Parameter settings of the polling systems used for the numerical study of Section 4.2.2.

	0-0.01%	0.01-1%	1-10%	>10%
% of accuracy errors $\Delta^{n\text{-}opt}$	59.03%	31.68%	8.85%	0.43%
% of accuracy errors Δ^{base}	0.26%	7.99%	33.07%	58.68%

Table 2: The accuracy differences $\Delta^{n\text{-}opt}$ and Δ^{base} categorised in bins.

For the sake of comparison, we also consider the base-line scenario where the routing and exhaustiveness probabilities are chosen in a naive manner, namely $p_i = \frac{1}{N}$ and $r_i = \frac{1}{d_i N}$ for all $i \in \mathcal{S}$. This leads to the weighted sum denoted by β^{base}, so that the accuracy error Δ^{base} is defined similarly to $\Delta^{n\text{-}opt}$. Note that this base-line scenario is optimal for completely symmetric systems. In Table 2, the errors $\Delta^{n\text{-}opt}$ and Δ^{base} pertaining to all model instances are summarised. In particular, we see that $\Delta^{n\text{-}opt}$ is smaller than 1% in more than 90% of all cases, and is even smaller than 0.01% in more than half of the cases. This suggests that the values $p_i^{n\text{-}opt}$ and $r_i^{n\text{-}opt}$ indeed virtually always lead to a weighted sum of mean waiting times that is close to optimal. They also seem to perform much better than the base-line scenario, since Table 2 shows almost completely opposite numbers for Δ^{base}. In particular, the accuracy errors of the base-line scenario are larger than 1% in more than 90% of all cases, and they even exceed 10% in more than half of the cases. This effect is also captured by the fact that the average value of $\Delta^{n\text{-}opt}$ equals 0.425%, and that of Δ^{base} equals 24.18%.

To give insight into parameter effects, Table 3 displays average values of $\Delta^{n\text{-}opt}$ categorised in some of the model parameters. From Table 3(a), we conclude that the accuracy of the near-optimal values is hardly influenced by the number of queues in the system. However, judging by Table 3(b), the accuracy is sensitive to the workload ρ offered to the server. As any choice for the routing and exhaustiveness probabilities is optimal in case of a zero workload, it makes sense that the accuracy degrades slowly when the workload increases. Table 3(c) suggests that the near-optimal values tend to perform better when there is less stochasticity in the system. Judging by Table 3(d), the performance is also increasing in the average duration of the switch-over times. This can be explained by the fact that the routing or exhaustiveness probabilities have less impact on the waiting time when the switch-over times become an increasing source of waiting time. Finally, Tables 3(e) and 3(f) suggest that a higher level of asymmetry in the model parameters leads to larger inaccuracies. This is as expected, as the near-optimal probabilities are optimal when the system to be optimised is completely symmetric.

Remark 4.3. In Sections 4.1.2 and 4.2.1, we have derived expressions for exhaustiveness probabilities that (nearly) optimise a weighted sum of the mean waiting times. However, in practice, one may also be interested in keeping the level of variation in the waiting times low. In an effort to reduce the level of variation, one may thus choose to adapt the exhaustiveness probability in a dynamic fashion at the start of every n-th visit period at that queue, based on the number of customers present in the queue at that particular polling epoch. More specifically, let r_i^{opt} be the expression of the (near-)optimal exhaustiveness probability at Q_i as found before, and let $f_{i,r_i^{opt}}(i)$ be the corresponding expected queue length at Q_i at the start of any visit period to Q_i, that can be computed through (13). By using (16), the expected number of customers that the server leaves behind at that queue when initiating the next switch-over period, is given by $f_{i,r_i^{opt}}(i)h_i(i) = f_{i,r_i^{opt}}(i)\left(1 - (1 - \rho_i \mathbb{1}_{\{i \in \mathcal{I}_{BG}\}})r_i^{opt}\right)$. Like-

14

N	2	3	4	5
Average $\Delta^{n\text{-}opt}$	0.41%	0.43%	0.43%	0.43%

(b)

ρ	0.1	0.5	0.99
Average $\Delta^{n\text{-}opt}$	0.00%	0.03%	1.24%

(c)

