Centrum voor Wiskunde en Informatica

REPORT *RAPPORT*

PNA

Probability, Networks and Algorithms

Probability, Networks and Algorithms

PNA

Traffic-splitting networks operating under alpha-fair sharing policies and balanced fairness

P.M.D. Lieshout

REPORT PNA-R0707 JULY 2007

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)

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

Traffic-splitting networks operating under alpha-fair sharing policies and balanced fairness

ABSTRACT
We consider a data network in which, besides classes of users that use specific routes, one class of users can split its traffic over several routes. We consider load balancing at the packet-level, implying that traffic of this class of users can be divided among several routes at the same time. Assuming that load balancing is based on an alpha-fair sharing policy, we show that the network has multiple possible behaviors. In particular, we show that some classes of users, depending on the state of the network, share capacity according to some Discriminatory Processor Sharing (DPS) model, whereas each of the remaining classes of users behaves as in a single-class single-node model. We compare the performance of this network with that of a similar network, where packet-level load balancing is based on balanced fairness. We derive explicit expressions for the mean number of users under balanced fairness, and show by conducting extensive simulation experiments that these provide accurate approximations for the ones under alpha-fair sharing.

2000 Mathematics Subject Classification: 60K25; 68M20
Keywords and Phrases: Traffic splitting; Alpha-fair sharing; Balanced fairness
Note: This research has been funded by the Dutch Bsik/BRICKS (Basic Research in Informatics for Creating the Knowledge Society) project.

Traffic-Splitting Networks Operating under Alpha-fair Sharing Policies and Balanced Fairness

P. Lieshout

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Email: lieshout@cwi.nl

July 7, 2007

Abstract

We consider a data network in which, besides classes of users that use specific routes, one class of users can split its traffic over several routes. We consider load balancing at the packet-level, implying that traffic of this class of users can be divided among several routes at the same time. Assuming that load balancing is based on an alpha-fair sharing policy, we show that the network has multiple possible behaviors. In particular, we show that some classes of users, depending on the state of the network, share capacity according to some Discriminatory Processor Sharing (DPS) model, whereas each of the remaining classes of users behaves as in a single-class single-node model.

We compare the performance of this network with that of a similar network, where packet-level load balancing is based on balanced fairness. We derive explicit expressions for the mean number of users under balanced fairness, and show by conducting extensive simulation experiments that these provide accurate approximations for the ones under alpha-fair sharing. [1]

[1]This research has been funded by the Dutch BSIK/BRICKS (Basic Research in Informatics for Creating the Knowledge Society) project.

1 Introduction

The performance of communication networks can be improved when the service demands are efficiently divided among the available resources, so-called *load balancing.* One can apply either *static* or *dynamic* load balancing. In the former case the balancing is not affected by the state of the network, whereas in the latter case it does depend on the system state. It is clear that better performance can be achieved when using dynamic load balancing, but it is often hard to find the optimal load balancing policy. Even for simple systems such a dynamic load balancing problem has non-trivial solutions [16].

In this paper we analyze load balancing in data networks carrying elastic traffic, as considered by [12]. Transfers in such networks can be represented by flows. We may distinguish between load balancing at the *flow-level* or the *packet-level*, depending on whether an arriving flow is entirely directed to a specific route (that it uses until the flow is finished) or a flow can be split between several routes, respectively. This paper deals with packet-level load balancing, i.e., we assume that packets of a flow can be divided among several routes.

Due to the dynamic nature of traffic, it is in general complicated to analyze the performance of such networks. Flows arrive according to some stochastic process and bring along a random amount of work. For each given number of flows present in the system, the allocated service rates are determined by some sharing policy. As soon as the number of flows changes, it is assumed that these rates are adapted instantly.

We analyze a network in which, besides classes of users that use specific routes, one class of users can split its traffic over several routes. We note that this network is the simplest system to analyze the performance and potential gains of load balancing at the packet-level, and it is therefore of particular interest to gain insight. In addition, this system also accounts for rather explicit results.

We assume that packet-level load balancing is based on an *alpha-fair bandwidth sharing* policy as introduced in [13]. The family of alpha-fair policies covers several common notions of fairness as special cases, such as *max-min fairness* ($\alpha \to \infty$), *proportional fairness* ($\alpha \to 1$) and *maximum throughput* ($\alpha \to 0$). In [14] it has also been shown that the case $\alpha = 2$, with additional class weights set inversely proportional to the respective round trip times, provides a reasonable modeling abstraction for the bandwidth sharing realized by TCP (Transmission Control Protocol) in the Internet.

We show that the above network has multiple possible behaviors. In particular, we show that packet-level load balancing based on alpha-fair sharing implies that some classes of users, depending on the state of the network, share capacity according to some Discriminatory Processor Sharing (DPS) model, whereas each of the remaining classes of users behaves as in a single-class single-node model.

The flow-level performance of the above network is compared with that of a similar network, where packet-level load balancing is based on *balanced fairness*, so-called *insensitive* load balancing at the packet-level. The term 'insensitive' refers to the fact that the corresponding steady-state distribution depends on the traffic characteristics through the traffic intensity only. Insensitive load balancing at the flow-level was first introduced in [4], and extended to insensitive load balancing at the packet-level in [10]. Optimal insensitive load balancing at the flow-level utilizing local state information was addressed in [1]. In [8] it was shown that one can achieve

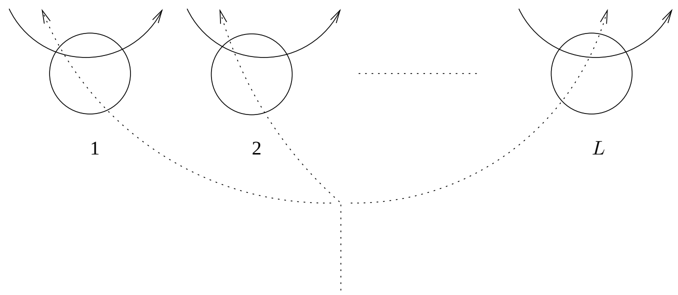

Figure 1: The bandwidth-sharing network.

even better performance if capacity allocation and load balancing are optimized jointly. A comparison between packet-level and flow-level insensitive load balancing was conducted in [11].

Assuming Poisson arrivals and exponentially distributed service requirements, the dynamics of the flow population may be described by a Markov process under both packet-level load balancing policies. We derive closed-form expressions for the mean number of users of each class under insensitive load balancing. Extensive simulation experiments show that these are also quite accurate approximations for the ones in a similar network where load balancing is based on alpha-fair sharing, for which no explicit expressions are available.

The above results are in line with the findings of [3], in which it was shown that the performance of networks operating under unweighted max-min fairness, unweighted proportional fairness and balanced fairness is nearly similar. The results in this paper suggest that balanced fairness is in fact a reasonable approximation for all unweighted alpha-fair sharing policies.

The remainder of this paper is organized as follows. In Section 2 we first provide a detailed model description, and introduce balanced fairness and alpha-fair sharing. In the next section we consider the model for a fixed flow population, and we characterize how bandwidth is allocated under both policies. In Section 4 we consider the model at large time-scales, such that the state of the network varies, and we derive explicit expressions for the mean number of users under balanced fairness, and show by conducting extensive simulation experiments that these provide accurate approximations for the ones under alpha-fair sharing. In the next section we examine the gain than one can achieve for both policies by using packet-level load balancing instead of static or flow-level load balancing. Section 6 concludes with some final observations.

2 Model

We consider the network as depicted in Figure 1. The network consists of L nodes, where node i has service rate C_i, $i = 1, \dots, L$. There are $L + 1$ classes of users. Class i requires service at node i, $i = 1, \dots, L$, whereas class 0 can be served at all nodes at the same time, i.e., class-0 users can split their traffic.

We assume that class-i users arrive according to a Poisson process of rate λ_i, and have exponentially distributed service requirements with mean μ_i^{-1}, $i = 0, \dots, L$. The arrival processes are all independent. The traffic load of class i is then $\rho_i = \lambda_i \mu_i^{-1}$. Let $n = (n_0, \dots, n_L)$ denote the state of the network, with n_i representing the number of class-i users.

