Queueing models for bandwidth-sharing disciplines
Lieshout, P.M.D.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam,Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Download date: 07 Apr 2019
Chapter 8

Flow-level performance
of traffic-splitting networks

In the previous two chapters we assumed that each class of users corresponded to a unique route in the network. In this chapter we consider the case that some class of users has multiple alternative paths through the network.

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 [173].

In this chapter we analyze load balancing in data networks carrying elastic traffic, as considered by [135]. 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 chapter deals with packet-level load balancing, i.e., we assume that packets of a flow can be divided among several routes.

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. This particular network is useful for analyzing the performance and potential gains of load balancing at the packet-level. In addition, this system allows for rather explicit results.

We assume that packet-level load balancing is based on an AFS policy. Under this policy, the above network can be shown to have multiple possible behaviors. In particular, we show that packet-level load balancing based on AFS implies that some classes of users, depending on the state of the network, share capacity according to
some 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 to that of a similar network, where packet-level load balancing is based on BFS, 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.

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 AFS, for which no explicit expressions are available.

The remainder of this chapter is organized as follows. In Section 8.1 we first provide a detailed model description, and introduce BFS and AFS. In the next section we consider the model for a fixed flow population, and characterize how bandwidth is allocated under both policies. In Section 8.3 we consider the model at large time-scales, so that the state of the network varies. We derive explicit expressions for the mean number of users under BFS, and show by conducting extensive simulation experiments that these provide accurate approximations for the ones under AFS. In the next section we examine the performance gain that one can achieve for both policies by using packet-level load balancing instead of static or flow-level load balancing.

8.1 Queueing model

We consider the network as depicted in Figure 8.1. The network consists of $L$ nodes, where node $i$ has service rate $c_i$, $i = 1, \ldots, L$. There are $L + 1$ classes of users. Class $i$ requires service at node $i$, $i = 1, \ldots, L$, 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 $\lambda_i$, and have exponentially distributed service requirements with mean $\mu_i^{-1}$, $i = 0, \ldots, L$. The arrival processes are all independent. The traffic load of class $i$ is then $\rho_i = \lambda_i \mu_i^{-1}$. Let $n = (n_0, \ldots, n_L)$ denote the state of the network, with $n_i$ representing the number of class-$i$ users.

8.1.1 BFS

We first assume that the bandwidth is shared according to BFS, see Chapter 1. Let $\phi_i(n)$ denote the service rate allocated to class $i$, $i = 0, \ldots, L$, under BFS, when the network is in state $n$ (here $\phi_0(n) = \sum_{i=1}^{L} \phi_{bi}(n)$). These service rates have to satisfy
the balance conditions

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

(8.1)

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

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

(8.2)

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 \( n \in \mathbb{N}_0^{L+1} \setminus \{\vec{0}\} \):

\[
\begin{align*}
\text{max} & \quad \Phi(n)^{-1} \\
\text{subject to} & \quad \sum_{j=1}^{L} \phi_{0j}(n) = \Phi(n - e_0) \Phi(n) \\
\phi_i(n) & = \frac{\Phi(n - e_i)}{\Phi(n)}, \ i = 1, \ldots, L \\
\phi_{0i}(n) + \phi_i(n) & \leq c_i, \ i = 1, \ldots, L \\
\text{over} & \quad \phi_{0i}(n), \phi_i(n) \geq 0, \ i = 1, \ldots, L.
\end{align*}
\]

(8.3)

It is clear that \( \Phi(n) \) can be obtained recursively: \( \Phi(n - e_i) \) is required to determine \( \Phi(n), i = 0, \ldots, L \). Also note that (8.3) is a simple linear programming (LP) problem, which can be solved using standard LP algorithms. In Section 8.2.1, however, we solve (8.3) by rewriting the LP problem in terms of a related network.

### 8.1.2 AFS

We next assume that the network operates under an AFS policy, as introduced in [140]. When the network is in state \( n \in \mathbb{N}_0^{L+1} \setminus \{\vec{0}\} \), the service rate \( x_i^n \) allocated to each of
the class- \( i \) users is obtained by solving the following optimization problem:

\[
\text{max } F(x) \\
\text{subject to } n_0 x_0 + n_i x_i \leq c_i, \quad i = 1, \ldots, L \\
\text{over } x_0, x_i \geq 0, \quad i = 1, \ldots, L,
\]

where the objective function \( F(x) \) is defined by

\[
F(x) := \begin{cases} 
  n_0 \kappa_0 \left( \frac{\prod_{i=1}^{L} x_0}{L} \right)^{1-\alpha} + \sum_{i=1}^{L} n_i \kappa_i \frac{x_i^{1-\alpha}}{1-\alpha}, & \text{if } \alpha \in (0, \infty) \setminus \{1\}; \\
  n_0 \kappa_0 \log \left( \frac{\prod_{i=1}^{L} x_0}{L} \right) + \sum_{i=1}^{L} n_i \kappa_i \log(x_i), & \text{if } \alpha = 1.
\end{cases}
\]