$(B_i)_{i\in\mathcal{S}}$	$(\text{Exponential}\,(1/2i))_{i\in\mathcal{S}}$	$(\text{Deterministic}\,(i))_{i\in\mathcal{S}}$
Average $\Delta^{n\text{-}opt}$	0.66%	0.19%

(d)

$(S_i)_{i\in\mathcal{S}}$	$(\text{Uniform}\,(0,1))_{i\in\mathcal{S}}$	$(\text{Uniform}\,(0,100))_{i\in\mathcal{S}}$
Average $\Delta^{n\text{-}opt}$	0.61%	0.24%

(e)

$(c_i)_{i\in\mathcal{S}}$	$(\rho_i^2)_{i\in\mathcal{S}}$	$(e^{N+1-i})_{i\in\mathcal{S}}$
Average $\Delta^{n\text{-}opt}$	0.23%	0.62%

(f)

$(d_i)_{i\in\mathcal{S}}$	$((N+1-i)^{-1})_{i\in\mathcal{S}}$	$(\sqrt{N+1-i})_{i\in\mathcal{S}}$
Average $\Delta^{n\text{-}opt}$	0.62%	0.23%

Table 3: Average accuracy error categorised in some of the model parameters as specified in Table 1.

wise, if one decides that the server at Q_i should adhere to the exhaustiveness probability $r_{i,n}^{dyn}$ instead during the n-th visit period at Q_i, at the start of which $z_{i,n}$ are present, the expected number of customers left behind at the start of the subsequent switch-over period equals $z_{i,n}\left(1 - (1 - \rho_i \mathbb{1}_{\{i\in\mathcal{I}_{BG}\}})r_{i,n}^{dyn}\right)$. To reduce variation in the waiting times, $r_{i,n}^{dyn}$ could thus be chosen such that these two numbers are the same:

$$z_{i,n}(1 - (1 - \rho_i \mathbb{1}_{\{i\in\mathcal{I}_{BG}\}})r_{i,n}^{dyn}) = f_{i,r_i^{opt}}(i)\left(1 - \left(1 - \rho_i \mathbb{1}_{\{i\in\mathcal{I}_{BG}\}}\right)r_i^{opt}\right).$$

By rewriting this equation, and observing that $r_{i,n}^{dyn}$ cannot exceed one, we have that

$$r_{i,n}^{dyn} = \min\left\{1, (1 - \rho_i \mathbb{1}_{\{i\in\mathcal{I}_{BG}\}})^{-1}\left(1 - \frac{f_{i,r_i^{opt}}(i)}{z_{i,n}}\left(1 - (1 - \rho_i \mathbb{1}_{\{i\in\mathcal{I}_{BG}\}})\right)r_i^{opt}\right)\right\}. \tag{36}$$

Observe that this expression only depends on model parameters that pertain to Q_i, and not to other queues. Choosing the exhaustiveness probabilities dynamically in this way makes customers waiting in a longer (shorter) queue than average have a higher (lower) probability of getting served during the current visit period than in the static case. This evidently reduces the variance of the waiting times. There is, however, no guarantee that the mean waiting times $\mathbb{E}[W_i]$ for $i \in \mathcal{S}$ will not increase as a result.

5 Optimisation in wireless random-access networks

In the previous section, we have derived expressions for certain model parameters that are optimal or nearly optimal in some sense. Among them, there are expressions for the routing probabilities p_j that (nearly) optimise a weighted sum of the mean waiting times (or equivalently, mean queue lengths) under the assumption of random routing. As seen in (27) and (34), these expressions are of the form

$$p_i^{opt} = \frac{\gamma_i}{\sum_{j\in\mathcal{S}} \gamma_j}, \tag{37}$$

where the coefficients γ_i are positive and only depend on parameters pertaining to Q_i. As such, the numerator of (37) only depends on Q_i-specific values, but the denominator pertains to parameters of all queues for normalisation purposes.