2.1 Balanced fairness

We first assume that the bandwidth is shared according to balanced fairness, as introduced in [4]. Let $\phi_i(n)$ denote the service rate allocated to class i, $i = 0, \ldots, L$, with balanced fairness, when the network is in state n (here $\phi_0(n) = \sum_{i=1}^{L} \phi_{0i}(n)$). These service rates have to satisfy the balance conditions

$$\frac{\phi_i(n-e_j)}{\phi_i(n)} = \frac{\phi_j(n-e_i)}{\phi_j(n)} \quad \forall i, j = 0, \ldots, L, \quad n_i, n_j > 0, \tag{1}$$

where e_i denotes the $(i+1)$th unit vector in $\mathbb{R}^{L+1}$. All balanced service rates can be expressed in terms of a unique balance function $\Phi(\cdot)$, so that $\Phi(0) = 1$ and

$$\phi_i(n) = \frac{\Phi(n-e_i)}{\Phi(n)} \quad \forall n : n_i > 0, \quad i = 0, \ldots, L. \tag{2}$$

Hence, characterization of $\Phi(n)$ implies that $\phi_i(n)$ is characterized as well. Define $\Phi(n) = 0$ if $n \notin \mathbb{N}_0^{L+1}$. In order to obtain $\Phi(n)$, we need to solve the following maximization problem for each $n \in \mathbb{N}_0^{L+1} \backslash \{0\}$:

$$\begin{aligned}
(BF) \quad \max \quad & \Phi(n)^{-1} \\
\text{s.t.} \quad & \sum_{j=1}^{L} \phi_{0j}(n) = \frac{\Phi(n-e_0)}{\Phi(n)} \\
& \phi_i(n) = \frac{\Phi(n-e_i)}{\Phi(n)}, \quad i = 1, \ldots, L \\
& \phi_{0i}(n) + \phi_i(n) \leq C_i, \quad i = 1, \ldots, L \\
& \phi_{0i}(n), \phi_i(n) \geq 0, \quad i = 1, \ldots, L.
\end{aligned}$$

It is clear that $\Phi(n)$ can be obtained recursively: the $\Phi(n - e_i)$s are required to determine $\Phi(n)$. Also note that (BF) is a simple LP-problem, which can be solved using standard LP algorithms. In Section 3.1, however, we solve (BF) by rewriting the LP-problem in terms of a related network.

2.2 Alpha-fair sharing

We next assume that the network operates under a so-called alpha-fair sharing policy, as introduced in [13]. When the network is in state $n \neq 0$, the service rate x_i^* allocated to each of the class-i users is obtained by solving the following optimization problem:

$$\begin{aligned}
(AF) \quad \max \quad & G(x) \\
\text{s.t.} \quad & n_0 x_{0i} + n_i x_i \leq C_i, \quad i = 1, \ldots, L \\
& x_{0i}, x_i \geq 0, \quad i = 1, \ldots, L,
\end{aligned}$$

where the objective function $G(x)$ is defined by

$$G(x) := \begin{cases} n_0 \kappa_0 \frac{(\sum_{i=1}^{L} x_{0i})^{1-\alpha}}{1-\alpha} + \sum_{i=1}^{L} n_i \kappa_i \frac{x_i^{1-\alpha}}{1-\alpha} & \text{if } \alpha \in (0, \infty) \backslash \{1\}; \\ n_0 \kappa_0 \log(\sum_{i=1}^{L} x_{0i}) + \sum_{i=1}^{L} n_i \kappa_i \log(x_i) & \text{if } \alpha = 1. \end{cases}$$

The κ_is are non-negative class weights, and $\alpha \in (0,\infty)$ may be interpreted as a fairness coefficient. The cases $\alpha \to 0$, $\alpha \to 1$ and $\alpha \to \infty$ correspond to allocations which achieve *maximum throughput*, *proportional fairness*, and *max-min fairness*, respectively. The value of x_{0i}^* denotes how much capacity is assigned to path i (that requires service at node i) of class 0. Here $x_0^* = \sum_{i=1}^{L} x_{0i}^*$ denotes how much capacity is assigned to a single class-0 user in the network. Let $s_i(n) := x_i^* n_i$ denote the total service rate allocated to class i, $i = 0,\ldots,L$.

3 Static setting

In this section we consider the model for a fixed flow population, i.e., the state $n \in \mathbb{N}_0^{L+1}\backslash\{0\}$ is fixed, and we derive how bandwidth is shared between the various classes in case of balanced fairness and alpha-fair sharing, respectively. The difficulty in solving problem (BF) and (AF), as presented in the previous section, is that no explicit expressions are available for their optimal solutions. We first show that the network depicted in Figure 1 is equivalent to another network. In order to do so, let us first introduce the notion of the capacity set.

The allocations $\phi(n) = (\phi_0(n),\ldots,\phi_L(n))$ and $s(n) = (s_0(n),\ldots,s_L(n))$ are clearly constrained by the capacity set $\mathcal{C} \subseteq \mathbb{R}_+^{L+1}$:

$$\mathcal{C} := \left\{ x \geq 0 : \exists a_1,\ldots,a_L \geq 0, \quad \sum_{j=1}^{L} a_j = 1, \quad a_i x_0 + x_i \leq C_i, \quad i = 1,\ldots,L \right\},$$

i.e., $\phi(n) \in \mathcal{C}$ and $s(n) \in \mathcal{C}$ for all $n \in \mathbb{N}_0^{L+1}$. It is straightforward to show that the capacity set $\mathcal{C}$ can also be expressed as

$$\tilde{\mathcal{C}} := \left\{ x \geq 0 : \sum_{j=0}^{L} x_j \leq \sum_{j=1}^{L} C_j, \quad x_i \leq C_i, \quad i = 1,\ldots,L \right\},$$

i.e., $\mathcal{C} = \tilde{\mathcal{C}}$. Since $\tilde{\mathcal{C}}$ is the capacity set corresponding to the tree network depicted in Figure 2, it follows that the networks depicted in Figures 1 and 2 are in fact equivalent. The tree has a common link with capacity $C_1 + \cdots + C_L$, and $L+1$ branches with capacities ∞, $C_1,\ldots,C_L$, respectively. In this network class-i users require service at the node with service rate C_i and at the common link, $i = 1,\ldots,L$, whereas class-0 users only require service at the common link. Note that each class of users corresponds to a specific route in the tree network.

As a side remark we mention that in general it is not true that a network (where some classes of users can split their traffic over several routes at the same time) can be converted in a tree network. In fact, if we extend the model depicted in Figure 1 by adding a class of users that requires service at all L nodes simultaneously, then it is already not possible to represent the network as a tree network. However, we note that in general one may still be able to convert a traffic-splitting network in some other network (with dummy nodes) without traffic splitting.

3.1 Balanced fairness

In this subsection we derive the balanced fairness allocation by solving problem (BF). Since the models depicted in Figures 1 and 2 are equivalent, it follows that the balance function $\tilde{\Phi}(\cdot)$

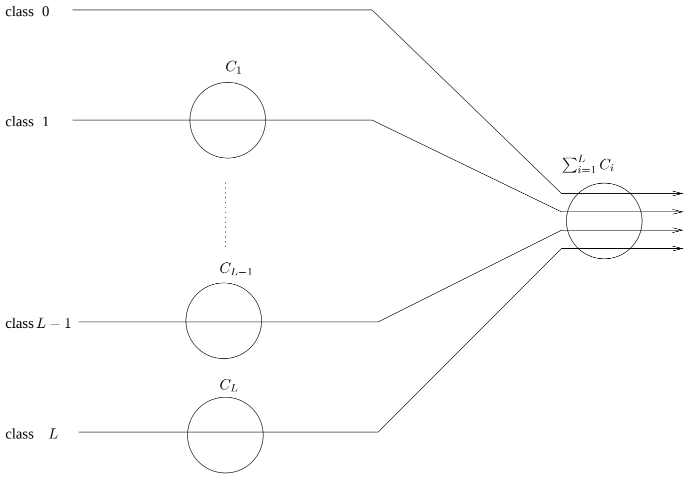

Figure 2: Tree network

corresponding to tree network coincides with $\Phi(\cdot)$, i.e., $\tilde{\Phi}(\cdot) = \Phi(\cdot)$, see [3]. In the following lemma we present the solution of the optimization problem (BF).

