

# Modeling, Scheduling and Simulation of Switched Processing Systems

YING-CHAO HUNG

Graduate Institute of Statistics

National Central University, Taiwan

and

GEORGE MICHAILIDIS

Department of Statistics and Electrical Engineering and Computer Science

University of Michigan, Ann Arbor

---

Switched Processing Systems (SPS) serve as canonical models in a wide area of applications such as high performance computing, wireless networking, call centers, and flexible manufacturing. In this paper, we model the SPS by considering both slotted and continuous time and analyze it under fairly mild stochastic assumptions. Two classes of scheduling policies are introduced and shown to maximize the throughput and maintain strong stability of the system. In addition, their performance with respect to the average job sojourn time is examined by simulating small SPS subject to different types of input traffic. By utilizing the simulation result of the proposed policies, a hybrid control policy is constructed to reduce the average job sojourn time when the system has unknown and changing input loads.

Categories and Subject Descriptors: C.2.3 [Computer-Communication Networks]: Network Operations

General Terms: Management, Performance, Theory

Additional Key Words and Phrases: Switched processing systems, maximal throughput, strong stability, scheduling policy, average sojourn time, simulation

---

## 1. INTRODUCTION: MODEL AND APPLICATIONS

Switched Processing Systems (SPS) serve as canonical models in a wide area of applications such as computing, wireless networking, call centers, and manufacturing. They are comprised of parallel FIFO queues whose service rates are specified by a pool of service configurations. An important feature of these models is that they allow flexible scheduling, i.e., the system can switch into one of possible service modes at any point in time. This induces a fundamental resource-sharing problem on how the appropriate service configu-

---

The research of Ying-Chao Hung was supported in part by NSC Grant 91-2119-M-008-026 and NSC Grant 92-2118-M-008-012. The research of George Michailidis was supported in part by NSF grants CCR-0325571 and DMS-0500535.

Author's address: Ying-Chao Hung, Graduate Institute of Statistics, National Central University, Zhongli 32049, Taiwan; E-mail: hungy@stat.ncu.edu.tw; George Michailidis, Department of Statistics, University of Michigan, Ann Arbor, MI 48109-1107; E-mail: gmichail@umich.edu.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20YY ACM 0000-0000/20YY/0000-0001 \$5.00

ration should be selected in order to optimize the performance metrics of interest. Specifically, consider a queueing system consisting of  $Q$  infinite capacity first-in-first-out (FIFO) queues in parallel, with each queue corresponding to a different class of job traffic. To illustrate the main features of the system, we first assume that it operates in slotted time and that job arrival process is given by a Bernoulli process  $\mathcal{A}_q$  with normalized rate  $\lambda_q$ , i.e., at each time slot class  $q$  receives a job with probability  $\lambda_q$ ,  $0 \leq \lambda_q \leq 1$ ,  $q = 1, \dots, Q$ . This implies that the interarrival times of the class  $q$  jobs are independent random variables from a common geometric distribution having mean  $E[\tau_q] = 1/\lambda_q$ ,  $q = 1, \dots, Q$ . Further, at any point in time, the system can be in one of  $M$  service modes. When it switches into service mode  $m$ , the head-of-line (HOL) jobs of queue  $q$  departs according to a Bernoulli process  $\mathcal{D}_{mq}$  with normalized rate  $\mu_{mq}$ , i.e., at each time slot the HOL job of queue  $q$  leaves with probability  $\mu_{mq}$ ,  $0 \leq \mu_{mq} \leq 1$ ,  $q = 1, \dots, Q$ , and  $m = 1, \dots, M$ . This implies that the service times of the class  $q$  jobs are independent random variables from a common geometric distribution having mean  $E[S_{mq}] = 1/\mu_{mq}$ ,  $q = 1, \dots, Q$ , and  $m = 1, \dots, M$ . Therefore, mode  $m$  is associated with the service rate vector  $\vec{\mu}_m = (\mu_{m1}, \mu_{m2}, \dots, \mu_{mQ})$ . We also assume that all departure processes  $\mathcal{D}_{mq}$  and arrival processes  $\mathcal{A}_q$  are mutually independent. The model is extended to operating in continuous time in Section 5.

This basic queueing model captures the essence of a fundamental resource allocation problem in many modern systems involving heterogeneous processors and multiple classes of job traffic flows. We discuss some interesting applications next. The first application is a wireless packet network with unreliable communication links that interface with each other but share the same channel. The queues are associated with transmitters in which the arriving packets are queued up and waiting to be transmitted to the receiver. At each time slot, a transmitter can transmit one packet from its queue at a particular power level. Thus, the service rate  $\mu_{mq}$  corresponds to the probability that the packet transmitted by queue  $q$  is successfully received at its receiver, and, each service mode is determined by the power levels.

The second application is that of the high-speed  $N$  by  $N$  crossbar switch with packets/cells arriving to input ports and destined for output ports. At any time, the switch can open the connection between each input port and exactly one output port. It is assumed that a cell from input port  $i$  to output port  $j$  arrives according to a Bernoulli process with rate  $\lambda_{ij}$  which corresponds to class  $q$  in our setting. Also, upon each connection a single cell is transmitted and the transmission takes one time slot. Thus,  $\mu_{mq}$  corresponds to the transmission rate of the connection between input port  $m$  and output port  $q$ , which is one for all  $m$  and  $q$ .

Another important application of this model is in the area of flexible manufacturing. Suppose a factory workstation is manned by a worker who operates a set of tools, working on multiple classes of products. Each class is queued up in a separate buffer. To complete a job of a certain product class, the worker needs to use a particular subset of the tools. Thus, each particular subset of the tools corresponds to one service mode. There are some other applications, which we briefly introduce below. In call centers, different types of jobs correspond to different requests (say airplanes, rental cars, cruises, etc). The service vectors correspond to different specializations of the workforce. In web services, one request may require a database search, another request may require a calculation, a third request may require loading an applet. The service vectors correspond to resources available (CPUs, memory, hardisks, etc) to service those requests.

The stability, throughput maximization, and resource allocation are the most fundamental problems when analyzing such systems. Over the years, an extensive literature has been developed to construct throughput-maximizing scheduling policies for the well-known input-queued switch (a special model of SPS) so that a certain level of system stability can be achieved (see [Dai and Prabhakar 2000; McKeown et al. 1999; Mekkitikul and McKeown 1996] and references therein). For example, in [McKeown et al. 1999; Mekkitikul and McKeown 1996] a 100% throughput and strong system stability were achieved by using the Maximal Weight Matching algorithms (called LQF and OCF) under Markovian settings. Strong stability in this context means that each queue has a finite expected length, independent of time. More recently, Armony and Bambos [Armony and Bambos 2003] showed that 100% throughput and rate stability of SPS can be achieved by extending the Maximal Weight Matching algorithm to a class of projective cone policies, called the MaxProduct policies. It is noted that many scheduling policies can be shown to belong to the class of cone policies in different state spaces; other stability related research for variants of a SPS under different stochastic assumptions can be found in [Andrews et al. 2004; Hung and Michailidis 2006; Ross and Bambos 2005; Stolyar 2004; 2003; Tassiulas and Ephremides 1992].

In this study, we focus on (i) modeling the SPS under fairly mild stochastic assumptions; (ii) constructing scheduling policies (i.e. selecting the service modes over time) to maximize the throughput and achieve strong stability of the system; and (iii) examining the performance of the proposed scheduling policies in terms of the average job sojourn time via computer simulation. The rest of the paper is organized as follows. In Section 2, the stability issue of the system is discussed. In Section 3, two classes of scheduling policies are introduced. The first class is the queue-length driven MaxProduct policy, which at any points in time switches the system into the service mode whose service-rate vector has the maximal inner product with the queue-length vector. The second class is called the Largest Weighted Waiting Time (LWWT) policy, which at any points in time switches the system into the service mode whose service-rate vector has the maximal inner product with the waiting-time vector of head-of-line (HOL) jobs. In Section 4, the proposed scheduling policies are shown to maximize the throughput and maintain strong stability of the system, using drift analysis ([Hajek 1982; Pemantle and Rosenthal 1999]). In Section 5, we model the SPS in continuous time and show that the same policies maximize the throughput and maintain system stability under fairly mild stochastic assumptions. Note that the stability here is a property of the “uniform mean recurrence time” to some compact set, which is different from that defined for the queue length. Owing to the non-Markovian nature of the states, a perturbed Lyapunov function method [Kushner 1967; Kushner and Yin 2003] is used to show the proof. In Section 6, we examine the quality of service (QoS) performance of the proposed policies via computer simulation. The goal is to determine which policy (the MaxProduct or LWWT) one has to use in order to minimize the average job sojourn time. The simulation result reveals that the solution is related to the input load of each queue. For practical purposes, we construct a hybrid scheduling policy (involving the MaxProduct and the LWWT policy) and show that it outperforms (in terms of the average job sojourn time) both the MaxProduct and the LWWT policy when the system has unknown and changing input loads.

## 2. SYSTEM STABILITY

We first focus on the following version of system stability. Let  $N_q(n)$  denote the length of queue  $q$  at time  $n$  by  $N_q(n)$ ; then, if

$$\sup_n E[N_q(n)] < \infty \text{ for all } q \in \{1, 2, \dots, Q\}, \quad (1)$$

the queueing system is characterized as *strongly stable*. Note that the notion of stability defined above (stability-in-the-mean) is stronger than other notions, such as *weak stability* or *rate (pathwise) stability* [Armony and Bambos 2003; Leonaridi et al. 2001; Dai and Lin 2005]. Further, for some systems the distribution of the state process can be shown to converge to a finite stationary process, independent of the initial condition, leading to an even stronger notion of stability. However, it usually requires the strong Markov property of the system [Walrand 1988; Meyn and Tweedie 1993]. For other definitions of stability such as *positive Harris recurrence* under some stronger assumptions, the readers can refer to [Dai 1995; Dai and Meyn 1995].