In the wireless random-access network setting as considered in Section 1, an important question is what the activation probability of each node should be in order to minimise the overall mean number of packets waiting to be

transmitted and hence the overall mean delay. Although each of the nodes in the network operates autonomously, it is reasonable to assume that the nodes are cooperative and in principle strive to achieve such a common goal. The found expressions of the form in (37) in principle offer a solution to this type of problem. However, these expressions are not directly applicable to the wireless setting. Recall that the nodes operate in a distributed way; i.e., they operate concurrently on the basis of the partial information that is known to them. The information known to each node includes the value γ_i and the observed inter-transmission times so far, but not the values γ_j pertaining to other nodes. In this section, we therefore propose an algorithm that makes each node update its activation probability in such a way that these probabilities tend toward their (near-)optimal values $\frac{\gamma_i}{\sum_{j \in S} \gamma_j}$, provided that all nodes in the network follow this algorithm. The algorithm works in a distributed fashion as desired: all the nodes execute this algorithm concurrently, but autonomously based on inter-transmission times observed thus far and their value of γ_i. In Section 5.1, we describe two possible variants of the algorithm. The first variant makes the nodes choose activation probabilities that converge to (values near) the desired values p_i^{opt} over time with probability one. Although in the second variant the activation probabilities converge to their limiting values in a weaker sense, we will see that this variant is more robust to a variable population of nodes in the network or changing values of γ_i. Section 5.2 subsequently examines both variants of the algorithm in more detail and elaborates on their convergence properties. Finally, we provide numerical examples for both variants in Section 5.3.

5.1 Description of the distributed algorithm

We now propose an algorithm, which prescribes for each node which activation probability it should adopt based on the information available to that specific node. We assume that the information known to any of the N present nodes, which we index by i, includes the value γ_i and the durations of the previous inter-transmission times. First, we introduce some additional notation. We index time by n, so that $X(n)$ refers to the duration of the n-th inter-transmission time. The activation probability of node i during the n-th inter-transmission time is denoted by $p_i(n)$. As in Section 2, we refer to the back-off rates of the nodes as ν_0. Thus, the inter-transmission time $X(n)$ is exponentially distributed with rate $\nu_0 \sum_{j \in S} p_j(n)$, where the set $S = \{1, \ldots, N\}$ represents the N nodes present in the network. Finally, for the sake of conciseness, we write $[x]_y^z$ for $\min\{\max\{x, y\}, z\}$, $(x)^-$ for $\min\{x, 0\}$ and $(x)^+$ for $\max\{x, 0\}$.

Now that the required additional notation has been introduced, we proceed to describe a distributed algorithm that makes the activation probabilities move towards their (near-)optimal values in the long run.

Algorithm 5.1. *Let α and M be positive constant coefficients, $\alpha < \gamma_i^{-1}$. Furthermore, let the i-th node ($i \in S$) have an initial activation probability of $p_i(1) = \gamma_i \theta_i(0)$, $i = 1, \ldots, N$, where $\theta_i(0)$ is assumed to be in the interval $[\alpha, \gamma_i^{-1}]$. After the n-th transmission time, it calculates*

$$\theta_i(n) = [(1 - \epsilon(n))\theta_i(n-1) + \epsilon(n)M(\nu_0 X(n) - 1)]_\alpha^{\gamma_i^{-1}}, \tag{38}$$

where the $\epsilon(n)$ are step sizes that depend on n. Subsequently, node i updates its activation probability according to

$$p_i(n+1) = \gamma_i \theta_i(n). \tag{39}$$

When each node adheres to this algorithm, the activation probabilities $p_i(n)$ of the various nodes will eventually converge in some sense to the value $\gamma_i \hat{\theta}$, with $\hat{\theta}$ given by

$$\hat{\theta} = \frac{-M}{2} + \sqrt{\frac{M^2}{4} + \frac{1}{\sum_{j \in S} \gamma_j}}, \tag{40}$$

provided that $\alpha < \hat{\theta}$.

In Section 5.2, we examine this algorithm in detail and focus on the convergence properties of this algorithm. However, we first study this algorithm to see why it forms a solution to our problem, and to explore the roles of the step sizes and the algorithm's coefficients. To this end, we observe that if the values $X(n)$ did not exhibit random noise, i.e. $X(n) = \mathbb{E}[X(n)] = \left(\sum_{j \in S} \gamma_j \nu_0 \theta_j(n-1)\right)^{-1}$, the N-dimensional difference equation in (38) would reduce to In Section 5.2, we examine this algorithm in detail and focus on the convergence properties

