CYBERNETICS AND PHYSICS, VOL. 7, NO. 1, 2018 , 26–34

ON STABILIZABILITY OF FLUID MULTI-SERVER POLLING SYSTEMS WITH SETUPS

Alexey Matveev
Department of Mathematics and Mechanics
Saint Petersburg State University
Russia
almat1712@yahoo.com

Article history:
Received 28.10.2017, Accepted 15.05.2018

Abstract
A time-invariant fluid model of a polling system is considered. It consists of finitely many servers and buffers with unlimited sizes. The buffers receive inflows of work from the outside, work leaves the system after processing by a server. Every server works only with buffers from an associated zone of service, which may overlap for various servers, is able to serve at most one buffer at a time and so has to switch, from time to time, among buffers, the switch-over times are nonzero. We present a criterion for existence of a scheduling and service protocol that makes the system stable in the sense that the total amount of work in the buffers remains bounded as time progresses. The necessity part of this result is concerned with the widest class of protocols, including dynamic ones that are centralized and have access to the full information about the events in the system. Meanwhile, we show that every stabilizable system can be stabilized in a fully decentralized fashion via a simple static protocol, e.g., by a protocol that is based on independent round robin scheduling of the servers and for every server, employs only time measurement.

Key words
Queues, service, planning and scheduling, stability.

1 Introduction

We consider *polling systems* [Kleinrock, 1976] where finitely many and mostly independent queues share common sources of service (finite capacity *servers*); every server is able to serve at most one queue at a time and has to switch among queues from time to time. Such systems have gained tremendous interest as models of some key aspects of functioning in a wide range of applications, such as computer and communication networks, flexible manufacturing systems, transportation, repair management, chemical kinetics, etc. [Levy and Sidi, 1990; Takagi, 1997, 2001; Vishnevskii and Semenova, 2006; Boon et al., 2014]. In many of these applications, every server incurs a switch-over period when changing a queue.

Control of polling systems is a twofold discipline: for every server, a *scheduling protocol* gives the queue to visit next, whereas a *service protocol* regulates the amount and rate of service given to the current queue. The stability of the closed-loop system is of paramount importance. Stochastic models of queueing theory are typically viewed as stable if the underlying Markov process describing the dynamics of the closed-loop network is positive Harris recurrent [Meyn and Tweedie, 1993]. Stability of a polling system in this sense can be linked to stability of an associated deterministic fluid model, see, e.g., [Rybko and Stolyar, 1992; Chen, 1995; Dai, 1995; Fricker and Jaïbi, 1998; Down, 1998; Foss and Kovalevskii, 1999; Bramson, 2008]. This serves as an extra incentive for study of such models whose general discussion is available in, e.g., [Chen and Mandelbaum, 1994]. In the case of zero switch-over times, definitions of stability of such a fluid model often require that the system eventually drains to zero and then stays empty regardless of the initial state [Dai, 1995]. However, this is unrealistic if the switch-over times are nonzero; then the requirement is weakened: the system's content should merely stay bounded as time progresses [Kumar and Seidman, 1990; Matveev

and Savkin, 2000; Lefeber and Rooda, 2006].

Although the main body of the vast literature on polling systems assumes a single server [Takagi, 1997, 2001; Vishnevskii and Semenova, 2006], nowadays the "second wave" of interest to the multiple server case is motivated by recent advances in multiple access technologies in passive optical networks [Antunes et al., 2011] and emerging intellectual transportation systems [Boon et al., 2014]. The multiplicity of servers brings strong extra challenges, which make the analysis much more involved and less tractable [Dai, 1995].

In [Dai, 1995], an unified approach to stability study is laid down for a general class of stochastic multi-server multi-class open queueing networks with a given queueing discipline by showing that such a network is positive Harris recurrent if the corresponding fluid limit model eventually reaches zero and stays there regardless of the initial system configuration. By using this result, it is shown that the usual traffic condition ensures positive Harris recurrence of the network under a number of specific standard disciplines. Multiple-server stochastic polling models with individual time-invariant stochastic Markov routing of every server over the buffers are examined in [Fricker and Jaïbi, 1998; Down, 1998; Delcoigne and Fayolle, 1999; Antunes et al., 2011]. In [Fricker and Jaïbi, 1998] necessary and sufficient conditions for stability are established for given service disciplines taken from a specific general class. Stability conditions for a somewhat more general case where the service protocol is randomly chosen from a given finite set are offered in [Down, 1998]. In [Delcoigne and Fayolle, 1999], polling systems are analyzed in the thermodynamic limit, i.e., as the numbers of the queues and servers grow without limits, with assuming some symmetry of the system. A stochastic model with two heterogeneous servers, two input streams of customers, and the exhaustive policy is studied in [Foss and Kovalevskii, 1999]. In [van der Mei and Borst, 1997; Borst and van der Mei, 1998], the notion of server limits is introduced: when visiting queues in a cyclic order, the server moves directly to the next queue if the maximal allowed number of servers is already present in the current queue. Formulas for the mean waiting time are obtained for the exhaustive and gated service disciplines. Stability conditions under server limits are established in [Down, 1998; Antunes et al., 2011].

Thus the previous research on multiple-server case was focused on stability of a closed-loop system under Markov routing and mostly dealt with pre-specified classes of static service disciplines. Meanwhile, recent research interest has largely turned to design of dynamic (interactive, feedback) protocols [Aytug et al., 2005; Ouelhadj and Petrovic, 2009; Terekhov et al., 2014], which make decisions on an ongoing basis from the current events in the system. For them, the servers may visit the queues in an a priori unknown, not necessarily regular order, with no guarantees that choices of the next queue happen to be in a Markov fashion. *Whether the breakthrough to this wider class of protocols may bring essential performance improvement?*

We address this issue in static environments, i.e., assume that the arrival rates to the queues, the server maximum productivity, and the changeover periods are time-invariant. A generic deterministic fluid polling system is examined, with negative queue content being meaningless.[1] The primary focus of our analysis is not so much on closed-loop stability as on stabilizability of the system, i.e., the possibility to design a scheduling and service protocol that makes the system stable. The main result of the paper gives necessary and sufficient conditions for stabilizability. In its necessity part, it in fact addresses the widest class of scheduling and service protocols and does not limit anyhow their features, except for inevitable requirements, such as logical consistency in closed-loop. Meanwhile, the sufficiency part shows that any system stabilizable in this widest class can be stabilized in a fully decentralized fashion via a simple static protocol that is based on independent round robin scheduling of the servers.

