Optimal File Splitting for Wireless Networks With Concurrent Access

Gerard Hoekstra[1,2], Rob van der Mei[1,3],
Yoni Nazarathy[1,4,5], and Bert Zwart[1,3,4,6]

[1] CWI, Amsterdam, The Netherlands
[2] Thales Nederland B.V., Huizen, The Netherlands
[3] VU University Amsterdam, The Netherlands
[4] Eindhoven University of Technology, EURANDOM, The Netherlands
[5] Eindhoven University of Technology, Department of Mechanical Engineering, The Netherlands
[6] Georgia Institute of Technology, Atlanta, GA, U.S.A.

Abstract. The fundamental limits on channel capacity form a barrier to the sustained growth on the use of wireless networks. To cope with this, multi-path communication solutions provide a promising means to improve reliability and boost Quality of Service (QoS) in areas that are covered by a multitude of wireless access networks. Today, little is known about how to effectively exploit this potential.

Motivated by this, we consider N parallel communication networks, each of which is modeled as a processor sharing (PS) queue that handles two types of traffic: foreground and background. We consider a foreground traffic stream of files, each of which is split into N fragments according to a fixed splitting rule $(\alpha_1, \ldots, \alpha_N)$, where $\sum \alpha_i = 1$ and $\alpha_i \geq 0$ is the fraction of the file that is directed to network i. Upon completion of transmission of all fragments of a file, it is re-assembled at the receiving end. The background streams use dedicated networks without being split. We study the sojourn time tail behavior of the foreground traffic. For the case of light foreground traffic and regularly varying foreground file-size distributions, we obtain a reduced-load approximation (RLA) for the sojourn times, similar to that of a single PS-queue. An important implication of the RLA is that the tail-optimal splitting rule is simply to choose α_i proportional to $c_i - \rho_i$, where c_i is the capacity of network i and ρ_i is the load offered to network i by the corresponding background stream. This result provides a theoretical foundation for the effectiveness of such a simple splitting rule. Extensive simulations demonstrate that this simple rule indeed performs well, not only with respect to the tail asymptotics, but also with respect to the mean sojourn times. The simulations further support our conjecture that the same splitting rule is also tail-optimal for non-light foreground traffic. Finally, we observe near-insensitivity of the mean sojourn times with respect to the file-size distribution.

Keywords: Concurrent Access, Processor Sharing Queues, Tail Asymptotics, File Splitting.

1 Introduction

Many of today's wireless networks have already closely approached the Shannon limit on channel capacity, leaving complex signal processing techniques room for only modest improvements in the data transmission rate [5]. An alternative to increase the overall data rate then becomes one in which multiple, likely different, networks are used concurrently because (1) the spectrum is regulated among various frequency bands and corresponding communication network standards, and (2) the overall spectrum usage remained to be relatively low over a wide range of frequencies [10]. The concurrent use of multiple networks simultaneously has opened up possibilities for increasing bandwidth, improving reliability, and enhancing Quality of Service (QoS) in areas that are covered by multiple wireless access networks. Despite the enormous potential for quality improvement, only little is known about how to fully exploit this potential. This motivates us to take a first step towards gaining fundamental insight regarding the implications of the choice of a splitting rule. In particular, we focus on the impact of static splitting rules on file download times. To this end, we study the flow-level performance of file transfers utilizing multiple networks simultaneously.

We study the splitting problem in a queueing theoretical context. Modeling network performance using processor sharing (PS) based models [2, 17, 23] is applicable to a variety of communication networks, including CDMA 1xEV-DO, WLAN, and UMTS-HSDPA. In fact, PS models can actually model file transfers over WLANs accurately [11], hence taking into account the complex dynamics of the file transfer application and its underlying protocol-stack, including their interactions.

The queueing model we consider is the *concurrent access network* model, see Figure 1. There are N PS queues that serve $N+1$ file streams. Stream 0 is called the foreground stream and streams $1, \ldots, N$ are called the background streams. Files of background stream i are served exclusively at PS queue i. Each files of the foreground stream is fragmented (split) upon arrival according to a fixed, non-negative, splitting rule $\underline{\alpha} = (\alpha_1, \ldots, \alpha_N)$ where $\sum_{i=1}^{N} \alpha_i = 1$ and $\alpha_i \geq 0$, $i = 1, \ldots, N$. After splitting, the fragments are routed to their corresponding queues. Thus, when a file of size B arrives at stream 0, a fragment of size $\alpha_i B$ is directed to each queue i. Once all fragments complete their service, the fragments are reunited, and this completes the file transfer.

Consider a tagged file of the foreground stream that arrives to a network in steady-state. Denote the sojourn time of its i'th fragment operating under the splitting rule $\underline{\alpha}$ by $V_i^{\underline{\alpha}}$. This is the time it takes the fragment to complete service at queue i. Denote $\underline{V}_{\underline{\alpha}} = (V_1^{\underline{\alpha}}, \ldots, V_N^{\underline{\alpha}})$. The sojourn time of the file through the network is $M_{\underline{\alpha}} = \max \underline{V}_{\underline{\alpha}}$. Our purpose is to analyze the distribution of $M_{\underline{\alpha}}$ and choose a splitting rule $\underline{\alpha}$ such that $M_{\underline{\alpha}}$ is kept minimal.

