ON OPTIMAL POLLING POLICIES*

Zhen LIU[1], Philippe NAIN[2], Don TOWSLEY[3]

[1,2]INRIA, Centre de Sophia Antipolis
06565 Valbonne Cedex, France

[3]Department of Computer and Information Science
University of Massachusetts, Amherst, MA 01003, USA

December 16, 1991

Abstract

In a single server polling system, the server visits the queues according to a routing policy and while at a queue serves some or all of the customers there according to a service policy. A polling (or scheduling) policy is a sequence of decisions on whether to serve a customer, idle the server, or switch the server to another queue. The goal of this paper is to find polling policies that stochastically minimize the unfinished work and the number of customers in the system at all time. This optimization problem is decomposed into three subproblems: determine the optimal action (i.e., serve, switch, idle) when the server is at a nonempty queue; determine the optimal action (i.e., switch, idle) when the server empties a queue; determine the optimal routing (i.e., choice of the queue) when the server empties a queue and decides to switch.

Under fairly general assumptions, we show for the first subproblem that optimal policies are greedy and exhaustive, i.e., the server should neither idle nor switch when it is at a nonempty queue. For the second subproblem, we prove that in symmetric polling systems patient policies are optimal, i.e., the server should stay idling at the last visited queue whenever the system is empty. When the system is slotted, we further prove that non-idling and impatient policies are optimal. For the third subproblem, we establish that in symmetric polling systems optimal policies belong to the class of Stochastically Largest Queue (SLQ) policies. A SLQ policy is one that never routes the server to a queue known to have a queue length that is stochastically smaller than that of another queue. This result implies, in particular, that the policy that routes the server to the queue with the largest queue length is optimal when all queue lengths are known and that the cyclic routing policy is optimal in the case that the only information available is the previous decisions.

*This work was supported in part by NSF under contract ASC-8802764

1 Introduction

In this paper, we consider the problem of scheduling a server in a polling system. We model this system as N queues attended by a single server. The server visits the queues according to some rule. While at a queue, the server serves some or all of the customers again according to some rule. A polling (or scheduling) policy consists of a series of decisions on whether to serve a customer, to idle the server, or to switch to another queue. These decisions are made based on partial knowledge of the state of the queue (i.e., queue occupancies, past arrival patterns) and on past scheduling decisions.

Polling systems are frequently used to model computer and communication networks (cf. Takagi [7]). The performance analysis of these systems has been the subject of a large number of papers (more than 450 references are included in Takagi's survey paper [8]). These analyses have focussed on a large number of routing policies including cyclic and probabilistic routing and on a large number of service policies such as exhaustive, gated and limited service.

However, very few papers have focused on the problem of developing optimal scheduling policies, except for the simple μc rule which minimizes some discounted cost function when there is no switch-over time. One of the first attempts to optimize polling systems with switch-over times was made by Hofri and Ross [4]. They show for a two queue model that the policy that minimizes the sum of discounted switch-over times and the holding cost is exhaustive service in a nonempty queue and of threshold type for switching from an empty queue to another. Browne and Yechiali [3] provide a semi-dynamic policy where the server chooses a visiting order of the queues at the beginning of each cycle in order to minimize the cycle time. A similar approach is presented by Browne and Yechiali [2] for queues with unit buffer and losses of customers. They show that an index rule policy minimizes the sum of holding costs and customer loss costs over a cycle. In Levy et al. [5], it is proved that the exhaustive policy dominates all other service policies, under the assumptions that the server does not wait idling at a queue and that the server switches to the next queue once a queue is emptied. In [1] Boxma et al. investigate the optimal polling table (ratios of occurence of all queues in the table, size of the table, order within the table) that minimizes the mean total workload in a periodic polling system. In particular, they show that the *golden ratio* policy for determining the exact visit order of the queues provides a good heuristic for this problem. In [6] Liu and Nain consider a particular polling system arising from the videotex system. Depending on the amount of information available to the controller, they identify optimal scheduling policies under fairly general assumptions in the case when there are no switch-over times. Last, in [9], Towsley, et al. prove that the policy that serves the queue with the largest queue length minimizes the number of customers that are lost when queues have finite and equal buffer cpacities, arrivals to each queue are governed by a class of statistically identical processes, and switching times are negligible.

In this paper, we study the case where the switch-over times are strictly positive. The aim is to find polling policies that stochastically minimize the unfinished work and the number of customers in the system at all time. This optimization problem reduces to answering the following questions:

(1) when the server is at a nonempty queue, should it serve a customer of that queue or not?

(2) when the server empties a queue, should it switch or not?

(3) when the server empties a queue and decides to switch to another queue, which queue should it switch to?

The paper is organized as follows. Notation and assumptions are introduced in Section 2. The answer to question (1) is given in Sections 3 and 4. We show, under fairly general assumptions, that optimal policies are greedy and exhaustive. In other words, when the server is at a nonempty queue, it should neither idle nor switch until this queue is empty.

Section 5 answers question (3) in the case where the polling system is symmetric. We establish that the optimal policy is a *Stochastically Largest Queue* (SLQ) policy. Here a SLQ policy is the one that never visits a queue known to have a queue length that is stochastically smaller than that of another queue. ¿From this result, we obtain that the routing policy that moves the server to the queue with the largest queue length is optimal when all queue lengths are known at all times. In the case that the only available information is the previous decisions, we show that the cyclic routing policy is optimal. We also consider other applications where the queue length information is delayed.

Last, in Sections 6 and 7 we address question (2). We prove that in symmetric polling system patient policies are optimal, i.e., the server always stays idling at the last visited queue whenever the system is empty. If in addition the system is slotted and the switch-over times are one time unit, then non-idling as well as impatient policies are shown to be optimal, i.e., the server always switches whenever it empties a queue.

2 Notation and Assumptions

All of the random variables (r.v.'s) considered in this paper are defined on a fixed probability triple $(\Omega, \mathcal{F}, P)$. Let $\mathbb{N}$ and $\mathbb{N}_+$ be the set of nonnegative integer numbers and strictly positive integer numbers, respectively. Define $\mathbb{R} := (-\infty, +\infty)$, $\mathbb{R}_+ := [0, +\infty)$ and $\mathbf{I} := \{1, 2, \ldots, N\}$.

Input sequences. For $n \geq 1$, $i \in \mathbf{I}$, let

- a_n be the arrival time of the n-th customer, $0 \leq a_1 < a_2 < \cdots$;
- u_n be the index of the queue at which the n-th customer arrives;
- $\sigma_n^i > 0$ be the amount of service required by the n-th arriving customer at queue i. We shall assume throughout this paper that $\{\sigma_n^i\}_{n=1}^{\infty}$ is a sequence of independent and identically distributed (i.i.d.) r.v.'s for all $i \in \mathbf{I}$. Further, $\{\sigma_n^1\}_{n=1}^{\infty}, \ldots, \{\sigma_n^N\}_{n=1}^{\infty}$ are assumed to be mutually independent sequences;

- $\theta_n^{i,j} > 0$ be the duration of the n-th switching period between queue i and queue j for all $j \in \mathbf{I}$, $i \neq j$.

In the sequel, the notation **A1**, **A2**, **A3**, **A4** and **A5** will be used to denote the following assumptions:

A1 The sequences of r.v.'s $\{a_n, u_n\}_{n=1}^{\infty}$, $\{\sigma_n^i,\, i \in \mathbf{I}\}_{n=1}^{\infty}$, and $\{\theta_n^{i,j},\, (i,j) \in \mathbf{I}^2, i \neq j\}_{n=1}^{\infty}$ are mutually independent;

A2 The sequences of r.v.'s $\{a_n\}_{n=1}^{\infty}$, $\{u_n\}_{n=1}^{\infty}$, $\{\sigma_n^i,\, i \in \mathbf{I}\}_{n=1}^{\infty}$, and $\{\theta_n^{i,j},\, (i,j) \in \mathbf{I}^2, i \neq j\}_{n=1}^{\infty}$ are mutually independent;

A3 The r.v.'s $\{\sigma_n^i, i \in \mathbf{I}\}_{n=1}^{\infty}$ are identically distributed;

A4 The sequence $\{\theta_n^{i,j}, (i,j) \in \mathbf{I}^2, i \neq j\}_{n=1}^{\infty}$ is an i.i.d. sequence of r.v.'s;

A5 The sequence $\{u_n\}_{n=1}^{\infty}$ is an i.i.d. sequence of r.v.'s such that $\mathrm{P}(u_n = i) = 1/N$ for all $i \in \mathbf{I}$.

The polling system fulfilling assumptions **A2**, **A3**, **A4** and **A5** is called a symmetric polling system.

Service policy. At every queue the service policy can be arbitrary as long as it is *nonpreemptive* and *work conserving* (i.e., no work can be created or destroyed). Instances of service policies are the exhaustive, gated and random service policies (see Takagi [7]). Because of our assumption that the service times at a given queue are i.i.d. r.v.'s, we shall not need to specify the order of service (e.g., FIFO, random).

Admissible polling policy. At any *decision epoch* (see below) the polling policy must decide either to *move* the server to another queue and to provide the queue index to which the server is sent, or to ask the server to *idle* at the queue at which he resides, or to ask the server to *serve* a customer at the queue at which he resides (provided it is nonempty) according to the enforced service policy at that queue.

The decision epochs are the service completion epochs, the epochs where the switching periods end (or equivalently, the instants when the server arrives at a queue), and the epochs where the idle periods end.

Define $\mathcal{A} := \{0,1,2\} \times \{1,2,\ldots,N\}$ to be the decision space.

For any polling policy π, let π_n^e denote the n-th decision epoch, and let $A_n^{\pi} := (\pi_n^a, \pi_n^q) \in \mathcal{A}$ denote the n-th decision, where

- $\pi_n^a = 0$ (resp. $\pi_n^a = 1$, $\pi_n^a = 2$) if the n-th decision is to keep the server idle (resp. serve a customer provided the queue is nonempty, move the server to another queue). We shall assume that the first decision takes place at time 0, i.e., $\pi_1^e = 0$;

- π_n^q is the index of the queue where the server is switching to if $\pi_n^a = 2$; if $\pi_n^a \in \{0,1\}$ then π_n^q indicates the current location of the server (i.e., $\pi_n^q = \pi_{n-1}^q$). Without loss of generality, we assume that the server is located at queue 1 at time 0.

