**REPORT**RAPPORT

*PNA*

Probability, Networks and Algorithms

*Probability, Networks and Algorithms*

Stability of size-based scheduling disciplines in resource-sharing networks

I.M. Verloop, S.C. Borst, R. Núñez-Queija

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

# Stability of size-based scheduling disciplines in resource-sharing networks

ABSTRACT

Size-based scheduling strategies such as Shortest Remaining Processing Time first (SRPT) and Least Attained Service first (LAS) provide popular mechanisms for improving the overall delay performance by favoring smaller service requests over larger ones. The performance gains from these disciplines have been thoroughly investigated for single-server systems, and have also been experimentally demonstrated in web servers for example. In the present paper we explore the fundamental stability properties of size-based scheduling strategies in multi-resource systems, such as bandwidth-sharing networks, where users require service from several shared resources simultaneously. In particular, we establish the exact stability conditions for the SRPT and LAS disciplines in various limiting regimes. The results indicate that size-based scheduling strategies may fail to use the available resources efficiently, and in fact cause instability effects, even at arbitrarily low traffic loads, and will therefore not yield optimal delay performance. The qualitative findings confirm the tendency for users with long routes and large service requirements to experience severe performance degradation.

# Stability of Size-Based Scheduling Disciplines in Resource-Sharing Networks

Maaike Verloop[†*], Sem Borst[†,⋆,‡], Rudesindo Núñez-Queija[†,⋆]

[†]CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

[⋆]Department of Mathematics & Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[‡]Bell Laboratories, Lucent Technologies
P.O. Box 636, Murray Hill, NJ 07974, USA

## Abstract

Size-based scheduling strategies such as Shortest Remaining Processing Time first (SRPT) and Least Attained Service first (LAS) provide popular mechanisms for improving the overall delay performance by favoring smaller service requests over larger ones. The performance gains from these disciplines have been thoroughly investigated for single-server systems, and have also been experimentally demonstrated in web servers for example. In the present paper we explore the fundamental stability properties of size-based scheduling strategies in multi-resource systems, such as bandwidth-sharing networks, where users require service from several shared resources simultaneously. In particular, we establish the exact stability conditions for the SRPT and LAS disciplines in various limiting regimes. The results indicate that size-based scheduling strategies may fail to use the available resources efficiently, and in fact cause instability effects, even at arbitrarily low traffic loads, and will therefore not yield optimal delay performance. The qualitative findings confirm the tendency for users with long routes and large service requirements to experience severe performance degradation.

---

[*]This work was done while the first author was a master student at Utrecht University.

# 1  Introduction

The past few years have witnessed a tremendous interest in the design and analysis of service disciplines for efficient delivery of digital documents in communication systems, such as page requests in web servers and file transfers in the Internet. These disciplines have mostly been proposed with two related but different objectives in mind: (i) improving the overall delay performance by exploiting the variability in service demands, and giving precedence to smaller requests over larger ones; (ii) supporting service differentiation among competing users based on Quality-of-Service requirements, traffic characteristics, or tariff structures.

(i) The Shortest Remaining Processing Time first (SRPT) discipline provides a canonical example of a size-based scheduling strategy. The SRPT discipline minimizes the mean response time (and in fact the number of jobs in the system, sample-path wise) among the class of all strategies [32]. The rationale for size-based scheduling has been greatly amplified by empirical findings indicating that file sizes in the Internet show extreme variability and commonly have infinite variance [13]. Note that in case of infinite variance the mean delay under the First-Come First-Served discipline is infinite for any load, whereas the SRPT discipline yields a finite mean delay as long as the system is stable [33]. For the latter reasons, the SRPT discipline has been widely adopted as an effective mechanism for improving the response time performance in web servers [11, 18]. To some extent, the huge variability in job sizes also alleviates the long-standing concerns that have surrounded SRPT regarding the perceived unfairness towards extremely long jobs [7, 16, 35]. It turns out that in case of heavy-tailed distributions only an exceedingly small fraction of the jobs is worse off than under Processor Sharing (PS) as the prototype of perfect fairness.

(ii) A typical example of differentiated scheduling is a strict priority discipline, either preemptive or non-preemptive. This form of service differentiation has for example been suggested as a potential mechanism for supporting low-priority data transfers without affecting regular TCP flows [20, 23]. The Discriminatory Processor Sharing (DPS) discipline offers a more flexible instrument for accomplishing service differentiation [21]. In DPS, jobs belong to one of several classes, and the total service rate is shared among competing requests in proportion to class-dependent weight factors. There are strong indications that DPS achieves a certain degree of isolation among classes and in particular immunity from highly variable service demands [6, 9], in contrast to strict priority disciplines.

Besides fairness aspects, a further major issue surrounding size-based scheduling in general and SRPT in particular, is that it relies on knowledge of (remaining) job sizes. While such information is typically available in web servers, it is usually nearly impossible to obtain in Internet routers. An alternative strategy which has therefore been advocated for scheduling data flows is the Least Attained Service first (LAS) discipline also known as Foreground-Background Processor Sharing (FBPS) [22, 27, 29, 30]. In case the service requirement distribution has a decreasing failure rate, it has been shown that LAS stochastically minimizes the number of jobs in the system among all strategies that use no knowledge of the remaining job sizes [31]. While this implies that the LAS discipline improves the performance for a rich class of highly variable service requirement distributions, there is no guarantee that it will do better than FCFS or PS in general, and in fact it performs worse for distributions with increasing failure rates. A discrete-class version of LAS that has recently attracted renewed attention is the Multi-Level Processor Sharing (MLPS) discipline [1, 2, 5, 15, 22].

The performance gains from size-based scheduling disciplines have been thoroughly analyzed in the literature for single-server systems. In practice however, users commonly require service

from several shared resources simultaneously, like traffic flows that traverse multiple links in bandwidth-sharing networks as considered in [9, 25] for example. While single-server systems provide tractable results and useful insights, they do not exhibit the potential non work-conserving behavior that may arise in systems with concurrent resource possession. The performance characteristics of size-based scheduling disciplines in such resource-sharing networks have only received limited attention so far. As one of the few exceptions, Yang & De Veciana [36, 37] evaluated the potential performance benefits from SRPT scheduling in bandwidth-sharing networks supporting best-effort traffic. They found that SRPT scheduling may yield considerable performance improvements in terms of mean transfer delays and user throughputs, but also observed that flows on long routes with large sizes may experience a marked performance degradation.

In the present paper, we explore the fundamental stability properties of size-based scheduling strategies in systems with simultaneous resource possession, like the bandwidth-sharing networks mentioned above. In particular, we establish the exact stability conditions for the SRPT and LAS disciplines in various limiting regimes. Due to concurrency constraints, these disciplines may use the available resources inefficiently and persistently leave critical resources underutilized, even when congestion builds up. As a result, the SRPT and LAS disciplines may fail to achieve stability, and will therefore certainly not yield optimal delay performance. In particular, instability effects may occur when the users with long routes have larger service requirements than the ones with shorter routes. For networks with sufficiently many nodes, it turns out that instability phenomena may in fact arise at arbitrarily low traffic loads. The latter conclusions corroborate the findings in [36, 37]. The results further suggest that low-priority transfer protocols as proposed in [20, 23] may not provide satisfactory throughput performance to users with long routes. In the opposite regime, where the users with long routes have smaller service requirements than the ones with shorter routes, size-based scheduling strategies are less prone to instability effects. However, prioritization of users with long routes will evidently not optimize the delay performance in general.

It is worth drawing a distinction with the situation in queueing networks with feedback where the usual necessary stability conditions are not sufficient either, as first exemplified in Lu & Kumar [24] for priority scheduling and later studied in Bramson [10] for FIFO (First-In First-Out). In these networks, users visit the various nodes along their route through the network in succession, whereas traffic flows in bandwidth-sharing networks require service at all nodes along their route simultaneously. The way in which the queues build up in these networks with feedback is also qualitatively different, and typically involves oscillatory behavior.

On a final note, instability is to a certain extent a theoretical concept that cannot occur in an actual system with an inherently finite user population. In practice, peak rate limitations will also inhibit strict prioritization and tend to promote fair bandwidth sharing, thus reducing the scope for instability effects. However, it is plausible that instability phenomena will be reflected in poor performance in terms of long delays, low throughputs, and user impatience in practical circumstances as well.

The remainder of the paper is organized as follows. In Section 2 we present a detailed model description and give some preliminary results. In Section 3 we examine the stability conditions for SERPT (Shortest Expected Remaining Processing Time first) scheduling in linear networks with exponentially distributed service requirements. We turn the attention to SRPT scheduling in Section 4. In Section 5 we derive stability properties for LAS scheduling. In Section 6 we make some concluding remarks.

# 2  Model description and preliminary results

We consider a linear network with $L$ nodes. For convenience, we assume each of the nodes to have a unit service rate. With minor modifications, most of the results extend to scenarios where the various nodes have heterogeneous service rates.

In order to present the results in the simplest possible setting, we focus on a traffic scenario with $L + 1$ classes, where class $i$ requires service at node $i$ only, $i = 1, \ldots, L$, while class 0 requires service at all $L$ nodes simultaneously. The above 'toy' scenario appears already sufficiently rich to exhibit many of the qualitative phenomena that may occur for general network topologies and route structures. Class-$i$ users arrive according to independent Poisson processes of rate $\lambda_i$, and have generally distributed service requirements $B_i$ with distribution function $B_i(x) = \mathbb{P}(B_i < x)$ and mean $\beta_i$, $i = 0, \ldots, L$. Define the traffic load of class $i$ as $\rho_i := \lambda_i \beta_i$. Thus the load at node $i$ is $\rho_0 + \rho_i$, $i = 1, \ldots, L$.

The queue of class-$i$ users is referred to as $Q_i$. Note that in bandwidth-sharing networks, the queue is a purely virtual entity in the sense that the users do not actually reside in physical queues, but rather keep the bulk of the backlogged work stored in their own buffers. Denote by $N_i(t)$ the length of $Q_i$ at time $t$, i.e., the number of class-$i$ users in the system at time $t$. Define $N_i$ as a random variable with the time-average distribution of $N_i(t)$ for $t \to \infty$, assuming it exists.