The body of the paper is organized as follows. The problem to be studied is introduced in Sect. 2. Section 3 introduces the notion of stabilizability, whereas Sect. 4 presents the main result, which proof is given in Sect. 5. Section 6 summarizes the findings of the paper.

2 Multi-Server Polling System

The system is assembled of $n \geq 2$ infinite size *buffers* and s *servers*. The content x_b of any buffer b is called *work* and interpreted as fluid; negative buffer levels $x_b < 0$ are meaningless. Work arrives at every buffer b at a constant rate $\lambda_b > 0$. Every server σ is able to work with at most one buffer at a time but is responsible for service of several buffers, which constitute its *operational zone* $\emptyset \neq Z_\sigma \subset [1:n]$. So any server has to change buffers from time to time; switch from buffer b to b' involves a switch-over activity and consumes a nonzero time $\delta_{b\to b'|\sigma} > 0$. When serving buffer $b \in Z_\sigma$, server σ withdraws its content at a rate $u_\sigma \in [0, \mu_{b,\sigma}]$, where u_σ is a control variable and the *maximal productivity* $\mu_{b,\sigma} > 0$ is given. No buffer b is refused service: $\Sigma_b := \{\sigma : b \in Z_\sigma\} \neq \emptyset$. The operational zones of various servers may overlap. We examine the case where several servers $\sigma_1, \ldots, \sigma_k \in \Sigma(b)$ may cooperate and simultaneously serve a common

[1] When modeling a manufacturing process, with inflows to the queues being interpreted as flows of demands, this means that our study is confined to make-to-order strategies.

buffer b; then the total rate of withdrawing its content x_b may vary from 0 to $\sum_{i=1}^{k} \mu_{b,\sigma_i}$.

A *process* is any feasible scenario of system's evolution over time, including a feasible initial state (X, Q). Here $X = \{x_b\}_{b=1}^{n}$ is a *continuous state* and $Q = \{q_\sigma\}_{\sigma=1}^{s}$ is a *discrete state*, where $q_\sigma = \circledast$ means that server σ is switching among buffers; otherwise, q_σ is the serial number of the attended buffer. The state (X, Q) is *feasible* if

$$x_b \geq 0 \ \forall b, \quad q_\sigma \in \widehat{Z}_\sigma := Z_\sigma \cup \{\circledast\} \ \forall \sigma. \qquad (1)$$

A formal definition of the process is as follows.

Definition 2.1. *A process is any pair of functions $p = [X(\cdot), Q(\cdot)]$ that are defined on $[0, \infty)$ and meet the following requirements:*

- **i)** *The function $Q(t) = \{q_\sigma(t)\}$ is piece-wise constant and $[X(t), Q(t)]$ is a feasible state for all t;*
- **ii)** *If server σ leaves buffer $b_1 := q_\sigma(t_* - 0)$ at a time t_*, the "switch" value $q_\sigma(t_* + 0) = \circledast$ is first assumed by $q_\sigma(\cdot)$ and then altered by some "buffer" value $q_\sigma(t_\star + 0) = b_2$ at $t_\star = t_* + \delta_{b_1 \to b_2 | \sigma}$;*
- **iii)** *The function $X(\cdot)$ is absolutely continuous and*

$$\dot{x}_b(t) = \lambda_b - \sum_{\sigma : b = q_\sigma(t)} u_\sigma(t) \qquad (2)$$

for any buffer $b \in [1 : n]$, where the sum over the empty set is defined to be zero and

$$0 \leq u_\sigma(t) \leq \mu_{q_\sigma(t),\sigma} \text{ whenever } q_\sigma(t) \neq \circledast. \quad (3)$$

Typically, polling systems are controlled according to a certain *scheduling and service protocol*. This is an algorithm that for any time and every server σ, gives 1) the current rate of service,[2] 2) the time when the server should (depending on the type of the current activity) either terminate service of the current buffer or complete the switch between the buffers, and 3) the buffer $b \in Z_\sigma$ to be served next. We consider only deterministic protocols that are arbitrary in other respects, except for the natural requirement that the current decision should not be based on future events. Mathematically, one more inevitable minimal requirement is that the system closed by the control algorithm at hands should be solvable and give rise to a specific process from any feasible initial state (X, Q).

[2] This rate is 0 whenever the server undergoes a switch-over.

3 Stabilizability of the Multi-Server Polling System

The concept of stabilizability addresses possibility to avoid a catastrophic explosion of the total amount of work in the system

$$w = \sum_{b=1}^{n} x_b. \qquad (4)$$

This possibility means that the productivity of the servers conforms to the challenge from the inflows: there is a way to process the inflows with finite queues.

Definition 3.1. **i)** *A process p is said to be* stable *if* $\sup_{t \geq 0} w(t) < \infty$.

- **ii)** *The system is said to be* slightly stabilizable *if there exists a stable process in it.*
- **iii)** *The system is said to be* stabilizable *if some scheduling and service protocol gives rise to a stable process from any feasible initial state.*
- **iv)** *The system is said to be* strongly stabilizable *if there exists a scheduling and service protocol and $c, d \in [0, \infty)$ such that for any feasible initial state and the associated process,*

$$w(t) \leq c\, w(0) + d \qquad \forall t \geq 0.$$

It is clear that ii) $\Leftarrow$ iii) $\Leftarrow$ iv).

If there exists a buffer b for which

$$\lambda_b > \sum_{\sigma \in \Sigma_b} \mu_{b,\sigma}, \qquad (5)$$

then $w(t) \geq x_b(t) \to \infty$ as $t \to \infty$ and so the system is not slightly stabilizable. Now we address the marginal case where $=$ is put in place of $>$ in (5).