Our probabilistic and load assumptions are as follows: Arrivals of files in all streams are according to independent Poisson processes with rates $\lambda_i, i = 0, 1, \ldots, N$. File sizes of stream i constitute an i.i.d. sequence of positive random variables with finite expectation. The $N+1$ sequences of processing times are mutually independent. Denote the mean file size of stream i by β_i and $\rho_i = \lambda_i \beta_i$

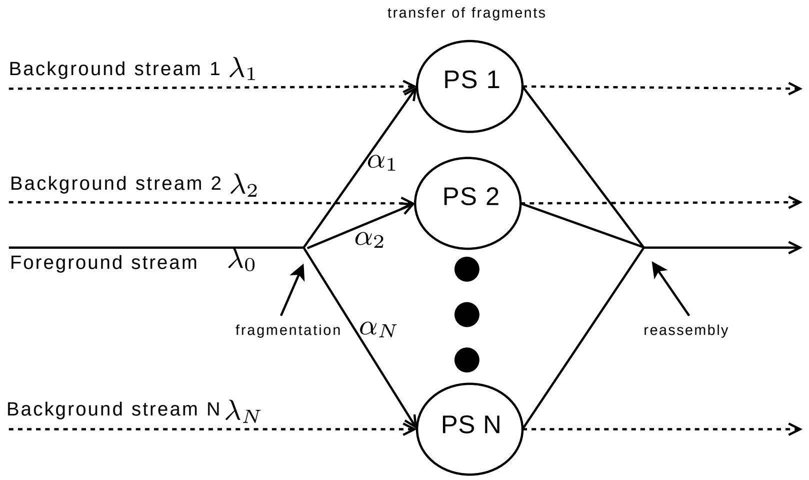

transfer of fragments
Background stream 1 λ_1
PS 1
α_1
Background stream 2 λ_2
PS 2
α_2
Foreground stream λ_0
fragmentation
α_N
reassembly
Background stream N λ_N
PS N

Fig. 1: The concurrent access network.

$i = 0, 1, \ldots, N$. We assume that processor sharing queue i operates at rate c_i. For the background streams and queues, denote the corresponding N dimensional vectors, $\underline{\rho}$ and $\underline{c}$. We assume that $\rho_0\mathbf{1}+\underline{\rho} < \underline{c}$. Here $\mathbf{1}$ denotes a vector of 1's. This condition ensures stability irrespective of our choice of splitting proportions.

The Splitting Rule $\boldsymbol{\alpha}^*$

Our main goal is to provide supporting arguments for using this simple splitting rule:

$$\alpha_i^* := \frac{c_i - \rho_i}{\sum_{j=1}^N (c_j - \rho_j)} \tag{1}$$

Note that $c_i - \rho_i$ is the unutilized capacity of queue i due to background traffic and $\sum_{j=1}^N (c_j - \rho_j)$ is the total unutilized capacity due to background traffic. Observe that $\underline{\alpha}^*$ does not depend on ρ_0.

To motivate this rule, consider the following heuristic argument: Observe that each queue in isolation is a two class M/G/1 PS queue, allowing us to compute means. It is well known (first shown in [18]) that the mean sojourn time of a job of size B in a processor sharing queue with rate $\tilde{c}$ and load $\tilde{\rho}$ is:

$$\mathbb{E}\,[\tilde{V}|B] = \frac{B}{\tilde{c} - \tilde{\rho}}.$$

Now upon arrival of a foreground job of size B, we have

$$\mathbb{E}\,[V_i^{\underline{\alpha}}|B] = \frac{\alpha_i B}{c_i - \rho_i}.$$

Setting α_i^* in the above, we see that

$$\mathbb{E}\left[V_i^{\underline{\alpha}^*}|B\right] = \frac{B}{\sum_{j=1}^{N}(c_j - \rho_j)}, i = 1, \ldots, N.$$

Thus the mean sojourn times of fragments are equal for all i. This essentially aggregates all processor sharing queues into a single "virtual queue" with capacity $\sum_{i=1}^{N} c_i$ and utilization based on the background traffic.

Theoretical Contribution

For our theoretical results, we shall further assume that the distribution of stream 0 files is regularly varying of index $\nu > 1$. This means that the tail of the distribution function has the form $P(B > x) = L(x)x^{-\nu}$, where $L(\cdot)$ is a slowly varying function: $L(ax)/L(x) \to 1$ as $x \to \infty$ for any $a > 0$. We do not require the background file sizes to be heavy-tailed, but do require that there exist $\epsilon_i > 0$ such that $\mathbb{E}\left[B_i^{1+\epsilon_i}\right] < \infty$, where we denote by B_i a generic random variable representing the file size of background stream i.

Denote,

$$\gamma_m^{\underline{\alpha}} := \min_{i=1,\ldots,N}\left(\frac{c_i - \rho_i}{\alpha_i}\right) - \rho_0. \tag{2}$$

Our key result is:

$$P(M_{\underline{\alpha}} > x) \sim P(B > \gamma_m^{\underline{\alpha}} x). \tag{3}$$

Here $f(x) \sim g(x)$ implies that $\lim_{x\to\infty} f(x)/g(x) = 1$. This is a form of a *Reduced Load Approximation* (RLA) (c.f. [13], [21]) which appears in our network. It is further evident that in this case, the splitting rule which maximizes $\gamma_m^{\underline{\alpha}}$ is $\underline{\alpha}^*$ and thus we have the tail asymptotic optimality:

$$\limsup_{x\to\infty} \frac{P(M^{\alpha^*} > x)}{P(M^{\alpha} > x)} \leq 1, \ \forall \text{ splitting rules } \underline{\alpha}. \tag{4}$$

This tail asymptotic optimality of the design parameter $\underline{\alpha}^*$ is similar to the tail optimality properties of scheduling disciplines discussed in [19].

In this paper we present a proof of (3) for the case of light foreground traffic. In this case we set $\lambda_0 = 0$ and assume that a single foreground job arrives to a steady state system. We further conjecture that (3) is true for the general case. Extensive simulation experiments demonstrate our conjecture to be true.