We define next the *stability region* of the system, which is the maximal set of all possible input rates so that the system can be stabilized under some queueing discipline  $\pi$ . Letting  $\vec{\lambda} = (\lambda_1, \dots, \lambda_Q)$ , it is defined for the system at hand as

$$S = \left\{ \vec{\lambda} \in \mathbb{R}_+^Q : \lambda_q < \sum_{m=1}^M \omega_m \mu_{mq}, \text{ for all } q = 1, \dots, Q \right\} \quad (2)$$

where  $\omega_m$  is interpreted as the long-term proportion that the system is in service mode  $m$ ,  $0 \leq \omega_m \leq 1$ , and  $\sum_{m=1}^M \omega_m \leq 1$ .

It can also be shown that the stability region is the *convex hull* that contains all service vectors  $\vec{\mu}_m$ . The left panel of Fig. 1 shows an example of the stability region  $S$  for a 2-queue system with 4 service modes  $\vec{\mu}_1 = (0, 1)$ ,  $\vec{\mu}_2 = (1/2, 3/4)$ ,  $\vec{\mu}_3 = (3/4, 0)$  and  $\vec{\mu}_4 = (1/4, 1/4)$ . Note that here the service mode  $\vec{\mu}_4$  can be omitted since it is dominated by  $\vec{\mu}_2$ . The right panel of Fig. 1 shows another example of the stability region for a 3-queue system with 6 service modes  $\vec{\mu}_1 = (5/6, 0, 0)$ ,  $\vec{\mu}_2 = (0, 1, 0)$ ,  $\vec{\mu}_3 = (0, 0, 1)$ ,  $\vec{\mu}_4 = (1/3, 1/2, 2/3)$ ,  $\vec{\mu}_5 = (0, 1/2, 5/6)$  and  $\vec{\mu}_6 = (2/3, 1/2, 0)$ .

On the other hand, a queueing system is said to be *unstable* if there exists at least one queue  $q$  such that

$$\liminf_{n \rightarrow \infty} N_q(n) = +\infty \text{ with probability 1.}$$

The instability of the system under consideration is shown in the following theorem.

**THEOREM 2.1.** *If  $\vec{\lambda} \notin S$ , then the queueing system is unstable for every control policy  $\pi$ .*

**PROOF.** The result is obtained in [Armony and Bambos 2003] and therefore is omitted.  $\square$

## 3. THE SCHEDULING POLICY

A scheduling policy is a rule that determines at any point in time which service mode the system has to switch into. We first introduce a scheduling policy that is motivated by the starvation-free algorithm in [Mekkittikul and McKeown 1996]. For the  $n$ -th time slot, we define the following quantities of interest.

(1)  $D_q$ : the set of job departure times in queue  $q$ ,  $q = 1, \dots, Q$ .

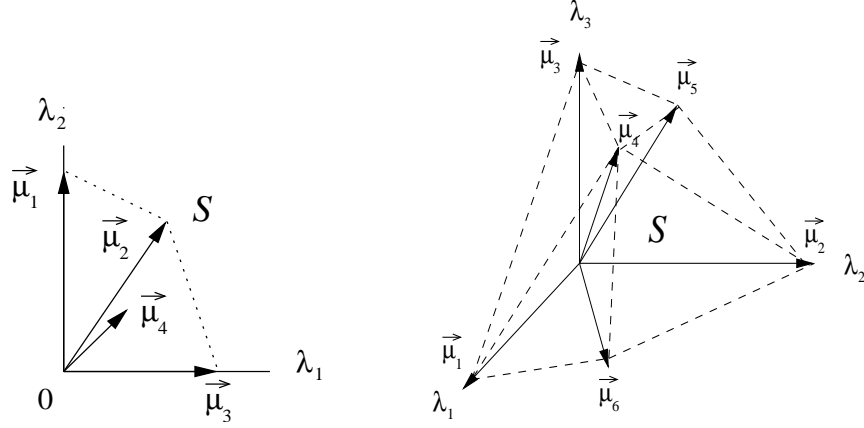


Fig. 1. (Left panel): The stability region for a 2-queue system with 4 service modes. (Right panel): The stability region for a 3-queue system with 6 service modes.

(2)  $Y_q(n)$  : the time that the head-of-the line (HOL) job has spent in the system by time  $n$ , which is typically the sum of the delay and service time.

(3)  $\tau_q(n)$  : the interarrival time between the HOL job of queue  $q$  at time  $n$  and the job behind it.

The evolution of the HOL job waiting time is best illustrated in Fig. 2, where  $C_q(n)$  denotes the HOL job of queue  $q$  at time  $n$ . Note that if  $C_q(n)$  does not leave the queue at time  $n + 1$ , its waiting time increases by one, that is,

$$Y_q(n + 1) = Y_q(n) + 1.$$

If  $C_q(n)$  leaves behind a non-empty queue at time  $n + 1$ , the job behind then becomes the new HOL job. In this case,

$$Y_q(n + 1) = Y_q(n) + 1 - \tau_q(n).$$

If  $C_q(n)$  leaves behind an empty queue at time  $n + 1$ , then we have that

$$Y_q(n + 1) = 0.$$

To summarize, the next-state waiting time for the HOL job of queue  $q$  is denoted by

$$Y_q(n + 1) = [Y_q(n) + 1 - \tau_q(n)I_{\{n+1 \in D_q\}}]^+ . \quad (3)$$

The Largest Weighted Waiting Time (LWWT) policy switches the system into service mode  $m^*$  at time  $n$  if

$$m^* = \arg \max_{m=1, \dots, M} \langle Y(n), \vec{\mu}_m \rangle_{\vec{\alpha}} = \arg \max_{m=1, \dots, M} \sum_{q=1}^Q \alpha_q Y_q(n) \mu_{mq}, \quad (4)$$

where  $Y(n) = (Y_1(n), \dots, Y_Q(n))$ , and  $\vec{\alpha} = (\alpha_1, \dots, \alpha_Q)$  is any positive weight vector chosen for all queues. If the solution of  $m^*$  is not unique, say,  $m^* = m'$ , the system is randomly switched into service mode  $m^*$  or mode  $m'$ . It is clear that the LWWT policy partitions the waiting time space  $\mathcal{Y}$  into exclusive subsets  $C_1, C_2, \dots$ . Specifically, the subset  $C_{m^*}$  is defined by  $C_{m^*} = \{y \in \mathcal{Y} : \arg \max_{m=1, \dots, M} \langle y, \vec{\mu}_m \rangle_{\vec{\alpha}} = m^*\}$ , where

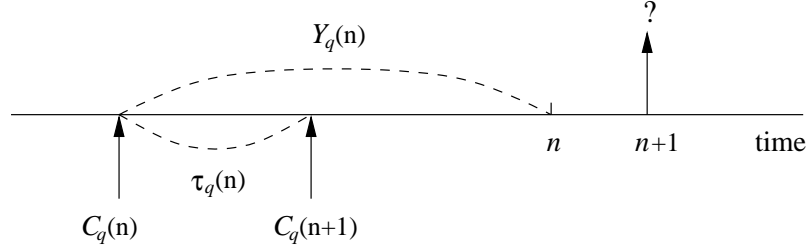


Fig. 2. The time line for the next-state waiting time for the HOL job of queue  $q$ . The arrivals are denoted below the line while the departures are denoted above the line.

by definition each element  $y$  in  $C_{m^*}$  has the maximal projection on  $\vec{\mu}_{m^*}$  (skewed by the weight vector  $\vec{\alpha}$ ) among all admissible service vectors. It is also noted that if  $y \in C_{m^*}$ , then  $ay \in C_{m^*}$  for any constant  $a > 0$ . Further, it can be seen that the boundary between any two subsets is a *hyperplane* through the origin in  $Q$ -dimensional space. These facts indicate that  $C_1, C_2, \dots$  are *polyhedral cones* in the waiting time space; thus, LWWT belongs to the class of so-called *cone policies* introduced in [Armony and Bambos 2003]. When  $Y(n)$  is in a certain cone  $C_m$ , the system is switched into the service mode associated with it (i.e.  $\vec{\mu}_m$ ). An example of such cones for a 2-queue system with 3 service modes  $\vec{\mu}_1 = (0, 1)$ ,  $\vec{\mu}_2 = (1/2, 3/4)$ , and  $\vec{\mu}_3 = (3/4, 0)$ , is shown in Fig. 3. If we replace the HOL job waiting

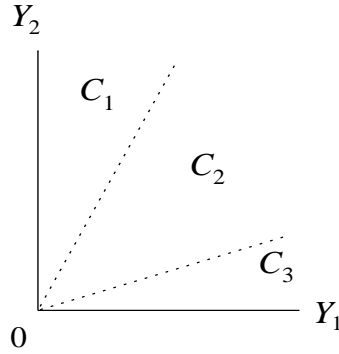


Fig. 3. The cone partition of the waiting time space  $\mathcal{Y}$  under the LWWT policy for a 2-queue system with service modes  $\vec{\mu}_1 = (0, 1)$ ,  $\vec{\mu}_2 = (1/2, 3/4)$ , and  $\vec{\mu}_3 = (3/4, 0)$ . Here all queue weights are equally placed (i.e.  $\vec{\alpha} = \vec{1}$ ).

time  $Y_q(n)$  by the queue length  $N_q(n)$  in (4), the policy then becomes the so-called (queue-length driven) MaxProduct policy introduced in [Armony and Bambos 2003]. Although the LWWT policy and the MaxProduct policy have similar cone structures in the state space, their sample paths turn out to be quite different. To illustrate, let's consider a 2-queue system and assume both queues are non-empty at time  $n$ . If there are no departures at time  $n + 1$ , it is clear that  $(Y_1(n + 1), Y_2(n + 1)) = (Y_1(n) + 1, Y_2(n) + 1)$ . That is,  $Y(n)$  always goes at a 45-degree angle pointing away from the current state until a departure happens. However, whenever a departure occurs,  $Y(n)$  will jump and probably

jump to a decision cone that is far away from the current one. This intuitively implies that sample paths under the LWWT policy are sensitive to job departures. On the other hand, the variation of sample paths under the (queue-length driven) MaxProduct policy is relatively small for this type of stationary process, since at each time slot the queue length will increase or decrease only by one.