Lemma 3.1 *The balanced fairness function $\Phi(n)$ satisfies, with $\Phi(0) = 1$,*

$$\Phi(n) = \max\left\{\frac{\Phi(n-e_1)}{C_1}, \ldots, \frac{\Phi(n-e_L)}{C_L}, \frac{\sum_{i=0}^{L}\Phi(n-e_i)}{\sum_{i=1}^{L} C_i}\right\}, \quad n \in \mathbb{N}_0^{L+1}\backslash\{0\}. \tag{3}$$

Proof: From the above it follows that we can obtain $\Phi(\cdot)$ by determining $\tilde{\Phi}(\cdot)$, as they are the same. Subsequently, $\tilde{\Phi}(\cdot)$ is obtained by using Equation (2) in [5]. □

We note that Lemma 3.1 is in agreement with Equation (19) in [10]. From Lemma 3.1 it follows that $\Phi(n)$ can be obtained recursively. The total service rate allocated to class i, $i = 0, \ldots, L$, in each state $n \in \mathbb{N}_0^{L+1}$ can be obtained using Lemma 3.1 and (2).

3.2 Alpha-fair sharing

In this subsection we focus on the alpha-fair allocation, that is obtained by solving problem (AF). Similar to the previous subsection, we can obtain the alpha-fair allocation $s(n)$ by determining the alpha-fair allocation $\tilde{s}(n)$ in the tree network, as both networks are the same, implying that $s(n) = \tilde{s}(n)$. In order to obtain $\tilde{s}(n)$ we need to solve the following maximization problem:

$$\begin{aligned} (AF2) \quad \max \quad & H(x) \\ \text{s.t.} \quad & \sum_{i=0}^{L} n_i x_i \le \sum_{i=1}^{L} C_i, \\ & n_i x_i \le C_i, \quad i = 1, \ldots, L, \\ & x_i \ge 0, \quad i = 0, \ldots, L, \end{aligned} \tag{4}$$

where the objective function $H(x)$ is defined by

$$H(x) := \begin{cases} \sum_{i=0}^{L} n_i \kappa_i \frac{x_i^{1-\alpha}}{1-\alpha} & \text{if } \alpha \in (0,\infty)\backslash\{1\}; \\ \sum_{i=0}^{L} n_i \kappa_i \log(x_i) & \text{if } \alpha = 1. \end{cases}$$

Below we show that $(AF2)$ is solvable, but the optimal solution strongly depends on the state $n \neq 0$. We present a simple algorithm for obtaining the alpha-fair allocation.

Lemma 3.2 *The alpha-fair allocation $s(n)$ can be obtained with the following algorithm:*

```
Set Stop:=False
Set S := {0,...,L}
WHILE Stop=False DO
  Determine the |S|-class DPS allocation:  s_i(n) := n_i κ_i^{1/α} Σ_{j∈S\{0}} C_j / Σ_{j∈S} n_j κ_j^{1/α},  i ∈ S
  IF s_i(n) ≤ C_i for all i ∈ S\{0} THEN set Stop:=True
  ELSE
    Take any i* ∈ S\{0} such that s_{i*}(n) > C_i
    Set S := S\{i*}
    Set s_{i*}(n) := C_i
  END
END
```

Proof: First consider the Karush-Kuhn-Tucker (KKT) necessary conditions for problem $(AF2)$. If x is an optimal solution to problem $(AF2)$, then there exist constants $p_i \geq 0$, $i = 0, \ldots, L$, such that,

$$\frac{n_0 \kappa_0}{x_0^{\alpha}} - n_0 p_0; \tag{5}$$

$$\frac{n_i \kappa_i}{x_i^{\alpha}} - n_i(p_0 + p_i), \quad i = 1, \ldots, L; \tag{6}$$

$$p_0 \left(\sum_{i=1}^{L} C_i - \sum_{i=0}^{L} n_i x_i \right) = 0; \tag{7}$$

$$p_i \left(C_i - n_i x_i \right) = 0, \quad i = 1, \ldots, L. \tag{8}$$

Note that (5) and (6) hold for any $\alpha \in (0, \infty)$. Solving (5)-(8) for $(x_0, \ldots, x_L)$ and $(p_0, \ldots, p_L)$ yields $\sum_{q=1}^{L} \frac{L!}{q!(L-q)!} = 2^L - 1$ possible solutions, however, depending on the state of the network n, only one of the $2^L - 1$ solutions, x^*, is such that $p_i \geq 0$, $i = 0, \ldots, L$, i.e., this is the optimal solution for $(AF2)$. For each of the other solutions there exists at least one Lagrange parameter that is negative, implying that these solutions cannot be optimal. Note that the existence of a unique optimal solution x^* for $(AF2)$ also follows as $H(x)$ is strictly concave and the constraints are linear. Straightforward calculus shows that the corresponding alpha-fair allocation $\tilde{s}_i(n) = s_i(n) = n_i x_i^*$, $i = 0, \ldots, L$, can be obtained by the above algorithm. The algorithm reflects that $2^L - 1$ solutions exist for (5)-(8), but it also shows that only one of these solutions, x^*, is found after termination of the algorithm. The Lagrange parameters corresponding to x^* are such that $p_i = 0$ if $i \in S\backslash\{0\}$, and $p_i > 0$ if $i \notin S\backslash\{0\}$, where S is the set obtained after termination of the algorithm. Furthermore, $p_0 = 0$ if $n_0 = 0$ and if there exists an i such that $n_i = 0$, $i = 1, \dots, L$, otherwise $p_0 > 0$. □

4 Flow-level dynamics

In the previous section we considered the model for a fixed flow population, and we derived expressions for the balanced fairness and alpha-fair allocations in each state of the network. In this section we analyze the model at sufficiently large time scales. In this case we also have to take the random nature of the traffic into account, i.e., the state of the network n varies at large time scales.

4.1 Balanced fairness

Let $N(t) = (N_0(t), \dots, N_L(t))$ denote the state of the network at time t. Since we assumed Poisson arrivals and exponentially distributed service requirements, $N(t)$ is a Markov process with transition rates:

$$q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \mu_i \phi_i(n), \quad i = 0, \dots, L,$$

in case of balanced fairness. In [3] it was shown that the process $N(t)$ is stable if there exists $(\tilde{\rho}_{01}, \dots, \tilde{\rho}_{0L})$ such that

$$\sum_{i=1}^{L} \tilde{\rho}_{0i} = \rho_0 \quad \text{and} \quad \tilde{\rho_{0i}} + \rho_i < C_i, \quad i = 1, \dots, L, \tag{9}$$

or equivalently, if

$$\sum_{i=0}^{L} \rho_i < \sum_{j=1}^{L} C_j \quad \text{and} \quad \rho_i < C_i, \quad i = 1, \dots, L. \tag{10}$$

It may be verified from (1) that the steady-state queue length distribution is given by

$$\pi(n) = \frac{1}{G(\rho)} \Phi(n) \prod_{i=0}^{L} \rho_i^{n_i}, \quad n \in \mathbb{N}_0^{L+1}, \tag{11}$$

where the normalization constant $G(\rho)$ equals

$$G(\rho) = G(\rho_0, \dots, \rho_L) = \sum_{n_0=0}^{\infty} \cdots \sum_{n_L=0}^{\infty} \Phi(n) \prod_{i=0}^{L} \rho_i^{n_i}.$$

As a side remark we mention that (11) in fact holds for much more general traffic characteristics, see [4] for a more detailed treatment.

When applying Little's formula we find that

$$\mathbb{E}N_i^{BF} = \rho_i \frac{\frac{\partial G(\rho)}{\partial \rho_i}}{G(\rho)} = \rho_i \frac{\partial \log G(\rho)}{\partial \rho_i}, \quad i = 0, \dots, L, \tag{12}$$

i.e., characterization of $G(\rho)$ implies that $\mathbb{E}N_i^{BF}$, $i = 0, \ldots, L$, is known as well.

