Centrum voor Wiskunde en Informatica

View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by CWI's Instituti

REPORTRAPPORT

PNA

Probability, Networks and Algorithms

Probability, Networks and Algorithms

Delay optimization in bandwidth-sharing networks

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

REPORT PNA-E0605 MARCH 2006

Centrum voor Wiskunde en Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation 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)

Copyright © 2006, Stichting Centrum voor Wiskunde en Informatica

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3711

Delay optimization in bandwidth-sharing networks

ABSTRACT

Bandwidth-sharing networks as considered by Massoulie & Roberts provide a natural modeling framework for describing the dynamic flow-level interaction among elastic data transfers. Although valuable stability results have been obtained, crucial performance metrics such as flow-level delays and throughputs in these models have remained intractable in all but a few special cases. In particular, it is not well understood to what extent flow-level delays and throughputs achieved by standard bandwidth-sharing mechanisms such as alpha-fair strategies leave potential room for improvement. In order to gain a better understanding of the latter issue, we set out to determine the scheduling policies that minimize the mean delay in some simple linear bandwidth-sharing networks. We compare the performance of the optimal policy with that of various alpha-fair strategies so as to assess the efficacy of the latter and gauge the potential room for improvement. The results indicate that the optimal policy achieves only modest improvements, even when the value of alpha is simply fixed, provided it is not too small.

2000 Mathematics Subject Classification: 68M20, 60K25,90B18
Keywords and Phrases: bandwidth-sharing networks; delay optimization; alpha-fair strategies
Note: This research is financially supported by The Netherlands Organization for Scientific Research (NWO) and the Dutch Bsik/BRICKS project.

Delay Optimization in Bandwidth-Sharing Networks[1]

Maaike Verloop[a], Sem Borst[a,b,c], Rudesindo Núñez-Queija[a,b]

[a] CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
[b] Dept. of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[c] Bell Laboratories, Lucent Technologies, P.O. Box 636, Murray Hill, NJ 07974, USA

Abstract

Bandwidth-sharing networks as considered by Massoulié & Roberts provide a natural modeling framework for describing the dynamic flow-level interaction among elastic data transfers. Although valuable stability results have been obtained, crucial performance metrics such as flow-level delays and throughputs in these models have remained intractable in all but a few special cases. In particular, it is not well understood to what extent flow-level delays and throughputs achieved by standard bandwidth-sharing mechanisms such as α-fair strategies leave potential room for improvement.

In order to gain a better understanding of the latter issue, we set out to determine the scheduling policies that minimize the mean delay in some simple linear bandwidth-sharing networks. We compare the performance of the optimal policy with that of various α-fair strategies so as to assess the efficacy of the latter and gauge the potential room for improvement. The results indicate that the optimal policy achieves only modest improvements, even when the value of α is simply fixed, provided it is not too small.

1 Introduction

Over the past several years, the processor-sharing discipline has emerged as a useful paradigm for evaluating the flow-level performance of elastic data transfers competing for bandwidth on a single bottle-neck link. Bandwidth-sharing networks as considered by Massoulié & Roberts [7] provide a natural extension for modeling the dynamic interaction among competing elastic flows that traverse several links along their source-destination paths. Bonald & Massoulié [1] showed that a wide class of α-fair bandwidth-sharing policies as introduced by Mo & Walrand [8] achieve stability in such networks under the simple (and necessary) condition that no individual link is overloaded, see also [11] for instance. While stability is arguably the most fundamental performance criterion, flow-level delays and throughputs are obviously crucial metrics too. Although useful approximations, bounds [2] and heavy-traffic limits [6] have been obtained, the latter performance metrics have largely remained intractable in all but a few special cases. In particular, it is not well understood to what extent the flow-level delays and throughputs achieved by common bandwidth-sharing mechanisms leave potential room for improvement.

The scope for improving flow-level delays and throughputs has been the focus of intense efforts in a somewhat distinct strand of research on size-based scheduling strategies. Several studies have demonstrated that the Shortest Remaining Processing Time first (SRPT) discipline can achieve significant performance improvements for heavy-tailed service requirements compared to First-Come First-Served or Processor Sharing. The SRPT discipline has therefore been adopted as an effective mechanism for improving the performance of web servers [3, 5]. A critical issue associated with size-based scheduling in general and SRPT in particular, is that it relies on (partial)

[1] This research is financially supported by The Netherlands Organization for Scientific Research (NWO) and the Dutch Bsik/BRICKS project.

knowledge of (remaining) service requirements. While such information is usually available in web servers, it is impractical to obtain in Internet routers. An alternative strategy which has hence been advocated for scheduling data flows is the Least Attained Service first (LAS) discipline also known as Foreground-Background Processor Sharing.