In the sequel, $Q_i$ is said to be stable if $\mathbb{P}(N_i = 0) > 0$. Node $i$, for $i = 1, \ldots, L$, will be called stable if both $Q_0$ and $Q_i$ are stable. Obviously, $\rho_0 + \rho_i < 1$, for $i = 1, \ldots, L$, are necessary conditions for the various nodes $i$ to be stable, and thus for stability of the entire system. It follows from the results in [9] that these conditions are in fact also sufficient for $\alpha$-fair bandwidth-sharing policies. For conciseness, these conditions will often be referred to as the 'standard' conditions.

Denote by $\sigma_i := \lim_{T \to \infty} \frac{1}{T} \int_{t=0}^{T} s_i(t) \mathrm{d}t$ the long-term average service rate of class $i$ (assuming it exists), with $s_i(t)$ denoting the service rate allocated to class $i$ at time $t$. Note that $\sigma_i = \rho_i$ when $Q_i$ is stable. We have the identity relation $\sigma_0 + \sigma_i + u_i = 1$, where $u_i := \lim_{T \to \infty} \frac{1}{T} \int_{t=0}^{T} u_i(t) \mathrm{d}t$, with $u_i(t) := 1 - s_0(t) - s_i(t)$, denotes the long-term unused average service rate at node $i$. Noting that $u_i(t) = 1$ when $N_0(t) = 0, N_i(t) = 0$, we may write $u_i = w_i + \mathbb{P}(N_0 = 0, N_i = 0)$, where $w_i := \lim_{T \to \infty} \frac{1}{T} \int_{t=0}^{T} u_i(t) \mathrm{I}_{\{N_0(t) > 0 \text{ or } N_i(t) > 0\}} \mathrm{d}t$ stands for the long-term average 'wasted' service rate at node $i$, (assuming it exists). Since $\mathbb{P}(N_0 = 0, N_i = 0) > 0$ implies stability of node $i$, it follows that node $i$ is stable if $\rho_0 + \rho_i + w_i < 1$. The above observation reflects the fact that the stability condition may not only depend on the first moments of the service requirements as captured through the loads, but also on higher-order characteristics, as was found in [8] as well.

The scheduling discipline prescribes the allocation of the service rates among competing users, depending on the state of the network. It relies on some underlying criterion which for every possible state of the network defines a priority ranking among all users, e.g. based on some class parameter (SERPT), remaining service requirement (SRPT), or amount of attained service (LAS). Since class-0 users require simultaneous service at all nodes, without any further arbitration mechanism, capacity can be left unused. Therefore the priority ranking needs to be augmented with a further arbitration mechanism to arrive at the allocation of service rates among competing users. We will distinguish between two options: (i) 'weak priority', which means that the capacity in node $i$ that is left unused, is re-allocated to class $i$; (ii) 'strict priority', which implies that this capacity is left unused. We have the following

4

useful property.

**Property 2.1** (i) *A sufficient condition for stability of the system is $\sum_{i=0}^{L} \rho_i < 1$, provided that at least one of the nodes operates at the full service rate whenever the system is non-empty.*
(ii) *A sufficient condition for stability of $Q_i$, for $i = 1, \ldots, L$, is $\rho_0 + \rho_i < 1$, provided that node $i$ operates at the full service rate whenever $Q_i$ is non-empty.*

Statement (i) follows from the fact that the total workload (of classes $i = 0, \ldots, L$), is stochastically dominated by that in a system where classes $i = 1, \ldots, L$ are never served at the same time. Statement (ii) deserves some more elaboration. Suppose that $\rho_0 + \rho_i < 1$, and hence $\sigma_0 + \sigma_i < 1$, while $Q_i$ is not stable. Since $Q_i$ is unstable, in the long run there are always class-$i$ users and because node $i$ operates at the full service rate whenever $Q_i$ is non-empty, we have $w_i = 0$. Therefore $\sigma_0 + \sigma_i = 1$, which is in contradiction with our initial supposition. For the scheduling disciplines considered in this paper (SERPT, SRPT, LAS), the first property is always satisfied, while the second property only holds for the variants with weak priority.

# 3   SERPT scheduling

In preparation for the analysis of the SRPT and LAS disciplines, we first consider the Shortest Expected Remaining Processing Time first (SERPT) discipline. We assume that class-$i$ users have exponentially distributed service requirements with mean $\beta_i = 1/\mu_i$. The SERPT discipline then simply gives priority to class-$i$ users over class-0 users when $\mu_0 < \mu_i$ and vice versa when $\mu_0 > \mu_i$. The stability results below in fact hold for generally distributed service requirements, general renewal arrival processes, and any static class-based priority discipline.

## 3.1   Large class-0 users

We start with the case $\mu_0 < \mu_1, \ldots, \mu_L$. Since class-$i$ users for $i = 1, \ldots, L$, always receive priority over class-0 users, they behave as in an isolated M/M/1 queue with class $i$ only. Therefore, $Q_i$ is stable if and only if $\rho_i < 1$, for $i = 1, \ldots, L$. Moreover, $N_1, \ldots, N_L$ are independent, so $\mathbb{P}(N_1 = n_1, \ldots, N_L = n_L) = (1 - \rho_1)\rho_1^{n_1} \cdots (1 - \rho_L)\rho_L^{n_L}$. Class-0 users can be served if and only if there are no class-$i$ users, for $i = 1, \ldots, L$. Hence, $Q_0$ is stable if and only if

$$\rho_0 < \mathbb{P}(N_1 = 0, \ldots, N_L = 0) = \Pi_{i=1}^{L}(1 - \rho_i).$$

Note that the above condition is more stringent than the standard conditions (unless $\rho_i > 0$ for at most one $i = 1, \ldots, L$). In fact, the system can be unstable for arbitrarily low values of $\rho_0$ if the number of traversed nodes is large. In Sections 4 and 5 we show that the SRPT and LAS disciplines inherit these difficulties.

## 3.2   Small class-0 users

Now focus on the case $\mu_0 > \mu_1, \ldots, \mu_L$. Because class-0 users receive priority over all class-$i$ users, $i = 1, \ldots, L$, $Q_0$ will be stable as long as $\rho_0 < 1$. Class-$i$ users, for $i = 1, \ldots, L$, behave as in an isolated priority queue with classes 0 and $i$ only, so that $Q_i$ will be stable when $\rho_0 + \rho_i < 1$.

### 3.3 Intermediate-size class-0 users

We now extend the model with class-$i'$ users which require service from resource $i$, arrive according to a Poisson process of rate $\lambda_{i'}$, and have exponentially distributed service requirements with mean $1/\mu_{i'}$. We assume $\mu_i > \mu_0 > \mu_{i'}$ for all $i = 1, \ldots, L$. Denote the traffic load of class $i'$ by $\rho_{i'} := \lambda_{i'}/\mu_{i'}$, $i = 1, \ldots, L$.

First observe that class-$i$ users, $i = 0, \ldots, L$, are not affected by the presence of class-$i'$ users, $i = 1, \ldots, L$. It thus follows from the results in Subsection 3.1 that $Q_i$ is stable if and only if $\rho_i < 1$ and that $Q_0$ is stable if and only if $\rho_0 < (1 - \rho_1) \cdots (1 - \rho_L)$.

In order to establish the stability condition for $Q_{i'}$, $i = 1, \ldots, L$, it is important to know whether we have weak or strict SERPT. 'Strict' SERPT only allows a class-$i'$ user to be served when there are no class-0 and class-$i$ users in the system. In contrast, 'weak' SERPT also allows a class-$i'$ user to be served when there are class-0 users in the system which are however blocked from service by class-$j$ users, $i \neq j$, and there are no class-$i$ users present.

#### 3.3.1 Weak SERPT

For weak SERPT, class-$i'$ users can be served during the time that $Q_i$ is empty. However, class-0 users may be served during this time as well. Thus, the stability condition for $Q_{i'}$ may be written as $\rho_{i'} < 1 - \rho_i - \sigma_0$, or equivalently $\rho_i + \rho_{i'} + \sigma_0 < 1$, where $\sigma_0$ denotes the fraction of time that class-0 users are served. In order to determine the value of $\sigma_0$, we need to distinguish whether $Q_0$ is stable or not, i.e., whether $\rho_0 < (1 - \rho_1) \cdots (1 - \rho_L)$ or not. If $Q_0$ is stable, then $\sigma_0 = \rho_0$, and thus the stability condition for $Q_{i'}$ becomes $\rho_{i'} < 1 - \rho_i - \rho_0$, or simply $\rho_0 + \rho_i + \rho_{i'} < 1$. If $Q_0$ is unstable, then $\sigma_0 = (1 - \rho_1) \cdots (1 - \rho_L)$, so that the stability condition for $Q_{i'}$ takes the form

$$\rho_{i'} < 1 - \rho_i - (1 - \rho_1) \cdots (1 - \rho_L) = (1 - \rho_i)(1 - \prod_{j \neq i}(1 - \rho_j)).$$

#### 3.3.2 Strict SERPT

For strict SERPT, the stability condition for $Q_{i'}$ may be expressed as $\rho_{i'} < \mathbb{P}(N_0 = 0, N_i = 0)$ for $i = 1, \ldots, L$. In general no tractable expression appears to exist for $\mathbb{P}(N_0 = 0, N_i = 0)$.

## 4 SRPT scheduling

We turn the attention to the Shortest Remaining Processing Time first (SRPT) discipline. A class-0 user receives the total capacity of all nodes whenever it has the smallest remaining service requirement among all users. Otherwise, in case of weak SRPT, in node $i$ the total capacity is given to the class-$i$ user with the smallest remaining service requirement. However, in case of strict SRPT the total capacity in node $i$ is only given to a class-$i$ user, if this user has indeed the smallest remaining size among all class-0 and class-$i$ users. Possible ties (which occur with non-zero probability in case of discrete service requirement distributions) are assumed to be broken at random.

For each class $i$, define $x_i^*$ as the smallest value of $x$ such that $Q_i$ is unstable in a reference system where all class-$i$ users with service requirement $x$ or larger are denied access. Denote by

$$\rho_i(x) := \lambda_i \mathbb{E}(B_i \mathrm{I}_{\{B_i < x\}}) = \lambda_i \int_{y=0}^{x^-} y \, \mathrm{d}B_i(y) \tag{1}$$

the traffic load of class $i$ when all class-$i$ users of size $x$ or larger are rejected. It is important to note that users of size exactly $x$ are excluded in this definition. It may be checked that, due to the mechanics of the SRPT discipline, in the original system all class-$i$ users of smaller size than $x_i^*$ eventually complete service and leave the system with probability one, whereas in the long run class-$i$ users of size larger than or equal to $x_i^*$ never complete service and stay in the system forever with non-zero probability. In fact, class-$i$ users of size strictly larger than $x_i^*$ will never even enter service in the long run. Note that $Q_i$ is stable in the original system when $\mathbb{P}(B_i < x_i^*) = 1$ (so, in particular when $x_i^* = \infty$ in case $B_i$ has infinite support).