By exploiting the results of [6] on tree networks we can determine $G(\rho)$, and it can be verified that this results in

$$G(\rho) = \frac{1}{1 - \frac{\sum_{i=0}^{L}\rho_i}{\sum_{i=1}^{L}C_i}} \frac{1 - \frac{\sum_{i=1}^{L}\rho_i}{\sum_{i=1}^{L}C_i}}{\prod_{i=1}^{L}\left(1 - \frac{\rho_i}{C_i}\right)}. \tag{13}$$

Then by using (12) we can obtain a closed-form expression for $\mathbb{E}N_i^{BF}$, $i = 0, \ldots, L$. The expression for $\mathbb{E}N_i^{BF}$, $i = 1, \ldots, L$, is in general quite complicated, in contrast to the expression for the mean number of class-0 users, which is given by

$$\mathbb{E}N_0^{BF} = \frac{\rho_0}{\sum_{i=1}^{L}C_i - \sum_{i=0}^{L}\rho_i}.$$

From (13) it follows that $\mathbb{E}N_i^{BF}$, $i = 0, \ldots, L$, is finite if the stability condition (10) holds.

4.2 Alpha-fair sharing

As before, let $N(t) = (N_0(t), \ldots, N_L(t))$ denote the state of the network at time t. In case of alpha-fair sharing $N(t)$ is a Markov process with transition rates:

$$q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \mu_i s_i(n), \quad i = 0, \ldots, L.$$

Since our network is equivalent to the tree network depicted in Figure 2, it follows from Theorem 1 in [2] that the process $N(t)$ is stable if (9) holds.

Lemma 3.2 shows that, depending on the state of the network $n \in \mathbb{N}_0^{L+1}$, the network has $2^L - 1$ possible behaviors. This illustrates the complication of finding closed-form expressions for the mean number of users of each class. In fact, so far no expressions for the mean number of users are available in case of alpha-fair sharing. To gain some insight, we derive in this section approximations for the mean number of users of each class, i.e., $\mathbb{E}N_i^{AF}$, $i = 0, \ldots, L$. The approximations are validated by means of simulation experiments. We consider the case where the network consists of $L = 2$ nodes, but we note that the approximations can be extended to the case $L > 2$ in a similar fashion.

Using Lemma 3.2 in Section 3.2, it follows that the network, depending on the state n, has three possible behaviors: (i) if

$$n_1 > \frac{C_1}{C_2}\left(\left(\frac{\kappa_2}{\kappa_1}\right)^{1/\alpha} n_2 + \left(\frac{\kappa_0}{\kappa_1}\right)^{1/\alpha} n_0\right),$$

then classes 0 and 2 behave as in a two-class DPS model with capacity C_2 and relative weights $\kappa_i^{1/\alpha}$, $i = 0, 2$, whereas class 1 behaves as an M/M/1 queue with arrival rate λ_1 and service rate $\mu_1 C_1$; (ii) If

$$n_1 < \frac{C_1}{C_2}\left(\frac{\kappa_2}{\kappa_1}\right)^{1/\alpha} n_2 - \left(\frac{\kappa_0}{\kappa_1}\right)^{1/\alpha} n_0,$$

then classes 0 and 1 behave as in a two-class DPS model with capacity C_1 and relative weights $\kappa_i^{1/\alpha}$, $i = 0, 1$, whereas class 2 behaves as an M/M/1 queue with arrival rate λ_2 and service rate $\mu_2 C_2$; (iii) otherwise the network will behave as in a three-class DPS model with capacity $C_1 + C_2$ and relative weights $\kappa_i^{1/\alpha}$, $i = 0, 1, 2$.

If the network were to behave as (i) all the time and if $\rho_1 < C_1$ and $\rho_0 + \rho_2 < C_2$ (stability conditions), then by exploiting the results of [7] we would obtain

$$\mathbb{E}N_0^{(i)} = \frac{\rho_0}{C_2 - \rho_0 - \rho_2}\left(1 + \frac{\mu_0 \rho_2 \left(\kappa_2^{1/\alpha} - \kappa_0^{1/\alpha}\right)}{\kappa_0^{1/\alpha}\mu_0(C_2 - \rho_0) + \kappa_2^{1/\alpha}\mu_2(C_2 - \rho_2)}\right);$$

$$\mathbb{E}N_1^{(i)} = \frac{\rho_1}{C_1 - \rho_1};$$

$$\mathbb{E}N_2^{(i)} = \frac{\rho_2}{C_2 - \rho_0 - \rho_2}\left(1 + \frac{\mu_2 \rho_0 \left(\kappa_0^{1/\alpha} - \kappa_2^{1/\alpha}\right)}{\kappa_0^{1/\alpha}\mu_0(C_2 - \rho_0) + \kappa_2^{1/\alpha}\mu_2(C_2 - \rho_2)}\right).$$

Likewise, when the network behaves as (ii) and if $\rho_2 < C_2$ and $\rho_0 + \rho_1 < C_1$ (stability conditions), we find

$$\mathbb{E}N_0^{(ii)} = \frac{\rho_0}{C_1 - \rho_0 - \rho_1}\left(1 + \frac{\mu_0 \rho_1 \left(\kappa_1^{1/\alpha} - \kappa_0^{1/\alpha}\right)}{\kappa_0^{1/\alpha}\mu_0(C_1 - \rho_0) + \kappa_1^{1/\alpha}\mu_1(C_1 - \rho_1)}\right);$$

$$\mathbb{E}N_1^{(ii)} = \frac{\rho_1}{C_1 - \rho_0 - \rho_1}\left(1 + \frac{\mu_1 \rho_0 \left(\kappa_0^{1/\alpha} - \kappa_1^{1/\alpha}\right)}{\kappa_0^{1/\alpha}\mu_0(C_1 - \rho_0) + \kappa_1^{1/\alpha}\mu_1(C_1 - \rho_1)}\right);$$

$$\mathbb{E}N_2^{(ii)} = \frac{\rho_2}{C_2 - \rho_2}.$$

If the network behaves as a three-class DPS model, i.e., as (iii), and if $\rho_0 + \rho_1 + \rho_2 < C_1 + C_2$ (stability condition), then one can obtain the mean number of users of each class by solving the following set of linear equations for $\mathbb{E}N_i^{(iii)}$, $i = 0, 1, 2$:

$$(C_1 + C_2)\mathbb{E}N_i^{(iii)} - \lambda \sum_{j=0}^{2} \kappa_j^{1/\alpha} \frac{\frac{\lambda_j}{\lambda}\mathbb{E}N_i^{(iii)} + \frac{\lambda_i}{\lambda}\mathbb{E}N_j^{(iii)}}{\kappa_j^{1/\alpha}\mu_j + \kappa_i^{1/\alpha}\mu_i} = \rho_i, \quad i = 0, 1, 2,$$

where $\lambda := \lambda_0 + \lambda_1 + \lambda_2$, see [7]. In this case there also exists a closed-form expression for $\mathbb{E}N_i^{(iii)}$, $i = 0, 1, 2$, but it is complicated.

We propose the following approximation: $\mathbb{E}N_i^{AF} \approx \mathbb{E}N_i^{AP}$, $i = 0, 1, 2$, where

$$\mathbb{E}N_0^{AP} := \mathbb{E}N_0^{(iii)}; \quad \mathbb{E}N_1^{AP} := \max\{\mathbb{E}N_1^{(i)}, \mathbb{E}N_1^{(iii)}\}; \quad \mathbb{E}N_2^{AP} := \max\{\mathbb{E}N_2^{(ii)}, \mathbb{E}N_2^{(iii)}\}.$$

It can be verified that $\mathbb{E}N_0^{AP}$ is bounded if $\rho_0 + \rho_1 + \rho_2 < C_1 + C_2$, $\mathbb{E}N_1^{AP}$ is bounded if $\rho_1 < C_1$ and $\rho_0 + \rho_1 + \rho_2 < C_1 + C_2$, and $\mathbb{E}N_2^{AP}$ is bounded if $\rho_2 < C_2$ and $\rho_0 + \rho_1 + \rho_2 < C_1 + C_2$. Hence, the $\mathbb{E}N_i^{AP}$s are only all bounded if (9) holds, i.e., if the process $N(t)$ is also stable.