Nearly all studies on the performance gains from size-based scheduling strategies such as SRPT and LAS have considered single-server settings. Single-server systems provide reasonable models for web servers, but they do not accurately capture scenarios where users require service from several resources simultaneously. Such concurrent resource possession arises in the above-mentioned bandwidth-sharing networks, where data flows traverse several links between their source-destination pairs and consume bandwidth on each of them for the duration of the transfer. (Even though individual packets travel across the network on a hop-by-hop basis, on a somewhat longer time scale a data flow claims roughly equal bandwidth on each of the links along its path since the amount of buffering at intermediate nodes is typically quite limited.)

While single-server systems provide tractable results and useful insights, they do not exhibit the potential non-work-conserving behavior that may occur in scenarios with concurrent resource possession. There are various indications that priority mechanisms in such scenarios may cause starvation effects with possibly severe consequences. For example, Yang & de Veciana [14, 15] demonstrated that SRPT scheduling in network scenarios may yield considerable performance improvements in terms of mean delays and throughputs, but also observed that flows on long routes with large sizes may sustain a marked performance degradation. Recently, it was shown that size-based scheduling strategies such as SRPT and LAS may in fact unnecessarily fail to achieve stability in network settings, even at arbitrarily low loads [13].

In conclusion, the results for size-based scheduling in single-server models do not provide a good indication for the scope for improvement over common bandwidth-sharing mechanisms in network scenarios. In order to gain better insight into the latter issue, we will set out to determine scheduling policies that minimize the mean delay in bandwidth-sharing networks with a linear topology. While admittedly simple, linear networks provide a useful model for flows that traverse several links and experience bandwidth contention from independent cross-traffic. Armed with the knowledge of the optimal policy, we then compare its performance with various α-fair strategies so as to assess the efficacy of the latter and gauge the potential room for improvement. Our results indicate that the optimal policy achieves only modest improvements over an α-fair strategy when the value of α is optimized. In its turn, an optimized α-fair strategy yields only marginal improvements compared to virtually any fixed value of α, as long as this value is not too small. This is particularly so for the important special cases $\alpha = 1$ (proportional fair strategy) and $\alpha = 2$ (which is a modeling abstraction of TCP).

The remainder of the paper is organized as follows. In Section 2 we provide a detailed model description and discuss some preliminaries. In Section 3 we derive some sample-path comparisons for the workload processes under various scheduling policies. We use these sample-path inequalities in Section 4 to show that in certain cases with exponential service requirements relatively simple priority-type policies minimize the mean number of users in the system. In Section 5 we examine cases where the optimal policy does not have a simple priority-type structure, and use dynamic programming techniques to prove that in these cases the optimal policy is characterized by a switching curve. Section 6 presents the numerical experiments that we conducted. We summarize our results in Section 7.

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. 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, $i = 1, \dots, L$, 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 λ_i, and have generally distributed service requirements B_i with mean β_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 obviously necessary conditions for stability $\rho_0 + \rho_i < 1$, for $i = 1, \ldots, L$, are known [1] to be sufficient as well for α-fair bandwidth-sharing policies. (For conciseness, these conditions will be referred to as the 'standard' conditions.) In order to examine the effectiveness of α-fair policies we seek policies that in some appropriate sense minimize the total number of active users in the above-described system. We only allow (possibly preemptive) policies that have no knowledge available of the remaining service requirements and denote this class of policies by Π. The following policies will play a central role.

- Policy π^* gives preemptive priority to class 0 whenever it is non-empty and, otherwise, serves any other class with at least one user.
- Policy π^{**} simultaneously serves all classes $i = 1, \ldots, L$ whenever at least one user of each class is present. Otherwise class 0 is served. When class 0 is empty, any other class with at least one user present is served.

For both these policies the system is stable under the standard conditions, since policies π^* and π^{**} ensure that each node operates at full rate when it is non-empty.

For a given policy π, denote by $N_i^\pi(t)$ the number of class-i users at time t and by $W_i^\pi(t)$ their total residual work. $N^\pi(t)$ is defined as $\sum_{i=1}^L N_i^\pi(t)$. We further define N_i^π, W_i^π and N^π as random variables with the corresponding time-average distributions (when they exist). For brevity, we use the superscripts * and ** for random variables corresponding to π^* and π^{**}.

Note that class 0 does not notice the presence of other classes under policy π^*. The mean amount of class-0 work is therefore given by the Pollaczek-Khintchine formula:

$$\mathbb{E}(W_0^*) = \frac{\lambda_0 \mathbb{E}({B_0}^2)}{2(1-\rho_0)}.$$

With policy π^*, any class $i \neq 0$ sees its service being interrupted by busy periods of class 0 so that [10]:

$$\mathbb{E}(W_i^*) = \frac{\lambda_0 \mathbb{E}({B_0}^2) + \lambda_i \mathbb{E}({B_i}^2)}{2(1-\rho_0-\rho_i)} - \frac{\lambda_0 \mathbb{E}({B_0}^2)}{2(1-\rho_0)}.$$

Note that these formulas hold for any service requirement distribution and scheduling discipline within classes.