**Observation 4.1** *For both weak and strict SRPT, $\rho_0(x_0^*) + \rho_i(x_i^*) \leq 1$, for $i = 1, \ldots, L$. For weak SRPT $\rho_0(x_0^*) + \rho_i(x_i^*) = 1$ if $Q_i$ is unstable and $B_0$ and $B_i$ have a continuous distribution.*

The observation follows from the properties that (i) $\sigma_0 + \sigma_i \leq 1$, $i = 1, \ldots, L$, with equality for weak SRPT in case $Q_i$ is unstable, and (ii) $\sigma_j \geq \rho_j(x_j^*)$, $j = 0, \ldots, L$, for both weak and strict SRPT, with equality in case $B_j$ has a continuous distribution.

The next proposition gives a general relationship between the values of $x_0^*$ and $x_i^*$, $i = 1, \ldots, L$.

**Proposition 4.2** *For weak SRPT, $x_0^* \leq x_i^*$, for $i = 1, \ldots, L$, and for strict SRPT, $x_0^* = x_i^*$, for $i = 1, \ldots, L$.*

**Proof** In order to avoid technicalities, we assume in this proof that the service requirement distributions of all classes have support everywhere. With minor modifications, the proof extends to distributions with zero density in some points by introducing 'fictitious' users and observing that $B_0$ and $B_i$ cannot both have zero density in $x_0^*$ and $x_i^*$, respectively.
The idea of the proof may be described as follows. Under SRPT, class-0 users with a given service requirement $x$ cannot enter service until all the class-$i$ users present at the time of their arrival with a service requirement smaller than or equal to $x$ have completed service. Thus, if class-0 users with a given service requirement eventually leave the system, then class-$i$ users with a smaller or equal service requirement must do so as well, which implies $x_0^* \leq x_i^*$. Moreover, for strict SRPT the reverse implication also holds.
In order to formalize the above arguments, suppose $x_0^* > x_i^*$. For both weak and strict SRPT, we can reach a contradiction as follows. By definition of $x_i^*$, in the long run, when a class-$i$ user arrives with service requirement $s_i$, $x_i^* < s_i < x_0^*$, it will never leave the system (and in fact never even enter service). Now suppose that subsequently a class-0 user arrives with service requirement $s_i < s_0 < x_0^*$. Because of SRPT, this user cannot enter service before the class-$i$ user leaves the system. Since the latter never happens, the class-0 user never leaves the system either, which contradicts the assumption $s_0 < x_0^*$ and the definition of $x_0^*$. Thus, for both weak and strict SRPT, $x_0^* \leq x_i^*$, for $i = 1, \ldots, L$.
For strict SRPT, it may be shown along similar lines that $x_0^* < x_i^*$ leads to a contradiction. By definition of $x_0^*$, in the long run, when a class-0 user arrives with service requirement $s_0$, $x_0^* < s_0 < x_i^*$, it will never leave the system (and in fact never even enter service). Now suppose that subsequently a class-$i$ user arrives with service requirement $s_0 \leq s_i < x_i^*$. Because of the scheduling rule, this user cannot enter service before the class-0 user leaves the system. Since the latter never happens, the class-$i$ user never leaves the system either, which contradicts the assumption $s_i < x_i^*$ and the definition of $x_i^*$. Thus, for strict SRPT, $x_0^* \geq x_i^*$, so that in fact $x_0^* = x_i^*$, for $i = 1, \ldots, L$. $\square$

Combining Observation 4.1 and Proposition 4.2, it follows that if $\rho_0 + \rho_i > 1$ for some $i = 1, \ldots, L$, then for strict SRPT $x_0^* = x_i^* < \infty$, so that both $Q_0$ and $Q_i$ are unstable in case $B_0$ and $B_i$ have infinite support. Unfortunately, the above results do not suffice to determine the exact values of $x_i^*$ in general, since Observation 4.1 gives equality only for weak SRPT, whereas the relation in Proposition 4.2 only holds with equality for strict SRPT. In order to establish exact stability conditions, we need to impose some additional assumptions on the service requirement distributions, as will be done in the next subsections.

## 4.1 Large class-0 users

In this subsection we consider class-0 users with large service requirements, compared to all other classes.

### 4.1.1 Stability of $Q_i$, for $i = 1, \ldots, L$

Define $m_i := \inf\{x : B_i(x) > 0\}$ and $M_i := \sup\{x : B_i(x) < 1\}$ as the minimum and maximum values of the class-$i$ service requirements, $i = 0, \ldots, L$. We focus on the case where class 0 has larger service requirements than all classes $i$, i.e., $m_0 > M_i$, for $i \neq 0$. Thus, a class-0 user can only enter service when there are no class-$i$ users in the system. When a class-0 user is in service and a class-$i$ user arrives, the service is preempted when the remaining service requirement of the class-0 user is larger than that of the arriving class-$i$ user.

Evidently, $\rho_i < 1$ is a necessary condition for stability of $Q_i$, $i = 1, \ldots, L$, because otherwise $Q_i$ would be unstable even in the absence of any class-0 users. The next proposition shows that for weak SRPT with $m_0 > M_i$ this condition is sufficient as well.

**Proposition 4.3** *Suppose the service discipline is weak SRPT and $m_0 > M_i$. Then the condition $\rho_i < 1$ is sufficient for stability of $Q_i$.*

**Proof** As observed above, the fact that $m_0 > M_i$ implies that class $i$ receives preemptive priority over class 0, unless a class-0 user has a smaller remaining service requirement than all class-$i$ users (so at most $M_i$) and is being served. In the presence of this class-0 user, it depends on the other classes whether class 0 or class $i$ is being served. But, as long as $Q_i$ remains non-empty after the arrival of a new class-$i$ user, it will be prevented from service for at most a duration $M_i$, since weak SRPT does not leave any capacity in node $i$ unused when class $i$ is present. When this class-0 user leaves the system, no new class-0 users are taken into service under SRPT as long as class $i$ is present, since we assumed that $m_0 > M_i$. It follows that the number of class-$i$ users behaves as in an isolated queue with class $i$ only and random service interruptions whose total duration during each busy period is bounded by $M_i$. Lemma 4.4 implies that a queue with service interruptions of deterministic size in each busy period, is stable for any $\rho_i < 1$. $\square$

**Lemma 4.4** *Consider an M/G/1 queue with traffic load $\rho$ and with service interruptions. Assume that the total duration of the service interruptions in any contiguous period during which the queue is continuously backlogged is stochastically bounded by the random variable $M$. Further assume that $M$ is independent of the arrival process and the service requirements and that $\mathbb{E}(M) < \infty$. Then for any work-conserving policy the queue is stable when $\rho < 1$.*

**Proof** In an ordinary M/G/1 queue without service interruptions we know that if $\rho < 1$, then $\mathbb{E}(BP) < \infty$, where $BP$ is the random variable denoting the length of a busy period.

Let the random variable $C$ denote the length of a contiguous period during which the queue is continuously backlogged. With each user we can associate a sub-busy period during which that user is served, as well as users that arrived during that service time (not counting those that arrive when that service time is interrupted), those that arrived during the service of those users and so on. The period $C$ can now be split into the following three components: the service interruptions, the sub-busy periods of the user that arrived at an empty system when there is no service interruption (this user may not be present) and the sub-busy periods of the users that arrived during a service interruption. The expected number of users that arrive while the service is interrupted is bounded by $\lambda\mathbb{E}(M)$. We can therefore write

$$\mathbb{E}(C) \leq \mathbb{E}(M) + (1 + \lambda\mathbb{E}(M))\mathbb{E}(BP) < \infty.$$

This implies $\mathbb{P}(N = 0) > 0$, which establishes the stability of the queue. $\square$

The next proposition indicates that for strict SRPT the condition $\rho_i < 1$ is not sufficient in general for $Q_i$ to be stable.

**Proposition 4.5** *Suppose the service discipline is strict SRPT and $m_0 > M_i$. Then the condition for stability of $Q_i$ is*

$$\rho_i < 1 \text{ and } \rho_j(M_i) < 1 \text{ for all } j \neq 0, i.$$

**Proof** We first prove that the above condition is sufficient. The fact that $m_0 > M_i$ implies that class $i$ receives preemptive priority over class 0, and will be entitled to service, unless a class-0 user is present with a smaller remaining service requirement than all class-$i$ users, regardless of whether it is being served or not. Although the service of such a class-0 user may repeatedly be interrupted by arriving class-$j$ users, $j = 1, \ldots, L$, the latter users all have service requirements of at most $M_i$.

At the moment that $Q_i$ is empty and there is an arrival of a class-$i$ user, the time that class $i$ is prevented from service while $Q_i$ remains non-empty is denoted by $D_i$. The number of class-$i$ users behaves as in an isolated queue with class $i$ only and random service interruptions with a total duration of $D_i$ during each busy period. By Lemma 4.4, such a queue is stable for any $\rho_i < 1$, when $D_i$ has a finite mean. Therefore it remains to be shown that $\mathbb{E}(D_i) < \infty$.

$T_i$ is defined as the time it takes for a class-0 user with a remaining service requirement of $r_0 = M_i$, to receive the last $M_i$ part of its service. $D_i$ can be bounded from above by $T_i$, since class $i$ only notices the class-0 user, when $r_0 \leq M_i$. Note that at the moment that the class-0 user is being served and $r_0$ reaches the level $M_i$, because of SRPT, it is necessary that there are no other users present with remaining service requirement smaller than $M_i$, that is there are no class-$1, \ldots, L$ users present.

Denote by $r_0(t)$ the smallest remaining service requirement of all the class-0 users present at time $t$. A class-0 user with remaining service requirement smaller than $M_i$ is being served until a user of size smaller than $r_0(t_1)$ arrives at time $t_1$ (so it necessarily is of class $i$, $i = 1, \ldots, L$). This user preempts the class-0 user. The class-0 user can resume its service when all newly arrived users with size not larger than $r_0(t_1)$ have left the system. This period is called a busy period of classes $1, \ldots, L$.