Related Work

In the context of telecommunication systems the concurrent use of multiple network resources in parallel was already described for a Public Switched Digital Network (PSDN)[6]. Here inverse multiplexing was proposed as a technique to perform the aggregation of multiple independent information channels across a network to create a single higher-rate information channel. Various approaches have appeared to exploit multiple transmission paths in parallel. For example by using multi-element antennas, as adopted by the IEEE802.11n draft [4] standard, at the physical layer or by switching datagrams at the link layer [3, 8], and also by using multiple TCP sessions in parallel to a file-server [20]. In the latter case each available network transports part of the requested data in a separate TCP session. Previous work has indicated that downloading from multiple networks concurrently may not always be beneficial [12], but in general significant performance improvements can be realized [14, 15, 16]. Under these circumstances of using a combination of different network types in particular the transport layer-approaches have shown their applicability [16] as they allow appropriate link layer adaptations for each TCP session.

In [22], the authors investigate the same queueing model in the context of web-server farms. A slight difference is that they do not consider background streams. The major difference is that they analyze the routing policy Join the Shortest Queue (JSQ) while we concentrate on a splitting policy. Note that as opposed to communication networks, splitting in the context of web-server farms is not always possible. Another related paper is [7] where the authors consider routing policies of the model in a distributed vs. centralized optimization. In general our queueing model falls within the framework of a fork-join queueing network [9]. To the best of our knowledge such a queueing network in which nodes are PS queues have not been investigated.

Organization of the Text

The rest of this paper is organized as follows: In Section 2 we heuristically deduce (3) and (4). In Section 3 we prove (3) for the light traffic case and conjecture it for the general case. In Section 4 we present our simulation results. These results put a strong basis regarding our conjecture. They further show ”near insensitivity” with regards to file size distributions and exhibit the fact in the case of light-tailed foreground file sizes our result does not hold. In Section 5 we discuss the relation between minimization of expected sojourn times and minimization of tails.

2 Heuristic Derivation of the Proposed Splitting Rule

Denote by B a random variable distributed as the file size of the foreground traffic files. Denote by $Q_i^{\underline{\alpha}}(t)$ the number of files in queue i at time t, operating under a splitting rule $\underline{\alpha}$. Define,

$$R_i^{\underline{\alpha}}(x) := \int_0^x \frac{1}{1 + Q_i^{\underline{\alpha}}(t)} dt,$$

this is the amount of service that a permanent customer obtains in queue i during the time $[0, x]$ when operating under the splitting rule $\underline{\alpha}$. Further denote by $\underline{R}^{\underline{\alpha}}(x)$ the N dimensional vector of $R_i^{\underline{\alpha}}(x)$. We have the following:

$$P(M_{\underline{\alpha}} > x) = 1 - P(M_{\underline{\alpha}} \leq x) = 1 - P(\underline{V}_{\underline{\alpha}} \leq x\mathbf{1}) = 1 - P(B\underline{\alpha} \leq \underline{R}^{\underline{\alpha}}(x)). \quad (5)$$

The first and second equalities are trivial. The third equality is due to the fact that in a processor sharing queue $P(\tilde{V} > \tilde{x}) = P(\tilde{B} > \tilde{R}(\tilde{x}))$. Observe now that,

$$\lim_{x\to\infty} \frac{1}{x}\underline{R}^{\underline{\alpha}}(x) = \underline{c} - \underline{\rho} - \rho_0\underline{\alpha} \quad \text{a.s..} \tag{6}$$

As a consequence, since for large x, $\underline{R}^{\underline{\alpha}}(x) \approx (\underline{c} - \underline{\rho} - \rho_0\underline{\alpha})x$, we can hope to have that for large x:

$$P(B\underline{\alpha} > \underline{R}^{\underline{\alpha}}(x)) \approx P(B\underline{\alpha} > (\underline{c} - \underline{\rho} - \rho_0\underline{\alpha})x). \tag{7}$$

Here we replaced the N dimensional random process $\underline{R}^{\underline{\alpha}}(x)$ by its asymptotic value. Heuristically, such an equivalence should hold when $\underline{R}^{\underline{\alpha}}(x)/x$ converges fast compared to the decay of the tail of B. In the next section we prove this relationship holds in the light traffic case and conjecture it also holds in the general case.

Assuming (7) to be true and continuing heuristically from (5) we have:

$$\begin{aligned} P(M_{\underline{\alpha}} > x) &\approx 1 - P(B\underline{\alpha} \le (\underline{c} - \underline{\rho} - \rho_0\underline{\alpha})x) \\ &= 1 - P(B \le \min_{i=1,\ldots,N}\left(\frac{c_i - \rho_i - \rho_0\alpha_i}{\alpha_i}\right) x) \\ &= P(B > \gamma_m^{\underline{\alpha}}\, x). \end{aligned}$$

Where $\gamma_m^{\underline{\alpha}}$ is given by (2). Thus we have heuristically arrived at our reduced load approximation (3).

Observe now that maximizing $\gamma_m^{\underline{\alpha}}$ minimizes $P(B > x\,\gamma_m^{\underline{\alpha}})$ for any x. As a result, finding the tail optimal $\underline{\alpha}$ means solving:

$$\begin{aligned} &\max_{\underline{\alpha}} \;\; \min_{i=1,\ldots,N}\left(\frac{c_i - \rho_i}{\alpha_i}\right) \\ &\text{s.t.} \;\; \sum_{i=0}^{N} \alpha_i = 1 \\ &\qquad \underline{\alpha} \ge 0. \end{aligned} \tag{8}$$

It is clear that an optimizer of (8) achieves the tail asymptotic optimality (4). We now show that this solution is easily found to be $\underline{\alpha}^*$ as in (1).