of this algorithm. However, we first study this algorithm to see why it forms a solution to our problem, and to explore the roles of the step sizes and the algorithm's coefficients. To this end, we observe that if the values $X(n)$ did not exhibit random noise, i.e. $X(n) = \mathbb{E}[X(n)] = \left(\sum_{j \in \mathcal{S}} \gamma_j \nu_0 \theta_j(n-1) \right)^{-1}$, the N-dimensional difference equation in (38) would reduce to

$$\theta_i(n) = \left[(1 - \epsilon(n)) \theta_i(n-1) + \epsilon(n) M \left(\frac{1}{\sum_{j \in \mathcal{S}} \gamma_j \theta_j(n-1)} - 1 \right) \right]_\alpha^{\gamma_i^{-1}}, \tag{41}$$

for each $i \in \mathcal{S}$. In this N-dimensional difference equation each of the $\theta_i(n)$ evolves in exactly same way, so that the fixed point $\boldsymbol{\theta}$ of this N-dimensional difference equation must satisfy $\theta_i = \hat{\theta}$ for all i for some value $\hat{\theta}$ as a result of symmetry. Thus, the problem of finding the fixed point of the N-dimensional difference equation can be reduced to finding the solution of the one-dimensional problem

$$\hat{\theta} = \left[(1 - \epsilon(n)) \hat{\theta} + \epsilon(n) M \left(\frac{1}{\hat{\theta} \sum_{j \in \mathcal{S}} \gamma_j} - 1 \right) \right]_\alpha^{\gamma_i^{-1}},$$

which is easily seen to be given by $\hat{\theta}$ as specified in (40) when α is smaller than the expression displayed in (40). Furthermore, it is easily verified that this expression is a unique fixed point of (41) and tends to $\frac{1}{\sum_{j \in \mathcal{S}} \gamma_j}$ when $M \to \infty$. By this observation and (39), it is thus not surprising that the activation probabilities p_i of the nodes will eventually be close to their (near-)optimal values p_i^{opt}, when taking M large enough. In fact, as M tends to infinity, the expression of (40) tends to its limit from below. This is a desired property, as the sum of the activation probabilities does not exceed one in that case for a finite M.

We consider two different variants of this algorithm. The first variant uses step sizes $\epsilon(n)$ that satisfy the conditions

$$\epsilon(n) \geq 0 \text{ for all } n \geq 1, \quad \epsilon(n) \to 0 \text{ if } n \to \infty, \qquad \sum_{n=1}^\infty \epsilon(n) = \infty \quad \text{and} \quad \sum_{n=1}^\infty (\epsilon(n))^2 = \infty. \tag{42}$$

As we will see in Section 5.2, the activation probabilities converge with probability one to $\gamma_i \hat{\theta}$ when using this variant. We also study a second variant of the algorithm, namely the one which assumes that the step sizes $\epsilon(n) = \epsilon$ are constant over time. We will see in Section 5.2 that although stationary iterates of (38) will then still be contained in a small area around $\gamma_i \hat{\theta}$, this variant does not exhibit convergence with probability one since the step sizes do not decrease over time. Due to the constant step sizes however, the second variant is more suitable for use in networks with a variable population of nodes or changing values of γ_i; i.e. for settings for which the (near-)optimal activation probability p_i^{opt} is of a variable nature. In the first variant, convergence of the activation probabilities to new values of p_i^{opt} would after some point become unacceptably slow, due to the decreasing step sizes. The second variant does not have this problem.

When deploying the algorithm, it is important to choose the coefficients of the algorithm well. In particular, the lower bound α needs to be chosen positive so as to keep the algorithm from producing negative control parameters $\theta_i(n)$, but smaller than $\hat{\theta}$ so as to preserve the desired limiting values. Due to the bounds α and γ_i^{-1}, the control parameters $\boldsymbol{\theta}(n)$ take values in the hypercube $\mathcal{H} = \{ \boldsymbol{\theta} : \alpha \leq \theta_i \leq \gamma_i^{-1} \}$. As for the coefficient M, we have already seen that the higher the value of M, the closer the limiting value $\gamma_i \hat{\theta}$ is to the desired value p_i^{opt}. However, a large M also implies that the iterates of (38) are prone to a significant amount of random noise. To prevent the latter, the step sizes should be chosen such that $\epsilon(n)M$, or in case of the second variant, ϵM is small enough. The step size should however not be taken too small, as this will result in a low convergence speed.