After this busy period the class-0 user can enter service again, until at a certain time $t_2$ a user arrives with size smaller than $r_0(t_2)$ (such a user is necessarily of class $i$, $i = 1, \ldots, L$). A new busy period starts of class-$i$ users, $i = 1, \ldots, L$, with sizes smaller than or equal to

9

$r_0(t_2)$. This pattern repeats itself until the class-0 user has received its complete service and leaves the system.

Note that an upper bound for these busy periods is obtained when instead we look at the busy periods of users with size smaller than or equal to $M_i$. It is sufficient to show that $\tilde{T}_i$, defined as the total length of interruption in a busy period of this upper-bounding queue, has a finite mean. Since the class-0 user needs a total of $M_i$ service, we can conclude that $(1 - \rho_1(M_i)) \cdots (1 - \rho_L(M_i)) \mathbb{E}(\tilde{T}_i) = M_i$. Hence, the class-0 user will eventually complete service, i.e. $\mathbb{E}(T_i) < \infty$, since $\rho_j(M_i) < 1$ for all $j = 1, \ldots, L$. The fact that $\mathbb{E}(D_i) \leq \mathbb{E}(T_i) < \infty$ concludes the proof that the above condition is sufficient.

It remains to be shown that the above condition is necessary as well. $\rho_i < 1$ is clearly a necessary condition. To show that the second condition is necessary too, suppose it is not satisfied. Then $\rho_j(M_i) > 1$ for some $j$. Define $s_j = \sup\{s : \rho_j(s) \leq 1\}$, hence $s_j < M_i$. There is an $b$, $s_j < b \leq M_i$, such that there arrive class-$j$ users with sizes in the interval $(s_j, b]$. For these class-$j$ users the queue is unstable. Consider the last time epoch $t^*$ that a class-$j$ user arrives with size in the interval $(s_j, b]$ and in $Q_j$ there are no users with size less than or equal to $b$. With a certain non-zero probability, there is a class-0 user in the system at time $t^*$ with remaining service requirement $r$, with $s_j < b \leq r \leq M_i$. The service of this class-0 user will be preempted by the newly arrived class-$j$ user at time $t^*$, and the service will never be resumed, since $Q_j$ will never empty of class-$j$ users with size less than or equal to $b$ again from time $t^*$ onward. In the presence of this class-0 user, a possible present class-$i$ user with remaining service requirement smaller than $r$ can still be served, but after this class-$i$ user has left, no class-$i$ users with size greater than $r$ will ever enter service again. Hence $Q_i$ will grow indefinitely from time $t^*$ onward. $\square$

### 4.1.2 Stability of $Q_0$

We now turn to the stability of $Q_0$. To determine the sufficient condition for stability of $Q_0$, we will consider the network in a limiting regime, obtained by scaling the dynamics of some classes with a common parameter $\epsilon$ and passing $\epsilon \downarrow 0$. This technique is usually referred to as analytic perturbation or nearly-complete decomposability and has successfully been applied to study steady-state performance as a function of $\epsilon$, as $\epsilon \downarrow 0$, see for instance [3, 12].

We assume $B_0$ and $B_i$, $i = 1, \ldots, L$ to be generally distributed. We will consider a sequence of systems, indexed by $\epsilon$, where the class-$i$ arrival rate in the $\epsilon$-system is $\lambda_i^{(\epsilon)} := \lambda_i / \epsilon$ and the class-$i$ service requirements are $B_i^{(\epsilon)} := \epsilon B_i$, for $i = 1, \ldots, L$. Note that the traffic load of class $i$ in the $\epsilon$-system is $\rho_i^{(\epsilon)} = \frac{\lambda_i}{\epsilon} \epsilon \beta_i = \rho_i$, hence it is independent of $\epsilon$. Furthermore, when we let $\epsilon \downarrow 0$, the class-$i$ service requirements will become extremely small compared to class 0, so we are indeed in the situation of large class-0 users.

In the $\epsilon$-system we will make a distinction between class-$i$ users with original size smaller or larger than $\sqrt{\epsilon}$.

For $i = 1, \ldots, L$, we define $I_i^{\sqrt{\epsilon}}$ as a period where class-$i$ users with original size smaller than $\sqrt{\epsilon}$ are not served in the $\epsilon$-system. Denote by $N_i^{\sqrt{\epsilon}}(t)$ the number of class-$i$ users in the $\epsilon$-system with original size smaller than $\sqrt{\epsilon}$ present at time $t$. It is possible that $N_i^{\sqrt{\epsilon}}(t) > 0$ during a period $I_i^{\sqrt{\epsilon}}$, but that these class-$i$ users are blocked from service by a class-0 user

or a class-$i$ user with an original size larger than $\sqrt{\epsilon}$, and a remaining service requirement smaller than $\sqrt{\epsilon}$. We define $A_i^{\sqrt{\epsilon}}$, for $i = 1, \ldots, L$, as a period where class-$i$ users with original size smaller than $\sqrt{\epsilon}$ are served in the $\epsilon$-system. Note that in this period the total capacity of node $i$ is allocated to a class-$i$ user with original size smaller than $\sqrt{\epsilon}$. With minor abuse of notation, $I_i^{\sqrt{\epsilon}}$ and $A_i^{\sqrt{\epsilon}}$ will also be used to indicate that the event occurs at an arbitrary time epoch. In these definitions we implicitly assume the stationary distributions of $I_i^{\sqrt{\epsilon}}$ and $A_i^{\sqrt{\epsilon}}$ to exist.

Note that class-$i$ users with original size smaller than $\sqrt{\epsilon}$, for $i = 1, \ldots, L$, receive preemptive priority over all other users, unless one of the other users is being served with a remaining service requirement smaller than or equal to $\sqrt{\epsilon}$. The latter will occur at most a fraction of order $\sqrt{\epsilon}$ of the time, so class-$i$ users with original size smaller than $\sqrt{\epsilon}$ will receive priority over the other users virtually all the time as $\epsilon \downarrow 0$. Thus, class $i$ restricted to $\sqrt{\epsilon}$ will approximately behave as in an isolated queue with class $i$ restricted to $\sqrt{\epsilon}$ only as $\epsilon \downarrow 0$. Moreover, since $\rho_i^{(\epsilon)}(\sqrt{\epsilon}) \to \rho_i$ and classes $i$ for $i = 1, \ldots, L$ will behave roughly independently, this suggests that $\mathbb{P}(I_1^{\sqrt{\epsilon}}, \ldots, I_L^{\sqrt{\epsilon}}) \to \Pi_{i=1}^L (1 - \rho_i)$, as is confirmed by the next proposition.

**Proposition 4.6** *For the network under consideration in the limiting regime, with the weak SRPT discipline and $\rho_0 + \rho_i < 1$ for $i = 1, \ldots, L$, it holds that*

$$\lim_{\epsilon \downarrow 0} \mathbb{P}(I_1^{\sqrt{\epsilon}}, \ldots, I_L^{\sqrt{\epsilon}}) = \Pi_{i=1}^L (1 - \rho_i).$$

**Proof** As described earlier, class $i$ restricted to $\sqrt{\epsilon}$ behaves as in an isolated queue with class $i$ only and random service interruptions. Let us now introduce a reference system with class $i$ only and with the same arrival process and service requirements as in the original system, and class-$i$ users with sizes larger than $\sqrt{\epsilon}$ are rejected. Define $\hat{A}_i^{\sqrt{\epsilon}}$ and $\hat{I}_i^{\sqrt{\epsilon}}$ as the active and idle periods of the reference system, respectively.
Note that $\mathbb{P}(A_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}}) = \mathbb{P}(\hat{A}_i^{\sqrt{\epsilon}}) - \mathbb{P}(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}})$ and $\mathbb{P}(A_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}}) = \mathbb{P}(A_i^{\sqrt{\epsilon}}) - \mathbb{P}(\hat{I}_i^{\sqrt{\epsilon}}, A_i^{\sqrt{\epsilon}})$, so $\mathbb{P}(\hat{A}_i^{\sqrt{\epsilon}}) - \mathbb{P}(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}}) = \mathbb{P}(A_i^{\sqrt{\epsilon}}) - \mathbb{P}(\hat{I}_i^{\sqrt{\epsilon}}, A_i^{\sqrt{\epsilon}})$. Property 2.1 (ii) gives that $Q_1, \ldots, Q_L$ are stable in the original system, hence $\mathbb{P}(A_i^{\sqrt{\epsilon}}) = \rho_i^{(\epsilon)}(\sqrt{\epsilon})$. Since $\mathbb{P}(\hat{A}_i^{\sqrt{\epsilon}}) = \rho_i^{(\epsilon)}(\sqrt{\epsilon}) = \mathbb{P}(A_i^{\sqrt{\epsilon}})$, it follows that $\mathbb{P}(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}}) = \mathbb{P}(\hat{I}_i^{\sqrt{\epsilon}}, A_i^{\sqrt{\epsilon}})$.
We now proceed to derive an upper bound for the latter probabilities. Let us denote by $(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}})$ the event that the original system is in $I_i^{\sqrt{\epsilon}}$ and the reference system is in $\hat{A}_i^{\sqrt{\epsilon}}$. Observe that when the reference system is active at time $t$, i.e. $\hat{N}_i^{\sqrt{\epsilon}}(t) > 0$, it holds that $N_i^{\sqrt{\epsilon}}(t) > 0$, because $N_i^{\sqrt{\epsilon}}(t) \geq \hat{N}_i^{\sqrt{\epsilon}}(t)$. Thus, in order for the event $(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}})$ to occur, it must be the case that $N_i^{\sqrt{\epsilon}}(t) > 0$, i.e. there is a class-$i$ user with original size smaller than $\sqrt{\epsilon}$, but it is not served. As noted earlier, this can only arise when a class-$i$ user with original size greater than $\sqrt{\epsilon}$ or a class-0 user is present with a remaining service requirement smaller than $\sqrt{\epsilon}$.
Define $T_{[0,t]}(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}})$ as the amount of time that the event $(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}})$ occurs during the interval $[0, t]$. We have the bound $T_{[0,t]}(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}}) \leq \sum_{n=1}^{N_{[0,t]}} D_{i,n}^{\sqrt{\epsilon}}$, with $N_{[0,t]}$ denoting the number of class-0 users and class-$i$ users with original size larger than $\sqrt{\epsilon}$, that are served during the interval $[0, t]$; the index $n$ is used to denote the $n$-th such user, and $D_{i,n}^{\sqrt{\epsilon}}$ is the amount of time that class-$i$ users with original size smaller than $\sqrt{\epsilon}$ are prevented from service

because of user $n$. For weak SRPT, we have $D_{i,n}^{\sqrt{\epsilon}} \leq \sqrt{\epsilon}$, since no capacity is left unused in the presence of class-$i$ users and user $n$ needs only its last $\sqrt{\epsilon}$ amount of service.