To complete the definition of a polling policy, we need to specify how the decision epochs $\{\pi_n^e\}_{n=1}^\infty$ are chosen by the policy. Define

$$s_n^\pi := \sum_{m=1}^{n-1} \mathbf{1}(\pi_m^a = 1), \tag{2.1}$$

to be the number of service periods (or departures) in $[0, \pi_n^e)$ under policy π. Note that s_n^π includes the customer that is served in $[\pi_{n-1}^e, \pi_n^e)$ if $\pi_{n-1}^a = 1$. We also define σ_m^π, $\sigma_m^\pi \in \{\sigma_n^i, i \in \mathbf{I}\}_{n=1}^\infty$, to be the m-th service time delivered by the server under policy π, $m \geq 1$.

Similarly, let

$$k_n^\pi := 1 + \sum_{m=1}^{n-1} \mathbf{1}(\pi_m^a = 2) \tag{2.2}$$

be the number of queues visited by the server in the interval of time $[0, \pi_n^e]$, $n \geq 1$. Note that k_n^π includes the queue to which the server arrives at time π_n^e (i.e., queue π_{n-1}^q) if $\pi_{n-1}^a = 2$. We also define θ_m^π, $\theta_m^\pi \in \{\theta_n^{i,j}, (i,j) \in \mathbf{I}^2, i \neq j\}_{n=1}^\infty$, to be the m-th switch-over time duration under policy π.

With the above notation, the difference $\pi_{n+1}^e - \pi_n^e$ when $\pi_n^a \neq 0$ is given by

$$\pi_{n+1}^e - \pi_n^e = \begin{cases} \sigma_{s_{n+1}^\pi}^\pi, & \text{if } \pi_n^a = 1; \\ \theta_{k_n^\pi}^\pi, & \text{if } \pi_n^a = 2. \end{cases} \tag{2.3}$$

Relation (2.3) directly follow from the definition of the service time and switch-over time processes. Note that the case $\pi_n^a = 1$ follows from the assumption that the service policy at each queue is nonpreemptive.

In order to define the duration of an idle period as well as the notion of admissible policies, we introduce the *history* of a policy. For all $n \geq 1$, let

$$H_n^\pi := \left\{ (A_m^\pi)_{m=1}^{n-1}, (\pi_m^e)_{m=1}^n, \left(Q_j^\pi(x_j)\right)_{j \in \mathbf{I}}; \;\; x_i \leq F_{i,n}^\pi, i \in \mathbf{I} \right\}, \tag{2.4}$$

be the history of policy π up to time π_n^e, where $Q_j^\pi(t)$ is the number of customers at queue j under π at time t including the customer in service, if any, and $F_{i,n}^\pi$ is the last time in $[0, \pi_n^e]$ that the controller has received queue-length information on queue i. More precisely,

$$F_{i,n}^\pi := f_{k_n^\pi, n}^i \left((j_m^\pi)_{m=1}^{k_n^\pi}, (\tau_m^\pi)_{m=1}^{k_n^\pi}, (t_m^\pi)_{m=1}^{k_n^\pi - 1}, \pi_n^e \right), \tag{2.5}$$

where

- j^π_m is the index of the m-th queue that has been visited by the server, $m \geq 1$;
- τ^π_m is the time when the m-th visit of the server to a queue has started, $m \geq 1$;
- t^π_m is the time when the m-th visit of the server to a queue has ended, $m \geq 1$.

Note that k^π_n, $(\tau^\pi_m)_{m=1}^{k^\pi_n}$ and $(t^\pi_m)_{m=1}^{k^\pi_n-1}$ can be computed from $(A^\pi_m)_{m=1}^{n-1}$ and $(\pi^e_m)_{m=1}^{n}$.

In (2.5) $f^i_{l,n}$ is any mapping $\mathbb{N}^l \times \mathbb{R}_+^{2l} \to \mathbb{R}_+$ such that for all $i \in \mathbf{I}$, $n \geq 1$, $l \geq 1$, $\mathbf{v} = (v_1, \ldots, v_{3l}) \in \mathbb{N}^l \times \mathbb{R}_+^{2l}$, $v_{l+1} \leq v_{l+2} \leq \ldots v_{2l}$, $v_{2l+1} \leq v_{2l+2} \leq \ldots \leq v_{3l}$:

P1 $f^i_{l,n}(\mathbf{v})$ is nondecreasing as a function of v_m for all $m = l+1, \ldots, 3l$;

P2 $f^i_{l,n}(\mathbf{v}) \leq f^i_{l,n+1}(\mathbf{v})$;

P3 $h^i_l(\mathbf{v}) \leq f^i_{l,n}(\mathbf{v}) \leq v_{3l}$, where

$$h^i_l(\mathbf{v}) := \begin{cases} \max_{1 \leq k \leq l}\{v_{2l+k} : v_k = i\}, & i \in \{v_1, \cdots, v_l\}; \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

In **P3**, $h^i_l(\mathbf{v})$ gives the time when the last visit of the server to queue i has ended in $[0, v_{3l}]$ if the server has already visited this queue in this time interval; if the server has not visited queue i in $[0, v_{3l}]$ then $h^i_l(\mathbf{v}) = 0$.

In terms of the history (2.4) these properties have the following interpretation: **P1** and **P2** imply that at each decision epoch the controller has as least as much information as at the previous decision epoch (the controller learns more and more about the system when the time goes on). The first inequality in **P3** implies that at any decision epoch the controller knows (at least) the history of each queue up to the last visit of the server to that queue. The second inequality in **P3** ensures that the policy is nonanticipative in the sense that no information on the future is available at any decision epoch. Further, **P3** also implies that at any decision epoch the server knows the state of the queue that it is visiting. This follows from the fact that $f^i_{l,n}(\mathbf{v}) = v_{3l}$ when $h^i_l(\mathbf{v}) = v_{3l}$. In particular, this information will enable the controller not to make the decision to serve an empty queue.

The mapping $f^i_{l,n}$ can describe various types of information structures, including those with:

1. *Complete information.* $f^i_{l,n}(\mathbf{v}) = v_{3l}$, $\forall i, l, n$. In other words, the queue lengths of all of the queues are known to the controller at all times.
2. *Partial information.* $f^i_{l,n}(\mathbf{v}) = h^i_l(\mathbf{v})$, $\forall i, l, n$. This corresponds to a system in which the controller knows the queue length history of each queue only up until the last time the server visited that queue.

3. *Periodic information.* $f^i_{l,n}(\mathbf{v}) = \max\{d\lfloor v_{3l}/d\rfloor, h^i_l(\mathbf{v})\}$, $\forall i, l, n$. Here the controller is periodically updated regarding the queue length history of all queues at intervals of length d.

4. *Delayed information.* $f^i_{l,n}(\mathbf{v}) = \max\{v_{3l} - d, h^i_l(\mathbf{v})\}$, $\forall i, l, n$. Here the controller knows the queue length history of each queue up until d time units in the past, or the last time that the queue was visited, whichever provides more information.

5. *Nearest neighbor information.* The queues are organized in a logical ring. Whenever the server switches to queue i, the controller becomes aware of the queue length history of queues $i-1$ and $i+1$ as well as that of i, $f^i_{l,n}(\mathbf{v}) = \max\{h^{i-1}_l(\mathbf{v}), h^i_l(\mathbf{v}), h^{i+1}_l(\mathbf{v})\}$, $\forall i, l, n$, where addition and subtraction on the queue index is modulo N.

We now come back to the duration of an idle period (i.e., value of $\pi^e_{n+1} - \pi^e_n$ when $\pi^a_n = 0$). We shall not give a formal definition of the process that generates the idle period duration since we shall allow for rather general idle periods, including interruptible idle periods. Roughly speaking, the duration of the idle period $[\pi^e_n, \pi^e_{n+1})$ is a r.v. that may depend on the history (H^π_n, A^π_n) as well as on the information that the controller may receive while the server is idling (customer arrivals, typically). However, the amount of information that the controller receives in $[\pi^e_n, \pi^e_{n+1})$ must belong to the history h^π_{n+1} (i.e., the information collected by the controller cannot be lost).

Let $\mathbb{H}^\pi_n$ be the set of all histories up to time π^e_n.

We define a *randomized* polling policy π to be a sequence $\{\pi_n\}_{n=1}^\infty$ of conditional probability measures on $\mathcal{A}$ given H^π_n satisfying the constraints $\pi_n(\mathcal{A} \mid H^\pi_n) = 1$ for all $H^\pi_n \in \mathbb{H}^\pi_n$, $n \geq 1$. A *deterministic* polling policy π is a sequence of measurable functions $\{\pi_n\}_{n=1}^\infty$ from $\mathbb{H}^\pi_n$ into $\mathcal{A}$. Any polling policy that is either randomized or deterministic is called an *admissible* policy.

Let Π be the set of all admissible policies. A policy $\pi \in \Pi$ is said to be:

- *non-idling* if the server never idles when the system is nonempty. Let $\Upsilon \subset \Pi$ be the set of all such policies;
- *greedy* if the server never idles at a nonempty queue. Let $\Gamma \subset \Pi$ be the set of all such policies;
- *exhaustive* if the server never leaves a nonempty queue. Let $\Xi \subset \Pi$ be the set of all such policies;
- *patient* if the server stays at the last visited queue when the system is empty. Let $\Psi \subset \Pi$ be the set of all such policies.
- *impatient* if the server leaves a queue as soon as it is empty. Let $\Delta \subset \Pi$ be the set of all such policies.

Performance metrics. The performance metrics to be considered in this paper are:

- $U^\pi(t)$, the total unfinished work in the system at time t under $\pi \in \Pi$;
- $Q^\pi(t) := \sum_{i\in \mathbf{I}} Q_i^\pi(t)$, the total number of customers in the system at time t under $\pi \in \Pi$.

We shall say in the sequel that a real-valued r.v. X is stochastically smaller than a real-valued r.v. if $\mathrm{P}(X \leq x) \geq \mathrm{P}(Y \leq x)$ for all $x \in \mathbb{R}$. In that case the notation $X \leq_{st} Y$ will be used.