Lemma 3.1. *Let $\lambda_b = \sum_{\sigma \in \Sigma_b} \mu_{b,\sigma}$ for buffer b. Then for any stable process, there exists time after which every server $\sigma \in \Sigma_b$ is constantly in service of buffer b.*

Proof. Suppose to the contrary that for some stable process, some server $\sigma \in \Sigma_b$ is not in b at arbitrarily large times. Since switching of σ to b consumes no less than $\delta_{\to b|\sigma} := \min_{b'} \delta_{b' \to b|\sigma} > 0$ time units, there exists an infinite time sequence $t_0 = 0 < t_1 < t_2 < \ldots$ such that $t_{i-1} < t_i - \delta_{\to b|\sigma}$ $\forall i \geq 1$ and σ is not in b for

all $t \in [t_i - \delta_{\to b|\sigma}, t_i], i \geq 1$. Due to (1)—(3), we have

$$\dot{x}_b(t) \geq \lambda_b - \sum_{\sigma' : b = q_{\sigma'}(t)} \mu_{b,\sigma'} \geq$$

$$\begin{cases} \lambda_b - \sum\limits_{\sigma' \in \Sigma_b} \mu_{b,\sigma'} = 0 & \forall t, \\ \lambda_b - \sum\limits_{\sigma' \in \Sigma_b : \sigma' \neq \sigma} \mu_{b,\sigma'} = \mu_{b,\sigma} & \forall t \in [t_i - \delta_{\to b|\sigma}, t_i], \end{cases}$$

where $i \geq 1$ is arbitrary. It follows that $x_b(t) \to \infty \Rightarrow w(t) \to \infty$ as $t \to \infty$, in violation of i) in Definition 3.1. This contradiction completes the proof.

Lemma 3.1 provides an evidence that in the marginal case, stability can be achieved only if all servers $\sigma \in \Sigma_b$ are affixed to the considered buffer b. Since the effect from their acceleration in productivity up to the maximum $\mu_{b,\sigma}$ is beneficial for stability, study of system's stability can be carried out under the condition that the servers $\sigma \in \Sigma_b$ are constantly in b and work at the respective maximal rates, thus keeping the level of buffer b constant. Hence stability depends entirely on capacity of the other servers $\sigma \notin \Sigma_b$ to cope with the outer inflows to buffers $b' \neq b$. This permits one to exclude buffer b and servers $\sigma \in \Sigma_b$ from consideration and focus on the thus obtained system. If in this simpler system, some buffer is refused service (does not belong to the operational zone of any of the remaining servers), instability surely holds, and so the analysis is completed. Otherwise, the simpler system meets the basic assumptions stated at the beginning of Section 2.

By consecutively applying this procedure, while possible, and invoking the remark on instability due to (5), stability analysis can be ultimately boiled down to that for a system with the following property:

$$\lambda_b < \sum_{\sigma \in \Sigma_b} \mu_{b,\sigma} \qquad \forall b. \tag{6}$$

So taking (6) as an assumption in fact does not cause any loss of generality in our stability analysis.

4 Criterion for Stabilizability of the Polling System

To state the main result, we introduce the following convex programming problem:

$$\text{maximize} \quad \sum_{b=1}^{n} \lambda_b z_b$$

$$\text{subject to} \quad \sum_{\sigma} \max_{b} (z_b \mu_{b,\sigma}) \leq 1, \ z_b \geq 0 \ \forall b, \tag{7}$$

where $\mu_{b,\sigma} := 0 \ \forall b \notin Z_\sigma$.

Lemma 5.1 will throw an extra light on this problem.

Theorem 4.1. *Suppose that* (6) *is true. Then the following statements hold:*

- **i)** *If the multi-server polling system is slightly stabilizable, then the maximum value of the cost functional in the problem* (7) *is less than* 1*;*
- **ii)** *Conversely, if the maximum value of the cost functional in the problem* (7) *is less than* 1*, the system is strongly stabilizable.*

Thus slight and strong stabilizability occur only simultaneously. Theorem 4.1 will be proved in Section 5, along with following fact.

Remark 4.1. Whenever the system is strongly stabilizable, its strong stability can be basically ensured via round robin scheduling of the servers: every of them repeatedly runs through its own fixed cycle of visits to the buffers from its operational zone, with serving every buffer during a pre-specified time slot. Every server basically acts independently of the others. However, the servers are periodically synchronized. To this end, progression of a given server through its cycle of services may be delayed at certain recurrent temporal "check points", which are common for all servers. As a result, all servers "depart" from these "check points" simultaneously and their distribution over the buffers at the departure times is periodically repeated.

The problem (7) can be rewritten as that of linear programming. Indeed, let us introduce extra variables γ_σ with the meaning of upper bounds on $\max_b(z_b \mu_{b,\sigma}) = \max_{b \in Z_\sigma}(z_b \mu_{b,\sigma})$, where the last equation holds since $\mu_{b,\sigma} = 0 \ \forall b \notin Z_\sigma$. Then (7) can be shaped into

$$\text{maximize} \quad \sum_{b=1}^{N} \lambda_b z_b \quad \text{subject to}$$

$$\sum_{\sigma} \gamma_\sigma \leq 1, \ \ z_b \geq 0 \ \forall b, \ z_b \leq \frac{\gamma_\sigma}{\mu_{b,\sigma}} \forall b \in Z_\sigma, \sigma. \tag{8}$$

In turns, focusing on the maximum value of z_b yields the following equivalent dual reformulation of (7):

$$\text{maximize} \quad \sum_{b} \lambda_b \min_{\sigma \in \Sigma_b} \frac{\gamma_\sigma}{\mu_{b,\sigma}} \quad \text{subject to}$$

$$\gamma_\sigma \geq 0 \ \forall \sigma, \quad \sum_{\sigma} \gamma_\sigma \leq 1. \tag{9}$$

The following examples demonstrate that the criterion from Theorem 4.1 can be transformed in a closed from in some cases. The first two of these examples aim

at displaying the conformity of Theorem 4.1 with the well-known traffic conditions for single-server polling systems; see, e.g., [Hopp and Spearman, 2001]. To make the things interesting, the number of the buffers $n \geq 2$ in all examples.