In [3] it was argued that the performance of a network under proportional fairness ($\alpha = 1$) and max-min fairness ($\alpha \to \infty$) is closely approximated by that under balanced fairness. Therefore, we also propose the following approximation: $\mathbb{E}N_i^{AF} \approx \mathbb{E}N_i^{BF}$, $i = 0, 1, 2$. The value of $\mathbb{E}N_i^{BF}$, $i = 0, 1, 2$, can be obtained using (12), and is independent of the value of α.

γ	$\mathbb{E}N_0^{AF}$	$\mathbb{E}N_1^{AF}$	$\mathbb{E}N_2^{AF}$	$\mathbb{E}N_0^{AP}$	$\mathbb{E}N_1^{AP}$	$\mathbb{E}N_2^{AP}$	$\mathbb{E}N_0^{BF}$	$\mathbb{E}N_1^{BF}$	$\mathbb{E}N_2^{BF}$
0.1	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	0.15	0.26	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	0.30	0.46	0.46	0.27	0.43	0.43	0.27	0.49	0.49
0.4	0.55	0.77	0.77	0.50	0.67	0.67	0.50	0.83	0.83
0.5	1.10	1.39	1.39	1.00	1.00	1.00	1.00	1.50	1.50
0.6	3.17	3.48	3.48	3.00	3.00	3.00	3.00	3.75	3.75

Table 1: Simulation results for scenario I.

γ	α	$\mathbb{E}N_0^{AF}$	$\mathbb{E}N_1^{AF}$	$\mathbb{E}N_2^{AF}$	$\mathbb{E}N_0^{AP}$	$\mathbb{E}N_1^{AP}$	$\mathbb{E}N_2^{AP}$	$\mathbb{E}N_0^{BF}$	$\mathbb{E}N_1^{BF}$	$\mathbb{E}N_2^{BF}$
0.1	1	0.06	0.12	0.12	0.06	0.11	0.11	0.06	0.11	0.11
0.2	1	0.13	0.28	0.27	0.12	0.25	0.25	0.14	0.27	0.27
0.3	1	0.23	0.54	0.49	0.22	0.43	0.43	0.27	0.49	0.49
0.4	1	0.39	0.97	0.83	0.35	0.67	0.67	0.50	0.83	0.83
0.5	1	0.68	1.95	1.46	0.59	1.43	1.00	1.00	1.50	1.50
0.6	1	1.55	5.93	3.47	1.38	4.82	2.80	3.00	3.75	3.75
0.1	2	0.06	0.12	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	2	0.14	0.27	0.26	0.13	0.25	0.25	0.14	0.27	0.27
0.3	2	0.26	0.50	0.48	0.24	0.43	0.43	0.27	0.49	0.49
0.4	2	0.47	0.88	0.81	0.42	0.67	0.67	0.50	0.83	0.83
0.5	2	0.87	1.71	1.44	0.77	1.23	1.00	1.00	1.50	1.50
0.6	2	2.35	4.81	3.66	2.06	3.95	2.98	3.00	3.75	3.75
0.1	5	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	5	0.15	0.26	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	5	0.28	0.48	0.46	0.26	0.43	0.43	0.27	0.49	0.49
0.4	5	0.52	0.82	0.78	0.46	0.67	0.67	0.50	0.83	0.83
0.5	5	1.00	1.51	1.40	0.90	1.09	1.00	1.00	1.50	1.50
0.6	5	2.84	3.95	3.61	2.60	3.38	3.01	3.00	3.75	3.75
0.1	∞	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	∞	0.15	0.26	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	∞	0.30	0.46	0.46	0.27	0.43	0.43	0.27	0.49	0.49
0.4	∞	0.55	0.77	0.77	0.50	0.67	0.67	0.50	0.83	0.83
0.5	∞	1.10	1.39	1.39	1.00	1.00	1.00	1.00	1.50	1.50
0.6	∞	3.17	3.48	3.48	3.00	3.00	3.00	3.00	3.75	3.75

Table 2: Simulation results for scenario II.

To examine the accuracy of the above approximations we have performed simulation experiments. We consider the setting with $C_1 = C_2 = 1$, and we take $\lambda_i = \gamma$, $\mu_i = 1$, $i = 0, 1, 2$, such that $\rho_0 = \rho_1 = \rho_2 = \gamma$. We first consider scenario I, where $\kappa_i = 1$, $i = 0, 1, 2$. Subsequently, we consider scenario II, where $\kappa_0 = 5$, $\kappa_1 = 1$ and $\kappa_2 = 2$. In scenario II we let the traffic load γ and the alpha-fair coefficient α vary, whereas in scenario I we only let γ vary, as it can be verified that the $\mathbb{E}N_i^{AF}$s and $\mathbb{E}N_i^{AP}$s are independent of the value of α in scenario I. To ensure stability we assume that $\gamma < \frac{2}{3}$. The results are reported in Tables 1 and 2. Each reported simulation value in these (and other) tables is measured over $4 \cdot 10^6$ events, i.e., arrivals or departures.

Remark: We have also determined a 95% confidence interval (CI) for each listed simulation value in this paper, but these are not presented. We note, however, that the relative efficiency, i.e., the ratio of the half-length of the CI to the reported simulation value, is less than 3% for all listed cases in Tables 1, 2, 5 and 6, and less than 10% for all listed cases in Tables 3 and 4.

Table 1 compares the value of $\mathbb{E}N_i^{AF}$ obtained by simulation with the approximations $\mathbb{E}N_i^{AP}$ and $\mathbb{E}N_i^{BF}$, $i = 0, 1, 2$, for scenario I. The results show that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AP}$, $i = 0, 1, 2$. Also, the table shows that $\mathbb{E}N_0^{AF} \geq \mathbb{E}N_0^{BF}$ and $\mathbb{E}N_i^{AF} \leq \mathbb{E}N_i^{BF}$, $i = 1, 2$. Overall we see that both approximations are accurate in case of equal class weights, especially for low traffic load.

Table 2 reports the results corresponding to scenario II, i.e., in case of unequal class weights. In this case $\mathbb{E}N_i^{AF}$ and $\mathbb{E}N_i^{AP}$ do depend on the value of α, as is shown in the table. Again, we see that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AP}$, $i = 0, 1, 2$. For low traffic loads both approximations perform quite well, but for high traffic loads we see that the balanced fairness approximation is less accurate than the other one.

Tables 1 and 2 show that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AP}$, $i = 0, 1, 2$, which may be explained as follows. First note that the rate allocated to class 1 is smaller than or equal to C_1 at all moments in time under alpha-fair sharing, whereas rate C_1 is continuously available to class 1 in (i). Clearly, this implies that $\mathbb{E}N_1^{AF} \geq \mathbb{E}N_1^{(i)}$. With similar reasoning, we find that $\mathbb{E}N_2^{AF} \geq \mathbb{E}N_2^{(ii)}$. Since class-$i$ users cannot be allocated more than C_i, $i = 1, 2$, under alpha-fair sharing, whereas in the three-class DPS model the upper bound is $C_1 + C_2$ for both classes, one may expect that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{(iii)}$, $i = 1, 2$. For any state $n \in \mathbb{N}_0^3 \backslash \{0\}$ it can be verified that the alpha-fair allocation to class 0 is larger or equal than the one obtained in the three-class DPS model, so one would expect $\mathbb{E}N_0^{AF} \leq \mathbb{E}N_0^{(iii)}$ at first sight. However, recall that we argued that the number of users of classes 1 and 2 in the model operating under alpha-fair sharing will (on average) be larger than in the three-class DPS model, which causes that the total service allocated to class 0 in the model operating under alpha-fair sharing is less than or equal to that in the three-class DPS model, i.e., we may also expect $\mathbb{E}N_0^{AF} \geq \mathbb{E}N_0^{(iii)}$. The above reasoning indeed suggests that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AP}$, $i = 0, 1, 2$.

4.2.1 Fluid and quasi-stationary regimes

To test the performance of the two approximations in case of extreme parameter values, we now assume that the flow dynamics of the various classes occur on widely separate time scales, i.e., in fluid and quasi-stationary regimes.