5.2 Convergence properties

Now that Algorithm 5.1 has been introduced properly, we study this algorithm in detail and establish the convergence properties of the two variants as considered in the previous section. Although both variants exhibit a different form of convergence, we will see that the arguments needed to establish these convergence properties are similar. In particular, results from [13] imply that the limiting result in both variants coincides with the unique asymptotically stable point of the same N-dimensional ordinary differential equation (ODE), which can informally be thought of as the continuous-time equivalent of (38). To be more specific, we can rewrite (38) in the

17

form

$$\theta_i(n) = \theta_i(n-1) + \epsilon(n)Y_i(n) + \epsilon(n)Z_i(n) \tag{43}$$

for each $i \in \mathcal{S}$, where the variables $Y_i(n)$ and $Z_i(n)$ are given by $Y_i(n) = M(\nu_0 X(n) - 1) - \theta_i(n-1)$ and $Z_i(n) = \left(\frac{\alpha - \theta_i(n-1)}{\epsilon(n)} - Y_i(n)\right)^+ + \left(\frac{\gamma_i^{-1} - \theta_i(n-1)}{\epsilon(n)} - Y_i(n)\right)^-$, respectively. Thus, $Z_i(n)$ is the number with the smallest absolute value needed to keep $\theta_i(n+1)$ between α and γ_i^{-1}. The N-dimensional ODE referred to in [13] can now be expressed as

$$\dot{\theta}_i = g_i(\boldsymbol{\theta}) + z_i(\boldsymbol{\theta}) \tag{44}$$

for all $i \in \mathcal{S}$, where $\boldsymbol{\theta}$ is a function of a continuous-time parameter t rather than the discrete-time parameter n as before. Here, $g_i(\boldsymbol{\theta}) = \mathbb{E}[Y_i(n) \mid \boldsymbol{\theta}(n-1) = \boldsymbol{\theta}] = M((\sum_{j \in \mathcal{S}} \gamma_j \theta_j)^{-1} - 1) - \theta_i$. Furthermore, $z_i(\boldsymbol{\theta})$ is again a number with the smallest absolute value needed to keep $\boldsymbol{\theta}$ from leaving the hypercube $\mathcal{H}$. Thus, $z_i(\boldsymbol{\theta})$ becomes positive (negative) whenever θ_i takes the boundary value of α (γ_i^{-1}) and needs to be 'pushed' back for $\boldsymbol{\theta}$ to stay in $\mathcal{H}$. More specifically, we have that $z_i(\boldsymbol{\theta}) = -g_i(\boldsymbol{\theta})\mathbb{1}_{\{(\theta_i = \alpha \wedge g_i(\boldsymbol{\theta}) < 0) \vee (\theta_i = \gamma_i^{-1} \wedge g_i(\boldsymbol{\theta}) > 0)\}}$.

To find the asymptotically stable points of (44), we first look for fixed points of (44); i.e. points for which $\dot{\theta}_i = 0$ for all $i \in \mathcal{S}$. To this end, note that $g_i(\boldsymbol{\theta})$ has a positive root $\boldsymbol{\theta}^*$ with elements given by $\theta_i^* = \hat{\theta}$ for all $i \in \mathcal{S}$ provided that $\alpha < \hat{\theta}$ and is as such contained in the interior of $\mathcal{H}$. Since $g_i(\boldsymbol{\theta})$ is decreasing in θ_i, $g_i(\boldsymbol{\theta})$ is positive when θ_i equals α, as this is a lower boundary of $\mathcal{H}$. Similarly, $g_i(\boldsymbol{\theta})$ is negative when θ_i equals the upper boundary γ_i^{-1}. As a result, we have that $z_i(\boldsymbol{\theta}) = 0$ for any $i \in \mathcal{S}$ and $\boldsymbol{\theta} \in \mathcal{H}$. Thus, any fixed point $\boldsymbol{\theta}^*$ of (44) satisfies $g_i(\boldsymbol{\theta}^*) = 0$ for all $i \in \mathcal{S}$. As such, $\boldsymbol{\theta}^* = (\theta_1^*, \ldots, \theta_N^*) = (\hat{\theta}, \ldots, \hat{\theta})$ is a fixed point of (44). This fixed point is moreover unique, as $g_i(\boldsymbol{\theta})$ only has one positive root due to its decreasingness in the non-negative orthant.