Example 1. Let the number of the servers $s = 1$, which permits us to drop the index $\sigma = 1$ in $\mu_{b,\sigma}$. The assumption (6) means that $\lambda_b < \mu_b \ \forall b$. The maximum value of the cost functional (9) equals $\sum_{b=1}^{n} \frac{\lambda_b}{\mu_b}$ and so the criterion from Theorem 4.1 comes to

$$\sum_{b=1}^{n} \frac{\lambda_b}{\mu_b} < 1.$$

Example 2. The number of the servers $s \geq 2$, their zones of service are disjoint $Z_{\sigma'} \cap Z_{\sigma''} = \emptyset \ \forall \sigma' \neq \sigma''$. Since the server $\sigma \in \Sigma(b)$ is determined by b, the index σ in $\mu_{b,\sigma}$ is dropped. The assumption (6) still means that $\lambda_b < \mu_b \ \forall b$. The problem (9) takes the form

$$\text{maximize} \quad \sum_{\sigma} \gamma_\sigma \sum_{b \in Z_\sigma} \frac{\lambda_b}{\mu_b} \quad \text{subject to}$$
$$\gamma_\sigma \geq 0 \ \forall \sigma, \quad \sum_{\sigma} \gamma_\sigma \leq 1.$$

The respective maximum equals $\max_\sigma \sum_{b \in Z_\sigma} \frac{\lambda_b}{\mu_b}$. Thus the criterion from Theorem 4.1 means that $\sum_{b \in Z_\sigma} \lambda_b / \mu_b < 1$ for all servers σ.

Example 3. Two servers $s = 2$, more than one buffer in the zone of responsibility of each of them, these zones contain a single common buffer $b^0 \in Z_1 \cap Z_2$.

The problem (9) takes the form:

$$\text{maximize} \quad \gamma_1 \sum_{b \in \Sigma_1 : b \neq b^0} \frac{\lambda_b}{\mu_{b,1}}$$
$$+ \lambda_{b^0} \min\left\{ \frac{\gamma_1}{\mu_{b^0,1}}; \frac{\gamma_2}{\mu_{b^0,2}} \right\} + \gamma_2 \sum_{b \in \Sigma_2 : b \neq b^0} \frac{\lambda_b}{\mu_{b,2}}$$
$$\text{subject to} \quad \gamma_1, \gamma_2 \geq 0, \quad \gamma_1 + \gamma_2 \leq 1.$$

Here $\gamma_1 + \gamma_2 \leq 1$ can be evidently replaced by $\gamma_1 + \gamma_2 = 1 \Leftrightarrow \gamma_2 = 1 - \gamma_1$. Being treated as a function of $\gamma_1 \in [0, 1]$, the continuous piecewise linear cost functional attains its maximum either at an end-point $\gamma_1 = 0$ or $\gamma_1 = 1$ of the interval $[0, 1]$, or at the point of the fracture, where $\gamma_1 / \mu_{b^0,1} = \gamma_2 / \mu_{b^0,2}$ and so $\gamma_1 = \mu_{b^0,1} / (\mu_{b^0,1} + \mu_{b^0,2}), \gamma_2 = \mu_{b^0,2} / (\mu_{b^0,1} + \mu_{b^0,2})$.

Thus the criterion from Theorem 4.1 takes the form

$$\sum_{b \in \Sigma_1 : b \neq b^0} \frac{\lambda_b}{\mu_{b,1}} < 1, \quad \sum_{b \in \Sigma_2 : b \neq b^0} \frac{\lambda_b}{\mu_{b,2}} < 1,$$
$$\frac{\mu_{b^0,1}}{\mu_{b^0,1} + \mu_{b^0,2}} \sum_{b \in \Sigma_1 : b \neq b^0} \frac{\lambda_b}{\mu_{b,1}}$$
$$+ \frac{\lambda_{b^0}}{\mu_{b^0,1} + \mu_{b^0,2}}$$
$$+ \frac{\mu_{b^0,2}}{\mu_{b^0,1} + \mu_{b^0,2}} \sum_{b \in \Sigma_2 : b \neq b^0} \frac{\lambda_b}{\mu_{b,2}} < 1.$$

Here the third inequality implies some of the first two inequalities.

5 Proof of Theorem 4.1

We first show that the criterion from Theorem 4.1 can be reformulated as feasibility of the following set of linear relations in unknowns $\tau_{b,\sigma}$:

$$\tau_{b,\sigma} \geq 0 \quad \forall b, \sigma,$$
$$\sum_{b} \tau_{b,\sigma} < 1 \ \forall \sigma, \ \sum_{\sigma} \mu_{b,\sigma} \tau_{b,\sigma} = \lambda_b \ \forall b. \quad (10)$$

Remark 5.1. Feasibility of (10) is equivalent to feasibility of the relaxed system that results from putting $\sum_\sigma \mu_{b,\sigma} \tau_{b,\sigma} \geq \lambda_b$ in place of $\sum_\sigma \mu_{b,\sigma} \tau_{b,\sigma} = \lambda_b$:

$$\tau_{b,\sigma} \geq 0 \quad \forall b, \sigma,$$
$$\sum_{b} \tau_{b,\sigma} < 1 \ \forall \sigma, \ \sum_{\sigma} \mu_{b,\sigma} \tau_{b,\sigma} \geq \lambda_b \ \forall b. \quad (11)$$

Indeed, on the one hand, (10) $\Rightarrow$ (11). On the other hand, any solution $\tau_{b,\sigma}$ of (11) is transformed into a solution of (10) by putting $\tau'_{b,\sigma} := a_b \tau_{b,\sigma}$, where $a_b := \lambda_b \left(\sum_\sigma \mu_{b,\sigma} \tau_{b,\sigma} \right)^{-1} \in [0, 1]$ and so $\tau'_{b,\sigma} \leq \tau_{b,\sigma}$.