Formally, let $\lambda_i^{(r)} := \lambda_i f_i(r)$ and $\mu_i^{(r)} := \mu_i f_i(r)$, where $f_i(r)$ represents the time scale associated with class i as function of r, $i = 0, \ldots, L$. Note that the traffic intensity of class i equals $\rho_i^{(r)} := \lambda_i^{(r)}/\mu_i^{(r)} = \rho_i$, $i = 0, \ldots, L$, so it is independent of r. Let $N_i^{(r)}$ be the number of class-i flows in the r-th system. Before analyzing the quality of the approximations, we first present the following useful proposition.

Proposition 4.1 *Assume that $L+1$ classes of users share C units of capacity according to DPS, where class i has relative weight κ_i, $i = 0, \ldots, L$. If $f_i(r)/f_{i-1}(r) \to 0$ as $r \to \infty$, $i = 1, \ldots, L$, i.e., higher indexed classes operate on faster time scales, then*

$$\mathbb{E}N_i^{(\infty)} = \frac{\rho_i}{C - \sum_{j=i}^{L} \rho_i} + \sum_{j=0}^{i-1} \frac{\kappa_j}{\kappa_i} \frac{\rho_i \rho_j}{\left(C - \sum_{r=j}^{L} \rho_r\right)\left(C - \sum_{r=j+1}^{L} \rho_r\right)}, \quad i = 0, \ldots, L.$$

Proof: In [9] the above result was already proved for $L = 1$. For $L > 1$ the authors showed that $\mathbb{E}N_j^{(\infty)}$, $j = 1, \ldots, L$, could be obtained by determining $\mathbb{E}N_i^{(\infty)}$, $i = 0, \ldots, j-1$, i.e., as a recursion. Straightforward calculus, however, shows that this recursion reduces to the above result. □

γ	$\mathbb{E}N_0^{AF}$	$\mathbb{E}N_1^{AF}$	$\mathbb{E}N_2^{AF}$	$\mathbb{E}N_0^{AP(\infty)}$	$\mathbb{E}N_1^{AP(\infty)}$	$\mathbb{E}N_2^{AP(\infty)}$	$\mathbb{E}N_0^{BF}$	$\mathbb{E}N_1^{BF}$	$\mathbb{E}N_2^{BF}$
0.1	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	0.14	0.25	0.25	0.14	0.25	0.25	0.14	0.27	0.27
0.3	0.27	0.45	0.45	0.27	0.43	0.43	0.27	0.49	0.49
0.4	0.51	0.76	0.76	0.50	0.67	0.67	0.50	0.83	0.83
0.5	1.02	1.34	1.34	1.00	1.00	1.00	1.00	1.50	1.50
0.6	3.06	3.30	3.30	3.00	3.00	3.00	3.00	3.75	3.75

Table 3: Results corresponding to the fluid and quasi-stationary regimes (scenario I).

Let us return to the setting with $L = 2$ nodes and $L+1 = 3$ classes of users. Proposition 4.1 allows us to obtain simple closed-form expressions for $\mathbb{E}_i^{AP}$, $i = 0, 1, 2$, when $r \to \infty$. Assuming that higher indexed classes operate on faster time scales and that the stability conditions (10) hold, we find

$$\mathbb{E}N_0^{AP(\infty)} := \frac{\rho_0}{C_1 + C_2 - \rho_0 - \rho_1 - \rho_2};$$

$$\mathbb{E}N_1^{AP(\infty)} := \max\left\{\frac{\rho_1}{C_1 - \rho_1}, \frac{\rho_1}{C_1 + C_2 - \rho_1 - \rho_2} + \frac{\kappa_0^{1/\alpha}\rho_0\rho_1}{\kappa_1^{1/\alpha}(C_1 + C_2 - \rho_0 - \rho_1 - \rho_2)(C_1 + C_2 - \rho_1 - \rho_2)}\right\};$$

$$\mathbb{E}N_2^{AP(\infty)} := \max\left\{\frac{\rho_2}{C_2 - \rho_2}, \frac{\rho_2}{C_1 + C_2 - \rho_2} + \sum_{j=0}^{1}\frac{\kappa_j^{1/\alpha}\rho_j\rho_2}{\kappa_2^{1/\alpha}(C_1 + C_2 - \sum_{r=j}^{2}\rho_r)(C_1 + C_2 - \sum_{r=j+1}^{2}\rho_r)}\right\}.$$

In case of equal class weights, $\kappa_i = \kappa$, $i = 0, 1, 2$, it is not hard to see that

$$\mathbb{E}N_0^{AP(\infty)} = \frac{\rho_0}{C_1 + C_2 - \rho_0 - \rho_1 - \rho_2};$$

$$\mathbb{E}N_1^{AP(\infty)} = \max\left\{\frac{\rho_1}{C_1 - \rho_1}, \frac{\rho_1}{C_1 + C_2 - \rho_0 - \rho_1 - \rho_2}\right\};$$

$$\mathbb{E}N_2^{AP(\infty)} = \max\left\{\frac{\rho_2}{C_2 - \rho_2}, \frac{\rho_2}{C_1 + C_2 - \rho_0 - \rho_1 - \rho_2}\right\}.$$

Clearly, the $\mathbb{E}N_i^{AP(\infty)}$s strongly depend on the ordering of the classes with respect to the time scales. In case of other orderings than the one mentioned above one can obtain expressions in a similar fashion.

The accuracy of the approximations in the fluid and quasi-stationary regimes is examined by performing simulation experiments. We take $C_1 = C_2 = 1$, $\lambda_0 = \gamma$, $\lambda_1 = 10\gamma$, $\lambda_2 = 100\gamma$, $\mu_0 = 1$, $\mu_1 = 10$, $\mu_2 = 100$, so that $\rho_i = \gamma$, $i = 0, 1, 2$, and assume that higher indexed classes operate on faster time scales.

Tables 3 and 4 report the results for scenario I and II, respectively. Recall that the $\mathbb{E}N_i^{AF}$s and $\mathbb{E}N_i^{AP(\infty)}$s are independent of the value of α in scenario I, whereas they are sensitive to the value of α in scenario II. The tables show that also in the fluid and quasi-stationary regimes the approximations are promising.

5 Comparison with static and flow-level load balancing

In the previous sections we considered load balancing at the packet-level. In this section we quantify how much better packet-level load balancing is than static and flow-level load balancing.

γ	α	$\mathbb{E}N_0^{AF}$	$\mathbb{E}N_1^{AF}$	$\mathbb{E}N_2^{AF}$	$\mathbb{E}N_0^{AP(\infty)}$	$\mathbb{E}N_1^{AP(\infty)}$	$\mathbb{E}N_2^{AP(\infty)}$	$\mathbb{E}N_0^{BF}$	$\mathbb{E}N_1^{BF}$	$\mathbb{E}N_2^{BF}$
0.1	1	0.06	0.12	0.12	0.06	0.11	0.11	0.06	0.11	0.11
0.2	1	0.14	0.31	0.28	0.14	0.25	0.25	0.14	0.27	0.27
0.3	1	0.26	0.63	0.52	0.27	0.51	0.43	0.27	0.49	0.49
0.4	1	0.45	1.23	0.92	0.50	1.17	0.71	0.50	0.83	0.83
0.5	1	0.89	2.85	1.82	1.00	3.00	1.67	1.00	1.50	1.50
0.6	1	2.49	10.28	5.44	3.00	12.00	6.21	3.00	3.75	3.75
0.1	2	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	2	0.14	0.27	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	2	0.27	0.51	0.48	0.27	0.43	0.43	0.27	0.49	0.49
0.4	2	0.49	0.93	0.83	0.50	0.71	0.67	0.50	0.83	0.83
0.5	2	1.03	1.94	1.58	1.00	1.62	1.24	1.00	1.50	1.50
0.6	2	2.69	5.53	4.17	3.00	5.78	4.21	3.00	3.75	3.75
0.1	5	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	5	0.14	0.26	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	5	0.27	0.47	0.46	0.27	0.43	0.43	0.27	0.49	0.49
0.4	5	0.52	0.82	0.79	0.50	0.67	0.67	0.50	0.83	0.83
0.5	5	1.00	1.51	1.42	1.00	1.19	1.08	1.00	1.50	1.50
0.6	5	2.86	4.06	3.65	3.00	3.85	3.41	3.00	3.75	3.75
0.1	∞	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	∞	0.14	0.25	0.25	0.14	0.25	0.25	0.14	0.27	0.27
0.3	∞	0.27	0.45	0.45	0.27	0.43	0.43	0.27	0.49	0.49
0.4	∞	0.51	0.76	0.76	0.50	0.67	0.67	0.50	0.83	0.83
0.5	∞	1.02	1.34	1.34	1.00	1.00	1.00	1.00	1.50	1.50
0.6	∞	3.06	3.30	3.30	3.00	3.00	3.00	3.00	3.75	3.75

Table 4: Results corresponding to the fluid and quasi-stationary regimes (scenario II).