#### 4. STABILITY RESULTS IN SLOTTED TIME

We start by examining the stability issue in slotted time with the SPS fed by independent Bernoulli processes and independent geometric service times, as described in Section 1. Our goal is to show that both the LWWT and the MaxProduct policy maximize the throughput and maintain strong stability (as defined in (1)) of the system. The result is established using *drift analysis* [Hajek 1982; Pemantle and Rosenthal 1999]. In subsequent developments, the strong assumptions of independent Bernoulli processes for the arrivals and departures are relaxed; nevertheless, the slotted time setting provides strong insight into the proof under milder stochastic assumptions, since many steps are very similar.

The following Lyapunov function is considered.

$$V(Y(n)) = \sum_{q=1}^Q \alpha_q \lambda_q Y_q^2(n). \quad (5)$$

Let  $\mathcal{F}_n$  denote the  $\sigma$ -field that contains all system information up to time  $n$  and let  $E_n$  denote the expected value conditioned on  $\mathcal{F}_n$ . We establish the following two lemmas.

**LEMMA 4.1.** *Under the LWWT policy, there exist some  $\varepsilon > 0$  and  $b > 0$  such that, for all  $\vec{\lambda} \in S$ , if  $V(Y(n)) > b$  then*

$$E_n[V(Y(n+1)) - V(Y(n))] \leq -\varepsilon. \quad (6)$$

**PROOF.** Here we use the approximate next-state for the HOL job waiting time

$$\tilde{Y}_q(n+1) = Y_q(n) + 1 - \tau_q(n)I_{\{n+1 \in D_q\}}. \quad (7)$$

Since  $E_n[V(Y(n+1)) - V(Y(n))] \leq E_n[V(\tilde{Y}(n+1)) - V(Y(n))]$ , it suffices to show that  $E_n[V(\tilde{Y}(n+1)) - V(Y(n))] \leq -\varepsilon$ . Let's start without considering the condition  $V(Y(n)) > b$ , so we have that

$$E_n[V(\tilde{Y}(n+1)) - V(Y(n))] = E_n \left[ \sum_{q=1}^Q \alpha_q \lambda_q \tilde{Y}_q^2(n+1) - \sum_{q=1}^Q \alpha_q \lambda_q Y_q^2(n) \right]. \quad (8)$$

Replacing  $\tilde{Y}_q(n+1)$  by (7) and expanding the above term yields

$$\begin{aligned} E_n[V(\tilde{Y}(n+1)) - V(Y(n))] &= 2 \sum_{q=1}^Q \alpha_q \lambda_q Y_q(n) E_n [1 - \tau_q(n)I_{\{n+1 \in D_q\}}] \\ &\quad + \sum_{q=1}^Q \alpha_q \lambda_q E_n [(1 - \tau_q(n)I_{\{n+1 \in D_q\}})^2]. \end{aligned} \quad (9)$$

Note that the last expression can be easily bounded since

$$E_n [(1 - \tau_q(n)I_{\{n+1 \in D_q\}})^2] \leq E_n [(1 + \tau_q(n))^2] \text{ for all } q, \quad (10)$$

and  $E_n[\tau_q^2(n)] = (2/\lambda_q^2 - 1/\lambda_q)$  directly from the assumption of geometric distribution. Thus, there exists an  $A > 0$  such that

$$E_n[V(\tilde{Y}(n+1)) - V(Y(n))] \leq 2 \sum_{q=1}^Q \alpha_q \lambda_q Y_q(n) E_n [1 - \tau_q(n) I_{\{n+1 \in D_q\}}] + A. \quad (11)$$

Since we assume that all arrival processes and departure processes are mutually independent, it is clear that

$$E_n [\tau_q(n) I_{\{n+1 \in D_q\}}] = \mu_{m^*q} / \lambda_q, \quad (12)$$

where  $m^* = \arg \max_{m=1, \dots, M} \sum_{q=1}^Q \alpha_q Y_q(n) \mu_{mq}$  by the LWWT policy. Expanding the bracketed term in (11) yields

$$E_n[V(\tilde{Y}(n+1)) - V(Y(n))] \leq A + 2 \left\{ \sum_{q=1}^Q \alpha_q Y_q(n) \lambda_q - \sum_{q=1}^Q \alpha_q Y_q(n) \mu_{m^*q} \right\}. \quad (13)$$

Note that for any given  $\vec{\lambda} \in S$ , there exist  $\omega_1, \dots, \omega_M$  satisfying  $0 \leq \omega_m \leq 1$  for all  $m$  and  $\sum_{m=1}^M \omega_m \leq 1$ , and  $c > 0$  such that  $\lambda_q - \sum_{m=1}^M \omega_m \mu_{mq} \leq -c$  for all  $q = 1, \dots, Q$ . The last expression in (13) can be easily bounded by

$$\sum_{q=1}^Q \alpha_q Y_q(n) \mu_{m^*q} \geq \sum_{q=1}^Q \alpha_q Y_q(n) \mu_{m^*q} \left( \sum_{m=1}^M \omega_m \right) \geq \sum_{q=1}^Q \sum_{m=1}^M \omega_m \alpha_q Y_q(n) \mu_{mq}. \quad (14)$$

Therefore, (13) can be expressed as

$$E_n[V(\tilde{Y}(n+1)) - V(Y(n))] \leq A + 2 \left\{ \sum_{q=1}^Q \alpha_q Y_q(n) [\lambda_q - \sum_{m=1}^M \omega_m \mu_{mq}] \right\}. \quad (15)$$

Denote  $q^* = \arg \max_q \{Y_q(n)\}$  and  $a = \max_q \{\alpha_q \lambda_q\}$ , the condition  $V(Y(n)) > b$  directly implies that  $Y_{q^*}(n) > \sqrt{\frac{b}{aQ}}$ . Therefore, when  $V(Y(n)) > b$ , (15) can be expressed as

$$\begin{aligned} & E_n[V(\tilde{Y}(n+1)) - V(Y(n))] \\ & \leq A + 2\alpha_{q^*} Y_{q^*}(n) [\lambda_{q^*} - \sum_{m=1}^M \omega_m \mu_{mq^*}] \leq -2c\alpha_{q^*} \sqrt{\frac{b}{aQ}} + A = -\varepsilon(b). \end{aligned} \quad (16)$$

Thus, we can always choose  $b$  to be large enough so that  $\varepsilon(b) > 0$  and  $E_n[V(\tilde{Y}(n+1)) - V(Y(n))] \leq -\varepsilon(b)$  when  $V(Y(n)) > b$ .  $\square$

LEMMA 4.2. *Under the LWWT policy, there exist some  $p > 2$  and  $D > 0$  such that*

$$E_n |V(Y(n+1)) - V(Y(n))|^p \leq D. \quad (17)$$

PROOF. First,

$$\begin{aligned} E_n |V(Y(n+1)) - V(Y(n))|^p &= E_n \left| \sum_{q=1}^Q \alpha_q \lambda_q [Y_q^2(n+1) - Y_q^2(n)] \right|^p \\ &\leq E_n \left| \sum_{q=1}^Q \alpha_q \lambda_q [Y_q(n+1) + Y_q(n)]^2 \right|^p. \end{aligned} \quad (18)$$

Replacing  $Y_q(n+1)$  in the last expression by (3),

$$E_n |V(Y(n+1)) - V(Y(n))|^p \leq E_n \left| \sum_{q=1}^Q \alpha_q \lambda_q [2Y_q(n) + 1 + \tau_q(n)]^2 \right|^p. \quad (19)$$

Note that by expanding the bracketed term, the last expression is easily bounded by the expected value of a polynomial of  $\tau_q(n)$  of order  $2p$ . Since all  $\tau_q(n)$  have independent geometric distributions, it is clear that they have finite moments of all orders. Therefore, there exists a finite  $D > 0$  such that (17) holds.  $\square$

**PROPOSITION 4.3.** *Under the LWWT policy, the queueing system is strongly stable for all  $\vec{\lambda} \in S$ .*

**PROOF.** According to the result in [Pemantle and Rosenthal 1999], LEMMA 4.1 and LEMMA 4.2 imply that the job waiting times are stable. We now show that this also implies that queue length occupancies are stable. Consider an arbitrary nonempty queue  $q$  at time  $n$  and denote the arrival time of its HOL job by  $m$ ,  $m \leq n$ . Thus,  $Y_q(n) = n - m$  if  $n > m$  and 0 otherwise. Since time is slotted, the queue length occupancy must satisfy that  $N_q(n) \leq n - m + 1$ . So bounded waiting times in the system imply bounded queue length occupancies, which shows that queue length occupancies are stable as well.  $\square$

Note that the (queue-length driven) MaxProduct policy chooses the service mode  $m^*$  at time  $n$  if

$$m^* = \arg \max_{m=1, \dots, M} \langle N(n), \vec{\mu}_m \rangle_{\vec{\alpha}} = \arg \max_{m=1, \dots, M} \sum_{q=1}^Q \alpha_q N_q(n) \mu_{mq}, \quad (20)$$

where  $N(n) = (N_1(n), \dots, N_Q(n))$  denotes the queue-length vector at time  $n$ , and  $\vec{\alpha} = (\alpha_1, \dots, \alpha_Q)$  is any positive weight vector chosen for all queues. We next show that the system is stable under the MaxProduct policy.