Lemma 1. *The unique solution of (8) is given by* $\underline{\alpha}^*$*.*

Proof. For clarity denote $f_i = c_i - \rho_i$. Denote by $\underline{\alpha}'$ an optimal solution such that (w.l.o.g.):

$$\frac{f_1}{\alpha'_1} \le \ldots \le \frac{f_N}{\alpha'_N}.$$

Observe that under α^*, the objective function is $\sum_{j=1}^{N} f_j$. Thus optimality of α' yields:

$$\frac{f_i}{\alpha'_i} \ge \frac{f_1}{\alpha'_1} \ge \sum_{j=1}^{N} f_j,$$

or,

$$f_i \geq \alpha_i' \sum_{j=1}^{N} f_j \;\; \forall i.$$

Summing over i we obtain an inequality thus equality holds for all i since the summands are non-negative.

3 The Reduced Load Equivalence

For ease of notation of this section, we fix an arbitrary splitting rule and remove the subscript/superscript $\underline{\alpha}$ from all variables defined previously. Denote,

$$\gamma_i := \frac{c_i - \rho_i - \alpha_i \rho_0}{\alpha_i},$$

and observe that as in (2), $\gamma_m = \min_{i=1,\ldots,N} \gamma_i$.

The following lemma states conditions under which the RLA (3) holds for our model. It is a direct application of results from [1] and [13]. See [21] for a survey.

Lemma 2. *Assume that*

$$\max(\frac{R_1(x)}{\alpha_1 x}, \ldots, \frac{R_N(x)}{\alpha_N x}) \rightarrow \max(\gamma_1, \ldots, \gamma_N) \;\; a.s., \tag{9}$$

and that there exists a positive finite constant K_m such that

$$P(\max(\frac{R_1(x)}{\alpha_1}, \ldots, \frac{R_N(x)}{\alpha_N}) \leq \frac{x}{K_m}) = o(P(B > \max(\gamma_1, \ldots, \gamma_N)x)), \tag{10}$$

then we have the reduced load approximation (3): $P(M > x) \sim P(B > \gamma_m x)$.

Proof. Each of the processor sharing queues is a multi-class queue with two classes: foreground and background. Since background file sizes have a $1+\epsilon$ finite moment and foreground file sizes have a regularly varying distribution, we apply Theorem 4.2 of [21] (originally from [1]) to obtain:

$$P(B > \frac{R_i(x)}{\alpha_i}) \sim P(B > \gamma_i x), \;\; i = 1, \ldots, N. \tag{11}$$

Now using the assumptions (9) and (10) we apply Theorem 1 of [13] to obtain:

$$P(B > \max(\frac{R_1(x)}{\alpha_1}, \ldots, \frac{R_N(x)}{\alpha_N})) \sim P(B > \max(\gamma_1, \ldots, \gamma_N)x). \tag{12}$$

The rest of the proof is for the case $N = 2$ (the general case is more tedious but not more complicated, it requires using the inclusion exclusion law for the union of N events). First observe:

$$\begin{aligned}
P(M > x) &= P(V_1 > x \text{ or } V_2 > x) \\
&= P(V_1 > x) + P(V_2 > x) - P(V_1 > x, V_2 > x) \\
&= P(\alpha_1 B > R_1(x)) + P(\alpha_2 B > R_2(x)) - P(\alpha_1 B > R_1(x), \alpha_2 B > R_2(x)) \\
&= P(B > \frac{R_1(x)}{\alpha_1}) + P(B > \frac{R_2(x)}{\alpha_2}) - P(B > \max(\frac{R_1(x)}{\alpha_1}, \frac{R_2(x)}{\alpha_2})).
\end{aligned}$$

Now assume that $\gamma_1 \leq \gamma_2$ and thus $\gamma_m = \gamma_1$ and $\max(\gamma_1, \gamma_2) = \gamma_2$:

$$\frac{P(M > x)}{P(B > \gamma_m x)} = \frac{P(B > \frac{R_1(x)}{\alpha_1}) + P(B > \frac{R_2(x)}{\alpha_2}) - P(B > \max(\frac{R_1(x)}{\alpha_1}, \frac{R_2(x)}{\alpha_2}))}{P(B > \gamma_1 x)}$$

$$= \frac{P(B > \frac{R_1(x)}{\alpha_1})}{P(B > \gamma_1 x)} + \frac{P(B > \gamma_2 x)}{P(B > \gamma_1 x)} \left(\frac{P(B > \frac{R_2(x)}{\alpha_2})}{P(B > \gamma_2 x)} - \frac{P(B > \max(\frac{R_1(x)}{\alpha_1}, \frac{R_2(x)}{\alpha_2}))}{P(B > \max(\gamma_1, \gamma_2) x)} \right).$$

Now,

$$\frac{P(B > \gamma_2 x)}{P(B > \gamma_1 x)} = \frac{L(\gamma_2 x)}{L(\gamma_1 x)} \left(\frac{\gamma_2}{\gamma_1} \right)^{-\nu} \rightarrow \left(\frac{\gamma_2}{\gamma_1} \right)^{-\nu},$$

and from (11) and (12) we have our result. The case of $\gamma_2 > \gamma_1$ is symmetric.

We are now in a position to establish the RLA (3) and the asymptotic optimality of α^*. Our result is for the light traffic case.