We consider the same parameter values as in the previous section (without considering fluid and quasi-stationary regimes), and calculate the mean number of users of each class under static and flow-level load balancing, so that we can make a comparison with packet-level load balancing. As before, we first assume that load balancing is based on balanced fairness, and subsequently on alpha-fair sharing.

5.1 Balanced fairness

When static or flow-level load balancing is used, that is based on balanced fairness, we now need to keep track of the number of class-0 users at node i, $i = 1, 2$. Let n_{0i} denote the number of class-i users at node i, $i = 1, 2$. Then the balance function is given by (see [1])

$$\Phi(n) = \frac{\binom{n_{01}+n_1}{n_1}\binom{n_{02}+n_2}{n_2}}{C_1^{n_1+n_{01}} C_2^{n_2+n_{02}}},$$

and we obtain

$$\phi_{0i}(n) = \frac{n_{0i}}{n_{0i}+n_i} C_i; \quad \phi_i(n) = \frac{n_i}{n_{0i}+n_i} C_i, \quad i = 1, 2.$$

Hence, at both nodes capacity is shared according to egalitarian Processor Sharing (PS).

Let us first consider static load balancing. Clearly, considering the symmetric parameter setting of the previous section, the optimal static policy is to route class-0 arrivals to node i, $i = 1, 2$, with probability $\frac{1}{2}$. Using the parameter values of the previous section, we thus find

γ	$\mathbb{E}N_0^{BFst}$	$\mathbb{E}N_1^{BFst}$	$\mathbb{E}N_2^{BFst}$	$\mathbb{E}N_0^{BFfl}$	$\mathbb{E}N_1^{BFfl}$	$\mathbb{E}N_2^{BFfl}$	$\mathbb{E}N_0^{BF}$	$\mathbb{E}N_1^{BF}$	$\mathbb{E}N_2^{BF}$
0.1	0.12	0.12	0.12	0.11	0.12	0.12	0.06	0.11	0.11
0.2	0.29	0.29	0.29	0.25	0.27	0.27	0.14	0.27	0.27
0.3	0.55	0.55	0.55	0.46	0.50	0.50	0.27	0.49	0.49
0.4	1.00	1.00	1.00	0.82	0.87	0.87	0.50	0.83	0.83
0.5	2.00	2.00	2.00	1.59	1.64	1.64	1.00	1.50	1.50
0.6	6.00	6.00	6.00	5.15	5.27	5.27	3.00	3.75	3.75

Table 5: Results for static, flow-level and packet-level load balancing in case of balanced fairness.

that class-i (class-0) users arrive according to a Poisson process of rate γ ($\frac{1}{2}\gamma$) at node i, and both class-0 and class-i users have exponentially distributed service requirements with mean 1, $i = 1, 2$. Recalling that $C_i = 1$, $i = 1, 2$, and since capacity is shared according to PS at both nodes, it is a straightforward exercise to show that

$$\mathbb{E}N_i^{BFst} := \frac{\gamma}{1 - \frac{3}{2}\gamma}, \quad i = 0, 1, 2,$$

where $\mathbb{E}N_0^{BFst}$ denotes the mean number of class-0 users in the network (at node 1 or node 2). In Table 5 we report the $\mathbb{E}N_i^{BFst}$s for different values of the load γ.

Using the closed-form expressions for $\mathbb{E}N_i^{BFst}$ and $\mathbb{E}N_i^{BF}$, $i = 0, 1, 2$, it is straightforward to derive that

$$\frac{\mathbb{E}N_0^{BFst}}{\mathbb{E}N_0^{BF}} = 2; \quad \frac{\mathbb{E}N_i^{BFst}}{\mathbb{E}N_i^{BF}} = \frac{4 - 4\gamma}{4 - 5\gamma} \geq 1, \quad i = 1, 2,$$

given that the load γ of each class is smaller than $\frac{2}{3}$.

In case of flow-level load balancing it is optimal (under the current setting) to route class-0 users to node 1 if $n_{01} + n_1 < n_{02} + n_2$, and to node 2 if $n_{01} + n_1 > n_{02} + n_2$. If $n_{01} + n_1 = n_{02} + n_2$ then an arriving class-0 user is sent to node i with probability $\frac{1}{2}$, $i = 1, 2$. In other words, an arriving class-0 user should join the shortest queue, see [15]. Since no explicit expressions are known for the mean number of users $\mathbb{E}N_i^{BFfl}$ of class i, $i = 0, 1, 2$, under flow-level load balancing, we have performed simulation experiments to obtain these values. The results are also reported in Table 5.

Table 5 shows that packet-level load balancing outperforms both static and flow-level load balancing, and flow-level load balancing is better than static load balancing, as was expected, i.e., $\mathbb{E}N_i^{BF} \leq \mathbb{E}N_i^{BFfl} \leq \mathbb{E}N_i^{BFst}$, $i = 0, 1, 2$. For low values of γ (low loads), the results are quite similar, but for higher loads the differences become more significant. We note that these results are in line with the findings of [11].

5.2 Alpha-fair sharing

In case static or flow-level load balancing is executed through alpha-fair sharing, we also need to be aware of the number of class-0 users at nodes 1 and 2. In case n_i class-i users and n_{0i} class-0 users are present at node i, the allocated service rates are

$$s_i^*(n) = \frac{\kappa_i^{1/\alpha} n_i C_i}{\kappa_0^{1/\alpha} n_{0i} + \kappa_i^{1/\alpha} n_i}, \quad s_{0i}^*(n) = \frac{\kappa_0^{1/\alpha} n_{0i} C_i}{\kappa_0^{1/\alpha} n_{0i} + \kappa_i^{1/\alpha} n_i}, \quad i = 1, 2.$$

γ	α	$\mathbb{E}N_0^{AFst}$	$\mathbb{E}N_1^{AFst}$	$\mathbb{E}N_2^{AFst}$	$\mathbb{E}N_0^{AFfl}$	$\mathbb{E}N_1^{AFfl}$	$\mathbb{E}N_2^{AFfl}$	$\mathbb{E}N_0^{AF}$	$\mathbb{E}N_1^{AF}$	$\mathbb{E}N_2^{AF}$
0.1	1	0.11	0.12	0.12	0.10	0.12	0.12	0.06	0.12	0.12
0.2	1	0.25	0.31	0.30	0.23	0.28	0.28	0.13	0.28	0.27
0.3	1	0.44	0.61	0.59	0.40	0.54	0.53	0.23	0.54	0.49
0.4	1	0.71	1.17	1.12	0.64	0.98	0.94	0.39	0.97	0.83
0.5	1	1.21	2.47	2.32	1.09	1.97	1.85	0.68	1.95	1.46
0.6	1	2.90	7.85	7.26	2.81	6.68	6.21	1.55	5.93	3.47
0.1	2	0.11	0.12	0.12	0.11	0.12	0.12	0.06	0.12	0.11
0.2	2	0.27	0.30	0.29	0.24	0.27	0.27	0.14	0.27	0.26
0.3	2	0.48	0.58	0.57	0.43	0.53	0.51	0.26	0.50	0.48
0.4	2	0.83	1.10	1.06	0.71	0.94	0.91	0.47	0.88	0.81
0.5	2	1.54	2.28	2.17	1.30	1.83	1.78	0.87	1.71	1.44
0.6	2	4.17	7.13	6.69	4.03	6.43	6.09	2.35	4.81	3.66
0.1	5	0.12	0.12	0.12	0.11	0.12	0.11	0.06	0.11	0.11
0.2	5	0.28	0.29	0.29	0.25	0.27	0.27	0.15	0.26	0.26
0.3	5	0.52	0.56	0.56	0.44	0.51	0.50	0.28	0.48	0.46
0.4	5	0.93	1.04	1.03	0.78	0.92	0.90	0.52	0.82	0.78
0.5	5	1.80	2.12	2.07	1.51	1.77	1.73	1.00	1.51	1.40
0.6	5	5.21	6.50	6.29	4.46	5.20	5.07	2.84	3.95	3.61
0.1	∞	0.12	0.12	0.12	0.11	0.12	0.12	0.06	0.11	0.11
0.2	∞	0.29	0.29	0.29	0.25	0.27	0.27	0.15	0.26	0.26
0.3	∞	0.55	0.55	0.55	0.46	0.50	0.50	0.30	0.46	0.46
0.4	∞	1.00	1.00	1.00	0.82	0.87	0.87	0.55	0.77	0.77
0.5	∞	2.00	2.00	2.00	1.59	1.64	1.64	1.10	1.39	1.39
0.6	∞	6.00	6.00	6.00	5.15	5.27	5.27	3.17	3.48	3.48