Using the strong law of large numbers, we can conclude that for weak SRPT,

$$\mathbb{P}(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}}) = \lim_{t \to \infty} \frac{T_{[0,t]}(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}})}{t} \leq \lim_{t \to \infty} \frac{N_{[0,t]}}{t}\sqrt{\epsilon} \leq (\lambda_0 + \lambda_i^{(\epsilon)}\mathbb{P}(B_i^{(\epsilon)} > \sqrt{\epsilon}))\sqrt{\epsilon}. \quad (2)$$

Furthermore we have $\lim_{\epsilon \downarrow 0} \lambda_i^{(\epsilon)}\mathbb{P}(B_i^{(\epsilon)} > \sqrt{\epsilon})\sqrt{\epsilon} = \lim_{\epsilon \downarrow 0} \frac{\lambda_i}{\sqrt{\epsilon}}\mathbb{P}(B_i > \frac{1}{\sqrt{\epsilon}})$. It can be shown that when $\mathbb{E}(B_i) \leq \infty$, the last limit is equal to 0. Together with (2) we can conclude that

$$\lim_{\epsilon \downarrow 0}\mathbb{P}(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}}) = 0 \quad \text{for all } i = 1, \dots, L. \quad (3)$$

It follows that $\lim_{\epsilon \downarrow 0}\mathbb{P}(I_1^{\sqrt{\epsilon}}, \dots, I_L^{\sqrt{\epsilon}}) = \lim_{\epsilon \downarrow 0}\mathbb{P}(\hat{I}_1^{\sqrt{\epsilon}}, \dots, \hat{I}_L^{\sqrt{\epsilon}}) = \Pi_{i=1}^L(1 - \rho_i)$, where we use $\mathbb{P}(I_i^{\sqrt{\epsilon}}, \hat{A}_i^{\sqrt{\epsilon}}) = \mathbb{P}(\hat{I}_i^{\sqrt{\epsilon}}, A_i^{\sqrt{\epsilon}})$, the fact that $\hat{I}_1^{\sqrt{\epsilon}}, \dots, \hat{I}_L^{\sqrt{\epsilon}}$ are independent, $\mathbb{P}(\hat{I}_i^{\sqrt{\epsilon}}) = 1 - \rho_i^{(\epsilon)}(\sqrt{\epsilon})$ for all $\epsilon > 0$ and $\rho_i^{(\epsilon)}(\sqrt{\epsilon}) \to \rho_i$, for $i = 1, \dots, L$. $\square$

From Proposition 4.6 we can now derive the stability condition for $\epsilon$ small enough in an $\epsilon$-system.

**Corollary 4.7** *For the network under consideration in the limiting regime, with the weak SRPT discipline, we have (i) if $\rho_0 < \Pi_{i=1}^L(1 - \rho_i)$, then there exists an $\bar{\epsilon}$ such that $Q_0$ is stable in the $\epsilon$-system for every $\epsilon < \bar{\epsilon}$.*
*(ii) Conversely, if $\rho_0 > \Pi_{i=1}^L(1 - \rho_i)$, then there exists an $\bar{\epsilon}$ such that $Q_0$ is unstable in the $\epsilon$-system for every $\epsilon < \bar{\epsilon}$.*

**Proof** For (i), observe that $\rho_0 + \sum_{i=1}^L(\rho_i - \rho_i^{(\epsilon)}(\sqrt{\epsilon})) < \mathbb{P}(I_1^{\sqrt{\epsilon}}, \dots, I_L^{\sqrt{\epsilon}})$ is a sufficient condition for $Q_0$ to be stable, since at least one of the nodes works at full rate whenever the system is non-empty. Since $\rho_i^{(\epsilon)}(\sqrt{\epsilon}) \to \rho_i$, for $i = 1, \dots, L$, Proposition 4.6 implies that for any $\rho_0 < \Pi_{i=1}^L(1 - \rho_i)$ there exists an $\bar{\epsilon}$ such that $Q_0$ is stable in the $\epsilon$-system for every $\epsilon < \bar{\epsilon}$. Conversely, for (ii), for any $\rho_0 > \Pi_{i=1}^L(1 - \rho_i)$ there exists an $\bar{\epsilon}$ such that $\rho_0 > \mathbb{P}(I_1^{\sqrt{\epsilon}}, \dots, I_L^{\sqrt{\epsilon}})$, for all $\epsilon \leq \bar{\epsilon}$. To complete the proof, note that $\mathbb{P}(I_1^{\sqrt{\epsilon}}, \dots, I_L^{\sqrt{\epsilon}}) \geq \mathbb{P}(I_1, \dots, I_L)$ and that $\rho_0 > \mathbb{P}(I_1, \dots, I_L)$ implies that $Q_0$ is unstable. $\square$

## 4.2 Small class-0 users

We now turn the attention to the case where class 0 has smaller service requirements than all classes $i$, i.e., $M_0 < m_i$, for $i = 1, \dots, L$.

### 4.2.1 Stability of $Q_0$

In contrast to Corollary 4.7, the minimal condition $\rho_0 < 1$ can in this case already be *sufficient* for stability of $Q_0$. Moreover, for strict SRPT it is the exact stability condition for $Q_0$.

**Observation 4.8** *When the service discipline is strict SRPT and $M_0 < m_i$ for all $i = 1, \dots, L$, then the condition for stability of $Q_0$ is $\rho_0 < 1$.*

This may be deduced as follows. The fact that $M_0 < m_i$ implies that class 0 receives preemptive priority over class $i$, and will be entitled to service, unless a class-$i$ user, for some $i = 1, \ldots, L$, has a smaller remaining service requirement than all class-0 users (so at most $M_0$). Class 0 has to wait until those users with remaining service requirement smaller than all class-0 users have left the network. Since it is strict SRPT, no new class-$1, \ldots, L$ users are taken into service. Thus, as long as $Q_0$ remains non-empty after the arrival of a new class-0 user, it will be prevented from service for at most a period $M_0$. From this it follows that the number of class-0 users behaves as in an isolated queue with class 0 only and random service interruptions whose total duration during each busy period is bounded by $M_0$. By Lemma 4.4, such a queue is stable for any $\rho_0 < 1$.

Under weak SRPT, $\rho_0 < 1$ is not necessarily sufficient for stability of $Q_0$. However, first we will illustrate a situation in which it is a sufficient condition. For that purpose we consider deterministic service requirements and $L = 2$ nodes.

**Proposition 4.9** *Assume class $i$ has a deterministic service requirement $d_i$, $i = 1, 2$, with $d_1 \neq d_2$ and $d_0 < d_1, d_2$, or $d_1 = d_2 > 2d_0$. For the network under consideration with the weak SRPT discipline, $Q_0$ is stable if and only if $\rho_0 < 1$.*

**Proof** The fact that $d_0 < d_i$ implies that class 0 receives preemptive priority over class $i$, and will be entitled to service, unless a class-$i$ user, for some $i = 1, \ldots, L$, has a smaller remaining service requirement than $d_0$. Although class-$1, \ldots, L$ users may continue to be served for a while, the delay incurred by a newly arrived class-0 user is bounded as will be shown below. Thus, as long as $Q_0$ remains non-empty after the arrival of a new class-0 user, it will be prevented from service for at most a bounded period. Now it follows that the number of class-0 users behaves as in an isolated queue with class 0 only and random service interruptions whose total duration during each busy period is bounded. By Lemma 4.4, such a queue is stable for any $\rho_0 < 1$.

It remains to be shown that the delay incurred by a newly arrived class-0 user is bounded. Suppose that class 0 could be prevented from entering service indefinitely. Then at a certain point in time we have for example a class-1 user with a remaining service requirement $r_1 < d_0$ as well as class-0 and class-2 users of which none have received any service. Because of weak SRPT, the class-1 and class-2 users are served. When the class-1 user leaves the system, the class-2 user has remaining service requirement of $r_2 = d_2 - r_1$. When $r_2$ is smaller than $d_0$, this class-2 user is served and because of weak SRPT, a class-1 user also receives service. In order for this to repeat indefinitely, it is necessary that

$$r_1 < d_0, \;\; 0 < d_2 - r_1 < d_0, \;\; 0 < d_1 - d_2 + r_1 < d_0,$$

$$2d_2 - d_1 - r_1 < d_0, \;\; 0 < 2d_1 - 2d_2 + r_1 < d_0, \ldots$$

or equivalently

$$k(d_1 - d_2) + r_1 < d_0 \;\; \text{and} \;\; k(d_2 - d_1) + d_2 - r_1 < d_0, \;\; \forall \, k \geq 0. \tag{4}$$

When $d_1 \neq d_2$, we can choose a $K$, such that for all $r_1 < d_0$ there exists a $k = k(r_1) < K$ for which (4) is not satisfied. When $d_1 = d_2 > 2d_0$, we may choose $K = 1$. We can conclude that at some point in time all class-1 and class-2 users have remaining service requirements greater than $d_0$, so a class-0 user can enter service. The delay for class 0 is therefore bounded

by $(K+1)d_0$, independent of $r_1$. $\square$

In general, $\rho_0 < 1$ is not a sufficient condition for stability of $Q_0$ under weak SRPT, as may be illustrated again with deterministic service requirements and $L = 2$ nodes. Take $d_1 = d_2 = d$ with $d_0 < d < 2d_0$ and assume that $Q_1$ and $Q_2$ are both unstable. In that case, the staggered service pattern of class-1 and class-2 users described in the proof of the above proposition may in fact replicate itself ad infinitum and class 0 can never return to service. Hence, $Q_0$ may also become unstable with non-zero probability. If $Q_1$ or $Q_2$ is stable, which is the case if $\rho_0 + \rho_1 < 1$ or $\rho_0 + \rho_2 < 1$, then with probability 1 the above cycle cannot repeat indefinitely, and it may in fact be checked that $Q_0$ is stable.