In the special case of exponentially distributed service requirements, scheduling within a class (without knowledge of the actual size of jobs) does not affect the distribution of the number of users. Letting $\mu_i = 1/\beta_i$ (and thus $\mathbb{E}({B_i}^2) = 2/\mu_i^2)$), the mean number of users can then simply be obtained from $\mathbb{E}(N_i^*) = \mu_i \mathbb{E}(W_i^*)$ for all classes i. In particular

$$\mathbb{E}(N_0^*) = \frac{\rho_0}{1-\rho_0},$$

and

$$\mathbb{E}(N_i^*) = \frac{\rho_i}{1-\rho_0-\rho_i} + \frac{\mu_i}{\mu_0}\left(\frac{\rho_0}{1-\rho_0-\rho_i} - \frac{\rho_0}{1-\rho_0}\right),$$

so that

$$\mathbb{E}(N^*) = \frac{\rho_0}{1-\rho_0} + \sum_{i=1}^{L}\Big(\frac{\frac{\lambda_i}{\mu_0}\rho_0 + \rho_i^2}{(1-\rho_0)(1-\rho_0-\rho_i)} + \frac{\rho_i}{1-\rho_0}\Big). \tag{1}$$

For policy π^{**} there is no closed-form expression available for the mean workloads. For $L = 2$, determining these is equivalent to solving a boundary-value problem [4]: the service rate allocated to any class i depends on the workloads of all other classes.

3 Workload

In this section we allow for general service requirement distributions and compare (sample-path wise) the workloads of the various classes under different policies.

Let $\bar{\pi}^i$ be a policy that is work-conserving in node i, i.e., the capacity of node i is fully used whenever that node is non-empty. Obviously, such a policy minimizes the total workload in node i at all times. More specifically, if $W_0^{\bar{\pi}^i}(0)+W_i^{\bar{\pi}^i}(0) \leq_{st} W_0^{\pi}(0)+W_i^{\pi}(0)$ for some arbitrary policy π, then

$$W_0^{\bar{\pi}^i}(t) + W_i^{\bar{\pi}^i}(t) \leq_{st} W_0^{\pi}(t) + W_i^{\pi}(t), \quad \forall t \geq 0. \tag{2}$$

Here $\leq_{st}$ denotes the usual stochastic ordering. Note that both policies π^* and π^{**} are work-conserving in each node, so inequality (2) holds for all $i = 1, \dots, L$, if $\bar{\pi}^i \in \{\pi^*, \pi^{**}\}$. We call $W_{0,j,k}^{\pi}(t) := W_0^{\pi}(t) + W_j^{\pi}(t) + W_k^{\pi}(t)$ the aggregate workload in nodes j and k. Besides minimizing the workload in one node, at any point in time, policy π^{**} also minimizes the aggregate workload in at least one pair of nodes (these need not always be the same) as is formalized in the following lemma. This result will be useful for the analysis in the next sections.

Lemma 3.1 *If for $t = 0$ there exist nodes j and k with $j \neq k$, such that*

$$W_{0,j,k}^{**}(t) \leq_{st} W_{0,j,k}^{\pi}(t), \tag{3}$$

then, for any $t > 0$, there exist j and k (not necessarily the same as at time $t = 0$) with $j \neq k$ such that (3) holds.

Hence, if $L = 2$, the lemma states that policy π^{**} stochastically minimizes the total workload in the system. We note that there is no policy that achieves the same for $L > 2$.

Proof of Lemma 3.1 By assuming the same sequence of arrivals and service requests, we can compare the two policies, π^{**} and π, in the same sample space. Let

$$u = \inf\{t > 0 : W_{0,j,k}^{**}(t) > W_{0,j,k}^{\pi}(t), \forall j, k \neq 0, j \neq k\}.$$

We show by contradiction that u cannot be finite. Let us suppose $u < \infty$. Inequality (3) can only be violated for all pairs j and k immediately after time u, if it holds with equality at time u for some j and k, which we fix for the remainder of the proof. In addition, for the equality to cease to be valid, policy π^{**} should not be serving both nodes j and k at full rate, so that $W_0^{**}(u) = 0$ and $W_i^{**}(u) = 0$ for either $i = j$ or $i = k$. From (2) we have $W_l^{**}(u) \leq W_0^{\pi}(u) + W_l^{\pi}(u)$ for all $l \neq 0$, $l \neq i$; we fix such an l and observe that this inequality is preserved until the next arrival from either class 0 or class i (in the mean time, π^{**} works at full rate in node l and π can not do better than that). Note that $W_0^{**}(t) = 0$ and $W_i^{**}(t) = 0$ until such an arrival occurs and, hence, $W_{0,i,l}^{**}(t) \leq W_{0,i,l}^{\pi}(t)$, which contradicts the definition of u. □

4 Small class-0 users

In the remainder of the paper we focus on exponentially distributed service requirements and write $\mu_i = 1/\beta_i$. For relatively 'large' values of μ_0, i.e., when class-0 users are relatively small, we show that either π^* or π^{**} *stochastically* minimizes the number of users at every point in time. More precisely: this is so when $\mu_0 > \sum_{i \geq 1, i \neq j} \mu_i$ for all $j \neq 0$. In Section 4.1 we first show that the results of the previous section allow us to readily prove that π^* and π^{**} minimize the *mean* number of users in the above-mentioned cases. Because of Little's law, such a policy automatically minimizes the mean overall sojourn time as well. We briefly comment on the stochastic optimality in Section 4.2.