3 Optimality of Greedy Policies

We first compare the decisions between "serving" and "idling" when the server is at a nonempty queue. Under appropriate assumptions, we show that any policy in Π can be improved by a greedy policy in the sense of stochastic minimization of the unfinished work and the number of customers in the system at all time. In other words, in order to minimize these criteria, the server cannot idle at a nonempty queue.

Proposition 3.1 *Assume that* **A1** *holds. Then, for any policy $\pi \in \Pi$ there exists a policy $\xi \in \Gamma$ such that*

$$U^\xi(t) \leq_{st} U^\pi(t), \tag{3.1}$$

for all $t \geq 0$.

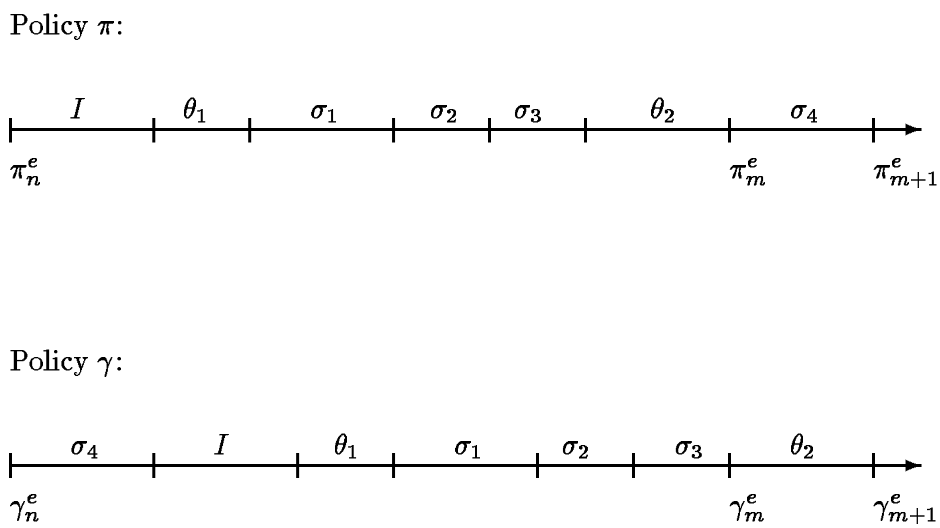
Proof. Let us consider a realization I of the input sequence $\{a_n, u_n, \sigma_n^i, \theta_n^{j,k}, (i,j,k) \in \mathbf{I}^3, j \neq k\}_{n=1}^\infty$. Let π be an arbitrary policy in $\pi \in \Pi - \Gamma$ (if $\pi \in \Gamma$ then take $\xi = \pi$) and let π run on the input sequence I.

On the sample path $\mathcal{I}$ generated by letting π run on the input sequence I, let $n \geq 1$ be the smallest integer such that $\pi_n^a = 0$ and $Q^\pi_{\pi_n^q}(\pi_n^e) > 0$. Let m be the smallest integer such that $n < m$, $\pi_m^a = 1$, and $\pi_m^q = \pi_n^q$. Note that if $m = \infty$ then the server does not serve anymore customers from queue π_n^q in $[\pi_n^e, \infty)$. By convention, we shall assume in the sequel that $\pi_{m+1}^e = \infty$ whenever $m = \infty$.

¿From $\mathcal{I}$ we construct a new policy γ as follows (cf. Figure 1):

- γ follows π in $[0, \pi_n^e)$ and γ follows π in $[\pi_{m+1}^e, \infty)$ if $m < \infty$, that is
  - $\star$ $\gamma_j^a = \pi_j^a$ and $\gamma_j^q = \pi_j^q$ for $j = 1, \ldots, n-1$, $\gamma_j^e = \pi_j^e$ for $j = 1, 2, \ldots, n$;
  - $\star$ $\gamma_j^a = \pi_j^a$, $\gamma_j^q = \pi_j^q$ and $\gamma_j^e = \pi_j^e$ for $j \geq m+1$ if $m < \infty$;
- $\gamma_n^a = 1$, $\gamma_n^q = \pi_n^q$. If $m < \infty$ then $\gamma_{n+1}^e - \gamma_n^e = \sigma^\pi_{s_m^\pi + 1}$; otherwise $\gamma_{n+1}^e - \gamma_n^e = \sigma_{v+1}^{\pi_n^q}$, where v is the number of customers served at queue π_n^q in $[0, \pi_n^e]$;
- For $j = n+1, \ldots, m$, $\gamma_j^a = \pi_{j-1}^a$, $\gamma_j^q = \pi_{j-1}^q$, and

- ⋆ $\gamma^e_{j+1} - \gamma^e_j = \pi^e_j - \pi^e_{j-1}$ if $\gamma^a_j \in \{0, 2\}$;
- ⋆ $\gamma^e_{j+1} - \gamma^e_j = \sigma^\pi_{s^\pi_{j-1}+1}$ if $\gamma^a_j = 1$.



I: Idle period; θ_1, θ_2: Switching periods; $\sigma_i := \sigma^\pi_{s^\pi_n+i}$, $i = 1, \ldots, 4$: Service periods.

Figure 1: An example of the behavior of π and γ in $[\pi^e_n, \pi^e_{m+1})$.

In other words, γ behaves like π except that during $[\pi^e_n, \pi^e_{m+1})$ γ first serves a customer and then does what π has been doing in $[\pi^e_n, \pi^e_m)$ if $m < \infty$ (resp. in $[\pi^e_n, \infty)$ if $m = \infty$). (Note that $\gamma^e_{m+1} = \pi^e_{m+1}$ if $m < \infty$.)

Let us first show that γ is admissible. This is true in $[0, \pi^e_n)$ since both policies π and γ are identical in this time interval. In $[\pi^e_n, \pi^e_{m+1})$ this result is a consequence of the properties **P1** and **P2**. Indeed, the above construction implies that for all $l = n, n+1, \ldots, m-1$, $i \in \mathbf{I}$,

$$\begin{aligned} & f^i_{k^\gamma_{l+1}, l+1}\left((j^\gamma_m)_{m=1}^{k^\gamma_{l+1}}, (\tau^\gamma_m)_{m=1}^{k^\gamma_{l+1}}, (t^\gamma_m)_{m=1}^{k^\gamma_{l+1}-1}, \gamma^e_{l+1}\right) \\ \geq\ & f^i_{k^\gamma_{l+1}, l}\left((j^\gamma_m)_{m=1}^{k^\gamma_{l+1}}, (\tau^\gamma_m)_{m=1}^{k^\gamma_{l+1}}, (t^\gamma_m)_{m=1}^{k^\gamma_{l+1}-1}, \gamma^e_{l+1}\right), \qquad (3.2) \\ \geq\ & f^i_{k^\pi_l, l}\left((j^\pi_m)_{m=1}^{k^\pi_l}, (\tau^\pi_m)_{m=1}^{k^\pi_l}, (t^\pi_m)_{m=1}^{k^\pi_l-1}, \pi^e_l\right), \qquad (3.3) \end{aligned}$$

The inequality (3.2) follows from **P2**. The inequality (3.3) follows from **P1** together with the obvious relations $k^\gamma_{l+1} = k^\pi_l$, $\tau^\gamma_l \geq \tau^\pi_l$, $t^\gamma_l \geq t^\pi_l$ and $\gamma^e_{l+1} \geq \pi^e_l$ for all $l = n, n+1, \ldots, m-1$. We may therefore deduce from (2.4) and (3.3) that at time γ^e_{l+1} the policy γ has always as much information as π at time π^e_l, for all $l = n, n+1, \ldots, m-1$. Further, at time γ^e_{l+1}, the policy γ is able to retrieve the history of policy π at time π^e_l (i.e., H^π_l) from its set of information, for all $l = n, n+1, \ldots, m-1$. This explains why γ is admissible in $[\pi^e_n, \pi^e_{m+1})$.

To show that the policy γ is admissible in $[\pi^e_{m+1}, \infty)$, observe that $k^\gamma_l = k^\pi_l$, $\gamma^e_l = \pi^e_l$, $\tau^\gamma_l = \tau^\pi_l$ and $t^\gamma_l = t^\pi_l$ for all $l \geq m+1$. This yields, for all $l \geq m+1$, $i \in \mathbf{I}$, cf. **P1**,

$$f^i_{k^\gamma_l, l}\left((j^\gamma_m)^{k^\gamma_l}_{m=1}, (\tau^\gamma_m)^{k^\gamma_l}_{m=1}, (t^\gamma_m)^{k^\gamma_l - 1}_{m=1}, \gamma^e_l\right) \geq f^i_{k^\pi_l, l}\left((j^\pi_m)^{k^\pi_l}_{m=1}, (\tau^\pi_m)^{k^\pi_l}_{m=1}, (t^\pi_m)^{k^\pi_l - 1}_{m=1}, \pi^e_l\right),$$

which shows that γ always has as much information as π in $[\pi^e_{m+1}, \infty)$, which in turn shows that γ may indeed follow π in this time interval.

Let us now prove (3.1). It is easily seen from the construction of γ that this policy belongs to Π. Further,

$$\begin{aligned} U^\gamma(t) &\leq U^\pi(t), \text{ for } t \in (\pi^e_n, \pi^e_{m+1}); \\ U^\gamma(t) &= U^\pi(t), \text{ for } t \in [0, \pi^e_n] \cup [\pi^e_{m+1}, \infty). \end{aligned}$$

Moreover, the policy γ is greedy up to γ^e_{n+1}, where $\gamma^e_{n+1} > \pi^e_n$. In the same manner, we can construct from γ a policy $\gamma' \in \Pi$ that is greedy up to γ'^e_{n+1}, where $\gamma'^e_{n+1} > \gamma^e_{n+1}$, and such that $U^{\gamma'}(t) \leq U^\pi(t)$ for all $t \geq 0$.