**THEOREM 4.4.** *Under the MaxProduct policy, the queueing system is strongly stable for all  $\vec{\lambda} \in S$ .*

**PROOF.** Although the proof is quite similar to that of the LWWT policy, we show it for the sake of completeness. We first prove that LEMMA 4.1 is true under the MaxProduct policy. Let us consider the approximate next-state for the queue length

$$\tilde{N}_q(n+1) = N_q(n) + A_q(n+1) - D_q(n+1) \quad (21)$$

where  $A_q(n+1)$  and  $D_q(n+1)$  are independent Bernoulli random variables denoting the number of arrivals at queue  $q$  and the number of departures from queue  $q$  at time  $n+1$ , respectively. Choosing an alternative Lyapunov function  $V(N(n)) = \sum_{q=1}^Q \alpha_q N_q^2(n)$ , an equation similar to (9) yields

$$\begin{aligned} E_n [V(\tilde{N}(n+1)) - V(N(n))] &= 2 \sum_{q=1}^Q \alpha_q N_q(n) E_n [A_q(n+1) - D_q(n+1)] \\ &\quad + \sum_{q=1}^Q \alpha_q E_n [A_q(n+1) - D_q(n+1)]^2, \end{aligned} \quad (22)$$

where the last expression is simply bounded by  $\sum_{q=1}^Q \alpha_q$ . Let  $B = \sum_{q=1}^Q \alpha_q$  and  $m^* = \arg \max_{m=1, \dots, M} < N(n), \bar{\mu}_m >_{\bar{\alpha}}$ , we then have

$$E_n[V(\tilde{N}(n+1)) - V(N(n))] \leq B + 2 \left\{ \sum_{q=1}^Q \alpha_q N_q(n) \lambda_q - \sum_{q=1}^Q \alpha_q N_q(n) \mu_{m^*q} \right\}. \quad (23)$$

Analogously, let's denote  $q^* = \arg \max_q \{N_q(n)\}$  and  $a = \max_q \{\alpha_q\}$ . Given that  $V(N(n)) > b$ , an inequality similar to (16) yields

$$E_n[V(\tilde{N}(n+1)) - V(N(n))] \leq -2c\alpha_{q^*} \sqrt{\frac{b}{aQ}} + B. \quad (24)$$

Thus, we can always choose  $b$  to be large enough so that the negative drift (similar to (6)) is obtained. Now we prove that LEMMA 4.2 is true under the MaxProduct policy. Following similar lines in the proof of LEMMA 4.2, it can be shown that

$$E_n |V(N(n+1)) - V(N(n))|^p \leq E_n \left| \sum_{q=1}^Q \alpha_q [2N_q(n) + 1]^2 \right|^p, \quad (25)$$

where the term on the right hand side is clearly a constant for any given  $p$ . Again, according to the result by Pemantle and Rosenthal, the queueing system is stable under the MaxProduct policy.  $\square$

## 5. STABILITY RESULTS FOR SPS UNDER RELAXED STOCHASTIC ASSUMPTIONS

We now consider SPS in continuous time under significantly relaxed stochastic assumptions. Specifically, let  $\mathcal{A}_q$  be the input process of queue  $q$  marked by a sequence of job interarrival times  $\tau_q^1, \tau_q^2, \dots$ . Define the mean input rate of each queue by

$$\lambda_q = \lim_{N \rightarrow \infty} \frac{N}{\sum_{i=1}^N \tau_q^i}, \quad q = 1, \dots, Q. \quad (26)$$

When the system is in service mode  $m$ , denote the sequence of service times for the jobs of queue  $q$  by  $S_{mq}^1, S_{mq}^2, \dots$  and define its mean service rate by

$$\mu_{mq} = \lim_{N \rightarrow \infty} \frac{N}{\sum_{i=1}^N S_{mq}^i}, \quad m = 1, \dots, M, \quad q = 1, \dots, Q. \quad (27)$$

Suppose now the system conditions are weakened so that (i) the job interarrival times and service times can be correlated; and (ii) all input processes can be interdependent. These assumption lead to non-Markovian dynamics for the system and thus the techniques used in the previous sections are not directly applicable. Thus, a more general framework is required, provided by the perturbed Lyapunov function method [Kushner 1967; Kushner and Yin 2003]. We start by introducing some additional necessary assumptions for future developments.

### 5.1 Assumptions and Stability

*Assumption 5.1.1.* There exists some  $0 < A < \infty$  such that  $E[\tau_q^2] \leq A$  for all  $q = 1, \dots, Q$ , where  $\tau_q$  denotes the random variable of job interarrival times in queue  $q$ .

*Assumption 5.1.2.* There exists some  $0 < U < \infty$  such that  $E[S_{mq}^2] \leq U$  for all  $m = 1, \dots, M$ ,  $q = 1, \dots, Q$ , where  $S_{mq}$  denotes the random variable of job service times in queue  $q$  when the system is in service mode  $m$ .

Recall that the set of all job departure times in queue  $q$  is denoted by  $D_q$ . Let us denote also the set of all possible service-mode switch times by  $\mathcal{T}$  and consider the sequence  $\{t_1, t_2, \dots\} = \cup_{q=1}^Q D_q \cup \mathcal{T}$ . This implies that the system will remain in the same service mode over each time interval  $[t_n, t_{n+1})$ . Let  $I_j(n)$  be an indicator random variable so that it has the value one if the system is in service mode  $j$  at time  $t_n$  and zero otherwise. Obviously,  $I_j(n)$  is determined by the scheduling policy. Here we focus on the *preemptive-resume* scheduling discipline. That is, jobs can be preempted and resumed without loss of work (i.e. the remaining service time of the HOL job is recorded when it is preempted). Although such a scheduling discipline requires more (memory) capacity to store the system information, it provides instantaneous and more flexible resource allocations so that the targeted service-level commitments can be met. For any given input processes, let  $r_j$  denote the long-term proportion that the system is in service mode  $j$  under the LWWT policy.

*Assumption 5.1.3.* There exists a function  $\rho(k)$  which goes to zero as  $k \rightarrow \infty$  such that under the LWWT policy,

$$E_n \left| \sum_{j=1}^M I_j(k) \mu_{jq} - \sum_{j=1}^M r_j \mu_{jq} \right| \leq \rho(k-n) \quad (28)$$

for all  $q = 1, \dots, Q$  and  $k \geq n$ .

This assumption says that the conditional mean service rate received by queue  $q$  at time  $t_k$  given all system information to time  $t_n$  converges to the long-term service rate as  $k - n \rightarrow \infty$ . Similarly, let's denote  $\tau_q(n)$  to be the interarrival time between the HOL job of queue  $q$  at time  $t_n$  and the job behind it.

*Assumption 5.1.4.* Under the LWWT policy,

$$E_n \left| \lambda_q \tau_q(k) \frac{I_{\{t_{k+1} \in D_q\}}}{E_n[t_{k+1} - t_k]} - \sum_{j=1}^M r_j \mu_{jq} \right| \leq \rho(k-n) \quad (29)$$

for all  $q = 1, \dots, Q$  and  $k \geq n$ .

This is simply a mixing condition on the data arrival and service processes. Suppose the job interarrival times of each queue  $q$  are such that  $E_n[\tau_q(k)]$  converges uniformly to  $1/\lambda_q$  and the conditional expected number of jobs processed for one time unit (i.e.  $E_n I_{\{t_{k+1} \in D_q\}}/E_n[t_{k+1} - t_k]$ ) converges uniformly to the long-term average service rate  $\sum_{j=1}^M r_j \mu_{jq}$ , it is clear that (29) is true when the interarrival times are independent of the departure times. However, *Assumption 5.1.4* corresponds to a more general condition on the arrival and service processes - it allows the interdependence (whatever its nature may be) between job interarrival times, job service times, and input traffic flows. When the future becomes less and less predictable, it is assumed that the conditional likelihood of such interdependence gets weaker.

Note that the stability region is the same as that was shown in (2), while now system stability is defined via the ‘‘uniform mean recurrence time’’ property [Kushner and Yin

2003]. That is, the system is said to be stable if there exist  $y_0 > 0$  and nonnegative real-valued function  $F(\cdot)$  such that for any  $n$  and  $s = \min\{t \geq t_n : |Y(t)| \leq y_0\}$ ,

$$E_n[s - t_n] \leq F(Y(n)) \text{ when } |Y(n)| \geq y_0. \quad (30)$$

This definition implies that when  $|Y(n)|$  reaches a level  $y_n > y_0$ , the conditional expectation of the time required to return to the value  $y_n$  (or smaller) is bounded above by a function of  $y_n$ , uniformly in  $n$  and in the past history. Note that if there are no job departures after time  $t_n$ ,  $Y(t)$  keeps moving away from  $y_0$  under the LWWT policy (see discussions at the end of Section 3). Therefore,  $s$  must be job departure times and thus belonging to the set  $\{t_1, t_2, \dots\}$ . We next introduce the Lyapunov function perturbations used to show system stability.

## 5.2 Perturbed Lyapunov Function

With the perturbed Lyapunov function method, one starts with a standard Lyapunov function  $V(Y(n))$  and then incorporates a perturbation  $\delta V(n)$  into  $V(Y(n))$  so that  $V(Y(n)) + \delta V(n)$  is used to show the desired stability of the non-Markovian system. Note that here  $\delta V(n)$  will be a sum of two terms, one corresponding to each departure process and the other corresponding to the mixing of arrival and departure processes. The design of its structure is basically motivated by the way it is used in the proof. For additional background and its applications, the readers can refer to [Kushner 1984; Kushner and Yin 2003].

We first define the ‘‘departure’’ Lyapunov function perturbation. For each queue  $q$ , let

$$\delta V_q^d(n) = 2\alpha_q Y_q(n) \sum_{k=n}^{n+h-1} E_n[t_{k+1} - t_k] E_n \left[ \sum_{j=1}^M I_j(k) \mu_{jq} - \sum_{j=1}^M r_j \mu_{jq} \right] \quad (31)$$

where the value of  $h$  will be chosen in the proof shown later. We next define the Lyapunov function perturbation for mixing the ‘‘arrival’’ and ‘‘departure’’ processes. For each queue  $q$ , let

$$\begin{aligned} \delta V_q^{mix}(n) = & \\ & -2\alpha_q \lambda_q Y_q(n) \sum_{k=n}^{n+h-1} E_n[t_{k+1} - t_k] E_n \left[ \tau_q(k) \frac{I_{\{t_{k+1} \in D_q\}}}{E_n[t_{k+1} - t_k]} - \frac{1}{\lambda_q} \sum_{j=1}^M r_j \mu_{jq} \right]. \end{aligned} \quad (32)$$

The full Lyapunov function perturbation  $\delta V(n)$  and the time-dependent Lyapunov function  $\tilde{V}(n)$  are then given by

$$\begin{aligned} \delta V(n) &= \sum_{q=1}^Q \delta V_q^d(n) + \sum_{q=1}^Q \delta V_q^{mix}(n) \text{ and} \\ \tilde{V}(n) &= V(Y(n)) + \delta V(n), \end{aligned} \quad (33)$$

where  $V(Y(n)) = \sum_{q=1}^Q \alpha_q \lambda_q Y_q^2(n)$ , the same as shown in (5).