Lemma 5.1. *Feasibility of* (10) *holds if and only if the maximum of the cost functional in* (7) *is less than* 1.

Proof. By Remark 5.1, attention can be switched to (11). By Motzkin's transposition theorem [Schrijver, 1999, Cor. 7.1k], the infeasibility of (11) is equivalent to existence of $z_b \geq 0, y_\sigma \geq 0$, and $x_{b,\sigma} \geq 0$ such that

$$y_\sigma = x_{b,\sigma} + \mu_{b,\sigma} z_b, \quad \sum_{\sigma} y_\sigma - \sum_{b} \lambda_b z_b \leq 0,$$
$$\text{and either} \ \sum_{\sigma} y_\sigma - \sum_{b} \lambda_b z_b < 0 \ \text{or} \ \sum_{\sigma} y_\sigma > 0.$$

Here the unknowns $x_{b,\sigma}$ can be evidently dropped by transforming the first equation into the inequality $y_\sigma \geq \mu_{b,\sigma} z_b$. So (11) is infeasible if and only if the following system has a solution $y_\sigma \geq 0, z_b \geq 0$:

$$y_\sigma \geq \mu_{b,\sigma} z_b \forall b, \sigma, \quad \sum_\sigma y_\sigma \leq \sum_b \lambda_b z_b,$$
$$\text{and either } \sum_\sigma y_\sigma < \sum_b \lambda_b z_b \text{ or } \sum_\sigma y_\sigma > 0.$$

Since the first inequality implies that $y_\sigma \geq \max_b(\mu_{b,\sigma} z_b)$, it is easy to see that the last system is feasible if and only if there exists a solution $z_b \geq 0$ to

$$\sum_\sigma \max_b(\mu_{b,\sigma} z_b) \leq \sum_b \lambda_b z_b,$$
$$\text{and either } \sum_\sigma \max_b(\mu_{b,\sigma} z_b) < \sum_b \lambda_b z_b$$
$$\text{or } q := \sum_\sigma \max_b(\mu_{b,\sigma} z_b) > 0. \quad (12)$$

Since $q = 0 \Rightarrow z_b = 0 \ \forall b$, in violation of the second row in (12), we see that $q > 0$. By putting $z_b := q^{-1} z_b \ \forall b$, we get another solution of (12) and make $q = 1$. These z_b's are feasible in (7) and for them, the value of the cost functional is no less than 1. Thus its maximal value is also no less than 1.

Conversely, suppose that this value is no less than 1. Let us consider a solution $z_1, \ldots, z_n$ for (7). Since $\sum_b \lambda_b z_b \geq 1 \Rightarrow \exists b : z_b > 0$ and so $q > 0$ in (12). Thus (10) is infeasible if and only if the maximum value of the cost functional in (7) is no less than 1, which completes the proof. □

5.1 Proof of i) in Theorem 4.1

Let the system be slightly stabilizable. We consider a stable process p, which exists by ii) in Definition 3.1, and the events that occur for p. For any time interval $I \subset [0, \infty)$, we introduce the following notations:

- $t_{b,\sigma}(I)$, the total time spent by server σ in buffer b within the time interval I;
- $\Delta_\sigma(I)$, the total time spent by server σ on switching among buffers within the time interval I;
- $|I|$, the length of the interval I.

An infinite sequence of time intervals $\mathcal{I} = \{I_j\}_{j=1}^\infty$ is said to be *proper* if the following statements hold:

p1) the intervals I_j are pairwise disjoint;
p2) $I_j \subset [0, \infty) \ \forall j$ and $|I_j| \to \infty$ as $j \to \infty$;
p3) there exist the following limits

$$\tau_{b,\sigma}(\mathcal{I}) := \lim_{j\to\infty} \frac{t_{b,\sigma}(I_j)}{|I_j|} \quad \forall b, \sigma,$$
$$\delta_\sigma^\infty(\mathcal{I}) := \lim_{j\to\infty} \frac{\Delta_\sigma(I_j)}{|I_j|} \quad \forall \sigma. \quad (13)$$

Since the ratios in (13) do not exceed 1, any sequence $\{I_j\}_{j=1}^\infty$ satisfying **p1)** and **p2)** has a proper subsequence. So proper sequences do exist. Any subsequence of a proper sequence is clearly proper and has the same characteristics (13) as the parent sequence. A proper sequence $\mathcal{I}' = \{I'_k\}_{k=1}^\infty$ is called an *abatement* of a proper sequence $\mathcal{I} = \{I_j\}_{j=1}^\infty$ if for any k there exists $j = j(k)$ such that $I'_k \subset I_{j(k)}$.

Definition 5.1. *Server σ is said to be* fixed *for a proper sequence $\mathcal{I}$ if there exists $b = b(\sigma) \in Z_\sigma$ such that server σ is in buffer b at any time $t \in I_j$ from any interval in this sequence.*

Lemma 5.2. *Let $\delta_\sigma^\infty(\mathcal{I}) = 0$ for some server σ and a proper sequence $\mathcal{I} = \{I_j\}_{j=1}^\infty$. Then there exists an abatement of $\mathcal{I}$ for which this server is fixed.*

Proof. Thanks to i) in Definition 2.1, the set $\{t \in I_j : \text{server } \sigma \text{ is in buffer } b\}$ is either empty or is assembled of finitely many disjoint subintervals $I_j^{b|\sigma}(1), \ldots, I_j^{b|\sigma}(m_j)$ interspersed by intervals where σ is not in b. Let $L_j(b)$ be zero in the former case and be the maximal length of these subintervals in the latter case. We are going to show that the sequence $\{L_j(b)\}_{j=1}^\infty$ is unbounded for some $b \in Z_\sigma$.