Theorem 1. *Consider the concurrent access network in light traffic: there is a single foreground arrival to steady state with* $\lambda_0 = 0$*. Then the reduced load approximation (3):* $P(M_{\underline{\alpha}} > x) \sim P(B > \gamma_m^{\underline{\alpha}} x)$ *holds.*

Proof. We apply Lemma 2: (9) follows from the SLLN. To see (10) observe that:

$$\begin{aligned}
P(\max(\frac{R_1(x)}{\alpha_1}, \dots, \frac{R_N(x)}{\alpha_N}) \leq \frac{x}{K_m}) &= P(\frac{R_1(x)}{\alpha_1} \leq \frac{x}{K_m}, \dots, \frac{R_N(x)}{\alpha_N} \leq \frac{x}{K_m}) \\
&= \prod_{i=1}^{N} P(\frac{R_i(x)}{\alpha_i} \leq \frac{x}{K_m})
\end{aligned}$$

Here we used the fact that under the light traffic assumption all queues are in steady state and there is a single arrival, thus $R_i(\cdot)$ are independent. Now as proved in [13] (Theorem 2), each of the terms can be made $o(P(B > x))$ by choosing K_m appropriately. Thus (10) is achieved.

Using this proof method to repeat the above for the non-light traffic case requires more care in obtaining (9) and (10). We conjecture that these conditions indeed hold and thus:

Conjecture 1. Theorem 1 holds also in the non-light traffic case and thus the splitting rule α^* is in general tail optimal.

In the next section we present simulation results that support the validity of this conjecture.

4 Simulation Results

We now summarize the results of some extensive simulations for evaluating $P(M_\alpha > x)$ on some examples with $N = 2$. For convenience we denote $\alpha := \alpha_1$ $(1 - \alpha = \alpha_2)$, similary for α^*. With respect to the tail probabilities, our primary purpose is to assert Conjecture 1 and the behavior of our tail optimality claim (4) by estimating,

$$\alpha^*(x) = \mathrm{argmin}_\alpha P(M_\alpha > x), \quad \text{and} \quad P^*(x) = P(M_{\alpha^*(x)} > x).$$

In this respect, we attempt to observe graphically that $\hat{\alpha}^*(x) \to \alpha^*$ as $x \to \infty$, where we denote estimators by hats. In addition it is fruitful to look at the relative suboptimality for a finite x when using α^* instead of $\alpha^*(x)$. For this purpose we plot:

$$\frac{\hat{P}(M_{\alpha^*} > x) - \hat{P}^*(x)}{\hat{P}^*(x)}. \tag{13}$$

In general, obtaining such results by simulation requires some long runs since we are trying to optimize probabilities of a rare event. In addition, we use the data of the simulation runs to analyze $\mathbb{E}\,[M_\alpha]$, show that it is nearly insensitive to the file size distributions and compare our splitting rule to the JSQ routing policy.

In all runs we set $\beta_0 = \beta_1 = \beta_2 = 1$ and $c_1 = c_2 = 1$. The types of file size distributions we consider are deterministic, exponential, Erlang 2 (a sum of two i.i.d. exponentials) and Pareto 3 (which is regularly varying with index $\nu = 3$). Here we take the case with support $[0, \infty)$, i.e. $P(B > x) = (1 + x/2)^{-3}$. We further parameterize the runs by the following:

$$\rho = \frac{\lambda_0 + \lambda_1 + \lambda_2}{2}, \quad \kappa = \frac{1 - \lambda_1}{1 - \lambda_2}, \quad \eta = \frac{\lambda_0}{\lambda_1 + \lambda_2}.$$

ρ is the total load on the system, κ is the ratio of free capacity and η is the ratio of foreground to background traffic. These 3 values uniquely define λ_0, λ_1 and λ_2. The table below specifies the parameters of the systems that we have simulated.

System	ρ	κ	η	Distribution 0	Distribution 1	Distribution 2	$(\lambda_0, \lambda_1, \lambda_2)$	α^*
1	0.5	1.5	0.5	Pareto 3	Pareto 3	Pareto 3	$(\frac{1}{3}, \frac{1}{5}, \frac{7}{15})$	0.6
2	0.5	1.5	0.5	Pareto 3	Deterministic	Deterministic	as System 1	-
3	0.5	1.5	0.5	Pareto 3	Exponential	Exponential	as System 1	-
4	0.5	1.5	0.5	Pareto 3	Exponential	Deterministic	as System 1	-
5	0.5	1.5	0.5	Deterministic	Deterministic	Deterministic	as System 1	-
6	0.5	1.5	0.5	Erlang 2	Erlang 2	Erlang 2	as System 1	-
7	0.5	1.5	0.5	Exponential	Pareto 3	Erlang 2	as System 1	-
8	0.5	2.0	0.5	Pareto 3	Pareto 3	Pareto 3	$(\frac{1}{3}, \frac{1}{9}, \frac{5}{9})$	$\frac{2}{3}$
9	0.5	1.0	0.5	Exponential	Exponential	Exponential	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	0.5

Systems 1 through 7 all have the same rate parameters but vary in the processing time distributions. System 8 is an additional example of an unbalanced system