**THEOREM 5.2.1.** *Under the LWWT policy and Assumption 5.1.1 - Assumption 5.1.4, the system is stable for all  $\vec{\lambda} \in S$ .*

PROOF. We will show that there exists a  $c > 0$  such that  $E_n[\tilde{V}(n+1) - \tilde{V}(n)] \leq -c$  when  $|Y(n)|$  is large enough. This inequality together with the bounds on  $\delta V(n)$  will then imply (30). Note that

$$\begin{aligned} E_n[\tilde{V}(n+1) - \tilde{V}(n)] &= E_n[V(Y(n+1)) - V(Y(n))] \\ &\quad + \sum_{q=1}^Q E_n[\delta V_q^d(n+1) - \delta V_q^d(n)] + \sum_{q=1}^Q E_n[\delta V_q^{mix}(n+1) - \delta V_q^{mix}(n)] \end{aligned}$$

and we will evaluate it term by term. Follow the lines in the proof of LEMMA 4.1, an inequality similar to (11) yields

$$\begin{aligned} E_n[V(Y(n+1)) - V(Y(n))] &\leq \\ &\quad 2 \sum_{q=1}^Q \alpha_q \lambda_q Y_q(n) E_n [t_{n+1} - t_n - \tau_q(n) I_{\{t_{n+1} \in D_q\}}] + B, \end{aligned} \quad (34)$$

where  $B > 0$  is a constant obtained from Assumption 5.1.1 and Assumption 5.1.2. Expanding the above bracketed term yields

$$\begin{aligned} E_n[V(Y(n+1)) - V(Y(n))] &\leq \\ &\quad 2E_n[t_{n+1} - t_n] \sum_{q=1}^Q \alpha_q \lambda_q Y_q(n) - 2 \sum_{q=1}^Q \alpha_q \lambda_q Y_q(n) E_n [\tau_q(n) I_{\{t_{n+1} \in D_q\}}] + B. \end{aligned} \quad (35)$$

Now we consider the ‘‘departure’’ perturbation term:

$$\begin{aligned} E_n[\delta V_q^d(n+1) - \delta V_q^d(n)] &= \\ &\quad 2\alpha_q E_n[Y_q(n+1)] \sum_{k=n+1}^{n+h} E_{n+1}[t_{k+1} - t_k] E_{n+1} \left[ \sum_{j=1}^M I_j(k) \mu_{jq} - \sum_{j=1}^M r_j \mu_{jq} \right] \\ &\quad - 2\alpha_q Y_q(n) \sum_{k=n}^{n+h-1} E_n[t_{k+1} - t_k] E_n \left[ \sum_{j=1}^M I_j(k) \mu_{jq} - \sum_{j=1}^M r_j \mu_{jq} \right]. \end{aligned} \quad (36)$$

This expression can be written as

$$\begin{aligned} &- 2\alpha_q Y_q(n) E_n[t_{n+1} - t_n] \left( \sum_{j=1}^M I_j(n) \mu_{jq} - \sum_{j=1}^M r_j \mu_{jq} \right) \\ &+ 2\alpha_q E_n[Y_q(n+1)] \sum_{k=n+1}^{n+h} E_{n+1}[t_{k+1} - t_k] E_{n+1} \left[ \sum_{j=1}^M I_j(k) \mu_{jq} - \sum_{j=1}^M r_j \mu_{jq} \right] \\ &- 2\alpha_q Y_q(n) \sum_{k=n+1}^{n+h-1} E_n[t_{k+1} - t_k] E_n \left[ \sum_{j=1}^M I_j(k) \mu_{jq} - \sum_{j=1}^M r_j \mu_{jq} \right]. \end{aligned} \quad (37)$$

From (3) we know that  $Y_q(n+1) = [Y_q(n) + t_{n+1} - t_n - \tau_q(n) I_{\{t_{n+1} \in D_q\}}]^+$ , this implies that  $E_n[Y_q(n+1)]$  is easily bounded above by  $Y_q(n) + E_n[t_{n+1} - t_n] + E_n[\tau_q(n)]$ . Further, by letting  $\mu_{min} = \min_{j,q} \{\mu_{jq}\}$  we know that  $E_n[t_{k+1} - t_k] \leq 1/\mu_{min}$  for all  $k \geq n$ ,

and by *Assumption 5.1.1* we know that there exists an  $A' > 0$  such that  $E_n[\tau_q(n)] \leq A'$ . Adding these results to (37) yields

$$E_n[\delta V_q^d(n+1) - \delta V_q^d(n)] = -2\alpha_q Y_q(n) E_n[t_{n+1} - t_n] \left( \sum_{j=1}^M I_j(n) \mu_{jq} - \sum_{j=1}^M r_j \mu_{jq} \right) + \varepsilon_q^d, \quad (38)$$

where  $\varepsilon_q^d$  is bounded above by  $2(\alpha_q/\mu_{min})Y_q(n)\rho(h) + hC_q^d$  for some constant  $C_q^d > 0$ . Analogously, we can show that

$$E_n[\delta V_q^{mix}(n+1) - \delta V_q^{mix}(n)] = 2\alpha_q Y_q(n) E_n[t_{n+1} - t_n] E_n \left[ \lambda_q \tau_q(n) \frac{I_{\{t_{n+1} \in D_q\}}}{E_n[t_{n+1} - t_n]} - \sum_{j=1}^M r_j \mu_{jq} \right] + \varepsilon_q^{mix}, \quad (39)$$

where  $\varepsilon_q^{mix}$  is bounded above by  $2(\alpha_q/\mu_{min})Y_q(n)\rho(h) + hC_q^{mix}$  for some constant  $C_q^{mix} > 0$ .

Adding all terms in (35), (38), (39), and canceling where possible yields

$$E_n[\tilde{V}(n+1) - \tilde{V}(n)] = 2E_n[t_{n+1} - t_n] \left( \sum_{q=1}^Q \alpha_q \lambda_q Y_q(n) - \sum_{q=1}^Q \alpha_q Y_q(n) \sum_{j=1}^M I_j(n) \mu_{jq} \right) + \varepsilon, \quad (40)$$

where  $\varepsilon$  is bounded above by

$$\rho(h) \frac{4}{\mu_{min}} \sum_{q=1}^Q \alpha_q Y_q(n) + h \left[ \sum_{q=1}^Q (C_q^d + C_q^{mix}) \right] + B. \quad (41)$$

As described, for any given  $\vec{\lambda} \in S$  there exist  $\omega_1, \dots, \omega_M$  satisfying  $0 \leq \omega_j \leq 1$  for all  $j$  and  $\sum_{j=1}^M \omega_j \leq 1$ , and  $c_0 > 0$  such that  $\lambda_q - \sum_{j=1}^M \omega_j \mu_{jq} \leq -c_0$  for all  $q = 1, \dots, Q$ . Let  $j^* = \arg \max_{j=1, \dots, M} \sum_{q=1}^Q \alpha_q Y_q(n) \mu_{jq}$ , which is the service mode chosen by the LWWT policy at time  $t_n$ . The last expression in the bracketed term of (40) then has the bound:

$$\begin{aligned} \sum_{q=1}^Q \alpha_q Y_q(n) \sum_{j=1}^M I_j(n) \mu_{jq} &= \sum_{q=1}^Q \alpha_q Y_q(n) \mu_{j^*q} \geq \sum_{q=1}^Q \alpha_q Y_q(n) \mu_{j^*q} \left( \sum_{j=1}^M \omega_j \right) \\ &= \sum_{q=1}^Q \sum_{j=1}^M \omega_j \alpha_q Y_q(n) \mu_{j^*q} \geq \sum_{q=1}^Q \sum_{j=1}^M \omega_j \alpha_q Y_q(n) \mu_{jq}. \end{aligned} \quad (42)$$

Adding these results to (40) together with (41) yields

$$\begin{aligned} E_n[\tilde{V}(n+1) - \tilde{V}(n)] &\leq 2E_n[t_{n+1} - t_n] \sum_{q=1}^Q \alpha_q Y_q(n) \left( \lambda_q - \sum_{j=1}^M \omega_j \mu_{jq} \right) + \varepsilon \\ &\leq -2 \frac{c_0}{\mu_{max}} \sum_{q=1}^Q \alpha_q Y_q(n) + \rho(h) \frac{4}{\mu_{min}} \sum_{q=1}^Q \alpha_q Y_q(n) + hC + B, \end{aligned} \quad (43)$$

where  $\mu_{max} = \max_{j,q} \{\mu_{jq}\}$  and  $C = \sum_{q=1}^Q (C_q^d + C_q^{mix})$ . Note that we can select the integer  $h$  (to be large enough) so that  $\rho(h)(4/\mu_{min}) \leq (c_0/\mu_{max})$ . Since now  $h$  is fixed, (43) can be written as

$$E_n[\tilde{V}(n+1) - \tilde{V}(n)] \leq -(c_0/\mu_{max}) \sum_{q=1}^Q \alpha_q Y_q(n) + c_1 \quad (44)$$

for some constant  $c_1 > 0$ . By (44), it is clear that  $E_n[\tilde{V}(n+1) - \tilde{V}(n)] \rightarrow -\infty$  as  $Y(n) \rightarrow \infty$ . Thus, there exist  $c > 0$  and  $y_0 > 0$  such that when  $|Y(n)| \geq y_0$ ,  $E_n[\tilde{V}(n+1) - \tilde{V}(n)] \leq -c$ . Since we also know that  $|\delta V(n)| = O(|Y(n)|)$ , given small  $\delta > 0$ , this implies that for sufficiently large  $y_0$ ,  $|\delta V(n)| = |V(Y(n)) - \tilde{V}(n)| \leq \delta[1 + V(Y(n))]$ . Let  $t_n$  be such that  $|Y(n)| > y_0$  and  $t_\sigma = \min\{t_i \geq t_n : |Y(i)| \leq y_0\}$ . We then have that  $E_n[\tilde{V}(\sigma) - \tilde{V}(n)] \leq -cE_n[t_\sigma - t_n]$ , which implies

$$\begin{aligned} -\delta E_n[1 + V(Y(\sigma))] + E_n V(Y(\sigma)) &\leq E_n \tilde{V}(\sigma) \\ &\leq -cE_n[t_\sigma - t_n] + \delta + V(Y(n))(1 + \delta), \end{aligned} \quad (45)$$

and thus

$$E_n[t_\sigma - t_n] \leq \frac{1}{c} [2\delta + V(Y(n))(1 + \delta) + \delta E_n V(Y(\sigma))]. \quad (46)$$

Since  $V(Y(\sigma)) \leq \sup_{|y| \leq y_0} V(y)$ , this then implies the stability defined in (30).  $\square$

*Remark 5.2.2.* Under the MaxProduct policy, the stability of the system can be shown in a similar fashion. The details of the proof are omitted.