Table 6: Results for static, flow-level and packet-level load balancing in case of alpha-fair sharing (scenario II).

Hence, capacity is shared according to DPS with relative weights $\kappa_0^{1/\alpha}$ and $\kappa_i^{1/\alpha}$ at node i, $i = 1, 2$.

Again, due to symmetric parameter values, in case of static load balancing it is optimal to route class-0 arrivals to node i, $i = 1, 2$, with probability $\frac{1}{2}$. Using the parameter values of the previous section, we thus find that class-i (class-0) users arrive according to a Poisson process of rate γ ($\frac{1}{2}\gamma$) at node i, and both class-0 and class-i users have exponentially distributed service requirements with mean 1, $i = 1, 2$. Using that $C_i = 1$, $i = 1, 2$, and since capacity is shared according to DPS at both nodes, the results of [7] imply that

$$\mathbb{E}N_0^{AFst} := \frac{\frac{1}{2}\gamma}{1-\frac{3}{2}\gamma}\left(2 + \frac{\gamma\left(\kappa_1^{1/\alpha} - \kappa_0^{1/\alpha}\right)}{\kappa_0^{1/\alpha}(1-\frac{1}{2}\gamma) + \kappa_1^{1/\alpha}(1-\gamma)} + \frac{\gamma\left(\kappa_2^{1/\alpha} - \kappa_0^{1/\alpha}\right)}{\kappa_0^{1/\alpha}(1-\frac{1}{2}\gamma) + \kappa_2^{1/\alpha}(1-\gamma)}\right);$$

$$\mathbb{E}N_i^{AFst} := \frac{\gamma}{1-\frac{3}{2}\gamma}\left(1 + \frac{\frac{1}{2}\gamma\left(\kappa_0^{1/\alpha} - \kappa_i^{1/\alpha}\right)}{\kappa_0^{1/\alpha}(1-\frac{1}{2}\gamma) + \kappa_i^{1/\alpha}(1-\gamma)}\right), \quad i = 1, 2.$$

Note that $\mathbb{E}N_i^{AFst} = \mathbb{E}N_i^{BFst}$, $i = 0, 1, 2$, in case of equal class weights. Therefore, we only focus on scenario II, and these results are shown in Table 6.

The optimal flow-level load balancing policy is as before to join the shortest queue, see [15]. As no explicit expressions for the mean number of users $\mathbb{E}N_i^{AFfl}$ of class i, $i = 0, 1, 2$, are available under flow-level load balancing, we resort to simulation experiments to obtain these values. Note that $\mathbb{E}N_i^{AFfl} = \mathbb{E}N_i^{BFfl}$, $i = 0, 1, 2$, in case of equal class weights, so we only report the results corresponding to scenario II, see Table 6.

Tables 6 shows that packet-level load balancing performs better than both static and flow-level load balancing: $\mathbb{E}N_i^{AF} \leq \mathbb{E}N_i^{AFfl} \leq \mathbb{E}N_i^{AFst}$, $i = 0, 1, 2$. Again, the results seem to vary more in case of high values of γ.

6 Conclusion

We analyzed a network consisting of L nodes, with $L + 1$ classes of users. Class-i users require service at node i only, $i = 1, \ldots, L$, whereas class-0 users can split their traffic over the L nodes. We considered load balancing at the packet-level, implying that class-0 users can split their traffic over the L nodes at the same time. We assumed that load balancing was based on balanced fairness and an alpha-fair bandwidth sharing policy, respectively. We characterized how bandwidth is allocated in each state of the network under these two policies. Assuming Poisson arrivals and exponentially distributed service requirements, we derived expressions (approximations) for the mean number of users of each class under these two policies. For both policies we also showed that one can achieve significant performance gains if one performs packet-level load balancing instead of static or flow-level load balancing, especially for highly loaded systems.

A topic for further research is extending the results to a more general network, e.g., so-called linear networks where some classes can split their traffic over multiple nodes at the same time. In this case it is considerably harder, if possible at all, to derive expressions for the mean number of users of each class under the above-mentioned policies, as the network does not reduce to a tree network.

Acknowledgments

The author wishes to thank Sem Borst and Michel Mandjes for valuable discussions.

References

[1] T. Bonald, M. Jonckheere, A. Proutière (2004). Insensitive load balancing. In: *Proceedings of the ACM SIGMETRICS/Performance 2004 Conference*, New York, USA, 367-377.

[2] T. Bonald, L. Massoulié (2001). Impact of fairness on Internet performance. In: *Proceedings of the ACM SIGMETRICS/Performance 2001 Conference*, Boston, USA, 82-91.

[3] T. Bonald, L. Massoulié, A. Proutière, J. Virtamo (2006). A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing Systems and Applications*, 53: 65-84.

[4] T. Bonald, A. Proutière (2003). Insensitive bandwidth sharing in data networks. *Queueing Systems*, 44: 69-100.

[5] T. Bonald, A. Proutière, J.W. Roberts, J. Virtamo (2003). Computational aspects of balanced fairness. In: *Proceedings of the 18th International Teletraffic Congress*, Berlin, Germany, 801-810.

[6] T. Bonald, J. Virtamo (2004). Calculating the flow level performance of balanced fairness in tree networks *Performance Evaluation*, 58: 1-14.

[7] G. Fayolle, I. Mitrani, R. Iasnogorodski (1980). Sharing a processor among many job classes. *Journal of the ACM*, 27: 519-532.

[8] M. Jonckheere, J. Virtamo (2005). Optimal insensitive routing and bandwidth sharing in simple data networks. In: *Proceedings of the ACM SIGMETRICS 2005 Conference*, Banff, Canada, 193-204.

[9] G. van Kessel, R. Nunez-Queija, S. Borst (2005). Differentiated bandwidth sharing with disparate flow sizes. In: *Proceedings of the IEEE INFOCOM 2005 Conference*, Miami, USA, 2425-2435.

[10] J. Leino, J. Virtamo (2005). Insensitive traffic splitting in data networks. In: *Proceedings of the 19th International Teletraffic Congress*, 1355-1364.

[11] J. Leino, J. Virtamo (2006). Insensitive load balancing in data networks. *Computer Networks*, 50: 1059-1068.

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

[13] J. Mo, J. Walrand (2000). Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8: 556-567.

[14] J. Padhye, V. Firoiu, D. Towsley, J. Kurose (2000). Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8: 133-145.

[15] D. Towsley, P.D. Sparaggis, C.G. Cassandras (1992). Optimal routing and buffer allocation for a class of finite capacity queueing systems. *IEEE Transactions on Automatic Control*, 37: 1446-1451.

[16] W. Whitt (1986). Deciding which queue to join: some counterexamples. *Operations Research*, 34: 226-244.