The \( \kappa_i \)s are non-negative class weights, and \( \alpha \in (0, \infty) \) may as before be interpreted as a fairness coefficient. The value of \( x^*_{0i} \) denotes how much capacity is allocated 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 \( s_i(n) := n_i x^*_i \) denote the total service rate allocated to class \( i, i = 0, \ldots, L \).

**8.2 Static setting**

In this section we consider the model for a fixed flow population, i.e., the state \( n \in \mathbb{N}^{L+1}_0 \setminus \{0\} \) is fixed, and we derive how bandwidth is shared between the various classes in case of BFS and AFS, respectively. We first show that the network depicted in Figure 8.1 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 8.2, it follows that the networks depicted in Figures 8.1 and 8.2 are in fact equivalent. The tree has a common link with capacity \( c_1 + \cdots + c_L \), and \( L + 1 \) branches with capacities \( \infty, c_1, \ldots, c_L \), 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 into a tree network. In fact, if we extend the model depicted in Figure 8.1 by adding a class of users that requires service at all \(L\) nodes simultaneously, then it is no longer 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 into some other network (with dummy nodes) without traffic splitting.

### 8.2.1 BFS

In this subsection we derive the BFS allocation by solving the problem (8.3). Since the models depicted in Figures 8.1 and 8.2 are equivalent, it follows that the balance function \(\tilde{\Phi}(\cdot)\) corresponding to the tree network coincides with \(\Phi(\cdot)\), i.e., \(\tilde{\Phi}(\cdot) = \Phi(\cdot)\), see [24]. In the following lemma we present the solution of the problem (8.3).

**Lemma 8.2.1** The BFS function \(\Phi(n)\) satisfies, with \(\Phi(0) = 1\),

\[
\Phi(n) = \max \left\{ \frac{\Phi(n - e_1)}{c_1}, \ldots, \frac{\Phi(n - e_L)}{c_L}, \frac{\sum_{i=0}^{L} \Phi(n - e_i)}{\sum_{i=1}^{L} c_i} \right\}, \quad n \in \mathbb{N}_{L+1}^L \setminus \{0\}. \tag{8.5}
\]

**Proof:** From the above it follows that we can obtain \(\Phi(\cdot)\) by determining \(\tilde{\Phi}(\cdot)\), as \(\Phi(\cdot) = \tilde{\Phi}(\cdot)\). Subsequently, \(\tilde{\Phi}(\cdot)\) is obtained by using Equation (2) in [27]. \(\square\)
We note that Lemma 8.2.1 is in agreement with Equation (19) in [108]. From Lemma 8.2.1 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 8.2.1 and (8.2).

### 8.2.2 AFS

In this subsection we focus on the AFS allocation, which is obtained by solving the problem (8.4). Similar to the previous subsection, we can obtain the AFS allocation \( s(n) \) by determining the AFS 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:

\[
\begin{align*}
\max & \quad H(x) \\
\text{subject to} & \quad \sum_{i=0}^{L} n_i x_i \leq \sum_{i=1}^{L} c_i \\
& \quad n_i x_i \leq c_i, \quad i = 1, \ldots, L \\
& \quad \text{over } x_i \geq 0, \quad i = 0, \ldots, L,
\end{align*}
\]

(8.6)

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

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

Below we show that (8.6) is solvable, but the optimal solution strongly depends on the state \( n \in \mathbb{N}_0^{L+1}\setminus\{0\} \). We present a simple algorithm for obtaining the AFS allocation.