## 6. THE AVERAGE SOJOURN TIME - A SIMULATION STUDY

So far, we have established the throughput maximizing property of two scheduling policies for the system under consideration. In this section, we investigate their performance with respect to the average sojourn time metric through a simulation study, since analytic results are hard to obtain due to the complex dynamics of SP systems. The work done in [Stolyar 2003] is an exception, it deals with the tail probability of delays under the largest weighted delay first (LWDF) discipline in a heavy traffic regime, where the input intensity is equal to the system's processing capacity.

In this section, 2- and 3-queue systems are simulated in slotted and continuous time fed by various types of input processes. Further, a hybrid scheduling policy is introduced and its performance examined through simulation of a 3-queue system with changing Poisson input processes.

In order to establish the average sojourn time of the system, each simulation run requires the collection of a large number of events (job arrivals/departures). It is clear that there is a burn-in period that would introduce bias. Hence, in order to eliminate such bias every simulation run was divided in two phases, the initialization phase and the steady state one when actual data collection occurs (see [Banks et al. 1999]). Estimation of the length of the initialization phase was based on the methods of *independent replications* and *ensemble averages* [Banks et al. 1999].

### 6.1 System 1

Consider a 2-queue system where jobs of each queue arrive according to an i.i.d. Bernoulli process. There are 3 normalized service rate vectors  $\vec{\mu}_1 = (0, 0.8)$ ,  $\vec{\mu}_2 = (0.4, 0.6)$ ,

and  $\vec{\mu}_3 = (0.6, 0)$ , with each vector corresponding to a combination of i.i.d. Bernoulli processes. This constitutes the stability region for the normalized input rates  $(\lambda_1, \lambda_2)$ , as shown in the left panel of Fig. 4. In order to obtain a thorough understanding about the average-sojourn-time performance under the LWWT and the MaxProduct policies, we superimpose on the stability region a regular grid of sufficient density and then simulate the system at all the grid points  $(\lambda_1, \lambda_2)$ . For comparison purposes, all queue weights here are chosen to be the same, say,  $\vec{\alpha} = \vec{1}$  in both policies. The simulation result is shown in the right panel of Fig. 4.

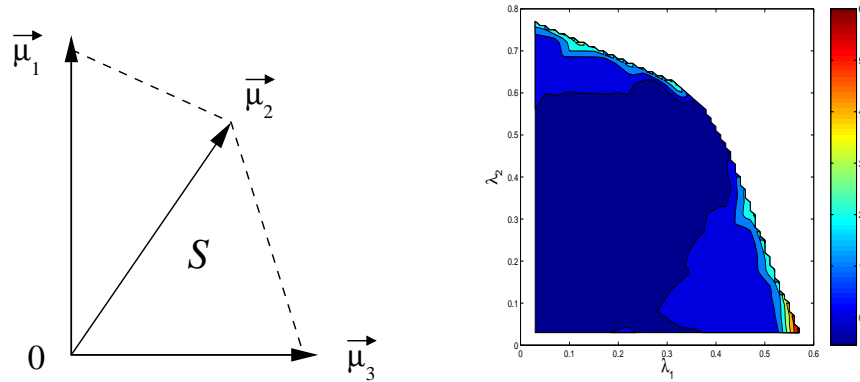


Fig. 4. (Left panel): The stability region of  $(\lambda_1, \lambda_2)$  for the system with 3 service vectors  $\vec{\mu}_1 = (0, 0.8)$ ,  $\vec{\mu}_2 = (0.4, 0.6)$ , and  $\vec{\mu}_3 = (0.6, 0)$ . (Right panel): The contour plot by taking the differences of average sojourn times for the two policies (MaxProduct-LWWT) over the stability region.

The right panel of Fig. 4 shows that, when the input rates belong to the upper-left and lower-right areas of the stability region (i.e. one of the classes experiences a very high loading, while the remaining class a fairly low one), the LWWT policy performs better than the MaxProduct policy in terms of the average job sojourn time. The numerical result reveals that, even for such a small system, the improvement by the LWWT policy can reach 20% when the system is heavily loaded. We explain this empirical finding next. When the input rates belong to the upper-left and lower-right area of the stability region, it is intuitive that the MaxProduct policy uses the service vector  $\vec{\mu}_1$  and  $\vec{\mu}_3$  (respectively) most of the time. However, this is not as efficient as using the service vector  $\vec{\mu}_2$  (which has a larger service rate  $0.4 + 0.6 = 1$  in total) when both queues are not empty. On the other hand, the LWWT policy is more likely to choose the service vector  $\vec{\mu}_2$  due to a larger variation of its sample path (as discussed in Section 4), thus leading to a significantly smaller average job sojourn time. When the input rates are close to the middle area of the stability region, it can be seen that the MaxProduct policy performs slightly better than the LWWT policy. The reason is that both queues are rarely empty (when the system is heavily loaded) and the MaxProduct policy uses the most efficient service vector  $\vec{\mu}_2$  most of the time. It is noted that when the system is lightly loaded, both policies exhibit almost identical performance in terms of average sojourn times.

## 6.2 System 2

We consider next an alternative 2-queue system, where jobs of each queue arrive according to an i.i.d. Bernoulli process and there are 3 normalized service rate vectors  $\vec{\mu}_1 = (0, 0.625)$ ,  $\vec{\mu}_2 = (0.5, 0.375)$ , and  $\vec{\mu}_3 = (0.625, 0.25)$ , with each corresponding to a combination of i.i.d. Bernoulli processes. The stability region for the normalized input rates  $(\lambda_1, \lambda_2)$  is shown in the left panel of Fig. 5. Analogously, we superimpose on the stability region a regular grid of sufficient density and then simulate the system at all the grid points  $(\lambda_1, \lambda_2)$  under the LWWT and the MaxProduct policy. The simulation result is shown in the right panel of Fig. 5.

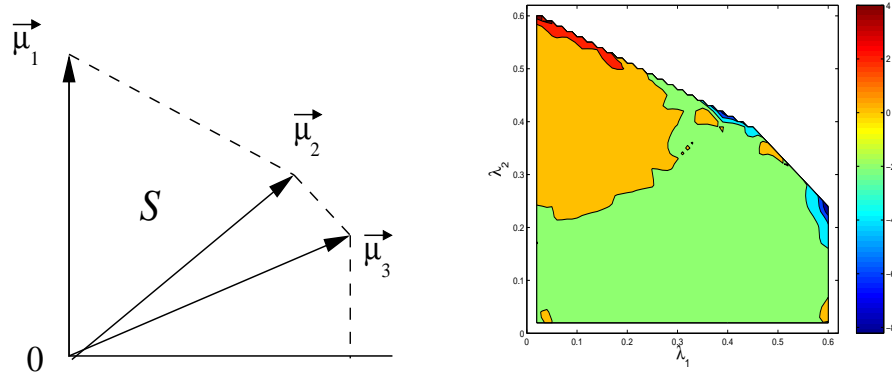


Fig. 5. (Left panel): The stability region of  $(\lambda_1, \lambda_2)$  for the system with 3 service vectors  $\vec{\mu}_1 = (0, 0.625)$ ,  $\vec{\mu}_2 = (0.5, 0.375)$ , and  $\vec{\mu}_3 = (0.625, 0.25)$ . (Right panel): The contour plot by taking the differences of average sojourn times for the two policies (MaxProduct-LWWT) over the stability region.

As it can be seen from the right panel of Fig. 5, the LWWT policy performs better than the MaxProduct policy when the input rates are in the upper-left area of the stability region. In particular, the difference of the average sojourn times between the two policies becomes significant when the system is heavily loaded. As interpreted in System 1, in this input area the MaxProduct policy uses the service vector  $\vec{\mu}_1$  most of the time, while the LWWT policy can possibly choose more efficient service vectors (i.e.  $\vec{\mu}_2$  and  $\vec{\mu}_3$ ) when both queues are not empty. On the other hand, the MaxProduct policy outperforms the LWWT policy in the remaining parts of the stability region. The reason is that, in this input area the MaxProduct policy uses  $\vec{\mu}_2$  or  $\vec{\mu}_3$  most of the time, while the LWWT policy is more likely to choose a less efficient service vector  $\vec{\mu}_1$  when both queues are not empty. Analogously, the numerical result shows that both policies have almost the same performance when the system is lightly loaded.

## 6.3 System 3

We consider a 3-queue system by feeding real network traces collected from the Internet 2 backbone links between Indianapolis and Cleveland. The data are from the Abilene-I collection of packet header traces, available from the National Laboratory for Advanced Network Research (NLNR). These traces were transformed into a sequence of arrival

bins (or packet lengths) corresponding to IP kilobytes with a fixed duration of 10 milliseconds (the same data set was used in [Rolls et al. 2005]). A time-series plot of the bin sizes is shown in the left panel of Fig. 6 while its auto-correlation function is shown in the right panel of Fig. 6.