To put our results in context, we recall that the μ-rule is known to stochastically minimize the number of users [9] in a single-server system. The rationale behind this rule is that it maximizes the output rate at all times. In the network we discuss, this can only be accomplished for certain parameter values. Besides trying to maximize the total output rate of the system, we must take into account that when serving class $i \neq 0$ while another class $j \neq 0$ is empty may leave node j underutilized if there are users of class 0. For example, if $\mu_i > \mu_0$ for all $i = 1, \dots, L$, then giving priority to classes $1, \dots, L$, myopically maximizes the total output rate of the system but such a discipline unnecessarily causes instability [13] when $\Pi_{j=1}^{L}(1-\rho_j) < \rho_0$. In general, there can be a trade-off between maximizing the output rate and using the full capacity in each node whenever that node is non-empty. It is precisely in those cases where these two objectives are compatible, that we can identify the policies that minimize the total number of users.

4.1 Minimizing the mean number of users

The next lemma, together with the results for the workload obtained in Section 3, can be used to prove that, in certain cases, policy π^* or π^{**} minimizes the *mean* total number of users at every point in time.

Lemma 4.1 *Let π, $\bar{\pi} \in \Pi$ and assume the service requirements of class i are exponentially distributed with mean $\beta_i = 1/\mu_i$. If for some $I \subseteq \{0, \dots, L\}$, $\sum_{i \in I} W_i^{\bar{\pi}}(t) \leq_{st} \sum_{i \in I} W_i^{\pi}(t)$, $\forall t \geq 0$, then*

$$\sum_{i \in I} \frac{1}{\mu_i} \mathbb{E}(N_i^{\bar{\pi}}(t)) \leq \sum_{i \in I} \frac{1}{\mu_i} \mathbb{E}(N_i^{\pi}(t)).$$

Proof Because of the memoryless property of the exponential distribution and the fact that policies π and $\bar{\pi}$ have no knowledge of the remaining service requirements, the workload, $W_i(t)$, is distributed as $\sum_{k=1}^{N_i(t)} E_k^i$. Here E_k^i are i.i.d. random variables from an exponential distribution with mean $1/\mu_i$. Hence,

$$\sum_{i \in I} \sum_{k=1}^{N_i^{\bar{\pi}}(t)} E_k^i \leq_{st} \sum_{i \in I} \sum_{k=1}^{N_i^{\pi}(t)} E_k^i, \quad \forall\, t \geq 0, \tag{4}$$

and the lemma is proved after taking expectations. □

This lemma paves the way for the following two propositions, which state that, in certain cases, π^* or π^{**} is optimal.

Proposition 4.2 *Assume $W_i{}^*(0) \leq_{st} W_i^{\pi}(0)$, for all i. If $\sum_{i=1}^{L} \mu_i \leq \mu_0$, then $\mathbb{E}(N^*(t)) \leq \mathbb{E}(N^{\pi}(t))$, $\forall \pi \in \Pi$ and $\forall t \geq 0$.*

Proof By (2), policy π^* minimizes the workload in each node, which implies by Lemma 4.1 that $\forall\ i = 1, \dots, L$,

$$\frac{1}{\mu_0}\mathbb{E}(N_0{}^*(t)) + \frac{1}{\mu_i}\mathbb{E}(N_i{}^*(t)) \ \leq \frac{1}{\mu_0}\mathbb{E}(N_0^{\pi}(t)) + \frac{1}{\mu_i}\mathbb{E}(N_i^{\pi}(t)). \tag{5}$$

Combining Lemma 4.1 with the fact that giving preemptive priority to class i minimizes the workload of class i, we have:

$$\mathbb{E}(N_0{}^*(t)) \leq \mathbb{E}(N_0^{\pi}(t)). \tag{6}$$

Multiplying (5) by $\mu_i \geq 0$, for all $i = 1, \dots, L$, multiplying (6) by $\frac{\mu_0 - \sum_{i=1}^{L}\mu_i}{\mu_0} \geq 0$ and summing these $L+1$ inequalities gives $\sum_{i=0}^{L} \mathbb{E}(N_i{}^*(t)) \leq \sum_{i=0}^{L} \mathbb{E}(N_i^{\pi}(t))$. □