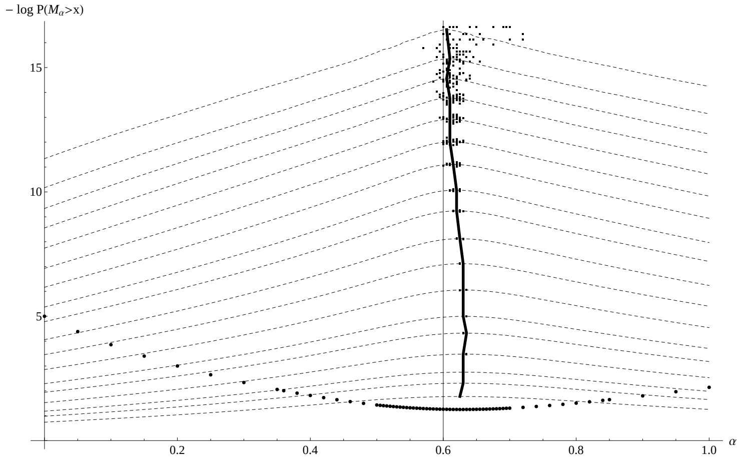


Fig. 2: An illustration of our data analysis approach: System 4 as an example. Dashed curves are plots of estimates of $-\log\ P(M_\alpha > x)$ for $x = 1, 2, 3, 5, 8, 11, 17, 25, 35, 48, 64, 85, 115, 160, 210, 270, 350, 500$. These curves are maximized by the thick trajectory of $\alpha^*(x)$ which converges to the vertical line at $\alpha^* = 0.6$. Clouds of optimizers over the 50 repetitions are plotted in order to present the dispersion in the argmax estimates. The convex dotted curve is the estimate of $\mathbb{E}\,[M_\alpha]$ drawn on the same scale.

having $\kappa = 2.0$ and thus $\alpha^* = 2/3$. System 9 is a balanced system which we have simulated for some additional sanity checking: we expect symmetric behavior of this system.

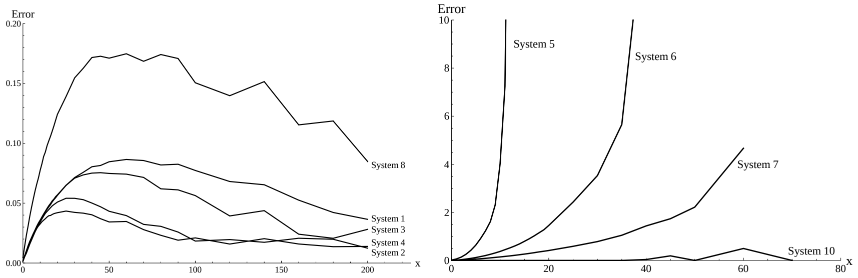
Simulation runs are composed of 5×10^7 foreground jobs, starting empty. For each system we repeated the simulation for various values of α, using the same seed for all values. We used a fine grid of steps of 0.005 for α within the range of $[\alpha^* - 0.10, \alpha^* + 0.10]$. Outside of this range but within the range $[\alpha^* - 0.25, \alpha^* + 0.25]$ we used a grid of steps of 0.02. In the remaining region we used a grid of 0.05. In addition we ran each system using the Join the Shortest Queue (non-splitting) policy.

Per system we repeated over the above specified range of α using 50 different seeds. Note that keeping the same seed while changing α is useful for optimizing the behavior of the queue given a single sample path of primitive processing times over α. The total number of runs that we performed is about $30,000$ and the total number of foreground jobs that have passed through the simulated system is of the order of 1.5×10^{12}. The simulations use a short and efficient C program which we have coded.

4.1 Tail Behavior

Figure 2 is a representative view of our results. It is a plot of some of the data collected in the simulation runs of System 4. We first estimate the tail probabilities $P(M_\alpha > x)$ for increasing values of x. These are plotted on a $-\log$ scale (dashed lines). We then optimize these over α for increasing values of x. This gives us the trajectory of $\hat{\alpha}^*(x)$ (thick curve). Obviously, as x grows the accuracy of this optimization is decreased due to the rarity of the tail event. We pictorially depict this in the figure by plotting the clouds of the 50 $(\text{argmax}_\alpha, \max_\alpha)$ pairs which result for increasing x's, one pair per seed. The thin vertical line in the figure is at $\alpha^* = 0.6$ and indeed, in agreement with the main conjecture and claim of this paper, it appears as the limiting value of $\alpha^*(x)$. We further plot the estimate $\mathbb{E}\,[M_\alpha]$ with a dot for every α in the grid. We comment on the mean in the next subsection.

Note that while Figure 2 shows that the argmax appears to converge rather slowly in x, it is more important to observe that the relative error (13) is always kept low. This can be observed in Figure 3a where we plot (13) for the systems in which the foreground files have a heavy-tailed regularly varying service distribution. The same quantity for systems with light-tailed foreground files is plotted in Figure 3b. Here it appears the relative error explodes. Thus suggesting that α^* is not tail optimal in the light-tailed foreground file size case.



(a) Heavy-tailed foreground file sizes. (b) Light-tailed foreground file sizes.

Fig. 3: Graphs of (13), the relative distance from optimality for finite x: (a) Heavy-tailed foreground file sizes. (b) Light tailed foreground file sizes.

4.2 Mean Behavior

In Figure 4 we plot the estimated values of $\mathbb{E}\,[M_\alpha]$ for systems $1-9$ for a range of α values. We also mark the values of α^* for the various systems by vertical dashed lines and on these lines we dot the mean sojourn times that are obtained for the systems using the JSQ routing policy. We note that at α^*, the width of

99% confidence intervals for the mean (using 50 observations) are is the order of 10^{-4}.

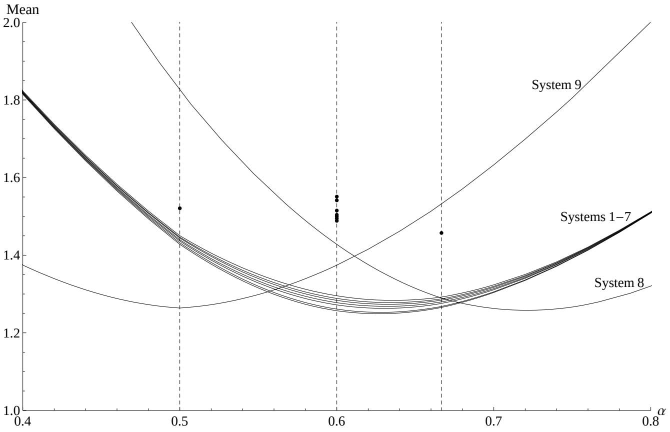


Fig. 4: Mean sojourn time curves. Vertical lines are at $\alpha^* = 0.5, 0.6, 2/3$. Dots on the vertical lines are mean sojourn times using JSQ for the corresponding systems.