### 4.2.2 Stability of $Q_i$, for $i = 1, \ldots, L$

We will now investigate the conditions for stability of $Q_i$.
Under weak SRPT, it follows from Property 2.1 that $\rho_0 + \rho_i < 1$ is a sufficient condition for stability of $Q_i$.
Under strict SRPT, $\rho_0 + \rho_i < 1$ will in general not be sufficient for stability of $Q_i$, $i = 1, \ldots, L$. We will show this by considering again deterministic service requirements and $L = 2$ nodes. As noted earlier, if $\rho_0 + \rho_i + w_i < 1$, then node $i$, and hence $Q_i$, is stable. Here the long-term average wasted service rate, $w_i$, is precisely the fraction of time that there are no class-$i$ users with remaining service requirement smaller than $d_0$ and there are new class-0 users in the system which cannot be served because of the presence of a class-$j$ user with a remaining service requirement smaller than $d_0$, $j \neq 0, i$. During this time no class-0 or class-$i$ users can be served. Observe that $w_i > 0$ since there is a non-zero probability that an arriving class-0 user finds $Q_0$ empty and a class-$j$ user in service with a remaining service requirement smaller than $d_0$, $j \neq 0, i$. An explicit expression for $w_i$ appears hard to find.

## 5 LAS scheduling

In this section we consider the Least Attained Service first (LAS) discipline. For the exposition, the assumption below will sometimes be convenient. With appropriate modifications, the results extend to distributions with atoms.

**Assumption 5.1** *The service requirements of all classes have general **continuous** distributions.*

As with SRPT, we again distinguish two variants of LAS. In each node, the users with the least attained service are granted the right to an equal share of the capacity at that node. Class-0 users only receive the minimum of the granted shares at the nodes. This may leave some capacity unused at the nodes with the larger relative proportion of class-0 users. With *weak* LAS, the unused capacity is re-allocated to the other class at that node (if there are users of that class). In case of *strict* LAS, the unused capacity is simply lost.
The subsequent analysis is facilitated by a particular property of LAS: the users with a total service requirement $x$ are not influenced by users that have received more than $x$ in service. It will be convenient to define the following quantities, which we refer to as truncated loads:

$$\tilde{\rho}_i(x) := \lambda_i \int_{y=0}^{x^-} y \, \mathrm{d}B_i(y) + \lambda_i x \mathbb{P}(B_i \geq x) = \rho_i(x) + \lambda_i x \mathbb{P}(B_i \geq x),$$

where $\rho_i(x)$ was previously defined in (1). Thus, $\tilde{\rho}_i(x)$ represents the load due to class-$i$ users *truncated* at size $x$ (users larger than or equal to $x$ contribute an amount $x$, rather than zero as in $\rho_i(x)$). We call the system obtained by truncating the sizes of class-$i$ users at $x_i$, $i = 0, \ldots, L$, the $(x_0, \ldots, x_L)$-truncated system. If $x_0 = \ldots = x_L = x$ we simply refer to the "$x$-truncated" system. The $\infty$-truncated system corresponds to the original one.

**Property 5.2** *From the perspective of users of size $x$, the system dynamics are identical to those of the $x$-truncated system. In addition, if $\mathbb{P}(B_0 \leq \bar{x}_0) = 1$, then from the perspective of class-$j$ users of size $x_j > \bar{x}_0$ for an $j = 1, \ldots, L$, the system behaves identically to the $(\infty, \bar{x}_0, \ldots, \bar{x}_0, x_j, \bar{x}_0, \ldots, \bar{x}_0)$-truncated system, with $x_j$ in the $j$-th component.*

While the first claim is immediate from the arguments above, the second statement deserves some elaboration. The influence of class $i$, with $i \neq 0, j$, on class $j$ is through class-0 users. If no class-0 user is larger than $\bar{x}_0$, then class-$i$ users larger than $\bar{x}_0$ have no effect on the class-0 users, and therefore no influence on the class-$j$ users either.

By choosing $x$ small enough, we can ensure that $\sum_{i=0}^{L} \tilde{\rho}_i(x) < 1$, i.e., there exists a stable $x$-truncated system. It follows from Property 5.2 that class-$i$ users of size at most $x$ experience a stable system if and only if $Q_i$ is stable in the $x$-truncated system, for $i = 0, \ldots, L$ and that stability is monotone with respect to truncation: if $(x_0, \ldots, x_L) \geq (y_0, \ldots, y_L)$ component-wise and $Q_i$ is stable in the $(x_0, \ldots, x_L)$-truncated system, then so is $Q_i$ in the $(y_0, \ldots, y_L)$-truncated system. Under Assumption 5.1 we can therefore define

$$x_i^* := \sup\{x : Q_i \text{ is stable in the } x-\text{truncated system}\},$$

i.e., all class-$i$ users of size smaller than $x_i^*$ complete service in the original system. In the long run, class-$i$ users of size larger than $x_i^*$ only receive an amount of service equal to $x_i^*$. Note that $Q_i$ is stable in the original system if $\mathbb{P}(B_i \leq x_i^*) = 1$.

**Proposition 5.3** *Suppose that Assumption 5.1 is satisfied and that all service requirement distributions have infinite supports, i.e. $\mathbb{P}(B_i > x) > 0$ for all $x$. For weak LAS, $x_0^* \leq x_i^*$, for $i = 1, \ldots, L$ and for strict LAS, $x_0^* = x_i^*$, for $i = 1, \ldots, L$.*

**Proof** The idea of the proof is similar to that of Proposition 4.2 and may be described as follows. Under LAS, class-0 users of a given size $x$ cannot complete service until all class-$i$ users of original size smaller than or equal to $x$ have been cleared from the system. Thus, if the class-0 users of a given size eventually leave the system, then class-$i$ users must do so as well, which implies $x_0^* \leq x_i^*$. Moreover, for strict LAS the reverse implication also holds.
In order to formalize the above arguments, suppose $x_0^* > x_i^*$, for some $i = 1, \ldots, L$. For both weak and strict LAS, we can reach a contradiction as follows. By definition of $x_i^*$, in the long run, when a class-$i$ user arrives of size $s_i$, $x_i^* < s_i < x_0^*$, it will never leave the system. Now suppose that subsequently a class-0 user arrives of size $s_0$, $s_i < s_0 < x_0^*$. Because of LAS, this user cannot complete service until the class-$i$ user does. Since the latter never happens, the class-0 user never leaves the system either, which contradicts the assumption $s_0 < x_0^*$ and the definition of $x_0^*$. Thus, for both weak and strict LAS, $x_0^* \leq x_i^*$.
For strict LAS, it may be shown using similar arguments that $x_0^* < x_i^*$ leads to a contradiction. Thus, for strict LAS we in fact have $x_0^* = x_i^*$. $\square$

## 5.1 Small class-0 users

In this section we consider class-0 users with small service requirements, compared to the service requirements of class-$i$ users, $i = 1, \ldots, L$. As before, we study a sequence of systems indexed by $\epsilon$ and let $\epsilon \downarrow 0$. In the $\epsilon$-system, class-0 users arrive according to a Poisson process of rate $\lambda_0^{(\epsilon)} := \lambda_0/\epsilon$ and the sizes are distributed as $\epsilon B_0$. The next proposition shows that the standard conditions can be arbitrarily close to sufficient for stability. We will consider the $\epsilon$-system with $B_0^{(\epsilon)}$ truncated at $h(\epsilon)$, with $\lim_{\epsilon \downarrow 0} h(\epsilon) = 0$.

**Proposition 5.4** *Suppose the service discipline is weak or strict LAS and assume $\lim_{\epsilon \downarrow 0} h(\epsilon) = 0$. If for some $i = 1, \ldots, L$, $\rho_0 + \rho_i < 1$, then there exists an $\bar{\epsilon}$ such that for all $0 < \epsilon < \bar{\epsilon}$, node $i$ is stable in the $\epsilon$-system with $B_0^{(\epsilon)}$ truncated at $h(\epsilon)$.*
*Hence, if $\rho_0 + \rho_i < 1$ for all $i = 1, \ldots, L$, then there exists an $\tilde{\epsilon}$ such that the $\epsilon$-system with $B_0^{(\epsilon)}$ truncated at $h(\epsilon)$ is stable for all $0 < \epsilon < \tilde{\epsilon}$.*

**Proof** Assume $\rho_0 + \rho_i < 1$, for an $i = 1, \ldots, L$. We have $\lim_{\epsilon \downarrow 0} h(\epsilon) = 0$, so there is an $\bar{\epsilon}$ such that

$$\rho_0 + \rho_i + \sum_{j=1, j \neq i}^{L} \tilde{\rho}_j(h(\epsilon)) < 1, \text{ for all } \epsilon < \bar{\epsilon}. \tag{5}$$

From this it follows that $\tilde{\rho}_0^{(\epsilon)}(h(\epsilon)) + \sum_{j=1}^{L} \tilde{\rho}_j(h(\epsilon)) < 1$, for all $\epsilon < \bar{\epsilon}$, which, according to Property 2.1, is a sufficient condition for stability of the $h(\epsilon)$-truncated system. The service requirements of class 0 in the $\epsilon$-system are bounded by $h(\epsilon)$. According to Property 5.2, $Q_0$ is stable, since $Q_0$ is stable in the $h(\epsilon)$-truncated system.
For class $i$, Property 5.2 implies that $Q_i$ is stable in the $(h(\epsilon), \infty, \ldots, \infty)$-truncated system if and only if $Q_i$ is stable in the $(h(\epsilon), \ldots, h(\epsilon), \infty, h(\epsilon), \ldots, h(\epsilon))$-truncated system, with $\infty$ in the $i$-th component. Because of Property 2.1, for the latter it is sufficient to have (5), which holds for $\epsilon < \bar{\epsilon}$. $\square$

**Remark 5.5** *The fact that we can choose $h(\epsilon)$ such that $\lim_{\epsilon \downarrow 0} h(\epsilon)/\epsilon = \infty$, and thus $\mathbb{P}(B_0^{(\epsilon)} \leq h(\epsilon)) \to 1$, as $\epsilon \downarrow 0$, suggests that the non-truncated $\epsilon$-system can be arbitrarily closely approximated by the truncated one. However, the proof of Proposition 5.4 relies on the truncation of $B_0^{(\epsilon)}$. In the particular case that $B_0$ is bounded from above by a constant $M$, Proposition 5.4 does imply that the condition $\rho_0 + \rho_i < 1$ is sufficient for stability of node $i$ in the $\epsilon$-system for $\epsilon$ small enough (take $h(\epsilon) = \epsilon M$).*

In the next proposition we compute the limit of $x_i^*(\epsilon)$, for $i = 1, \ldots, L$, in the $\epsilon$-system as $\epsilon \downarrow 0$, in the case that the distribution of $B_0$ has bounded support. Note that Proposition 5.3 does not apply because the distribution of $B_0$ does not have infinite support. That is why we do not have $x_i^* = x_j^*$.