Proposition 4.3 *Assume $W_i{}^{**}(0) \leq_{st} W_i^{\pi}(0)$, for all i. If $\sum_{i=1}^{L} \mu_i \geq \mu_0 \geq \sum_{i=1, i \neq j}^{L} \mu_i$ for all $j \neq 0$, then $\mathbb{E}(N^{**}(t)) \leq \mathbb{E}(N^{\pi}(t))$, $\forall \pi \in \Pi$ and $\forall t \geq 0$.*

Proof As in the previous proof we have by (2) and Lemma 4.1 that

$$\frac{1}{\mu_0}\mathbb{E}(N_0{}^{**}(t)) + \frac{1}{\mu_i}\mathbb{E}(N_i{}^{**}(t)) \leq \frac{1}{\mu_0}\mathbb{E}(N_0^{\pi}(t)) + \frac{1}{\mu_i}\mathbb{E}(N_i^{\pi}(t)). \tag{7}$$

Similarly, we can conclude from Lemmas 3.1 and 4.1 that at time t there are classes j and k, $j \neq k \in \{1, \ldots, L\}$, such that

$$\begin{aligned} &\frac{1}{\mu_0}\mathbb{E}(N_0{}^{**}(t)) + \frac{1}{\mu_j}\mathbb{E}(N_j{}^{**}(t)) + \frac{1}{\mu_k}\mathbb{E}(N_k{}^{**}(t)) \\ &\leq \frac{1}{\mu_0}\mathbb{E}(N_0^\pi(t)) + \frac{1}{\mu_j}\mathbb{E}(N_j^\pi(t)) + \frac{1}{\mu_k}\mathbb{E}(N_k^\pi(t)). \end{aligned} \tag{8}$$

Now multiply (7) by $\mu_0 - \sum_{l=1, l\neq i}^{L} \mu_l \geq 0$, for $i = j, k$ and by μ_i for all $i \neq 0, j, k$; multiply inequality (8) by $\sum_{i=1}^{L} \mu_i - \mu_0 \geq 0$ and sum these $L+1$ inequalities to obtain $\sum_{i=0}^{L} \mathbb{E}(N_i{}^{**}(t)) \leq \sum_{i=0}^{L} \mathbb{E}(N_i^\pi(t))$. □

4.2 Stochastic optimality

It is worth noting that despite the stochastic inequality (4) the above arguments can not be strengthened to prove that π^* and π^{**} in fact stochastically minimize the number of users for the given parameter values. This can, however, be accomplished using a dynamic programming (DP) approach similar to that in Section 5 below. For the case $L = 2$ the following two results are proved in [12]:

Proposition 4.4 *If $\mu_1 + \mu_2 \leq \mu_0$, then policy π^* stochastically minimizes the total number of users.*

Proposition 4.5 *If $\mu_1, \mu_2 \leq \mu_0$ and $\mu_1 + \mu_2 \geq \mu_0$, then policy π^{**} stochastically minimizes the total number of users.*

5 Large class-0 users

Again assuming exponential service requirements, we now explore the uncovered case when there exists an $j = 1, \ldots, L$, such that $\sum_{i=1, i\neq j}^{L} \mu_i \geq \mu_0$. Since a stochastically optimal policy may in general not exist, we focus instead on the average-optimal policy, i.e., the policy that minimizes $\mathbb{E}(N^\pi)$ over all policies $\pi \in \Pi$.

We again focus on the case of two nodes and hence consider service rates such that $\mu_0 < \mu_i$ for at least one $i = 1, 2$. Intuitively it is clear that when there are users of both classes 1 and 2 present, serving them will be optimal. When there are only users of classes 0 and 1 present and $\mu_1 < \mu_0$, serving class 0 seems appropriate, since it is work-conserving in both nodes and it maximizes the total output rate. However, when $\mu_0 < \mu_1$, there is no obvious rule which class to serve. The next proposition states that in such situations, there exists a switching curve that determines which class is optimal to serve, i.e. there exists a function $h(\cdot)$ such that it is optimal to serve class 0 at full rate if $N_1(t) \leq h(N_0(t))$ and to serve class 1 at full rate otherwise.

Proposition 5.1 *Assume that $\mu_1 > \mu_0$. If both classes 1 and 2 are non-empty, then the expected average-optimal stationary policy serves these simultaneously. While class 2 is empty, the optimal policy is characterized by a switching curve (class 1 is only served if there are sufficient class-1 users). If, in addition, $\mu_0 \geq \mu_2$, then class 0 is served while class 1 is empty.*

In the remainder of this section we outline the proof of this proposition. We denote by i, j and k the numbers of class-0, class-1 and class-2 users, respectively. It will be convenient to focus on the uniformized Markov chain. That is, transition epochs (possibly 'dummy' transitions that do not alter the system state) are generated by a Poisson process of uniform rate $\nu = \lambda_0 + \lambda_1 + \lambda_2 + \mu_0 + \mu_1 + \mu_2$. We assume $\nu = 1$ without loss of generality. Using DP, we minimize the mean number of users for the embedded uniformized process, which is equivalent to minimizing that of the original process.