In order to apply the results from [13], it remains to be shown that the unique fixed point $\boldsymbol{\theta}^*$ is asymptotically stable. To this end, we consider the Lyapunov function

$$L(\boldsymbol{\theta}) = (\max_{i \in \mathcal{S}}\{\theta_i\} - \min_{j \in \mathcal{S}}\{\theta_j\})^2 + \left(\sum_{k \in \mathcal{S}} \gamma_k(\theta_k - \theta_k^*)\right)^2. \tag{45}$$

It is evident that $L(\boldsymbol{\theta}^*) = 0$ and $L(\boldsymbol{\theta}) > 0$ for all $\boldsymbol{\theta} \in \mathcal{H}\backslash\{\boldsymbol{\theta}^*\}$. Furthermore, we see that the time-derivative of $L(\boldsymbol{\theta})$ satisfies

$$\dot{L}(\boldsymbol{\theta}) = 2(\dot{\theta}_{\arg\max_{i \in \mathcal{S}}\{\theta_i\}} - \dot{\theta}_{\arg\min_{j \in \mathcal{S}}\{\theta_j\}})(\max_{i \in \mathcal{S}}\{\theta_i\} - \min_{j \in \mathcal{S}}\{\theta_j\}) + 2\sum_{k \in \mathcal{S}} \dot{\theta}_k(\theta_k - \theta_k^*)$$

$$= -2(\max_{i \in \mathcal{S}}\{\theta_i\} - \min_{j \in \mathcal{S}}\{\theta_j\})^2 + 2\sum_{k \in \mathcal{S}} \gamma_k g_k(\boldsymbol{\theta})\sum_{l \in \mathcal{S}} \gamma_l(\theta_l - \theta_l^*), \tag{46}$$

where the second inequality follows from (44) and the fact that $z_i(\boldsymbol{\theta})$ equals zero for all $\boldsymbol{\theta} \in \mathcal{H}$. Note that the first term of (46) is negative, except when $\theta_i = \theta_j$ for all $i, j \in \mathcal{S}$, in which case the first term equals zero. As for the second term, observe that any $\boldsymbol{\theta} \in \mathcal{H}$ that satisfies $\sum_{l \in \mathcal{S}} \gamma_l \theta_l = \sum_{l \in \mathcal{S}} \gamma_l \theta_l^*$ is a root of $\sum_{k \in \mathcal{S}} \gamma_k g_k(\boldsymbol{\theta})$. As $\sum_{k \in \mathcal{S}} \gamma_k g_k(\boldsymbol{\theta})$ is decreasing in $\sum_{l \in \mathcal{S}} \gamma_l \theta_l$, it thus follows that the second term is negative, except when $\sum_{l \in \mathcal{S}} \gamma_l \theta_l = \sum_{l \in \mathcal{S}} \gamma_l \theta_l^*$. Combining these observations, we have that $\dot{L}(\boldsymbol{\theta}^*) = 0$ and $\dot{L}(\boldsymbol{\theta}) < 0$ for all $\boldsymbol{\theta} \in \mathcal{H}\backslash\{\boldsymbol{\theta}^*\}$. By standard theory on Lyapunov functions (see e.g. [11]), and the properties of the particular Lyapunov function $L(\boldsymbol{\theta})$ as established above, we conclude that the fixed point $\boldsymbol{\theta}^*$ is asymptotically stable.

Now that we have identified the unique asymptotically stable point of (44), we can apply the results from [13] to obtain the convergence properties of both variants of the algorithm. As proved in [13, Theorem 5.2.1], the iterates of (38), or equivalently, (43), converge under very broad assumptions (which are satisfied here) with probability one to the asymptotically stable point $\boldsymbol{\theta}^* = (\hat{\theta}, \ldots, \hat{\theta})$ of (44), in case the step sizes $\epsilon(n)$ decay over time subject to the conditions given in (42). Thus, in the first variant of the algorithm, the activation probabilities $p_i(n)$ converge with probability one to the value $\gamma_i \theta_i^*$ for all $i \in \mathcal{S}$, which we have already seen to be close to the desired value p_i^{opt}.