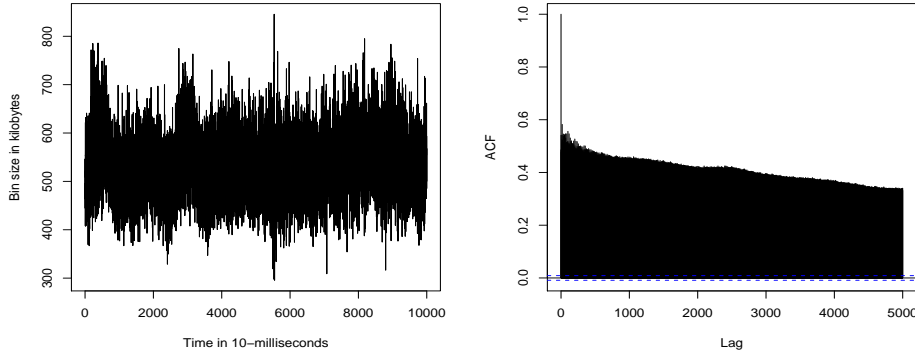


Fig. 6. (Left panel): The Abilence-I packet traces in kilobytes. (Right panel): The auto-correlation function of the bin sizes.

The right panel of Fig. 6 shows a slow decay in the auto-correlation function, which indicates that the process exhibits long-range dependence. In order to make this data set suitable for our analysis, we will transform the sequence of bin sizes into a sequence of service times when the system is in a particular service mode. Thus, each bin is treated as an arriving job and the fixed duration is treated as a deterministic job interarrival time. Suppose now the system has six service modes  $\vec{\mu}_1 = (0, 0, 6)$ ,  $\vec{\mu}_2 = (6, 0, 0)$ ,  $\vec{\mu}_3 = (0, 8, 0)$ ,  $\vec{\mu}_4 = (4, 0, 3)$ ,  $\vec{\mu}_5 = (4, 5, 0)$ , and  $\vec{\mu}_6 = (0, 5, 3)$ , where the service rate  $\mu_{mq}$  corresponds to the mean number of jobs in queue  $q$  that service mode  $m$  can process for one time unit (i.e. 10 milliseconds). Denote the sequence of bin sizes for queue  $q$  by  $b_q^i$ ,  $i = 1, \dots, N$ , and consider their normalized values  $b_q^i/\bar{b}_q$  where  $\bar{b}_q = \sum_{i=1}^N b_q^i/N$ . When the system is in service mode  $m$ , the corresponding service time of  $b_q^i$  is then taken to be  $b_q^i/(\bar{b}_q\mu_{mq})$ . Note that it is sometimes necessary to rescale the time duration (i.e. the job interarrival times) so that different input arrival rates can be constructed. However, this will not change the underlying correlation structure between service times. The stability region for this system is shown in the left panel of Fig. 7.

To examine the overall average-sojourn-time performance of the two policies, we simulate the system by feeding the traffic traces with seven input-rate combinations  $\vec{\lambda}_1 = (0.2, 0.2, 5.6)$ ,  $\vec{\lambda}_2 = (5.6, 0.2, 0.2)$ ,  $\vec{\lambda}_3 = (0.2, 7.6, 0.2)$ ,  $\vec{\lambda}_4 = (3.9, 0.2, 2.9)$ ,  $\vec{\lambda}_5 = (3.9, 4.9, 0.1)$ ,  $\vec{\lambda}_6 = (0.1, 4.9, 2.9)$ , and  $\vec{\lambda}_7 = (2.5, 2.5, 2.5)$ . Note that these seven input loads are allocated uniformly and close to the boundary of the stability region, where differences in the performance of the two policies would be easier to be distinguished. In order to gain better insight of the simulation results, all chosen input-rate combinations along with the decision cones of the MaxProduct policy are projected and sketched on the

triangular plane generated by the queue-length space coordinates, as shown in the right panel of Fig. 7. The numerical results are given in Table 1.

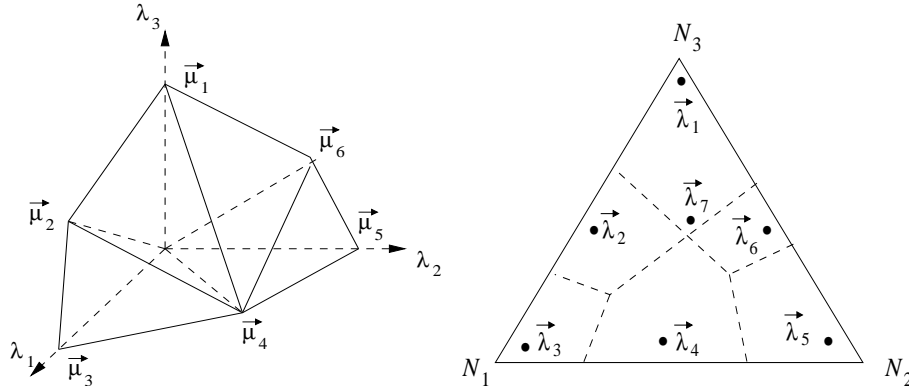


Fig. 7. (Left panel): The stability region formed by  $\vec{\mu}_1 = (0, 0, 6)$ ,  $\vec{\mu}_2 = (6, 0, 0)$ ,  $\vec{\mu}_3 = (0, 8, 0)$ ,  $\vec{\mu}_4 = (4, 0, 3)$ ,  $\vec{\mu}_5 = (4, 5, 0)$ , and  $\vec{\mu}_6 = (0, 5, 3)$ . (Right panel): Seven input-rate combinations (with sign “ $\cdot$ ”) and the decision cones (partitioned by dash lines) of the MaxProduct policy projected on the triangular plane formed by the queue-length space coordinates.

From Table 1, it can be seen that the LWWT policy outperforms the MaxProduct policy when the input loads are  $\vec{\lambda}_1$ ,  $\vec{\lambda}_2$ ,  $\vec{\lambda}_3$ ,  $\vec{\lambda}_4$ , and  $\vec{\lambda}_6$ . The reason is that, for these input loads the MaxProduct policy uses one service mode most of the time while this particular service mode is not the most efficient when all queues are not empty. For example, when  $\vec{\lambda}_1$  is used, it is expected that the MaxProduct policy will use  $\vec{\mu}_1$  most of the time. This then leads to a total service rate  $0 + 0 + 6 = 6$ , even when all queues are not empty. On the other hand, the LWWT policy is more likely to use  $\vec{\mu}_2$ ,  $\vec{\mu}_4$ , and  $\vec{\mu}_6$ , which are located in the neighborhood of  $\vec{\mu}_1$  and have larger (or at least equal) service rates in total (say, 6, 7, and 8, respectively). Table 1 shows that by using the LWWT policy, the improvement of the average sojourn time can be up to 20% for these particular input loads. When the input load corresponds to  $\vec{\lambda}_5$ , the numerical result shows that the MaxProduct policy outperforms the LWWT policy. The reason is that, for this input load the MaxProduct policy uses the service mode  $\vec{\mu}_5$  most of the time, which is the most efficient since it has the largest total service rate (say, 9). When  $\vec{\lambda}_7$  is used, the numerical results show that both policies exhibit almost identical performance. The reason is that, the system switches between service modes frequently under both policies since the input load is allocated near the boundaries of several decision cones, as shown in the right panel of Fig. 7.

## 7. CONSTRUCTING A HYBRID POLICY - A PRACTICAL EXTENSION

The simulation results in Section 6 provide us with guidelines as to best manipulate the two existing scheduling policies so that the average job sojourn time can be reduced. The main ideas are summarized next. When the system is lightly loaded, one can choose either the MaxProduct or the LWWT policy since both policies exhibit almost identical performance in terms of the average job sojourn time. When the system is heavily loaded, the simulation results reveal that the best policy is the one which is more likely to maximize the

Table I. The average sojourn times for both policies with seven input loads.

Input Load	The Average Sojourn Time	
	The MaxProduct Policy	The LWWT Policy
$\vec{\lambda}_1=(0.2,0.2,5.6)$	10.3787	8.7409
$\vec{\lambda}_2=(5.6,0.2,0.2)$	17.6234	14.1499
$\vec{\lambda}_3=(0.2,7.6,0.2)$	36.5881	35.3358
$\vec{\lambda}_4=(3.9,0.2,2.9)$	42.0322	38.5929
$\vec{\lambda}_5=(3.9,4.9,0.1)$	40.8925	43.8119
$\vec{\lambda}_6=(0.1,4.9,2.9)$	28.7145	25.4479
$\vec{\lambda}_7=(2.5,2.5,2.5)$	15.6810	15.7095

system's *instantaneous throughput*. For example, suppose the input-rate vector  $\vec{\lambda}$  belongs to a particular decision cone  $C_i$  in the queue-length space; then, the MaxProduct policy is more likely to choose the service mode  $\vec{\mu}_i$ , whose total service rate is  $T_i = \sum_{q=1}^Q \mu_{iq}$ . Consider next a collection of service modes whose corresponding decision cones are adjacent to  $C_i$  and denote it by the set  $\mathcal{N}_i = \{\vec{\mu}_{i(1)}, \dots, \vec{\mu}_{i(n_i)}\}$ , where  $\{i(1), \dots, i(n_i)\}$  is a subset of  $\{1, \dots, M\}$ . Each service vector in  $\mathcal{N}_i$  is then called a *neighbor* of  $\vec{\mu}_i$  and can be identified using a local-search algorithm introduced in [Ross and Bambos 2004]. Compute next the total service rate  $T_j$  for each  $\vec{\mu}_j \in \mathcal{N}_i$  and compare it with  $T_i$ . If  $T_i \geq T_j$  for all  $j \in \{i(1), \dots, i(n_i)\}$ , the MaxProduct policy should outperform the LWWT policy since it is more likely to maximize (locally) the system's instantaneous throughput. On the other hand, if  $T_j \geq T_i$  for all  $j \in \{i(1), \dots, i(n_i)\}$ , the LWWT policy should perform better since it is more likely to switch the system into the service modes in the neighborhood of  $\vec{\mu}_i$  (remember the LWWT policy has larger variations on its sample path), thus increasing the system's instantaneous throughput. Therefore, we propose the following rule of thumb for selecting the scheduling policy:

$$\begin{aligned} \text{choose the MaxProduct policy if } T_i &\geq \frac{1}{n_i} \sum_{j=i(1)}^{i(n_i)} T_j; \\ \text{choose the LWWT policy if } T_i &< \frac{1}{n_i} \sum_{j=i(1)}^{i(n_i)} T_j. \end{aligned}$$

In response to the input-rate vector  $\vec{\lambda} \in C_i$ , this rule suggests that if the total service rate of  $\vec{\mu}_i$  is greater than the "average" total service rate of its neighbors, the MaxProduct policy should be used. Otherwise, the LWWT policy should be used.