The direct costs that are incurred each time state (i,j,k) is visited, are $i+j+k$, which implies that the objective is to find a policy π that minimizes $\mathbb{E}(N^\pi)$. The DP equation can be written as:

$$\begin{aligned}
&V_{n+1}(i,j,k) = i+j+k\\
&+\lambda_0 V_n(i+1,j,k) + \lambda_1 V_n(i,j+1,k) + \lambda_2 V_n(i,j,k+1)\\
&+\min\{\mu_0 V_n((i-1)^+,j,k) + (\mu_1+\mu_2)V_n(i,j,k),\\
&\mu_0 V_n(i,j,k) + \mu_1 V_n(i,(j-1)^+,k) + \mu_2 V_n(i,j,(k-1)^+)\},
\end{aligned}$$

with $V_0(i,j,k) = i+j+k$.

The existence of an optimal switching curve when there are no class-2 users present is equivalent to the value function, $V(i,j,k)$, satisfying Properties 1 and 2 below. By symmetry, similar properties need to hold for the existence of a switching curve when there are no class-1 users.

Property 1: If it is optimal to serve class 1 in state $(i,j,0)$, then this is optimal in state $(i,j+1,0)$ as well, or equivalently, if

$$\begin{aligned}
&\mu_0 V(i,j,0) + \mu_1 V(i,j-1,0) + \mu_2 V(i,j,0)\\
&\leq \mu_0 V(i-1,j,0) + \mu_1 V(i,j,0) + \mu_2 V(i,j,0),
\end{aligned}$$

then

$$\begin{aligned}
&\mu_0 V(i,j+1,0) + \mu_1 V(i,j,0) + \mu_2 V(i,j+1,0)\\
&\leq \mu_0 V(i-1,j+1,0) + (\mu_1+\mu_2) V(i,j+1,0).
\end{aligned}$$

Note that this property is implied by the following inequality:

$$\begin{aligned}
&\mu_0 V(i,j+1,0) + \mu_0 V(i-1,j,0) + 2\mu_1 V(i,j,0)\\
&\leq \mu_0 V(i,j,0) + \mu_0 V(i-1,j+1,0)\\
&\quad +\mu_1 V(i,j-1,0) + \mu_1 V(i,j+1,0).
\end{aligned}$$

Property 2: If it is optimal to serve class 0 in state $(i,j,0)$, then this is optimal in state $(i+1,j,0)$ as well, or equivalently, if

$$\begin{aligned}
&\mu_0 V(i-1,j,0) + \mu_1 V(i,j,0) + \mu_2 V(i,j,0)\\
&\leq \mu_0 V(i,j,0) + \mu_1 V(i,j-1,0) + \mu_2 V(i,j,0),
\end{aligned}$$

then

$$\begin{aligned}
&\mu_0 V(i,j,0) + \mu_1 V(i+1,j,0) + \mu_2 V(i+1,j,0)\\
&\leq (\mu_0+\mu_2) V(i+1,j,0) + \mu_1 V(i+1,j-1,0).
\end{aligned}$$

This property is implied by

$$\begin{aligned}
&2\mu_0 V(i,j,0) + \mu_1 V(i+1,j,0) + \mu_1 V(i,j-1,0)\\
&\leq \mu_0 V(i+1,j,0) + \mu_0 V(i-1,j,0)\\
&+\mu_1 V(i+1,j-1,0) + \mu_1 V(i,j,0).
\end{aligned}$$

These properties can be established for $V(i,j,k)$ by proving them for all $V_n(i,j,k)$ using induction on the time index n, see [12] for details.

6 Numerical experiments

We now compare the performance of the optimal policy with that of α-fair bandwidth-sharing policies. We denote by $N_i^{(\alpha)}$ the mean number of class-i users as function of α. In the linear network, the α-fair allocation is

$$s_0 = \frac{n_0}{n_0 + (\sum_{l=1}^{L} n_l^\alpha)^{1/\alpha}} \quad \text{and} \quad s_i = 1 - s_0,$$

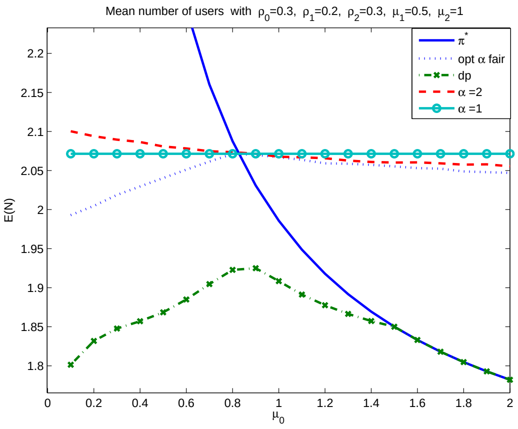

Figure 1: Total mean number of users in case A.