Some comments are due: First observe that in all these examples, $\mathbb{E}\,[M_{\alpha^*}] < \mathbb{E}\,[M_{\text{JSQ}}]$.

Secondly, observe that $\min_\alpha \mathbb{E}\,[M_\alpha] \approx \mathbb{E}\,[M_{\alpha^*}]$. This is a key result: The simple splitting rule that we propose (which is tail optimal) is nearly optimal with respect to the mean. We further comment on this in the next section.

A third observation that appears from Systems $1-7$ is that the mean sojourn times (and mean queue sizes) are quite insensitive to the processing time distribution. This property of JSQ was first observed and heavily investigated in [22] (for a system without background streams). Obviously using our file splitting rule and taking $\alpha = 0$ or 1 yields two multi-class PS queues which are known to be exactly insensitive (one of the two queues is single class). When $\alpha \neq 0, 1$ this is no longer the case, yet the figure show that even when using $\alpha = \alpha^*$, the queues are "nearly insensitive". It is important to note that in [22] the authors show that not all routing policies have this "near insensitivity" property (even though a single PS queue is insensitive). Note that the "magnitude" of the sensitivity of our splitting rule is similar to that of JSQ: The maximum difference in mean sojourn times due to the file size distribution is of the order of 4%.

5 Tail Behavior vs. Mean Behavior

Following Theorem 1 and Conjecture 1, we know that α^* is a tail optimal splitting rule. In addition, as observed in Figure 4 it nearly optimizes the mean. We now present two possible reasons for this "*buy one, get an approximate one for free*" relation between the optimization of the sojourn time tail and optimization of the mean sojourn time. Explanation 1 below is specific to our model and uses the asymptotic properties of the processes $R_i(x)$. Explanation 2 that follows presents a simple general result regarding performance analysis of tails and means.

Explanation 1 Fix an arbitrary splitting rule α. Denote $R(x) := \min_{i=1,\dots,N} \frac{R_i(x)}{\alpha_i}$. Observe that $\frac{R(x)}{x} \to \gamma_m$ and $\frac{R^{-1}(x)}{x} \to \frac{1}{\gamma_m}$, where the convergences are a.s.

We have that $P(M > x) = P(B > R(x))$ and thus defining $M(b)$ as the sojourn time of a foreground file of size b, we have that $M(b) = R^{-1}(b)$. Define $\mu(b) := \mathbb{E}\ [M(b)]$. Since the underlying queue is regenerative, the almost sure convergence implies, $\frac{\mu(b)}{b} \to \frac{1}{\gamma_m}$ as $b \to \infty$. As a result, for large b:

$$\mu(b) \approx \frac{b}{\gamma_m}. \tag{14}$$

Thus selecting α such that γ_m is maximal minimizes $\mu(b)$ when b is large. It thus also approximately minimizes the unconditional sojourn time $\mathbb{E}\ [M] = \mathbb{E}\ _B[\mu(B)]$ where B is distributed as a foreground file size.

Further observe that the relation (14) is similar to the distinctive feature of a standard processor sharing queue where the approximate equality is exact. This property also sheds light on the near insensitivity of our system since for large b it behaves similarly to a processor sharing queue.

A further observation is that the splitting rule α^* ensures $\mathbb{E}\ [V_i]$ equal. We know that $\mathbb{E}\ [M] \geq \mathbb{E}\ [V_i]$ and also for a job of size b, we have $\mathbb{E}\ [M(b)] \geq \mathbb{E}\ [V_i(b)]$. The auxiliary results we get for the reduced load equivalence suggest that, especially for large jobs, $\mathbb{E}\ [M(b)]$ and $\mathbb{E}\ [V_i(b)]$ are not too far apart.

Explanation 2 Consider an arbitrary stochastic model parameterized by α. Assume that the choice of α induces a non-negative distribution $1 - \overline{F}_\alpha(x)$ with mean μ_α. For simplicity assume that α is scalar and that $1 - \overline{F}_\alpha(x)$ is absolutely continuous. In the case of our model (for $N = 2$), $\alpha = \alpha_1$ and the distribution is that of the sojourn time.

Lemma 3. *Assume that $\overline{F}_\alpha(x)$ is unimodal in α and that $\overline{F}_\alpha(x)$ and μ_α are differentiable in α, then there exists an $x > 0$ such that*

$$argmin_\alpha \mu_\alpha = argmin_\alpha \overline{F}_\alpha(x)$$

The above is result may be observed in Figure 2 where the trajectory of $\alpha^*(x)$, appears to cross the dotted $\mathbb{E}\ [M_\alpha]$ curve at its minimum. While typically finding the x at which these two curves cross, is difficult and not of practical importance, systems in which $\alpha^*(x)$ does not vary greatly in x will nearly optimize the mean when optimizing the tail. This appears to be the case in our system. Since $\alpha^*(x)$ trajectories do not vary greatly in x.