Suppose to the contrary that there exists $L \in (0, \infty)$ such that $L_j(b) \leq L \ \forall j, b \in Z_\sigma$, and put

$$\delta_\sigma^- := \min_{b' \neq b''} \delta_{b' \to b''|\sigma} > 0, \ \delta_\sigma^+ := \max_{b' \neq b''} \delta_{b' \to b''|\sigma} > 0.$$

Now for given j, we gather all intervals of the form $I_j^{b|\sigma}(p), p = 1, \ldots, m_j, b \in Z_\sigma$ and put them in the ascending order. By ii) in Definition 2.1, any two adjoint terms in the resultant sequence are separated by an interval of switch-over activity; its duration is thus no less than δ_σ^- and no more than δ_σ^+. Such an interval may also separate the left end of I_j and the first interval in this sequence, as well as the the right end of I_j and the last interval. It follows that $|I_j| \leq \delta_\sigma^+ + N_j(L + \delta_\sigma^+)$, where N_j is the number of elements in this sequence. Hence $N_j \geq (|I_j| - \delta_\sigma^+)/(L + \delta_\sigma^+)$ and $\Delta_\sigma(I_j) \geq$

$\delta_\sigma^-(N_j-1)$. By invoking (13) and **p2)**, we see that

$$\delta_\sigma^\infty(\mathcal{I}) = \lim_{j\to\infty}\frac{\Delta_\sigma(I_j)}{|I_j|} \geq \lim_{j\to\infty}\frac{\delta_\sigma^-}{|I_j|}\left[\frac{|I_j|-\delta_\sigma^+}{L+\delta_\sigma^+}-1\right] = \frac{\delta_\sigma^-}{(L+\delta_\sigma^+)} > 0,$$

in violation of the assumption $\delta_\sigma^\infty(\mathcal{I})=0$ of the lemma.

The contradiction obtained proves that the sequence $\{L_j(b)\}$ is unbounded for some $b\in Z_\sigma$. Then there exists a subsequence such that $L_{j(k)}(b)\to\infty$ as $j\to\infty$. Meanwhile, $L_{j(k)}(b) = |I^{b|\sigma}_{j(k)}[p(k)]|$ for some $p(k)$ by the definition of $L_j(b)$. The sequence $\{I'_k := I^{b|\sigma}_{j(k)}[p(k)]\}_{k=1}^\infty$ clearly meets **p1)**, **p2)**, whereas **p3)** can be ensured by passing to a subsequence, as was remarked just after (13). This subsequence is clearly proper. By construction, it is an abatement of the initial sequence $\mathcal{I}$ and server σ is fixed for it. □

If server σ is fixed for a proper sequence I, it is evidently fixed for any its abatement as well. So Lemma 5.2 can be applied recurrently until arrival at the situation described in the following.

Corollary 5.1. *For any proper sequence $\mathcal{I}=\{I_j\}_{j=1}^\infty$, there exists an abatement of $\mathcal{I}$ for which every server σ is either fixed or has $\delta_\sigma^\infty(\mathcal{I})>0$.*

Completion of the proof of **i)** *in Theorem 4.1.* By Corollary 5.1, there exists a proper sequence $\mathcal{I}$ for which every server σ is either fixed or has $\delta_\sigma^\infty(\mathcal{I})>0$. Let Σ_{fix} stand for the set of fixed servers and let $b(\sigma)$ be taken from Definition 5.1 for $\sigma\in\Sigma_{\text{fix}}$. We are going to show that $\tau_{b,\sigma} := \tau_{b,\sigma}(\mathcal{I})$ obey the following relations:

$$\begin{gathered}\tau_{b,\sigma}\geq 0 \quad \forall b,\sigma,\\ \tau_{b(\sigma),\sigma}\leq 1, \tau_{b,\sigma}=0 \quad \forall b\neq b(\sigma), \sigma\in\Sigma_{\text{fix}},\\ \sum_b \tau_{b,\sigma}<1\ \forall\sigma\notin\Sigma_{\text{fix}},\ \sum_\sigma \mu_{b,\sigma}\tau_{b,\sigma}=\lambda_b\ \forall b. \quad (14)\end{gathered}$$

Here the relations from the first and second row are evident. For every $\sigma\notin\Sigma_{\text{fix}}$, any time interval I_j from $\mathcal{I}$ is fully composed of times when σ either serves some buffer or switches among buffers. So

$$\sum_b t_{b,\sigma}(I_j)+\Delta_\sigma(I_j)=|I_j|.$$

Dividing this equation by $|I_j|$, letting $j\to\infty$, and invoking (13) show that

$$\sum_b \tau_{b,\sigma}(\mathcal{I})+\delta_\sigma^\infty(\mathcal{I})=1,$$

where $\delta_\sigma^\infty(\mathcal{I})>0$. Thus the first inequality in the third row from (14) is true.

By i) of Definition 3.1, there exists $w_\infty\in[0,\infty)$ such that $w(t)\leq w_\infty\ \forall t\geq 0$, where $w(t)\geq x_b(t)\ \forall b$ by (4). For any buffer b, the total amount of work that is brought by the outer inflow to this buffer during the time interval $I_j=[t_j^-,t_j^+]$ equals $\lambda_b|I_j|$. Meanwhile the amount of work withdrawn by all servers during this interval does not exceed $\sum_\sigma \mu_{b,\sigma}t_{b,\sigma}(I_j)$. So

$$\begin{gathered}2w_\infty\geq x_b(t_j^+)-x_b(t_j^-)\geq\lambda_b|I_j|-\sum_\sigma\mu_{b,\sigma}t_{b,\sigma}(I_j)\\ \Rightarrow\sum_\sigma\mu_{b,\sigma}\frac{t_{b,\sigma}(I_j)}{|I_j|}\geq\lambda_b-2\frac{w_\infty}{|I_j|}\\ \xRightarrow{j\to\infty}\sum_\sigma\mu_{b,\sigma}\tau_{b,\sigma}\geq\lambda_b\quad\forall b.\end{gathered}$$

To complete the proof of (14), it suffices to correct $\tau_{b,\sigma}$ by putting $\tau'_{b,\sigma} := a_b\tau_{b,\sigma}$, where $a_b := \lambda_b\left(\sum_\sigma\mu_{b,\sigma}\tau_{b,\sigma}\right)^{-1}\in[0,1]$ and so $\tau'_{b,\sigma}\leq\tau_{b,\sigma}$.