Iterating this procedure we finally end up with a policy $\xi \in \Pi$ that is greedy in $[0, \infty)$, and such that $U^\xi(t) \leq U^\pi(t)$ for all $t \geq 0$. The proof is then concluded by removing the conditioning on the input sequence I. ∎

Proposition 3.2 *Assume that* **A1** *and* **A3** *hold. Then, for any policy $\pi \in \Pi$ there exists a policy $\xi \in \Gamma$ such that*

$$Q^\xi(t) \leq_{st} Q^\pi(t), \tag{3.4}$$

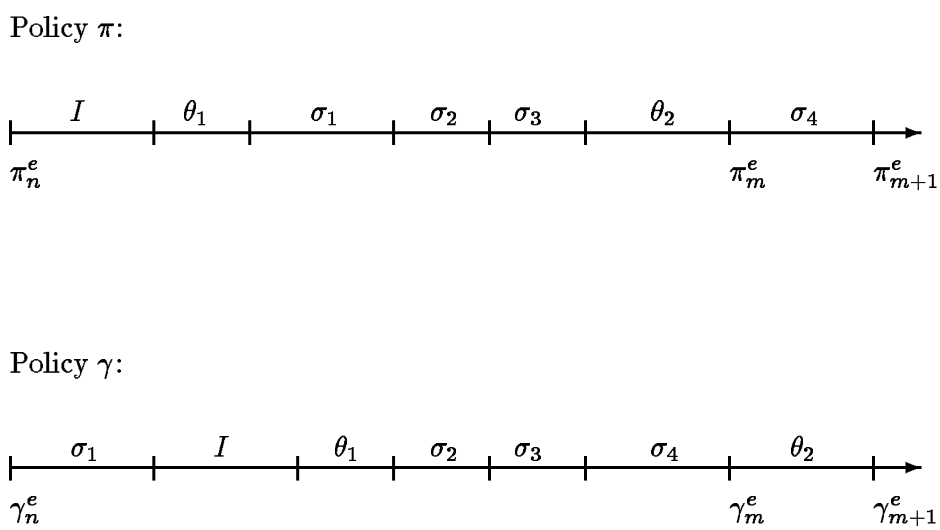
for all $t \geq 0$.

Proof. Let us consider a realization I of the input sequence $\{a_n, u_n, \sigma^i_n, \theta^{j,k}_n, (i,j,k) \in \mathbf{I}^3, j \neq k\}^\infty_{n=1}$. Let π be an arbitrary policy in $\pi \in \Pi - \Gamma$ (if $\pi \in \Gamma$ then take $\xi = \pi$) and let π run on the input sequence I.

On the sample path $\mathcal{I}$ generated by letting π run on the input sequence, let $n \geq 1$ be the smallest integer such that $\pi^a_n = 0$ and $Q^\pi_{\pi^q_n}(\pi^e_n) > 0$. Let m be the smallest integer such that $n < m$, $\pi^a_m = 1$, and $\pi^q_m = \pi^q_n$. Note that if $m = \infty$ then the server does not serve anymore customers from queue π^q_n in $[\pi^e_n, \infty)$. By convention, we shall assume in the sequel that $\pi^e_{m+1} = \infty$ whenever $m = \infty$.

¿From $\mathcal{I}$ we construct a new policy γ as follows (cf. Figure 2):

- γ follows π in $[0, \pi_n^e)$ and γ follows π in $[\pi_{m+1}^e, \infty)$ if $m < \infty$, that is
  - ⋆ $\gamma_j^a = \pi_j^a$ and $\gamma_j^q = \pi_j^q$ for $j = 1, \ldots, n-1$, $\gamma_j^e = \pi_j^e$ for $j = 1, 2, \ldots, n$;
  - ⋆ $\gamma_j^a = \pi_j^a$, $\gamma_j^q = \pi_j^q$ and $\gamma_j^e = \pi_j^e$ for $j \geq m+1$ if $m < \infty$;
- $\gamma_n^a = 1$, $\gamma_n^q = \pi_n^q$ and $\gamma_{n+1}^e - \gamma_n^e = \sigma_{s_n^\pi + 1}^\pi$;
- For $j = n+1, \ldots, m$, $\gamma_j^a = \pi_{j-1}^a$, $\gamma_j^q = \pi_{j-1}^q$ and
  - ⋆ $\gamma_{j+1}^e - \gamma_j^e = \pi_j^e - \pi_{j-1}^e$ if $\gamma_j^a \in \{0, 2\}$;
  - ⋆ $\gamma_{j+1}^e - \gamma_j^e = \sigma_{s_j^\pi + 1}^\pi$ if $\gamma_j^a = 1$.



I: Idle period; θ_1, θ_2: Switching periods; $\sigma_i := \sigma_{s_n^\pi + i}^\pi$, $i = 1, \ldots, 4$: Service periods.

Figure 2: An example of the behavior of π and γ in $[\pi_n^e, \pi_{m+1}^e)$.

In words, γ behaves exactly like π except that in the time interval $[\pi_n^e, \pi_{m+1}^e)$ γ first serves a customer (whereas π idles) and then does what π has been doing in $[\pi_n^e, \pi_m^e)$ if $m < \infty$ (resp. in $[\pi_n^e, \infty)$ if $m = \infty$) (observe that the consecutive service times delivered by both policies in $[\pi_n^e, \pi_m^e)$ are stochastically — but not pathwise — identical; this is allowed because of **A3**).

The proof that γ is admissible is identical to the proof given in Proposition 3.1.

With this construction, it is easily seen that $\gamma \in \Pi$ and that $\gamma^e_{m+1} = \pi^e_{m+1}$. Further,

$$\begin{aligned} Q^\gamma(t) &\leq Q^\pi(t), \text{ for } t \in [\gamma^e_{n+1}, \pi^e_{m+1}); \\ Q^\gamma(t) &= Q^\pi(t), \text{ for } t \in [0, \gamma^e_{n+1}) \cup [\pi^e_{m+1}, \infty). \end{aligned}$$

Moreover, the policy γ is greedy up to γ^e_{n+1}, where $\gamma^e_{n+1} > \pi^e_n$.

Iterating this procedure we finally end up with a policy $\xi \in \Gamma$ that is greedy in $[0, \infty)$, and such that $Q^\gamma(t) \leq Q^\pi(t)$ for all $t \geq 0$. The proof is then concluded by removing the conditioning on the input sequence I. ∎

4 Optimality of exhaustive policies

In view of the above results, it remains to compare the decisions between "serving" and "switching" in order to determine the optimal decision when the server is at a nonempty queue. We show that any policy in Π can be improved by a greedy and exhaustive policy. In other words, in order to stochastically minimize the unfinished work and the number of customers in the system at all time, the server can neither idle at a nonempty queue nor leave the queue before it is empty.

Proposition 4.1 *Assume that* **A1** *holds. Then, for any policy* $\pi \in \Pi$ *there exists a policy* $\xi \in \Gamma \cap \Xi$ *such that*

$$U^\xi(t) \leq_{st} U^\pi(t), \tag{4.1}$$

for all $t \geq 0$.

Proof. Owing to Proposition 3.1 we may restrict ourselves to the policies in Γ. The proof is identical to the proof of Proposition 3.1 except that n is now defined to be the first integer such that $\pi^a_n = 2$ and $Q^\pi_{\pi^q_{n-1}}(\pi^e_n) > 0$ (i.e., at time π^e_n the decision is made to move the server although the queue where it is located is not empty). The rest of the proof is analogous to that of the proof of Proposition 3.1, and it is therefore omitted. ∎

Proposition 4.2 *Assume that* **A1** *and* **A3** *hold. Then, for any policy* $\pi \in \Pi$ *there exists a policy* $\xi \in \Gamma \cap \Xi$ *such that*

$$Q^\xi(t) \leq_{st} Q^\pi(t), \tag{4.2}$$

for all $t \geq 0$.

Proof. The proof is identical to the proof of Proposition 4.1 (simply replace "Proposition 3.1" by "Proposition 3.2"), and it is therefore omitted. ∎

5 Optimality of Stochastically Largest Queue Policies

This section focuses on the choices of the queues to be visited when the server leaves a queue. Specifically, we define the class of *Stochastically Largest Queue* (SLQ) policies, Σ, and prove that for any policy in Π there exists a policy in Σ that will perform at least as well if the polling system is symmetric.

In general, it is not easy to identify the best policy within Σ. We will conclude this section with a number of interesting examples where it is possible to identify the optimal policy within Σ and, thus, within Π.

With any policy $\pi \in \Pi$, we associate the relations $\{R_n^\pi\}_{n=1}^\infty$, where $(j,k) \in R_n^\pi$ at the n-th decision epoch if one of the following two conditions holds:

i) $F_{j,n}^\pi > F_{k,n}^\pi$ and $Q_j^\pi(F_{j,n}^\pi) \leq Q_k^\pi(F_{k,n}^\pi)$;

ii) $F_{j,n}^\pi = F_{k,n}^\pi$ and $Q_j^\pi(F_{j,n}^\pi) < Q_k^\pi(F_{k,n}^\pi)$.

Definition 5.1 *A policy $\pi \in \Pi$ is a SLQ policy if, at every decision epoch such that $\pi_{n-1}^q \neq \pi_n^q$, there exists no $k \in \mathbf{I}$ such that $(\pi_n^q, k) \in R_n^\pi$.*

In the remainder of this paper, we shall assume that the polling system under consideration is *totally symmetric*, i.e., the assumptions **A2-A5** are satisfied (in the literature, a *symmetric* polling system usually refers to a system where the arrival patterns and the service time distributions are stochastically identical at all queues; however, the switching times may be different, see Takagi [7]). As a consequence of **A3**, we may now assume without loss of generality that the service times are associated with the server. Let σ_n be the n-th service time delivered by the server (observe that σ_n does not depend on the enforced polling policy). Likewise, because of assumption **A4** we may also assume that the duration of the consecutive switching periods do not depend on the routing policy. Let θ_n be the duration of the n-th switching period.

The main result in this section is the following.

Proposition 5.1 *Assume* **A2**, **A3**, **A4** *and* **A5** *hold. Then, for any policy $\pi \in \Pi$, there exists a policy $\xi \in \Sigma$ such that*

$$U^\xi(t) \quad \leq_{st} \quad U^\pi(t); \tag{5.1}$$

$$Q^\xi(t) \quad \leq_{st} \quad Q^\pi(t), \tag{5.2}$$

for all $t \geq 0$.