Proof. Denote $\tilde{\alpha}$ a minimizer of μ_α. Denote $\mu'(\alpha) = \frac{d}{d\alpha}\mu_\alpha$. Then we have $\mu'(\tilde{\alpha}) = 0$. We also know that $\mu_\alpha = \int_0^\infty \overline{F}_\alpha(u)du$. Denote $\overline{F}'(\alpha, u) = \frac{d}{d\alpha}\overline{F}_\alpha(u)$ Combining the above we have,

$$0 = \int_0^\infty \overline{F}'(\tilde{\alpha}, u)du,$$

Thus $\overline{F}'(\tilde{\alpha}, u)$ is either constantly 0 or has to be both negative and positive and thus there must be a $\tilde{u}$ for which it equals 0. Thus since $\overline{F}_\alpha(x)$ is unimodal in α then for $x = \tilde{u}$ it is optimized by $\tilde{\alpha}$.

6 Acknowledgments

We would like to thank Yoav Kerner for useful discussions. The work reported in this paper was supported, in part, by the Netherlands Organization for Scientific Research (NWO) under the Casimir project: Analysis of Distribution Strategies for Concurrent Access in Wireless Communication Networks. Bert Zwart's research is partly supported by NSF grants 0727400 and 0805979, an IBM faculty award, and a VIDI grant from NWO.

References

[1] A.P.Zwart. Sojourn times in a multiclass processor sharing queue. In *Proceedings of the 16th International Teletraffic Congress - ITC16, eds. P. Key, D. Smith (North-Holland, Amsterdam)*, pages 335–344, Edinburgh, UK, 1999.

[2] S.C. Borst, O.J. Boxma, and N. Hegde. Sojourn times in finite-capacity processor-sharing queues. In *Proceedings NGI 2005 Conference*, 2005.

[3] R. Chandra, P. Bahl, and P. Bahl. Multinet: Connecting to multiple IEEE 802.11 networks using a single wireless card. In *Proceedings of IEEE INFOCOM*, 2004.

[4] IEEE Unapproved Draft Std P802.11n D3.00. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY), amendment 4: Enhancements for higher throughput. September 2007.

[5] D.Cox. Fundamental limitations on the data rate in wireless systems. *IEEE Communications Magazine*, 46(12):16–17, 2008.

[6] J. Duncanson. Inverse multiplexing. *IEEE Communications Magazine*, 32(4):34–41, 1994.

[7] E.Altman, U. Ayesta, and B. Prabhu. Load balancing in processor sharing systems. In *Proceedings of the Second International Workshop on Game Theory in Communication Networks 2008, GameComm 2008, October 20, 2008, Athens Greece.* HAL - CCSD, 2008.

[8] Koudouris et al. Generic link layer functionality for multi-radio access networks. In *Proceedings 14th IST Mobile and Wireless Communications Summit*, 2005.

[9] F.Baccelli, W.A.Massey, and D.Towsley. Acyclic fork-join queuing networks. *Journal of the ACM*, 36(3):615–642, 1989.

[10] Federal Communications Commission Spectrum Policy Task Force. Report of the spectrum efficiency working group. Technical report, FCC-Federal Communications Commission, November 2002.

[11] G.J.Hoekstra and R.D. van der Mei. On the processor sharing of file transfers in wireless lans. In *Proceedings of the 69th IEEE Vehicular Technology Conference, VTC Spring 2009, 26-29 April 2009, Barcelona, Spain*. IEEE, 2009.

[12] C. Gkantsidis, M. Ammar, and E. Zegura. On the effect of large-scale deployment of parallel downloading. In *WIAPP '03: Proceedings of the The Third IEEE Workshop on Internet Applications*, page 79, Washington, DC, USA, 2003. IEEE Computer Society.

[13] F. Guillemin, Ph. Robert, and A.P. Zwart. Tail asymptotics for processor sharing queues. *Advances in Applied Probability*, 36:525–543, 2004.

[14] Y. Hasegawa, I. Yamaguchi, T. Hama, H. Shimonishi, and T. Murase. Deployable multipath communication scheme with sufficient performance data distribution method. *Computer Communications*, 30(17):3285–3292, 2007.

[15] G.J. Hoekstra and F.J.M Panken. Increasing throughput of data applications on heterogeneous wireless access networks. In *Proceedings 12th IEEE Symposium on Communication and Vehicular Technology in the Benelux*, 2005.

[16] H.Y. Hsieh and R. Sivakumar. A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 83–94, New York, NY, USA, 2002. ACM.

[17] R. Litjens, F. Roijers, J.L. Van den Berg, R.J. Boucherie, and M.J. Fleuren. Performance analysis of wireless LANs: An integrated packet/flow level approach. In *Proceedings of the 18th International Teletraffic Congress - ITC18*, pages 931–940, Berlin, Germany, 2003.

[18] L.Kleinrock. Time-shared systems: a theoretical treatment. *Journal of the ACM*, 14(2):242–261, 1967.

[19] O.Boxma and B.Zwart. Tails in scheduling. *SIGMETRICS Performance Evaluation Review*, 34(4):13–20, 2007.

[20] P. Rodriguez, A. Kirpal, and E. Biersack. Parallel-access for mirror sites in the internet. In *INFOCOM*, pages 864–873, 2000.

[21] S.Borst, R.Nú nez-Queija, and B.Zwart. Sojourn time asymptotics in processor-sharing queues. *Queueing Systems: Theory and Applications*, 53(1-2):31–51, 2006.

[22] V.Gupta, M.Harchol Balter, K.Sigman, and W.Whitt. Analysis of join-the-shortest-queue routing for web server farms. *Performance Evaluation*, 64(9-12):1062–1081, 2007.

[23] Y. Wu, C. Williamson, and J. Luo. On processor sharing and its applications to cellular data network provisioning. *Performance Evaluation*, 64(9-12):892–908, 2007.