By (14), $\tau_{b,\sigma}\leq 1$ for all b and σ. The equation from the third row in (14) and (6) imply that $\tau_{b,\sigma}<1$ for some $\sigma\in\Sigma_b$ irrespective of buffer b. So if $\tau_{b(\sigma),\sigma}=1$ for some server $\sigma\in\Sigma_{\text{fix}}$, there is a possibility to slightly reduce $\tau_{b(\sigma),\sigma}=1$, with slightly increasing $\tau_{b,\sigma'}<1$ for $b=b(\sigma)$ and some other $\sigma'\in\Sigma_b$, so that (14) remains true, along with all initially strict inequalities in the second row, whereas $\tau_{b(\sigma),\sigma}=1$ becomes less than 1. By consecutively repeating this procedure, we ultimately see that (14) is feasible with $<$ put in place of $\leq$ in the second row.

In turns, this means that (10) is feasible. Lemma 5.1 completes the proof.

5.2 Proof of Remark 4.1 and Part ii) in Theorem 4.1

By Lemma 5.1, there exists a solution $\{\tau_{b,\sigma}\}$ for (10). We put $\tau_{b,\sigma}:=0$ whenever $\mu_{b,\sigma}=0$, which does not violate (10). Also, we order the operational zone $Z_\sigma=\{b_1^\sigma\neq\cdots\neq b_{k(\sigma)}^\sigma\}$ of every server σ and put

$$\Delta_\sigma := \sum_{i=1}^{k(\sigma)-1}\delta_{b_i^\sigma\to b_{i+1}^\sigma|\sigma}+\delta_{b_{k(\sigma)}^\sigma\to b_1^\sigma|\sigma}. \quad (15)$$

This is the total time that server σ spends on switching among buffers when running through one cycle depicted in Fig. 1. Since $\sum_{b\in Z_\sigma}\tau_{b,\sigma}<1$ by (10), there exists a large enough T such that

$$T\sum_{b\in Z_\sigma}\tau_{b,\sigma}+\Delta_\sigma\leq T\ \forall\sigma,\quad T>\max_{b'\neq b''}\delta_{b'\to b''}. \quad (16)$$

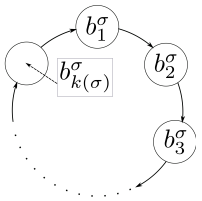

Figure 1. The cycle run by server σ.

The proposed protocol consists of the following rules:

r1) Within the time interval $[0, T]$, every server σ is switched to buffer b_1^σ, where it idles $u_{b_1^\sigma,\sigma} \equiv 0$ until T if the switch is completed before T;

r2) Within any time interval of the form $[kT, (k+1)T], k \geq 1$, every server σ with $k(\sigma) > 1$ performs the following operations:

- **A)** It makes a tour over its operational zone in the cyclic order $b_1^\sigma \mapsto \cdots \mapsto b_{k(\sigma)}^\sigma \mapsto b_1^\sigma$ illustrated in Fig. 1, starting and finishing in the same buffer b_1^σ;
- **B)** When visiting buffer b_i^σ, it serves this buffer at the maximal rate $\mu_{b_i^\sigma,\sigma}$ during $\tau_{b_i^\sigma,\sigma} \cdot T$ units of time;
- **C)** After this, it is switched to b_{i+1}^σ if $i < k(\sigma)$, and to b_1^σ if $i = k(\sigma)$;
- **D)** In the latter case, the server idles in buffer b_1^σ until $(k+1)T$ if the switch to b_1^σ is completed before $(k+1)T$.

r3) Within any time interval of the form $[kT, (k+1)T], k \geq 1$, every server σ with $k(\sigma) = 1$ first serves the buffer b_1^σ at the maximal rate $\mu_{b_1^\sigma,\sigma}$ during $\tau_{b_1^\sigma,\sigma} \cdot T$ units of time and then idles in buffer b_1^σ until $(k+1)T$.

To prove that **r1)—r3)** are always executable, it suffices to show that

a) on $[0, T]$, every server has enough time to complete switching requested in **r1)**,

b) the rules **r2B)** and **r3)** are executable on the time interval $[kT, (k+1)T]$ for any $k \geq 1$, i.e., the content of buffer b is large enough so that any concerned server can work at the maximal rate during the requested time slot,

c) on any interval $[kT, (k+1)T], k \geq 1$, every server σ with $k(\sigma) > 1$ completes the tour from Fig. 1 before $(k+1)T$.

All these are true indeed.

a) This is true due to the second inequality from (16).

b) It suffices to prove that for all k,

$$x_b[kT] \geq \sum_{\sigma \in \Sigma(b)} x_\downarrow(b, \sigma), \tag{17}$$

where $x_\downarrow(b, \sigma) := \mu_{b,\sigma} \cdot \tau_{b,\sigma} \cdot T$ is the amount of work to be withdrawn by server σ from buffer $b \in Z_\sigma$ when working at the maximal rate $\mu_{b,\sigma}$ during $\tau_{b,\sigma} T$ units of time, as is required by the protocol. We shall argue via induction on k. Thanks to the last equation from (10),

$$\sum_{\sigma \in \Sigma(b)} x_\downarrow(b, \sigma) = T \sum_{\sigma \in \Sigma(b)} \mu_{b,\sigma} \tau_{b,\sigma} = \lambda_b T.$$

So for $k = 1$, (17) is true thanks to **r1)**. Suppose that (17) holds for some $k \geq 1$. During the time interval $[kT, (k+1T)]$ the servers withdraw $\sum_{\sigma \in \Sigma(b)} x_\downarrow(b, \sigma)$ units of work from buffer b due to **r2), r3)** and the induction hypothesis, whereas $\lambda_b T$ units are brought to b by the outer inflow. Thanks to (17), this yields that

$$x_b[(k+1)T] = x_b[kT] \tag{18}$$

and so (17) holds with $k := k + 1$.

c) To fully serve all buffers within one cycle from Fig. 1, server σ needs $T \sum_{b \in Z_\sigma} \tau_{b,\sigma}$ time units, whereas Δ_σ time units more are needed for switching among the buffers. The proof is completed by the first inequality from (16).