**Proposition 5.6** *Let Assumption 5.1 be satisfied, $\rho_0 < 1$ and $B_0$ be bounded by a constant $M$. For weak and strict LAS*

$$\lim_{\epsilon \downarrow 0} x_i^*(\epsilon) = x_i^*(0) := \sup\{x_i : \rho_0 + \tilde{\rho}_i(x_i) < 1\}, \quad i = 1, \ldots, L.$$

16

**Proof** Class 0 in the $\epsilon$-system does not notice a truncation at a constant level $x$ for $\epsilon$ small enough, because $B_0^{(\epsilon)}$ is bounded by $\epsilon M$ which approaches 0 as $\epsilon \downarrow 0$. So for $\epsilon$ small enough, we have $\tilde{\rho}_0^{(\epsilon)}(x) = \rho_0$.

Take an $x_i < x_i^*(0)$, so $\rho_0 + \tilde{\rho}_i(x_i) < 1$. From Remark 5.5 it follows that there is an $\bar{\epsilon}$ such that for $\epsilon < \bar{\epsilon}$ the $x_i$-truncated system is stable, or equivalently $x_i < x_i^*(\epsilon)$ (by definition of $x_i^*(\epsilon)$). So for every $x_i < x_i^*(0)$, there is an $\bar{\epsilon}$ such that $x_i < x_i^*(\epsilon)$ for all $\epsilon < \bar{\epsilon}$. We also have $x_i^*(\epsilon) \leq x_i^*(0)$. Together this implies that $\lim_{\epsilon \downarrow 0} x_i^*(\epsilon) = x_i^*(0)$. $\square$

## 5.2 Large class-0 users

We now show that the standard conditions $\rho_0 + \rho_i < 1$, $i = 1, \ldots, L$, are in general not sufficient for stability under LAS scheduling. Loosely speaking, we show that if class 0 has extremely large service requirements compared to all other classes, then it is necessary that $\rho_0 \leq \prod_{i=1}^{L}(1 - \rho_i)$. We consider an $\epsilon$-system, in which class-0 users arrive according to a Poisson process of rate $\lambda_0^{(\epsilon)} := \epsilon \lambda_0$ and sizes are distributed as $B_0/\epsilon$. The proof of the next proposition uses that, in the limit as $\epsilon \downarrow 0$, all classes $i \neq 0$ behave as if there is no class-0 traffic.

**Proposition 5.7** *Assume the service discipline is either weak or strict LAS. If there exists an $\bar{\epsilon}$ such that $Q_0$ is stable in the $\epsilon$-system, for all $0 < \epsilon < \bar{\epsilon}$, then it must be that $\rho_0 \leq \prod_{i=1}^{L}(1-\rho_i)$.*

**Proof** Let us focus on the $\epsilon$-system. If $Q_0$ is stable, there must be sufficient capacity to serve all its traffic. In particular, this must be true for traffic due to class-0 users of total size larger than $h$. Once these users have received an amount of service equal to $h$, they can only be served when no users are present with attained service less than $h$. Therefore

$$\rho_0 - \tilde{\rho}_0^{(\epsilon)}(h) \leq \mathbb{P}(N_1^{(\epsilon,h)} = N_2^{(\epsilon,h)} = \cdots = N_L^{(\epsilon,h)} = 0),$$

where $N_i^{(\epsilon,h)}$ denotes the number of class-$i$ users with attained service less than $h$.

Choose $h = h(\epsilon) = 1/\sqrt{\epsilon}$. Since $\tilde{\rho}_0^{(\epsilon)}(h(\epsilon)) \to 0$ and $\tilde{\rho}_i(h(\epsilon)) \to \rho_i$ as $\epsilon \downarrow 0$, it is sufficient to show that $\mathbb{P}(N_1^{(\epsilon,h)} = N_2^{(\epsilon,h)} = \cdots = N_L^{(\epsilon,h)} = 0) \leq \prod_{i=1}^{L}(1 - \tilde{\rho}_i(h))$ for all $\epsilon > 0$ and $h > 0$. We will show this by comparing the backlogs of classes $i = 1, \ldots, L$, with those in the same system without class 0. The backlog is the sum of the remaining service requirements of all users. Since $\epsilon$ will remain fixed in the remainder of the proof, we suppress the dependence on $\epsilon$ for notational convenience. Let us denote the backlog of class $i$ in the $h$-truncated system at time $t$ by $V_i^h(t)$, and that in the $h$-truncated reference system by $\hat{V}_i^h(t)$. We further represent – both for the original and the reference system – the amount of traffic of class $i = 1, \ldots, L$, truncated at $h$ that arrives in the time interval $(s, t)$ by $A_i^h(s, t)$. In the original system we also define the amount of service given to class-0 users in $(s, t)$ in the $h$-truncated system by $B_0^h(s, t)$ and the capacity wasted in $(s, t)$ at node $i$ while there is at least one class-$i$ user that has received at most $h$ in service by $W_i^h(s, t)$. Assume both systems are empty at time 0. Then for $i = 1, \ldots, L$,

$$V_i^h(t) = \sup_{s \in [0,t]} \{A_i^h(s,t) + B_0^h(s,t) + W_i^h(s,t) - (t - s)\} \geq \sup_{s \in [0,t]} \{A_i^h(s,t) - (t - s)\} = \hat{V}_i^h(t),$$

so that

$$
\begin{aligned}
\mathbb{P}(N_1^h = N_2^h = \cdots = N_L^h = 0) &= \lim_{t \to \infty} \mathbb{P}(V_1^h(t) = V_2^h(t) = \cdots = V_L^h(t) = 0) \\
&\leq \lim_{t \to \infty} \mathbb{P}(\hat{V}_1^h(t) = \hat{V}_2^h(t) = \cdots = \hat{V}_L^h(t) = 0) \\
&= \prod_{i=1}^{L}(1 - \tilde{\rho}_i(h)),
\end{aligned}
$$

where the last equality follows from the independence of the various classes in the reference system. $\square$

For weak LAS, we can even proof that in the limiting regime considered here, $\rho_0 < \Pi_{i=1}^{L}(1-\rho_i)$ is also a sufficient condition for stability of the $\epsilon$-system.

**Proposition 5.8** *Assume the service discipline is weak LAS and the class-0 users are large compared to classes $i = 1, \ldots, L$. If $\rho_0 < \Pi_{i=1}^{L}(1-\rho_i)$, then for $\epsilon$ small enough the $\epsilon$-system is stable.*

**Proof** We will use the same notation as in the proof of Proposition 5.7. $N_i^h > 0$ implies that $1 - s_0^h(u) - s_i^h(u) = 0$, because the unused capacity is re-allocated to class-$i$ users with attained service less than $h$, when the service discipline is weak LAS. $W_i^h(0,t)$ can be written as $\int_0^t (1 - s_0^h(u) - s_i^h(u))\mathbf{I}_{\{N_i^h(u)>0\}}du$, so for weak LAS, $W_i^h(0,t)$ is equal to 0.

Again focus on the $\epsilon$-system. We can conclude that

$$
\begin{aligned}
V_i^{h(\epsilon)}(t) &= \sup_{s \in [0,t]} \{A_i^{h(\epsilon)}(s,t) + B_0^{h(\epsilon)}(s,t) - (t-s)\} \\
&\leq \sup_{s \in [0,t]} \{A_i^{h(\epsilon)}(s,t) - (1 - g(\epsilon))(t-s)\} \\
&\quad + \sup_{s \in [0,t]} \{B_0^{h(\epsilon)}(s,t) - g(\epsilon)(t-s)\}. \quad (6)
\end{aligned}
$$

Choose $h(\epsilon) = \frac{1}{\sqrt{\epsilon}}$, so that $\tilde{\rho}_0^{(\epsilon)}(h(\epsilon)) \to 0$ and $\tilde{\rho}_i(h(\epsilon)) \to \rho_i$, for $i = 1, \ldots, L$. In addition, let $g(\epsilon)$ be such that $\lim_{\epsilon \downarrow 0} g(\epsilon) = 0$ and $\lim_{\epsilon \downarrow 0} \tilde{\rho}_0^{(\epsilon)}(h(\epsilon))/g(\epsilon) = 0$. By (6) we have

$$
\begin{aligned}
&\mathbb{P}(N_1^{h(\epsilon)}(t) = \ldots = N_L^{h(\epsilon)}(t) = 0) \\
&= \mathbb{P}(V_1^{h(\epsilon)}(t) = \ldots = V_L^{h(\epsilon)}(t) = 0) \\
&\geq \mathbb{P}(\sup_{s \in [0,t]} \{A_i^{h(\epsilon)}(s,t) - (1 - g(\epsilon))(t-s)\} = 0, \; \forall i = 1, \ldots, L; \\
&\qquad \sup_{s \in [0,t]} \{B_0^{h(\epsilon)}(s,t) - g(\epsilon)(t-s)\} = 0) \\
&\geq \mathbb{P}(\sup_{s \in [0,t]} \{A_i^{h(\epsilon)}(s,t) - (1 - g(\epsilon))(t-s)\} = 0, \; \forall i = 1, \ldots, L) \quad (7) \\
&\quad -\mathbb{P}(\sup_{s \in [0,t]} \{B_0^{h(\epsilon)}(s,t) - g(\epsilon)(t-s)\} > 0).
\end{aligned}
$$