For practical purposes, the above findings can be utilized to improve the average-sojourn-time performance when the system has unknown and changing input rates. Specifically, we can estimate the input rate of each queue over time by tracking the job interarrival times. Suppose that time is divided into non-overlapping intervals  $(t_k, t_{k+1}]$ , so that  $N_q(k)$  ( $N_q(k) \geq 30$ ) jobs are observed arriving to each queue  $q$  in the  $k$ -th interval. Denote the

job interarrival times of queue  $q$  in the  $k$ -th interval by  $\tau_q^1, \dots, \tau_q^{N_q(k)}$ , then the input rate can be estimated by

$$\hat{\lambda}_q(k) = \frac{1}{\sum_{j=1}^{N_q(k)} \tau_q^j / N_q(k)}, \quad (47)$$

where  $q = 1, \dots, Q$ , and  $k = 1, 2, \dots$ . Hence, a sequence of estimated input-rate vectors  $\{\hat{\Lambda}(k), k = 1, 2, \dots\}$  can be calculated. Based on the estimated input rates, at certain points in time we decide whether the MaxProduct or the LWWT policy we have to choose according to the preceding rule of thumb. This then constitutes a *hybrid* scheduling policy that is adaptive to input traffic fluctuations. We summarize the steps of this hybrid policy in the following algorithm.

### Algorithm for implementing the hybrid policy

#### Step 1.

Calculate the total service rate for each service mode  $\vec{\mu}_i$ , say, let  $T_i = \sum_{j=1}^Q \mu_{ij}$ ,  $i = 1, \dots, M$ .

Identify the set of neighbors for each  $\vec{\mu}_i$  and denote it by  $\mathcal{N}_i = \{\vec{\mu}_{i(1)}, \dots, \vec{\mu}_{i(n_i)}\}$ .

Calculate the average of the total service rates for each  $\mathcal{N}_i$ , say, let  $\bar{T}_{\mathcal{N}_i} = \sum_{j=i(1)}^{i(n_i)} T_j / n_i$ .

#### Step 2.

Set  $k = 1$  and  $t_0 = 0$ .

Choose the MaxProduct policy (or the LWWT policy) at time  $t_0$ .

#### Step 3.

Record all subsequent job interarrival times after  $t_{k-1}$  and determine the value of  $t_k$  so that  $N_q(k) \geq 30$  for all  $q = 1, \dots, Q$ .

Set

$$\hat{\lambda}_q(k) = \frac{1}{\sum_{j=1}^{N_q(k)} \tau_q^j / N_q(k)}.$$

#### Step 4.

Identify the decision cone  $C_i$  of the MaxProduct policy so that  $\hat{\Lambda}(k) \in C_i$  in the queue-length space.

If  $T_i \geq \bar{T}_{\mathcal{N}_i}$ , then choose the MaxProduct policy at time  $t_k$ .

Otherwise, choose the LWWT policy at time  $t_k$ .

Set  $k = k + 1$ , go to Step 3.

Note that this hybrid policy can easily accommodate non-stationary input processes. Now we examine how this strategy performs by simulating a 3-queue system characterized by randomly modulated service modes that are defined by combinations of three possible input processes  $\mathcal{A}_i = (\mathcal{A}_i^1, \mathcal{A}_i^2, \mathcal{A}_i^3)$ ,  $i = 1, 2, 3$ . Suppose each  $\mathcal{A}_i^q$  is a Poisson process and three possible input-rate combinations are  $\vec{\lambda}_1 = (5.6, 0.2, 0.2)$ ,  $\vec{\lambda}_2 = (3.9, 0.2, 2.9)$ , and  $\vec{\lambda}_3 = (3.9, 4.9, 0.1)$ . Further, we assume that the combinations are mutually independent and each occurs with equal probability every 10,000 time units. There are six service modes with rates  $\vec{\mu}_1 = (0, 0, 6)$ ,  $\vec{\mu}_2 = (6, 0, 0)$ ,  $\vec{\mu}_3 = (0, 8, 0)$ ,  $\vec{\mu}_4 = (4, 0, 3)$ ,  $\vec{\mu}_5 = (4, 5, 0)$ , and  $\vec{\mu}_6 = (0, 5, 3)$ . When the system is in service mode  $m$ , the job service times in queue  $q$

are i.i.d. exponential random variables with mean  $\mu_{mq}$ . The simulation result shows that the average sojourn times under the MaxProduct and the LWWT policy are 35.0817 and 34.2172, respectively, while the average sojourn time under the hybrid policy is merely 32.5307, which shows a 5% to 7% improvement.

*REMARK 7.1. It is noted that the proposed hybrid policy switches between the Max-Product and the LWWT policy over time in accordance with the estimated input rates. For systems with stationary input processes, it is clear that this strategy uses almost only one policy in order to reduce the average job sojourn time. In this case, there is no doubt that the throughput of the system can be maximized. For systems with non-stationary input processes, this strategy seeks for an “optimal” convex combination of using these two policies to reduce the average job sojourn time. However, from the theoretical point of view, whether or not the system’s throughput can be still maximized needs to be further investigated.*

## 8. CONCLUDING REMARKS

In this study, we consider switched processing systems both in slotted and continuous times under a number of stochastic assumptions for the arrival and service processes. Further, two scheduling policies are considered; the existing MaxProduct one and the proposed Largest Weighted Waiting Time (LWWT). It is shown that both policies maximize the system’s throughput. Specifically, for systems with independent Bernoulli processes, the proof was established using drift analysis, while in the case of more general correlated processes that induce non-Markovian dynamics, the proof was based on a perturbed Lyapunov function method. The performance of the two policies under different input processes in terms of their average job sojourn times was examined through a simulation study and insights in their workings obtained. Based on the simulation results, a hybrid scheduling policy was proposed in order to further reduce the average job sojourn time, especially for systems with unknown and changing input rates. We may conclude that the hybrid policy is best suited for fairly heavily loaded systems that experience frequent changes of their input rates, a feature in real life operations. The following issues are currently under investigation: (i) how to improve the proposed hybrid policy by allocating the best convex combination of the MaxProduct and the LWWT policy so that different performance measures (such as throughput and average sojourn time) can be optimized; (ii) the computational issues raised for systems with a large number of inputs and service modes.

## ACKNOWLEDGMENTS

The authors would like to thank the Associate Editor and two referees for many useful comments and suggestions.

## REFERENCES

- ANDREWS, M., KUMARAN, K., RAMANAN, K., STOLYAR, A., VIJAYAKUMAR, R., AND WHITING, P. 2004. Scheduling in a queueing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences* 18, 191–217.
- ARMONY, M. AND BAMBOS, N. 2003. Queueing dynamics and maximal throughput scheduling in switched processing systems. *Queueing Systems: Theory and Applications* 44, 209–252.
- BANKS, J., CARSON, II, J. S., AND NELSON, B. L. 1999. *Discrete-Event System Simulation*. Prentice-Hall, 2nd Edition.

- DAI, J. G. 1995. On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *Annals of Applied Probability* 5, 49–77.
- DAI, J. G. AND LIN, W. 2005. Maximum pressure policies in stochastic processing networks. *Operations Research* 53, 2, 197–218.
- DAI, J. G. AND MEYN, S. P. 1995. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Trans. on Automatic Control* 40, 1889–1904.
- DAI, J. G. AND PRABHAKAR, B. 2000. The throughput of data switches with and without speedup. In *Proceedings of IEEE INFOCOM*. 556–564.
- HAJEK, B. 1982. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advanced Applied Probability* 14, 502–525.
- HUNG, Y. C. AND MICHAELIDIS, G. 2006. Improving quality of service for switched processing systems. In *Proceedings of 11th International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks*.
- KUSHNER, H. J. 1967. *Stochastic Stability and Control*. Academic Press, New York.
- KUSHNER, H. J. 1984. *Approximation and Weak Convergence Methods for Random Processes with Applications to Stochastic Systems Theory*. MIT Press, Cambridge, Mass.
- KUSHNER, H. J. AND YIN, G. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, Berlin and New York.
- LEONARIDI, E., MELLIA, M., NERI, F., AND MARSON, M. A. 2001. On the stability of input-queued switches with speed-up. *IEEE Trans. on Networking* 9, 1, 104–118.
- MCKEOWN, N., MEKKITTIKUL, A., ANANTHARAM, V., AND WALRAND, J. 1999. Achieving 100% throughput in an input-queued switch. *IEEE Trans. on Communications* 47, 8, 1260–1267.
- MEKKITTIKUL, A. AND MCKEOWN, N. 1996. A starvation-free algorithm for achieving 100% throughput in an input-queued switch. In *Proceedings of ICCCN*. 226–231.
- MEYN, S. P. AND TWEEDIE, R. L. 1993. *Markov Chains and Stochastic Stability*. Springer, London.
- PEMANTLE, R. AND ROSENTHAL, J. S. 1999. Moment conditions for a sequence with negative drift to be uniformly bounded in  $l^r$ . *Stochastic Processes and their Applications* 82, 143–155.
- ROLLS, D. A., MICHAELIDIS, G., AND HERNANDEZ-CAMPOS, F. 2005. Queueing analysis of network traffic: Methodology and visualization tools. *Computer Networks* 48, 447–473.
- ROSS, K. AND BAMBOS, N. 2004. Local search scheduling algorithms for maximal throughput in packet switches. In *Proceedings of IEEE INFOCOM*.
- ROSS, K. AND BAMBOS, N. 2005. Dynamic quality of service control in packet switch scheduling. In *Proceedings of IEEE International Conference on Communications*.
- STOLYAR, A. L. 2003. Control of end-to-end delay tails in a multiclass network: Lwdf discipline optimality. *Annals of Applied Probability* 13, 3, 1151–1206.
- STOLYAR, A. L. 2004. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Annals of Applied Probability* 14, 1, 1–53.
- TASSIULAS, L. AND EPHREMIDES, A. 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. on Automatic Control* 37, 12, 1936–1949.
- WALRAND, J. 1988. *An Introduction to Queueing Networks*. Prentice Hall.