Before proving this assertion, we state the following lemma :

Lemma 5.1 *Let $\pi \in \Pi$ be an arbitrary policy. Fix $n \geq 1$ and $p, q \in \{1, 2, \ldots, N\}$. ¿From the sequences $\{a_l\}_{l=1}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$, $\{\theta_l\}_{l=1}^{\infty}$ and $\{u_l\}_{l=1}^{\infty}$, we generate a new sequence $\{u'_l\}_{l=1}^{\infty}$ as follows:*

$$u'_l = u_l, \quad \text{for all } l \text{ such that } a_l < F^{\pi}_{p,n}; \tag{5.3}$$

$$u'_l = u_l \mathbf{1}(u_l \neq p, u_l \neq q) + p \mathbf{1}(u_l = q) + q \mathbf{1}(u_l = p), \quad \text{for all } l \text{ such that } a_l \geq F^{\pi}_{p,n}. \tag{5.4}$$

Then, $\{u_l\}_{l=1}^{\infty}$ and $\{u'_l\}_{l=1}^{\infty}$ are identical in law and $\{u'_l\}_{l=1}^{\infty}$ is independent of $\{a_l, \sigma_l, \theta_l\}_{l=1}^{\infty}$.

The proof of Lemma 5.1 is given in Appendix A.

Proof of Proposition 5.1.

Let us consider a realization I of the input sequence $\{a_n, u_n, \sigma_n, \theta_n\}_{n=1}^{\infty}$. As a result of Propositions 3.2 and 4.2 we can restrict ourselves to greedy and exhaustive policies. Let π be an arbitrary policy in $\Gamma \cap \Xi \cap (\Pi - \Sigma)$ (if $\pi \in \Gamma \cap \Xi \cap \Sigma$ then take $\xi = \pi$) such that the SLQ rule is violated on the input sequence I.

On the sample path $\mathcal{I}$ generated by letting policy π run on the input sequence I, let $n \geq 1$ be the smallest integer such that $\pi^a_n = 2$ and such that there exists some $k \in \mathbf{I}$ with $(\pi^q_n, k) \in R^{\pi}_n$ and where $(k, k') \notin R^{\pi}_n$ for all $k' \in \mathbf{I} - \{k\}$.

Let $j := \pi^q_n$ (i.e., the server switches to queue j in the interval of time (π^e_n, π^e_{n+1})). Also define

$$K := Q^{\pi}_k(F^{\pi}_{j,n}) - Q^{\pi}_j(F^{\pi}_{j,n}). \tag{5.5}$$

Observe that $K \geq 0$. This follows from the fact that $(j, k) \in R^{\pi}_n$ as well as from the fact that the mapping $x \to Q^{\pi}_k(x)$ is nondecreasing in $[F^{\pi}_{k,n}, F^{\pi}_{j,n}]$ because of **P3**.

Let

$$M_j := \inf\{l \in \mathbb{N} \,|\, l \geq n+1,\, Q^{\pi}_j(\pi^e_l) = 0\}.$$

In words, $\pi^e_{M_j}$ is the first time in $[\pi^e_{n+1}, \infty)$ when queue j is empty (recall that the server arrives at queue j at time π^e_{n+1}). If $Q^{\pi}_j(t) > 0$ for all $t \geq \pi^e_{n+1}$, then we take $M_j = \infty$.

Let

$$M^k := \inf\{l \in \mathbb{N} \,|\, \pi^q_{M_j+l} = k \text{ and } \pi^a_{M_j+l} = 1\}. \tag{5.6}$$

In words, $\pi^e_{M_j+M^k}$ is the first time when a customer of queue k is served in $[\pi^e_{M_j}, \infty)$ under policy π. In fact, as π is greedy and exhaustive, $\pi^e_{M_j+M^k}$ is the first time when the server arrives at queue k in $[\pi^e_{M_j}, \infty)$. We assume that $M^k = \infty$ if queue k is not visited by the server in $[\pi^e_{M_j}, \infty)$ under policy π.

¿From $\mathcal{I}$ we construct a new policy γ as follows (cf. Figure 3):

- For $1 \leq m \leq n-1$, $\gamma^a_m = \pi^a_m$, $\gamma^q_m = \pi^q_m$, $\gamma^e_{m+1} - \gamma^e_m = \pi^e_{m+1} - \pi^e_m$;

- $\gamma_n^a = \pi_n^a$, $\gamma_n^q = k$, $\gamma_{n+1}^e - \gamma_n^e = \pi_{n+1}^e - \pi_n^e$;
- For $n+1 \le m \le M_j + K - 1$, $\gamma_m^a = 1$, $\gamma_m^q = k$, $\gamma_{m+1}^e - \gamma_m^e = \sigma_{s_{m+1}^\gamma}$;
- For $M_j + K \le m \le M_j + K + M^k - 1$,
    - ⋆ $\gamma_m^a = \pi_{m-K}^a$;
    - ⋆ $\gamma_m^q = \mathbf{1}(\pi_{m-K}^q \notin \{j,k\})\, \pi_{m-K}^q + \mathbf{1}(\pi_{m-K}^q = j)\, k + \mathbf{1}(\pi_{m-K}^q = k)\, j$;
    - ⋆ $\gamma_{m+1}^e - \gamma_m^e = \pi_{m+1-K}^e - \pi_{m-K}^e$ if $\gamma_m^a \in \{0,2\}$;
    - ⋆ $\gamma_{m+1}^e - \gamma_m^e = \sigma_{s_{m+1}^\gamma}$ if $\gamma_m^a = 1$;
- For $m \ge M_j + K + M^k$,
    - ⋆ $\gamma_m^a = \pi_m^a$;
    - ⋆ $\gamma_m^e = \pi_m^e$;
    - ⋆ $\gamma_m^q = \mathbf{1}(\pi_m^q \notin \{j,k\})\, \pi_m^q + \mathbf{1}(\pi_m^q = j)\, k + \mathbf{1}(\pi_m^q = k)\, j$.

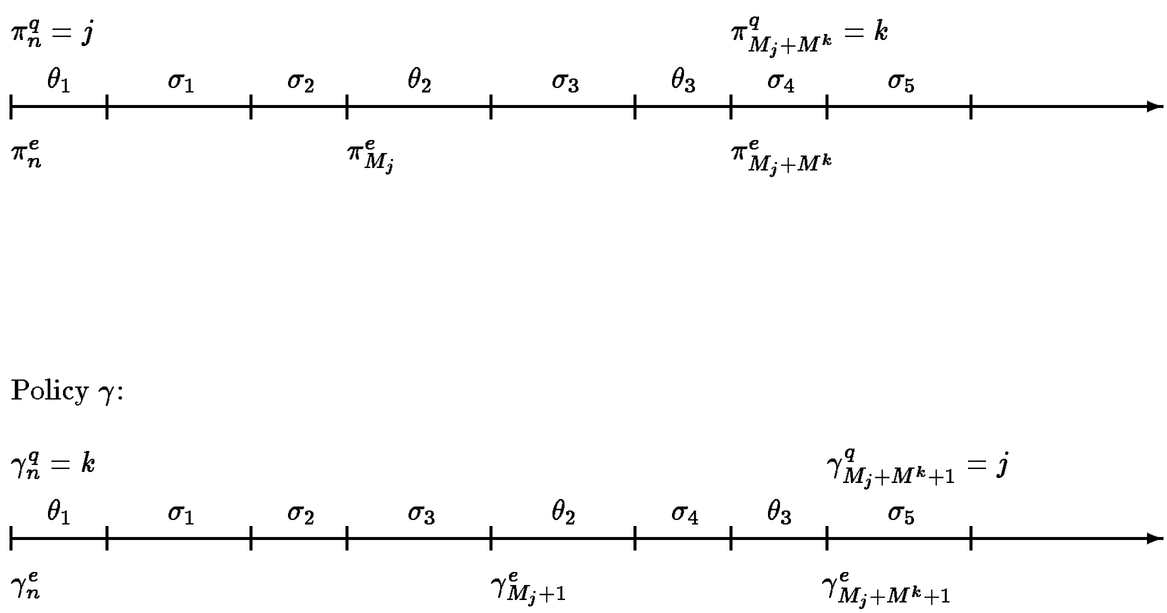


Figure 3: An example of the behavior of π and γ in $[\pi_n^e, \pi_{M_j+M^k+K+1}^e)$ with $K = 1$.

The policy γ operates on the sequences $\{a_l\}_{l=1}^\infty$, $\{u_l'\}_{n=0}^\infty$, $\{\sigma_l\}_{l=1}^\infty$ and $\{\theta_l\}_{l=1}^\infty$, where

$$u_l' = \begin{cases} u_l, & \text{if } a_l < F_{j,n}^\pi \text{ or } u_l \notin \{j,k\}; \\ k, & \text{if } a_l \ge F_{j,n}^\pi \text{ and } u_l = j; \\ j, & \text{if } a_l \ge F_{j,n}^\pi \text{ and } u_l = k, \end{cases} \tag{5.7}$$

for all $l \geq 1$. Owing to Lemma 5.1, the sequences $\{a_l\}_{l=1}^{\infty}$, $\{u'_l\}_{n=0}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$ and $\{\theta_l\}_{l=1}^{\infty}$ satisfy the assumptions **A2-A5** and have the same joint distribution as $\{a_l\}_{l=1}^{\infty}$, $\{u_l\}_{l=1}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$ and $\{\theta_l\}_{l=1}^{\infty}$.

In words, policies π and γ are identical in $[0, \pi_n^e)$; in $[\pi^e_{M_j+K+M^k}, \infty)$ they are also identical except that the server moves to queue j (resp. k) under policy γ each time he switches to queue k (resp. j) under policy π. In $[\pi_n^e, \pi^e_{M_j+K+M^K})$ policy γ first serves as many customers at queue k as policy does at queue j plus the K extra customers that were present at queue k at time $F^{\pi}_{j,n}$ (see (5.5)); this period terminates at time $\pi^e_{M_J+K+M^K}$. In $[\gamma^e_{M_j+K}, \pi^e_{M_j+K+M^K})$ γ follows what policy π has been doing in $[\pi^e_{M_j}, \pi^e_{M_j+M^k})$ with the exception that the server moves to queue k each time he moves to queue j under policy π in this time interval.