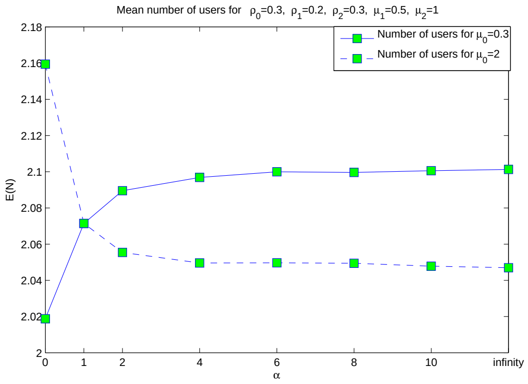

Figure 2: Total mean number of users in case A.

where s_j is the rate allocated to class j, see [1].

For the proportional fair allocation ($\alpha = 1$), the mean number of users is given by

$$\mathbb{E}(N_0^{(1)}) = \frac{\rho_0}{1-\rho_0}(1 + \sum_{i=1}^{L} \frac{\rho_i}{1-\rho_0-\rho_i})$$

and $\mathbb{E}(N_i^{(1)}) = \frac{\rho_i}{1-\rho_0-\rho_i}$, $i = 1, \dots, L$, see [7]. For general α-fair allocations ($\alpha \neq 1$) we conducted simulations in order to estimate the mean number of users. In our experiments we chose $\alpha \in A = \{0, 1, 2, 4, 6, 8, 10, \infty\}$. Besides $\alpha = 1$, the case $\alpha = 2$ will receive particular attention as well, because it is a common abstraction for TCP's bandwidth allocation.

Comparing the mean number of users for the proportional fair allocation and policy π^* already provides important insight. For $L = 2$ we have that $\mathbb{E}(N^*) - \mathbb{E}(N^{(1)})$ equals

$$\frac{\rho_0}{1-\rho_0}\Big(\frac{\rho_1}{1-\rho_0-\rho_1}\Big(\frac{\mu_1}{\mu_0} - 1\Big) + \frac{\rho_2}{1-\rho_0-\rho_2}\Big(\frac{\mu_2}{\mu_0} - 1\Big)\Big).$$

Note that for $\mu_0 < \bar{\mu}_0 := \frac{\lambda_1(1-\rho_0-\rho_2)+\lambda_2(1-\rho_0-\rho_1)}{\rho_1(1-\rho_0-\rho_2)+\rho_2(1-\rho_0-\rho_1)}$ (relatively large class-0 users), the proportional fair allocation does better than π^*, and that the difference is unbounded as $\mu_0 \to 0$. For $\mu_0 > \bar{\mu}_0$ (relatively small class-0 users), it is better to prioritize class 0. In fact, π^* achieves the minimum

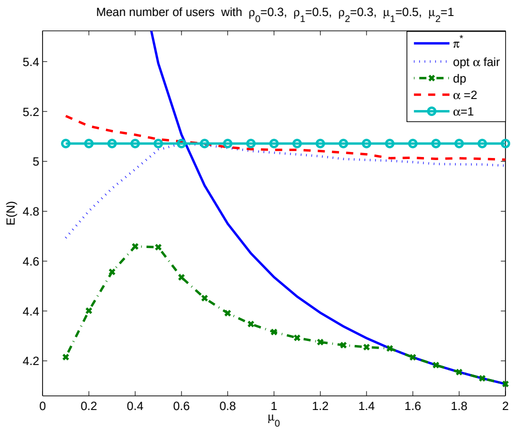

Figure 3: Total mean number of users in case B.

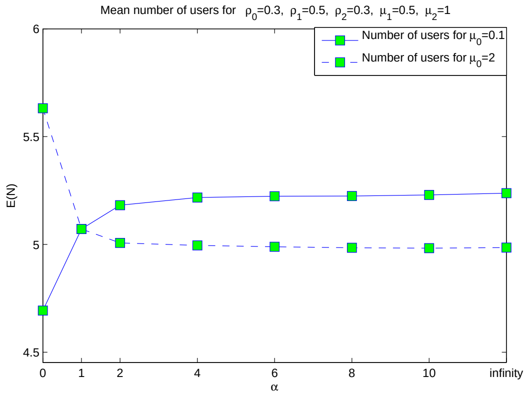

Figure 4: Total mean number of users in case B.

mean number of users among all strategies in Π, if $\mu_0 \geq \mu_1$ and $\mu_0 \geq \mu_2$. Still, the difference is limited by $-\frac{\rho_0}{1-\rho_0}\left(\frac{\rho_1}{1-\rho_0-\rho_1}+\frac{\rho_2}{1-\rho_0-\rho_2}\right)$. Thus, the proportional fair allocation performs well over a wide range of parameter values.