As for the second variant, it is stated in [13, Theorem 8.2.1] that for similar algorithms with constant step sizes, the iterates of (38) will not converge with probability one anymore, but will still in the long run fluctuate around the asymptotically stable point $\boldsymbol{\theta}^*$ of the ODE (44). More specifically, the theorem implies there always exists an $\epsilon > 0$ small enough so that the probability that a stationary value $\theta_i(n)$ is contained in any arbitrarily small area around this fixed point exceeds any given positive value smaller than one. Thus, the $\theta_i(n)$ converge to the same limiting values as in the first variant, but in a weaker sense. However, as discussed in Section 5.1, the second variant can handle changing values of $\sum_{j \in \mathcal{S}} \gamma_j$ better due to the constant step size.
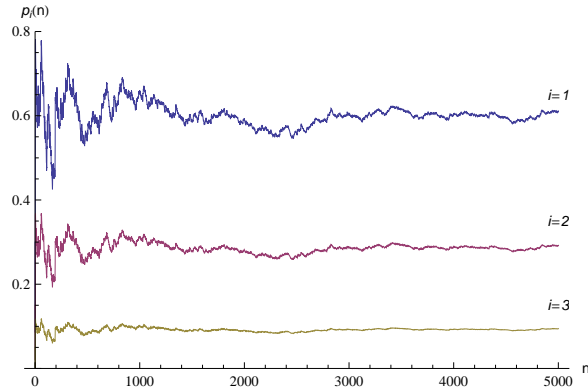
Figure 1: Evolution of the activation probabilities in the first example.

5.3 Numerical examples

We end the study of the distributed algorithm with two numerical examples illustrating the discussed variants of Algorithm 5.1. First, we consider three interfering nodes in a network with $\gamma_1 = \theta_1(0) = 0.3$, $\gamma_2 = \theta_2(0) = 0.15$ and $\gamma_3 = \theta_3(0) = 0.05$. To control their activation probabilities, the nodes adopt the first variant of the algorithm. The coefficients of the algorithm are given by $\alpha = 1/1000$ and $M = 100$, while the step sizes $\epsilon(n)$ are chosen according to $\epsilon(n) = \frac{1}{n + \log(n) + 10M}$ and as such satisfy the conditions in (42). Figure 1 plots the activation probabilities as generated by the three nodes adhering to the algorithm with these settings. As expected, the three back-off rates converge to (values close to) their optimal values $p_1^{opt} = 0.6$, $p_2^{opt} = 0.3$ and $p_3^{opt} = 0.1$. Furthermore, the back-off rates become less volatile as time progresses, due to the decaying step sizes.

To illustrate the second variant of the distributed algorithm, we again consider three nodes, this time with $\gamma_1 = \theta_1(0) = 0.1$, $\gamma_2 = \theta_2(0) = 0.05$ and $\gamma_3 = \theta_3(0) = 0.075$, respectively. To show that the second variant allows for changing settings in the network, we assume that after 5000 packet inter-transmissions, the third node disappears from the network. Furthermore, after 10000 packet inter-transmissions a third node reappears, this time with parameter $\gamma_3 = \theta_3(0) = 0.1$. To control the activation probabilities of the various nodes over time, we adopt Algorithm 5.1 with coefficients $\alpha = 1/1000$, $M = 35$ and constant step sizes $\epsilon(n) = \epsilon = 1/2000$.

Figure 2 plots the resulting evolution of the various activation probabilities $p_i(n)$. Initially, the activation probabilities fluctuate around values that are slightly smaller than the optimal activation probabilities $p_1^{opt} = 4/9$, $p_2^{opt} = 2/9$ and $p_3^{opt} = 1/3$. Thus, the sum of the activation probabilities rarely exceeds one, as desired. When the third node disappears after 5000 packet inter-transmission, the activation probabilities of the remaining nodes adapt to the new situation and correctly move towards new limiting values. When a third node reappears after the 10000-th packet inter-transmission, the activation probabilities once again adjust to the new situation. In particular, we see that the activation probabilities of the first and third node eventually coincide, since $\gamma_1 = \gamma_3$ for $n \geq 10000$.