We are going to show that the following relations are satisfied:

$$
\begin{aligned}
\gamma^e_{M_j+K+l} &\geq \pi^e_{M_j+l}, \quad \text{for } l = 0, 1, \ldots, M^k; &\quad (5.8)\\
\gamma^e_{M_j+K+M^k} &= \pi^e_{M_j+K+M^k}; &\quad (5.9)\\
Q^{\gamma}_{\gamma^q_l}(\gamma^e_l) &> 0 \ , \text{ if } \gamma^a_l = 1, \ \ l \geq 1, &\quad (5.10)
\end{aligned}
$$

which will ensure that γ is admissible. Indeed, because of the structure of the information set (2.4), relation (5.8) ensures that, when γ makes its $(l+1)$-st $(l \geq 0)$ decision in $[\gamma^e_{M_j+K}, \infty)$ (which occurs at time $\gamma^e_{M_j+K+l}$), it possesses at least as much information as π when π makes the corresponding decision at time $\pi^e_{M_j+l}$. Property (5.10) ensures that γ will never make the decision to serve an empty queue.

Proof of (5.8). Let X_l be the total duration of the idling and switching periods in $[\pi^e_{M_j}, \pi^e_{M_j+l})$. Let L_l be the number of customers served by policy π in the interval of time $[\pi^e_{M_j}, \pi^e_{M_j+l})$, $l \geq 0$.

We have for $l = 0, 1, \ldots, M^k - 1$,

$$
\begin{aligned}
\pi^e_{M_j+l} &= \pi^e_{M_j} + X_l + \sum_{i=1}^{L_l} \sigma_{s^{\pi}_{M_j}+i}; &\quad (5.11)\\
\gamma^e_{M_j+K+l} &= \gamma^e_{M_j} + X_l + \sum_{i=1}^{K+L_l} \sigma_{s^{\gamma}_{M_j}+i}. &\quad (5.12)
\end{aligned}
$$

Subtracting (5.11) from (5.12) yields

$$
\gamma^e_{M_j+K+l} - \pi^e_{M_j+l} = \sum_{i=L_l+1}^{K+L_l} \sigma_{s^{\gamma}_{M_j}+i} := W_l \geq 0, \qquad (5.13)
$$

by noting that $\gamma^e_{M_j} = \pi^e_{M_j}$ by construction of γ and that $s^{\gamma}_{M_j} = s^{\pi}_{M_j}$ from the definition (2.1).

Proof of (5.9). Since $Q_k^\pi(\pi^e_{M_j+M^k}) \geq K$ by definition of M^k, K and π, we have

$$\pi^e_{M_j+M^k+K} = \pi^e_{M_j+M^k} + \sum_{m=1}^{K} \sigma_{s^\pi_{M_j}+L_{M^k}+m},$$

since π is exhaustive and greedy, which in turn implies from (5.13) that

$$\pi^e_{M_j+M^k+K} = \gamma^e_{M_j+M^k+K}, \tag{5.14}$$

which establishes (5.9).

Proof of (5.10). First, it is clear that (5.10) holds for $l \leq M_j - 1$ by definition of policy γ. That (5.10) is true for $M_j \leq l \leq M_j + K - 1$ comes from the fact that

$$Q_k^\gamma(\pi^e_n) = Q_j^\pi(\pi^e_n) + K, \tag{5.15}$$

which follows from (5.5) and from the definition of the sequence $\{u'_m\}_{m=1}^\infty$. Therefore $Q^\gamma_{\gamma^q_l}(\gamma^e_l) \geq K + M_j - l$ for all $M_j \leq l \leq M_j + K - 1$.

It remains to examine (5.10) for $l \geq M_j + K$. Let $A_s(t_1, t_2)$ denote the number of customers that arrive at queue $s \in \mathbf{I}$ in the time interval $(t_1, t_2]$ operating under π, $0 < t_1 < t_2$. Assume that $\gamma^a_{M_j+K+r} = 1$ for some $r \geq 0$. Define $i := \gamma^q_{M_j+K+r}$.

Then, it is easily seen by definition of γ that

$$Q_i^\gamma(\gamma^e_{M_j+K+r}) = \begin{cases} Q_i^\pi(\pi^e_{M_j+r}) + A_i(\pi^e_{M_j+r}, \gamma^e_{M_j+K+r}), & i \neq k; \\ Q_j^\pi(\pi^e_{M_j+r}) + A_j(\pi^e_{M_j+r}, \gamma^e_{M_j+K+r}), & i = k, \end{cases} \tag{5.16}$$

if $i \neq j$. Property (5.10) then follows for $i \neq j$ because $Q_i^\pi(\pi^e_{M_j+l}) > 0$ (indeed, since π is admissible and $\pi^a_{M_j+l} = 1$ then necessarily $Q_i^\pi(\pi^e_{M_j+l}) > 0$).

We now examine the case where $i = j$. Observe that queue j is never served in $[\gamma^e_n, \gamma^e_{M_j+K+M^K})$ under policy γ. Therefore, (5.10) holds for $\gamma^q_l = j$ and for all $l = 1, 2, \ldots, M_j + M^K + K - 1$. The proof of (5.10) is now concluded by noting that (5.16) and the definition of γ yield

$$Q_j^\gamma(t) = Q_k^\pi(t), \tag{5.17}$$

for all $t \geq \gamma^e_{M_j+K+M^k}$.

We also deduce from (5.16) that for $t \geq \gamma^e_{M_j+K+M^k}$,

$$\begin{aligned} Q_i^\gamma(t) &= Q_i^\pi(t), \text{ for } i \in \mathbf{I} - \{j,k\}; & (5.18) \\ Q_j^\pi(t) &= Q_k^\gamma(t), & (5.19) \end{aligned}$$

that is both systems are again synchronized from time $\pi^e_{M_j+M^k+K}$.

In particular, relations (5.17)-(5.19) indicate that $Q^\gamma(t) \leq Q^\pi(t)$ and $U^\gamma(t) \leq U^\pi(t)$ for $t \geq \gamma^e_{M_j+K+M^k}$.

We now establish that $Q^\gamma(t) \leq Q^\pi(t)$ and $U^\gamma(t) \leq U^\pi(t)$, for all $0 \leq t < \gamma^e_{M_j+M^k+K}$.

Define $D^\rho(t)$ to be the number of departures in $[0,t)$ under policy ρ. Recall that s^ρ_l is the number of departures in $[0,\rho^e_l)$ under ρ (cf. (2.1)). It suffices to show that

$$D^\gamma(t) \geq D^\pi(t), \text{ for all } t < \gamma^e_{M_j+M^k+K}. \tag{5.20}$$

Proof of (5.20). The inequality (5.20) is clearly true for $0 \leq t < \gamma^e_{M_j+K}$ by definition of γ in $[0,\gamma^e_{M_j+K})$. Fix $\gamma^e_{M_j+K} \leq t < \gamma^e_{M_j+M^k+K}$, and let $0 \leq l \leq M^k - 1$ be such that

$$\gamma^e_{M_j+K+l} \leq t < \gamma^e_{M_j+K+l+1}. \tag{5.21}$$

¿From (5.21) and the definition of γ in $[\gamma^e_{M_j}, \gamma^e_{M_j+K+l}]$, $D^\gamma(t)$ and s^γ_l, we see that

$$\begin{aligned} D^\gamma(t) &= s^\gamma_{M_j+K+l}, \\ &= s^\pi_{M_j+l} + K. \end{aligned} \tag{5.22}$$

On the other hand, (5.13) and (5.21) yield $t < \pi^e_{M_j+l+1} + W_{l+1}$.

Therefore,

$$\begin{aligned} D^\pi(t) &< s^\pi(\pi^e_{M_j+l+1} + W_{l+1}), &(5.23)\\ &\leq s^\pi_{M_j+l+1} + K, &(5.24)\\ &\leq s^\pi_{M_j+l} + 1 + K, &(5.25) \end{aligned}$$

and so $D^\pi(t) \leq s^\pi_{M_j+l} + K = D^\gamma(t)$ from (5.22) (note that (5.23) and (5.24) follow from (5.13); (5.25) follows from the fact that at most one customer may be served between two consecutive decision epochs).

In summary, we have shown that $Q^\gamma(t) \leq Q^\pi(t)$ and $U^\gamma(t) \leq U^\pi(t)$ for all $t \geq 0$.

Although γ may not be a greedy policy, arguments used in Propositions 3.1 and 3.2 can be used to construct a greedy policy γ' which is a SLQ policy up until at least time $\gamma^e_{M_j+K} > \pi^e_n$ such that $U^{\gamma'}(t) \leq U^\gamma(t) \leq U^\pi(t)$ and $Q^{\gamma'}(t) \leq Q^\gamma(t) \leq Q^\pi(t)$, for all $t \geq 0$.

This procedure can be iterated until we finally obtain a policy $\xi \in \Sigma$ such that $U^\xi(t) \leq U^\pi(t)$ and $Q^\xi(t) \leq Q^\pi(t)$, for all $t \geq 0$. The proof is then concluded by removing the conditioning on the input sequence. ∎

In general, for a given set of functions $\{f^i_{l,n}\}$ we are unable to identify the best SLQ policy. This is due to the fact that the relation R^π_n may not contain a single maximal element for all $n \geq 1$. However, in the first four information structures described in section 2, we do identify the best SLQ policies:

1. *Complete information.* Here the SLQ policy is one that always switches the server to the queue with the largest queue size.
2. *Partial information.* Here the optimal SLQ policy is the *cyclic policy* that begins with the first queue. Observe that this policy requires no information except whether or not the queue currently being served is empty or not.
3. *Periodic information.* Here the optimal SLQ policy switches the server to the queue with the largest queue length at the time of the last update. If all of the queues have been visited since that update, then the server is switched to the queue visited the farthest in the past.
4. *Delayed information.* The optimal SLQ policy behaves similarly to that for the system with periodic updates. The server is switched to the queue with the largest queue length that has not been visited since the last update and otherwise to the queue visited the farthest in the past.