We now proceed to numerically investigate whether the latter finding holds in greater generality. The optimal policy is computed by DP after truncating the state space. In cases where the optimal policy is known explicitly, we verified that the results from DP are accurate. We examined a wide range of scenarios in terms of the values of the parameters λ_i and μ_i, $i = 0, 1, 2$. Since the results were qualitatively similar in the various scenarios, we only present the results for the cases with $\rho_0 = 0.3$, $\rho_2 = 0.3$, $\mu_1 = 0.5$, $\mu_2 = 1$, with either (A) $\rho_1 = 0.2$ or (B) $\rho_1 = 0.5$, and varying μ_0.

In Figures 1 and 3 we plot the total mean number of users under different policies as a function of μ_0 for cases A and B, respectively. The smallest mean number of users among all α-fair policies $(\min_{\alpha\in A}(\mathbb{E}(N^{(\alpha)})))$ is labeled with "opt α fair", the mean number of users for the optimal policy in each point is indicated by "dp", and the curve "π^*", corresponds to the function in (1). The other curves correspond to proportional fairness ($\alpha = 1$) and an abstraction of TCP ($\alpha = 2$).

From Figures 1 and 3 we see that the performance of α-fair policies compares well with that of the optimal policy. The gap does not exceed 20%. Apparently, α-fair policies succeed in *dynamically* adjusting the rate allocation in an efficient manner, without any knowledge of the service rate parameters. It is also striking that the differences within the class of α-fair policies are small, and that the mean total number of users is fairly insensitive to the value of μ_0 (for fixed ρ_0). In all cases, the optimal value of α is either 0 (for small values of μ_0) or ∞ (for large values of μ_0). The transition point occurs approximately at $\mu_0 = \bar{\mu}_0$.

In Figures 2 and 4 we plot the total mean number of users as a function of α for two values of μ_0, for cases A and B, respectively. Again, the results agree with what could be expected: for large μ_0 it is optimal to prioritize class 0, while for small small μ_0 it is better to achieve a large degree of parallelization. The difference between the best and the worst α-fair allocations is roughly 5% and 10% in cases A and B, respectively.

7 Summary and conclusions

In order to investigate the efficiency of standard allocation mechanisms such as α-fair policies, we have determined the delay-optimal allocation policies in a simple linear network with exponential service requirements. The optimal scheduling policies require a high degree of coordination within the network as well as knowledge of the service requirement distributions, which may prohibit actual implementation. As a benchmark, though, they are extremely useful to assess the effectiveness of other bandwidth-sharing strategies. In all our experiments we observed that (i) the differences within the class of α-fair allocations are not significant, and (ii) these allocations compare well with the optimal strategies.

The above-mentioned results concern rate allocation *across* flow classes (corresponding to flows sharing a common route), and do not account for scheduling *within* classes. As mentioned in the introduction, it was shown in [13] that standard size-based scheduling strategies such as SRPT and LAS applied across all flows can cause instability effects. However, size-based scheduling *within* flow classes may still produce substantial performance benefits, provided the rate allocation *across* flow classes is carefully arbitrated to avoid the above instability phenomena. Exactly how to combine size-based scheduling within classes with a stable rate arbitration across classes, and what the potential gains might be, is non-trivial and remains as a challenging topic for further research.

References

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

[2] Bonald, T., Proutière, A. (2004). On stochastic bounds for monotonic processor sharing networks. *Queueing Systems* **47**, 81–106.

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

[4] Cohen, J.W., Boxma, O.J. (1983). *Boundary Value Problems in Queueing System Analysis.* North-Holland, Amsterdam.

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

[6] Kelly, F.P., Williams, R.J. (2004). Fluid model for a network operating under a fair bandwidth-sharing policy. *Ann. Appl. Prob.* **14**, 1055–1083.

[7] Massoulié, L. Roberts, J.W. (2000). Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems* **15** 185–201.

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

[9] Righter, R., Shanthikumar, J.G. (1989). Scheduling multiclass single-server queueing systems to stochastically maximize the number of successful departures. *Prob. Eng. Inf. Sc.* **3**, 323–333.

[10] Takagi, H. (1991). *Queueing Analysis, Vol. I: Vacation and Priority Systems.* North-Holland, Amsterdam.

[11] de Veciana, G., Lee, T.-L., Konstantopoulos, T. (2001). Stability and performance analysis of networks supporting elastic services. *IEEE/ACM Trans. Netw.* **9**, 2–14.

[12] Verloop, I.M. (2005). Efficient flow scheduling in resource-sharing networks. Master thesis, Utrecht University.

[13] Verloop, I.M., Borst, S.C., Núñez Queija, R. (2005). Stability of size-based scheduling disciplines in resource-sharing networks. In *Proc. Performance 2005.*

[14] 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.

[15] 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.