Remark 5.1. In practice, it may happen that a node has no packets to transmit. In such a case, the 'empty' node will deactivate immediately after activation, due to its lack of packets to be transmitted. However, other nodes might not be able to detect such an activation followed by an immediate deactivation. This would then result in the (de)activating node updating its control parameter $\theta_i(n)$, while the other nodes do not update their control parameter. This may cause problems, as the algorithm requires all of the nodes to update their control parameter simultaneously. To avoid these problems, one may adapt the algorithm such that a node now sets its own activation probability equal to zero when it has no packets to transmit, and in accordance with the original Algorithm 5.1 otherwise. The nodes however continue to update their control parameter $\theta_i(n)$ according to (38) at all times, irrespective of the number of packets waiting to be transmitted. Simply put, an 'empty' node no longer activates if it has no packets to transmit. This minor adjustment has the advantage that, when certain nodes remain empty for a larger amount of time, the activation probabilities of the other nodes will adapt to this situation accordingly.

References

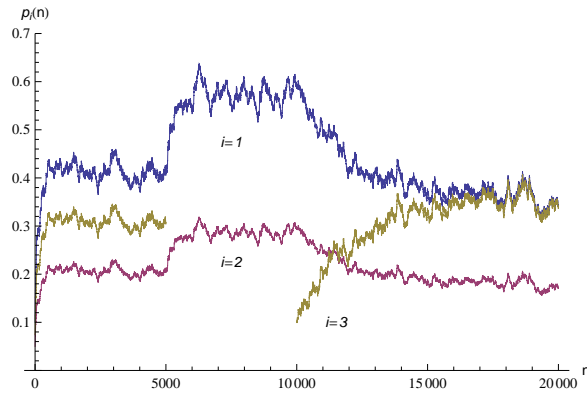[1] K.B. Athreya and P.E. Ney. *Branching Processes*. Springer, New York, 1972.

Figure 2: Evolution of the activation probabilities in the second example.

[2] M.A.A. Boon, R.D. Van der Mei, and E.M.M. Winands. Applications of polling systems. *Surveys in Operations Research and Management Science*, 16:67–82, 2011.

[3] O. J. Boxma. Workloads and waiting times in single-server systems with multiple customer classes. *Queueing Systems*, 5:185–214, 1989.

[4] O.J. Boxma and W.P. Groenendijk. Pseudo-conservation laws in cyclic-service systems. *Journal of Applied Probability*, 24:949–964, 1987.

[5] O.J. Boxma, O. Kella, and K.M. Kosiński. Queue lengths and workloads in polling systems. *Operations Research Letters*, 39:401–405, 2011.

[6] O.J. Boxma, H. Levy, and J. Weststrate. Optimization of polling systems. In P.J.B. King, I. Mitrani, and R.J. Pooley, editors, *Proceedings of the 14th IFIP WG 7.3 International Symposium on Computer Performance Modelling, Measurement and Evaluation*, pages 349–361. North-Holland, 1990.

[7] O.J. Boxma and J. Weststrate. Waiting times in polling systems with Markovian server routing. In G. Stiege and J.S. Lie, editors, *Messung, Modellierung und Bewertung von Rechensystemen und Netzen*, pages 89–104. Springer Berlin, 1989.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

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

[10] J. Keilson and L.D. Servi. The distributional form of Little's law and the Fuhrmann-Cooper decomposition. *Operations Research Letters*, 9:239–247, 1990.

[11] H.K. Khalil. *Nonlinear systems*. Prentice-Hall, Englewood Cliffs, New Jersey, third edition, 2002.

[12] L. Kleinrock and H. Levy. The analysis of random polling systems. *Operations Research*, 36:716–732, 1988.

[13] H.J. Kushner and G.G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, second edition, 2003.

[14] H. Levy. Analysis of cyclic polling systems with binomial gated service. In T. Hasegawa, H. Takagi, and Y. Takahashi, editors, *Performance of Distributed and Parallel Systems*, pages 127–139. North-Holland, Amsterdam, 1989.

[15] H. Levy and M. Sidi. Polling systems: application, modeling and optimization. *IEEE Transactions on Communications*, 38:1750–1760, 1990.

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

[17] M.M. Srinivasan. Nondeterministic polling systems. *Management Science*, 37:667–681, 1991.

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

[19] V.M. Vishnevskii and O.M. Semenova. Mathematical models to study the polling systems. *Automation and Remote Control*, 67:173–220, 2006.

[20] J.A. Weststrate. *Analysis and optimization of polling models*. PhD thesis, Katholieke Universiteit Brabant, 1992.