For $\epsilon$ small enough, the limit of (7) in the time-average sense, as $t \to \infty$, is $\Pi_{i=1}^{L}(1 - \frac{\tilde{\rho}_i(h(\epsilon))}{1-g(\epsilon)})$, using the independence of the arrival processes. Interpreting $\sup_{s \in [0,t]} \{B_0^{h(\epsilon)}(s,t) - g(\epsilon)(t-s)\}$

as the workload in a queue with input process $B_0^{h(\epsilon)}(s,t)$ and constant service rate $g(\epsilon)$, we have that the time-average limit of $\mathbb{P}(\sup_{s \in [0,t]}\{B_0^{h(\epsilon)}(s,t) - g(\epsilon)(t-s)\} > 0)$ can not be more than $\limsup_{t \to \infty} \frac{B_0^{h(\epsilon)}(0,t)}{g(\epsilon)t} \leq \lim_{t \to \infty} \frac{A_0^{h(\epsilon)}(0,t)}{g(\epsilon)t} = \tilde{\rho}_0^{(\epsilon)}(h(\epsilon))/g(\epsilon)$, where the latter limit holds with probability 1. Therefore

$$\mathbb{P}(N_1^{h(\epsilon)} = \ldots = N_L^{h(\epsilon)} = 0) \geq \Pi_{i=1}^L (1 - \frac{\tilde{\rho}_i(h(\epsilon))}{1-g(\epsilon)}) - \frac{\tilde{\rho}_0^{(\epsilon)}(h(\epsilon))}{g(\epsilon)}.$$

By choice of $h(\epsilon)$ and $g(\epsilon)$, the quantity on the right-hand side tends to $\Pi_{i=1}^L(1 - \rho_i)$, as $\epsilon \downarrow 0$. The capacity available in the $\epsilon$-system to serve class-$i$ users with attained service greater than $h(\epsilon)$, $i = 1, \ldots, L$, and for work of class 0, is not less than $\mathbb{P}(N_1^{h(\epsilon)} = \ldots = N_L^{h(\epsilon)} = 0)$. By Property 2.1 it is therefore sufficient for stability to have

$$\rho_0 + \sum_{i=1}^L (\rho_i - \tilde{\rho}_i(h(\epsilon))) < \mathbb{P}(N_1^{h(\epsilon)} = \ldots = N_L^{h(\epsilon)} = 0),$$

which is true for $\epsilon$ small enough. $\square$

# 6   Conclusion

We have explored the fundamental stability properties of size-based scheduling strategies in resource-sharing networks with simultaneous resource possession. In particular, we established the exact stability conditions for the SRPT and LAS disciplines in various limiting regimes. The results indicate that size-based scheduling strategies may fail to use the available resources efficiently, and in fact cause instability effects, even at arbitrarily low traffic loads. The qualitative findings specifically confirm the tendency for users with long routes and large service requirements to experience severe performance degradation.

The results imply that the prototypical size-based scheduling strategies will certainly not yield optimal delay performance in resource-sharing networks. Instead, proper tuning of the parameters of so-called (weighted) $\alpha$-fair bandwidth-sharing policies as introduced in [26] might provide a more promising approach for improving the delay performance. As is proved in [9], the usual necessary conditions are in fact sufficient for these policies to achieve stability. It is noteworthy that in single-link scenarios $\alpha$-fair policies essentially reduce to DPS disciplines, which are known to cover the entire achievable mean-delay region in that case [17].

Because of scalability issues in the core and the fact that congestion tends to occur at the edge, implementation of size-based scheduling strategies would be particularly appropriate in edge routers. A further interesting thread would thus be the study of size-based scheduling in so-called star topologies, which provide a natural paradigm for modeling networks with a high-capacity transparent core.

# References

[1] Aalto, S., Ayesta, U., Nyberg-Oksanen, E. (2004). Two-level processor sharing scheduling disciplines: mean delay analysis. In: *Proc. ACM Sigmetrics & Performance 2004 Conf.*, New York NY, USA, 97–105.

[2] Aalto, S., Ayesta, U., Nyberg-Oksanen, E. (2004). M/G/1 MLPS compared to M/G/1 PS. *INRIA Technical Report* RR-5219.

[3] Altman, E., Avrachenkov, K.E., Núñez-Queija, R. (2004). Perturbation analysis for denumerable Markov chains with application to queueing models. *Adv. Appl. Prob.* **36**, 839–853.

[4] Altman, E., Jimenez, T., Kofman, D. (2004). DPS queues with stationary ergodic service times and the performance of TCP in overload. In: *Proc. IEEE Infocom 2004*, Hong Kong.

[5] Avrachenkov, K., Ayesta, U., Brown, P. (2003) Batch arrival processor sharing with application to multi-level processor sharing scheduling. *INRIA Technical Report* RR-5043.

[6] Avrachenkov, K., Ayesta, U., Brown, P., Núñez-Queija, R. (2005). Discriminatory Processor Sharing revisited. In: *Proc. IEEE Infocom 2005*, Miami FL, USA.

[7] Bansal, N., Harchol-Balter, M. (2001). Analysis of SRPT scheduling: investigating unfairness. In: *Proc. ACM Sigmetrics & Performance 2001 Conf.*, Boston MA, USA, 279–290.

[8] Bonald, T., Borst, S.C., Hegde, N., Proutière, A. (2004). Wireless data performance in multi-cell scenarios. In: *Proc. ACM Sigmetrics & Performance 2004 Conf.*, New York NY, USA, 378–388.

[9] Bonald, T., Massoulié, L. (2001). Impact of fairness on Internet performance. In: *Proc. ACM Sigmetrics & Performance 2001 Conf.*, Boston MA, USA, 82–91.

[10] Bramson, M. (1994). Instability of FIFO queueing networks. *Ann. Appl. Prob.* **4**, 414–431.

[11] Chen, X., Heidemann, J. (2003). Preferential treatment for short flows to reduce Web latency. *Comp. Netw.* **41**, 779–794.

[12] Courtois, P.J. (1977). *Decomposability: Queueing and Computer System Applications.* Academic Press, New York.

[13] Crovella, M., Bestavros, A. (1996). Self-similarity in World Wide Web traffic: evidence and possible causes. In: *Proc. ACM Sigmetrics '96 Conf.*, 160–169.

[14] Fayolle, G., Mitrani, I., Iasnogorodski, R. (1980). Sharing a processor among many job classes. *J. ACM* **27**, 519–532.

[15] Feng, H., Misra, V. (2003). Mixed scheduling disciplines for network flows. *ACM SIGMETRICS Performance Evaluation Review* **31**, 36–39.

[16] Friedman, E., Henderson, S.G. (2003). Fairness and efficiency in Web servers. In: *Proc. ACM Sigmetrics 2003 Conf.*, San Diego CA, USA, 229–237.

[17] Gelenbe, E., Mitrani, I. (1980). *Analysis and Synthesis of Computer Systems.* Academic Press, London.

[18] Harchol-Balter, M., Schroeder, B., Bansal, N., Agrawal, M. (2003). Size-based scheduling to improve Web performance. *ACM Trans. Comp. Syst.* **21**, 207–233.

[19] Harchol-Balter, M. Sigman, K., Wierman, A. (2002). Asymptotic convergence of scheduling policies with respect to slowdown. In: *Proc. Performance 2002 Conf.*, Rome, Italy.

[20] Key, P.B., Massoulié, L., Wang, B. (2004). Emulating low-priority transport at the application layer: a background transfer service. In: *Proc. ACM Sigmetrics & Performance 2004 Conf.*, New York NY, USA, 118–129.

[21] Kleinrock, L. (1967). Time-shared systems: a theoretical treatment. *J. ACM* **14**, 242–261.

[22] Kleinrock, L. (1976). *Queueing Systems, Vol. II: Computer Applications.* Wiley, New York.

[23] Kuzmanovic, A., Knightly, E.W. (2003). TCP-LP: a distributed algorithm for low-priority data transfer. In: *Proc. IEEE Infocom 2003*, San Francisco CA, USA.

[24] Lu, S.H., Kumar, P.R. (1991). Distributed scheduling based on due dates and buffer priorities. *IEEE Trans. Aut. Control* **36**, 1406–1416.

[25] Massoulié, L., Roberts, J.W. (1999). Bandwidth sharing: objectives & algorithms. In: *Proc. IEEE Infocom '99*, New York NY, USA, 1395–1403.

[26] Mo, J., Walrand, J. (2000). Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Netw.* **8**, 556–567.

[27] Rai, I.A., Urvoy-Keller, G., Biersack, E.W. (2003). Analysis of LAS scheduling for job size distributions with high variance. In: *Proc. ACM Sigmetrics 2003 Conf.*, San Diego CA, USA, 218–228.

[28] Rai, I.A., Urvoy-Keller, G., Biersack, E.W. (2002). Size-based scheduling with differentiated services to improve response time of highly-varying flow sizes. In: *Proc. 15th ITC Specialist Seminar on Internet Traffic Engineering and Traffic Management*, Würzburg, Germany.

[29] Rai, I.A., Urvoy-Keller, G., Biersack, E.W. (2004). LAS scheduling approach to avoid bandwidth hogging in heterogeneous TCP networks. In: *7th IEEE International Conference on High-Speed Networks and Multimedia Communications HSNMC'04*, Toulouse, France.

[30] Rai, I.A., Urvoy-Keller, G., Vernon, M.K., Biersack, E.W. (2004). Performance analysis of LAS-based scheduling disciplines in a packet-switched network. In: *Proc. ACM Sigmetrics & Performance 2004 Conf.*, New York NY, USA, 106–117.

[31] Righter, R. (1994). Scheduling. In: *Stochastic Orders and their Applications*, eds. M. Shaked and J.G. Shanthikumar, Academic Press, 381–432.

[32] Schrage, L.E. (1968). A proof of the optimality of the shortest remaining processing time discipline. *Oper. Res.* **16**, 687–690.

[33] Schrage, L.E., Miller, L.W. (1966). The queue M/G/1 with the shortest remaining processing time discipline. *Oper. Res.* **14**, 670–684.

[34] Schroeder, B., Harchol-Balter, M. (2003). Web servers under overload: How scheduling can help. In: *Proc. ITC-18*, 171–180.

[35] Wierman, A., Harchol-Balter, M. (2003). Classifying scheduling policies with respect to unfairness in an M/GI/1. In: *Proc. ACM Sigmetrics 2003 Conf.*, San Diego CA, USA, 238–249.

[36] Yang, S., De Veciana, G. (2002). Size-based adaptive bandwidth allocation: optimizing the average QoS for elastic flows. In: *Proc. IEEE Infocom 2002*, New York NY, USA, 657–666.

[37] Yang, S., De Veciana, G. (2004). Enhancing both network and user performance for networks supporting best-effort traffic. *IEEE/ACM Trans. Netw.* **12**, 349–360.