Note that in the above four cases, the SLQ policies may not be unique since there may be several maximal elements in R^π_n for some $n \geq 1$. However, such maximal elements j and k have the property that $F^\pi_{j,n} = F^\pi_{k,n}$ and that $Q^\pi_j(F^\pi_{j,n}) = Q^\pi_k(F^\pi_{j,n})$. Therefore, all these SLQ policies are optimal.

It is not possible to characterize the optimal SLQ policy for the system in which the controller obtains information regarding a queue and its neighbors (local information). However, the previous theorem does reduce the number of decisions that must be considered by the optimal policy.

6 Optimality of Patient Policies

We are now concerned with the optimal decision to be made when the server is at an empty queue. Should the server idle at the last visited queue or should it move to another queue? Consider first the case that the system is empty at the decision epoch. We will therefore assume that the controller has complete (i.e., instantaneous) information on the state of system, i.e., for all $\pi \in \Pi$, $n \geq 1$,

$$F^\pi_{i,n} = \pi^e_n,$$

for all $i \in \mathbf{I}$ (cf. (2.5).

For symmetric polling systems, we prove that optimal polling policies are within the class of patient policies Ψ, i.e., policies that decide to idle in the last visited queue when the system is empty.

Proposition 6.1 *Assume* **A2**, **A3**, **A4** *and* **A5** *hold. Then, for any policy* $\pi \in \Pi$*, there exists a policy* $\xi \in \Gamma \cap \Xi \cap \Psi$ *such that*

$$U^{\xi}(t) \;\le_{st}\; U^{\pi}(t); \tag{6.1}$$

$$Q^{\xi}(t) \;\le_{st}\; Q^{\pi}(t), \tag{6.2}$$

for all $t \geq 0$.

Proof. Consider a realization I of the input sequence $\{a_n, u_n, \sigma_n, \theta_n\}_{n=1}^{\infty}$. As in the previous section, we restrict ourselves to greedy and exhaustive policies. Let π be an arbitrary policy in $\Gamma \cap \Xi \cap (\Pi - \Psi)$ (if $\pi \in \Gamma \cap \Xi \cap \Psi$ then take $\xi = \pi$) such that π is impatient on the input sequence I.

On the sample path $\mathcal{I}$ generated by letting policy π run on the input sequence I, let $n \geq 1$ be the smallest integer such that $\pi_n^a = 2$ and $Q_i^{\pi}(\pi_n^e) = 0$ for all $i \in \mathbf{I}$. Let $j = \pi_n^q$ and $k = \pi_{n-1}^q$. ¿From $\mathcal{I}$ we construct a new policy γ as follows:

- For $1 \leq m \leq n-1$, $\gamma_m^a = \pi_m^a$, $\gamma_m^q = \pi_m^q$, $\gamma_m^e = \pi_m^e$;
- $\gamma_n^a = 0$, $\gamma_n^q = k$, $\gamma_n^e = \pi_n^e$;
- For $m \geq n+1$;
  - ⋆ $\gamma_m^a = \pi_m^a$;
  - ⋆ $\gamma_m^e = \pi_m^e$;
  - ⋆ $\gamma_m^q = \mathbf{1}(\pi_m^q \notin \{j,k\})\, \pi_m^q + \mathbf{1}(\pi_m^q = j)\, k + \mathbf{1}(\pi_m^q = k)\, j$.

The policy γ operates on the sequences $\{a_l\}_{l=1}^{\infty}$, $\{u_l'\}_{n=0}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$ and $\{\theta_l'\}_{l=1}^{\infty}$, where for all $l \geq 1$,

$$u_l' = \begin{cases} u_l, & \text{if } a_l \leq \pi_n^e \text{ or } u_l \notin \{j,k\}; \\ k, & \text{if } a_l > \pi_n^e \text{ and } u_l = j; \\ j, & \text{if } a_l > \pi_n^e \text{ and } u_l = k, \end{cases}$$

and

$$\theta_l' = \begin{cases} \theta_l, & \text{if } l < k_n^{\pi}; \\ \theta_{l+1}, & \text{if } l \geq k_n^{\pi}. \end{cases}$$

Recall that k_n^{π} is the number of queues visited by the server during $[0, \pi_n^e]$.

In words, γ follows π in $[0, \pi_n^e)$, keeps the server idling at queue k at time π_n^e and follows π in $[\pi_{n+1}^e, \infty)$ with the exception that the server moves to queue j (resp. k) each time he moves to queue k (resp. j) under policy π.

It can be shown (cf. Lemma 5.1) that the sequences $\{a_l\}_{l=1}^{\infty}$, $\{u_l'\}_{n=0}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$ and $\{\theta_l'\}_{l=1}^{\infty}$ satisfy the assumptions **A2**-**A5** and have the same joint distribution as $\{a_l\}_{l=1}^{\infty}$, $\{u_l\}_{l=1}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$ and $\{\theta_l\}_{l=1}^{\infty}$.

It is easy to see from the definition of γ and that of the sequences $\{a_l\}_{l=1}^{\infty}$, $\{u_l'\}_{n=0}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$ and $\{\theta_l'\}_{l=1}^{\infty}$, that

$$\begin{aligned} Q_i^{\gamma}(t) &= Q_i^{\pi}(t), \quad \text{for all } 0 \leq t \leq \pi_n^e, i \in \mathbf{I}; \\ Q_i^{\gamma}(t) &= Q_i^{\pi}(t), \quad \text{for all } t \geq 0, i \in \mathbf{I} - \{j,k\}; \\ Q_j^{\gamma}(t) &= Q_k^{\pi}(t), \;\; t > \pi_n^e; \\ Q_k^{\gamma}(t) &= Q_j^{\pi}(t), \;\; t > \pi_n^e, \end{aligned}$$

which imply that γ is admissible and that $Q^{\gamma}(t) = Q^{\pi}(t)$ and $U^{\gamma}(t) = U^{\pi}(t)$ for all $t \geq 0$.

Note that γ may not be greedy as customers may arrive at queue k during $(\pi_n^e, \pi_{n+1}^e]$. Using the arguments of Propositions 3.2 and 4.2 yields a polling policy γ' which is greedy, exhaustive and patient until π_{n+1}^e and such that

$$Q^{\gamma'}(t) \leq Q^{\gamma}(t) = Q^{\pi}(t), \qquad t \geq 0;$$

$$U^{\gamma'}(t) \leq U^{\gamma}(t) = U^{\pi}(t), \qquad t \geq 0.$$

This procedure can be iterated until we finally obtain a policy $\xi \in \Psi$ such that $U^{\xi}(t) \leq U^{\pi}(t)$ and $Q^{\xi}(t) \leq Q^{\pi}(t)$, for all $t \geq 0$. The proof is then concluded by removing the conditioning on the input sequence I. ∎

7 Optimality of Non-Idling Policies

We now consider the optimal decision to be made when the server is at an empty queue while the system is not necessarily empty. Recall that a polling policy is an idling policy if the server stays idling at that empty queue. It is non-idling if the server switches to another queue (even if the latter is empty).

In general, non-idling policies are not optimal. However, when the polling system is symmetric and slotted, we show below that optimal policies are within the class of non-idling policies Υ.

Proposition 7.1 *Assume* **A2**, **A3**, **A4** *and* **A5** *hold. If $\theta_n = 1$ a.s., and $\sigma_n, \tau_n \in \mathbb{N}_+$, then, for any policy $\pi \in \Pi$, there exists a policy $\xi \in \Gamma \cap \Xi \cap \Upsilon$ such that*

$$U^{\xi}(t) \leq_{st} U^{\pi}(t); \tag{7.1}$$

$$Q^{\xi}(t) \leq_{st} Q^{\pi}(t), \tag{7.2}$$

for all $t \geq 0$.

Proof.

The proof is analogous to the proof of Proposition 5.1.

Consider a realization I of the input sequence $\{a_n, u_n, \sigma_n, \theta_n\}_{n=1}^{\infty}$. Owing to Propositions 3.2 and 4.2, we can restrict ourselves to greedy and exhaustive policies. Let π be an arbitrary policy in $\Gamma \cap \Xi \cap (\Pi - \Upsilon)$ (if $\pi \in \Gamma \cap \Xi \cap \Upsilon$ then take $\xi = \pi$) such that π is idling on the input sequence I.

Under the assumptions of the proposition, we can assume without loss of generality that the durations of idling periods are all equal to one. Therefore, if for some $n \geq 1$, $\pi_n^a = 0$, then $\pi_{n+1}^e = \pi_n^e + 1$.

On the sample path $\mathcal{I}$ generated by letting policy π run on the input sequence I, let $n \geq 1$ be the smallest integer such that $\pi_n^a = 0$ and $Q_{\pi_n^q}^{\pi}(\pi_n^e) = 0$. Let $j = \pi_n^q$ and $k \in \mathbf{I} - \{j\}$. Denote $K := Q_k^{\pi}(\pi_n^e) \geq 0 = Q_j^{\pi}(\pi_n^e)$.

As in the proof of Proposition 5.1, we define

$$
\begin{aligned}
M_j &= \inf\{l \in \mathbb{N} \,|\, l \geq n+1,\, Q_j^{\pi}(\pi_l^e) = 0\}; \\
M^k &= \inf\{l \in \mathbb{N} \,|\, \pi_{M_j+l}^a = k \text{ and } \pi_{M_j+l}^a = 1\}.
\end{aligned}
$$

In words, $\pi_{M_j}^e$ is the first time in $[\pi_{n+1}^e, \infty)$ when queue j is empty. Note that there may be an arrival at queue j at time π_{n+1}^e, in which case the server has to serve queue j until the queue is empty, as π is greedy and exhaustive. If $Q_j^{\pi}(t) > 0$ for all $t \geq \pi_{n+1}^e$, then we take $M_j = \infty$. The symbol M^k has the interpretation that $\pi_{M_j+M^k}^e$ is the first time when a customer of queue k is served in $[\pi_{M_j}^e, \infty)$ under policy π. We assume that $M^k = \infty$ if queue k is not visited by the server in $[\pi_{M_j}^e, \infty)$ under policy π.