To prove strong stability, we define $k(t)$ as the integer floor of t/T and note that $\tau(t) := t - Tk(t) \in [0, T) \Rightarrow x_b[t] \leq x_b[Tk(t)] + \lambda_b T$. Meanwhile, $x_b(t) = x_b(0) + \lambda_b t \leq x_b(0) + \lambda_b T \ \forall t \in [0, T]$ thanks to **r1)**. So (18) implies that that $x_b(kT) \leq x_b(0) + \lambda_b T \ \forall k \geq 0$ and

$$w(t) \overset{(4)}{=} \sum_{b=1}^n x_b(t) \leq \sum_{b=1}^n \Big\{ x_b[Tk(t)] + \lambda_b T \Big\}.$$
$$\overset{(18)}{=} \sum_{b=1}^n \big[x_b(0) + 2\lambda_b T \big] \overset{(4)}{=} w(0) + d,$$

where $d := 2T \sum_{b=1}^n \lambda_b$. The proof is completed by **iv)** in Definition 3.1. □

6 Conclusion

A criterion for stabilizability of multiple-server stationary fluid model of a polling system was obtained. In general, this criterion refers to solution of a linear/convex programming problem; it was shown that it can be transformed in a closed form in some special cases. It was also shown that any stabilizable fluid model can be stabilized in a fully decentralized fashion via a simple static protocol that is based on independent round robin scheduling of the servers and for every server, employs only time measurement.

Acknowledgements

This research was supported by the Russian Science Foundation under the grant 14-21-00041p and was performed in Saint Petersburg State University.

References

Antunes, N., Fricker, C. and Roberts, J. [2011]. Stability of multi-server polling system with server limits, *Queueing systems* **68**: 229–235.

Aytug, H., Lawley, M., McKay, K. and Mohan, S. [2005]. Executing production schedules in the face of uncertainties: A review and some future directions, *Europ. J. of Oper. Research* **161**(1): 86–110.

Boon, M., van der Mei, R. and Winands, E. [2014]. Applications of polling systems, arXiv. 1408.0136v1.

Borst, S. and van der Mei, R. [1998]. Waiting time approximations for multiple-server polling systems, *Performance Evaluation* **31**: 163–182.

Bramson, M. [2008]. Stability of queueing networks, Vol. 1950 of *Lecture Notes in Mathematics*, Springer-Verlag.

Chen, H. [1995]. Fluid approximations and stability of multiclass queueing networks: work-conserving disciplines, *Ann. Appl. Probab.* **5**: 637–655.

Chen, H. and Mandelbaum, A. [1994]. Hierarchical modeling of stochastic networks, Part I: Fluid models, *in* D. Yao (ed.), *Stochastic Modeling and Analysis of Manufacturing Systems*, Springer-Verlag, Berlin.

Dai, J. [1995]. On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models, *Ann. Appl. Probab.* **5**: 49–77.

Delcoigne, F. and Fayolle, G. [1999]. Thermodynamical limit and propagation of chaos in polling systems, *Markov Processes and Related Fields* **5**(1): 89–124.

Down, D. [1998]. On the stability of polling models with multiple server, *Journal of Applied Probability* **35**(4): 925–935.

Foss, S. and Kovalevskii, A. [1999]. A stability criterion via fluid limits and its application to a polling system, *Queueing systems* **32**(1–3): 131–168.

Fricker, C. and Jaïbi, M. [1998]. Stability of multi-server polling models, *Technical Report RR-3347*, INRIA.

Hopp, W. and Spearman, M. [2001]. *Factory physics*, second edn, McGraw-Hill, New York.

Kleinrock, L. [1976]. *Queueing Systems*, Vol. 2, John Wiley and Sons, New York.

Kumar, P. R. and Seidman, T. I. [1990]. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems, *IEEE Transactions on Automatic Control* **35**(3): 289–298.

Lefeber, E. and Rooda, J. [2006]. Controller design for a reentrant network of servers with setup times: the Kumar-Seidman case, *45th IEEE Conference on Decision and Control*, San Diego, CA.

Levy, H. and Sidi, M. [1990]. Polling systems: applications, modeling, and optimization, *IEEE Transactions on Communications* **38**(10): 1750–1760.

Matveev, A. S. and Savkin, A. V. [2000]. *Qualitative Theory of Hybrid Dynamical Systems*, M. A., Birkhauser, Boston.

Meyn, S. and Tweedie, R. [1993]. *Markov Chains and Stochastic Stability*, Springer-Verlag, London.

Ouelhadj, D. and Petrovic, S. [2009]. A survey of dynamic scheduling in manufacturing systems, *Journal of Scheduling* **12**(4): 417–431.

Rybko, A. and Stolyar, A. [1992]. Ergodicity of stochastic processes describing the operation of open queueing networks, *Problems of Information Transmission* **28**: 199–220.

Schrijver, A. [1999]. Theory of linear and integer programming, Wiley-Interscience series in discrete mathematics and optimization, Wiley & Sons, NY.

Takagi, H. [1997]. Queueing analysis of polling models: progress in 1990-1994, *in* J. Dshalalow (ed.), *Frontiers in Queueing: Models, Methods and Problems*, CRC Press, Boca Raton, pp. 119–146.

Takagi, H. [2001]. Bibliography on polling models, *Technical report*, Institute of Socio-Economic Planning, University of Tsukuba, Tsukuba-shi, Japan. Available at `http://www.sk.tsukuba.ac.jp/takagi/polling.html`.

Terekhov, D., Down, D. and Beck, J. [2014]. Queueing-theoretic approaches for dynamic scheduling: A survey, *Surveys in Operations Research and Management Science* **19**(2): 105–129.

van der Mei, R. and Borst, S. [1997]. Analysis of multiple-server polling systems by means of the powerseries algorithm, *Stochastic Models* **13**(2): 339–369.

Vishnevskii, V. and Semenova, O. [2006]. Mathematical methods to study the polling systems, *Automation and Remote Control* **67**(2): 173–220.