¿From $\mathcal{I}$ we construct a new policy γ as follows:

- For $1 \leq m \leq n-1$, $\gamma_m^a = \pi_m^a$, $\gamma_m^q = \pi_m^q$, $\gamma_{m+1}^e - \gamma_m^e = \pi_{m+1}^e - \pi_m^e$;
- $\gamma_n^a = 2$, $\gamma_n^q = k$, $\gamma_{n+1}^e - \gamma_n^e = \pi_{n+1}^e - \pi_n^e = 1$;
- For $n+1 \leq m \leq M_j + K - 1$, $\gamma_m^a = 1$, $\gamma_m^q = k$, $\gamma_{m+1}^e - \gamma_m^e = \sigma_{s_{m+1}^{\gamma}}$;
- For $M_j + K \leq m \leq M_j + K + M^k - 1$,
  - ⋆ $\gamma_m^a = \pi_{m-K}^a$;
  - ⋆ $\gamma_m^q = \mathbf{1}(\pi_{m-K}^q \notin \{j,k\})\, \pi_{m-K}^q + \mathbf{1}(\pi_{m-K}^q = j)\, k + \mathbf{1}(\pi_{m-K}^q = k)\, j$;
  - ⋆ $\gamma_{m+1}^e - \gamma_m^e = \pi_{m+1-K}^e - \pi_{m-K}^e = 1$ if $\gamma_m^a \in \{0, 2\}$;
  - ⋆ $\gamma_{m+1}^e - \gamma_m^e = \sigma_{s_{m+1}^{\gamma}}$ if $\gamma_m^a = 1$;

- For $m \geq M_j + K + M^k$,
  - ⋆ $\gamma_m^a = \pi_m^a$;
  - ⋆ $\gamma_m^e = \pi_m^e$;
  - ⋆ $\gamma_m^q = \mathbf{1}(\pi_m^q \notin \{j,k\})\,\pi_m^q + \mathbf{1}(\pi_m^q = j)\,k + \mathbf{1}(\pi_m^q = k)\,j$.

The policy γ operates on the sequences $\{a_l\}_{l=1}^{\infty}$, $\{u_l'\}_{n=0}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$ and $\{\theta_l\}_{l=1}^{\infty}$, where

$$u_l' = \begin{cases} u_l, & \text{if } a_l \leq \pi_n^e \text{ or } u_l \notin \{j,k\}; \\ k, & \text{if } a_l > \pi_n^e \text{ and } u_l = j; \\ j, & \text{if } a_l > \pi_n^e \text{ and } u_l = k, \end{cases}$$

for all $l \geq 1$.

In words, policy γ behaves as policy γ in the proof of Proposition 5.1 except that the server moves to queue k at time π_n^e whereas he idles at queue j at time π_n^e under policy π.

Similar to Lemma 5.1, one can show that the sequences $\{a_l\}_{l=1}^{\infty}$, $\{u_l'\}_{n=0}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$ and $\{\theta_l\}_{l=1}^{\infty}$ satisfy the assumptions **A2**-**A5** and have the same joint distribution as $\{a_l\}_{l=1}^{\infty}$, $\{u_l\}_{l=1}^{\infty}$, $\{\sigma_l\}_{l=1}^{\infty}$ and $\{\theta_l\}_{l=1}^{\infty}$.

It is now immediate (by mimicking the proof of Proposition 5.1) that the policy γ is admissible, and that

$$Q^{\gamma}(t) \leq Q^{\pi}(t), \qquad U^{\gamma}(t) \leq U^{\pi}(t), \qquad t \geq 0.$$

Using again the same argument as those in the proof of Proposition 5.1, we obtain a policy $\xi \in \Upsilon$ such that $U^{\xi}(t) \leq U^{\pi}(t)$ and $Q^{\xi}(t) \leq Q^{\pi}(t)$, for all $t \geq 0$. ∎

Observe that the above proposition indicates only that the non-idling policies do at least as well as idling policies. In order to decide which queue to switch to, we have to use Proposition 5.1.

Combining the results of Propositions 6.1 and 7.1 we obtain that patient and impatient policies are both optimal in slotted systems:

Corollary 7.1 *Assume* **A2**, **A3**, **A4** *and* **A5** *hold. If $\theta_n = 1$ a.s., and $\sigma_n, \tau_n \in \mathbb{N} - \{0\}$, then, for any policy $\pi \in \Gamma \cap \xi \cap \Delta$, there exists a policy $\psi \in \Gamma \cap \xi \cap \Psi$, such that*

$$U^{\pi}(t) =_{st} U^{\psi}(t), \qquad Q^{\pi}(t) =_{st} Q^{\psi}(t), \qquad \forall t \geq 0.$$

A Appendix

Proof of Lemma 5.1.

Let us first show that the sequences $\{u_l\}_{l=1}^{\infty}$ and $\{u'_l\}_{l=1}^{\infty}$ are identical in law. Consider the case where $x_1 = p$, $x_2 \neq p$, $x_2 \neq q$, $n_1 < n_2$. We have, cf. (5.3), (5.4),

$$\begin{aligned}
&\mathrm{P}\,(u'_{n_1} = x_1, u'_{n_2} = x_2) \\
&= \mathrm{P}\,(u'_{n_1} = x_1, u'_{n_2} = x_2, F^{\pi}_{p,n} < a_{n_1}) + \mathrm{P}\,(u'_{n_1} = x_1, u'_{n_2} = x_2, a_{n_1} \le F^{\pi}_{p,n} < a_{n_2}) \\
&\quad + \mathrm{P}\,(u'_{n_1} = x_1, u'_{n_2} = x_2, a_{n_2} \le F^{\pi}_{p,n}), \\
&= \mathrm{P}\,(u_{n_1} = q, u_{n_2} = x_2, F^{\pi}_{p,n} < a_{n_1}) + \mathrm{P}\,(u_{n_1} = x_1, u_{n_2} = x_2, a_{n_1} \le F^{\pi}_{p,n} < a_{n_2}) \\
&\quad + \mathrm{P}\,(u_{n_1} = x_1, u_{n_2} = x_2, a_{n_2} \le F^{\pi}_{p,n}), \\
&= \mathrm{P}\,(u_{n_1} = x_1, u_{n_2} = x_2),
\end{aligned}$$

where we have used the independence assumption between $\{u_l\}_{l=1}^{\infty}$ and $\{a_l, \sigma_l, \theta_l\}_{l=1}^{\infty}$, together with **A5**. The general proof is similar and is omitted for sake of conciseness.

It remains to establish that $\{u'_l\}_{l=1}^{\infty}$ is independent of $\{a_l, \sigma_l, \theta_l\}_{l=1}^{\infty}$. Let us consider a realization I of the input sequence $\{a_l, \sigma_l, \theta_l\}_{l=1}^{\infty}$.

For $x_1 = p$, $x_2 \neq p$, $x_2 \neq q$, $n_1 < n_2$, it is readily seen that, cf. (5.3), (5.4),

$$\begin{aligned}
&\mathrm{P}\,(u'_{n_1} = x_1, u'_{n_2} = x_2 \mid I) \\
&= \mathrm{P}\,(u_{n_1} = q, u_{n_2} = x_2, F^{\pi}_{p,n} < a_{n_1} \mid I) + \mathrm{P}\,(u_{n_1} = x_1, u_{n_2} = x_2, F^{\pi}_{p,n} \ge a_{n_1} \mid I), \\
&= \mathrm{P}\,(u_{n_1} = q, u_{n_2} = x_2, \mid F^{\pi}_{p,n} < a_{n_1}, I)\mathrm{P}\,(F^{\pi}_{p,n} < a_{n_1} \mid I) \\
&\quad + \mathrm{P}\,(u_{n_1} = x_1, u_{n_2} = x_2 \mid F^{\pi}_{p,n} \ge a_{n_1}, I)\mathrm{P}\,(F^{\pi}_{p,n} \ge a_{n_1} \mid I), \\
&= \mathrm{P}\,(u_{n_1} = x_1, u_{n_2} = x_2), \qquad \text{(A.1)}
\end{aligned}$$

where we used **A5** and the fact that $\{u_l\}_{l=1}^{\infty}$ is independent of $\{a_l, \sigma_l, \theta_l\}_{l=1}^{\infty}$ (assumption **A2**). Next, using the property that $\{u_l\}_{l=1}^{\infty}$ and $\{u'_l\}_{l=1}^{\infty}$ are identical in law, we get from (A.1),

$$\mathrm{P}\,(u'_{n_1} = x_1, u'_{n_2} = x_2 \mid I) = \mathrm{P}\,(u'_{n_1} = x_1, u'_{n_2} = x_2).$$

Again, the general proof is omitted for sake of conciseness. ∎

References

[1] O. J. Boxma, H. Levy and J. A. Weststrate, "Optimization of polling systems," Proc. of *Performance '90*, P.J.B. King, I. Mitrani and R. J. Pooley, (Eds.), pp. 349-361, North-Holland, 1990.

[2] S. Browne and U. Yechiali, "Dynamic scheduling in single server multi-class service systems with unit buffers," *Naval Res. Logist.* **38**, pp. 383-396, 1991.

[3] S. Browne and U. Yechiali, "Dynamic priority rules for cyclic-type queues," *Adv. Appl. Prob.* **21**, pp. 432-450, 1989.

[4] M. Hofri and K. W. Ross, "On the optimal control of two queues with server set-up times and its analysis," *SIAM J. on Computing* **16**, pp. 399-419, 1987.

[5] H. Levy, M. Sidi, and O. J. Boxma, "Dominance relations in polling systems," *Queueing Systems* (QUESTA) **6**, pp. 155-172, 1990.

[6] Z. Liu and P. Nain, "Optimal scheduling in some multi-queue single server systems," INRIA Report No. 1147, Dec. 1989. To appear in *IEEE Trans. on Automat. Contr.*

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

[8] H. Takagi, "Queueing analysis of polling models," *Stochastic analysis of computer and communication Systems*, H. Takagi (Ed.), North Holland, 1990.

[9] D. Towsley, S. Fdida, H. Santoso, "Design and analysis of flow control protocols for metropolitan area networks," *High-Capacity Local and Metropolitan Area Networks* (ed. G. Pujolle), Springer Verlag, pp. 471-492, 1991.