**Lemma 8.2.2** The AFS allocation \( s(n) \) can be obtained with the following algorithm:

Set Stop:=False
Set \( S := \{0, \ldots, L\} \)
\[\text{WHILE Stop=False DO}\]
\[\text{Determine the } |S|\text{-class DPS rates: } s_i(n) := \frac{n_i \kappa_i^{1/\alpha} \sum_{j \in S \setminus \{n\}} c_j}{\sum_{j \in S} n_j \kappa_j^{1/\alpha}}, \quad i \in S\]
\[\text{IF } s_i(n) \leq c_i \text{ for all } i \in S\setminus\{0\} \text{ THEN set Stop:=True}\]
\[\text{ELSE}\]
\[\text{Take any } i^* \in S\setminus\{0\} \text{ such that } s_{i^*}(n) > c_i\]
\[\text{Set } S := S\setminus\{i^*\}\]
\[\text{Set } s_{i^*}(n) := c_i\]
\[\text{END}\]
\[\text{END}\]
8.3 Flow-level dynamics

Proof: First consider the Karush-Kuhn-Tucker (KKT) [104] necessary conditions for the problem (8.6). If \( x \) is an optimal solution to the problem (8.6), then there exist constants \( p_i \geq 0, i = 0, \ldots, L \), such that,

\[
\frac{n_0 x_0}{x_0} - n_0 p_0; \tag{8.7}
\]

\[
\frac{n_i x_i}{x_i} - n_i (p_0 + p_i), \quad i = 1, \ldots, L; \tag{8.8}
\]

\[
p_0 \left( \sum_{i=1}^{L} c_i - \sum_{i=0}^{L} n_i x_i \right) = 0; \tag{8.9}
\]

\[
p_i (c_i - n_i x_i) = 0, \quad i = 1, \ldots, L. \tag{8.10}
\]

Note that (8.7) and (8.8) hold for any \( \alpha \in (0, \infty) \). Solving (8.7)-(8.10) 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 (8.6). For each of the other solutions there exists at least one Lagrange multiplier that is negative, implying that these solutions cannot be optimal. Note that the existence of a unique optimal solution \( x^* \) for (8.6) also follows as \( H(x) \) is strictly concave and the constraints are linear.

Straightforward calculus shows that the corresponding AFS allocation \( s_i(n) = s_i(n) = n_i x^*_i, i = 0, \ldots, L \), can be obtained by the above algorithm. The algorithm reflects that \( 2^L - 1 \) solutions exist for (8.7)-(8.10), but it also shows that only one of these solutions, \( x^* \), is found after termination of the algorithm. The Lagrange multipliers corresponding to \( x^* \) are such that \( p_i = 0 \) if \( i \in S \setminus \{0\} \), and \( p_i > 0 \) if \( i \notin S \setminus \{0\} \), 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 \( n_i = 0, i = 1, \ldots, L \), otherwise \( p_0 > 0 \). \( \square \)

8.3 Flow-level dynamics

In the previous section we considered the model for a fixed flow population, and we derived expressions for the BFS and AFS 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.

8.3.1 BFS

Let \( N(t) = (N_0(t), \ldots, N_L(t)) \) 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:

\[
q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \mu_i \phi_i(n), \quad i = 0, \ldots, L,
\]
in case of BFS. In [24] it was shown that the process \( N(t) \) is stable if there exists \((\tilde{\rho}_0, \ldots, \tilde{\rho}_L)\) such that
\[
\sum_{i=1}^{L} \tilde{\rho}_i = \rho_0 \quad \text{and} \quad \tilde{\rho}_0 + \rho_i < c_i, \quad i = 1, \ldots, L,
\]
or equivalently, if
\[
\sum_{i=0}^{L} \rho_i < \sum_{j=1}^{L} c_j \quad \text{and} \quad \rho_i < c_i, \quad i = 1, \ldots, L. \tag{8.11}
\]
It may be verified from (8.1) that the steady-state queue length distribution is given by
\[
\pi(n) = \frac{1}{G(\rho)} \Phi(n) \prod_{i=0}^{L} \rho_i^{n_i}, \quad n \in \mathbb{N}_0^{L+1}, \tag{8.12}
\]
where the normalization constant \( G(\rho) \) equals
\[
G(\rho) = G(\rho_0, \ldots, \rho_L) = \sum_{n_0=0}^{\infty} \cdots \sum_{n_L=0}^{\infty} \Phi(n) \prod_{i=0}^{L} \rho_i^{n_i}.
\]
As a side remark we mention that (8.12) in fact holds for much more general traffic characteristics, see [25] for a more detailed treatment.

When applying Little’s formula we find that
\[
\mathbb{E} N_i^{BF} = \rho_i \frac{\partial G(\rho)}{\partial \rho_i} = \rho_i \frac{\partial \log G(\rho)}{\partial \rho_i}, \quad i = 0, \ldots, L, \tag{8.13}
\]
i.e., characterization of \( G(\rho) \) implies that \( \mathbb{E} N_i^{BF}, \quad i = 0, \ldots, L, \) is known as well.

By exploiting the results of [29] on tree networks we can determine \( G(\rho) \), and it can be verified that this results in
\[
G(\rho) = \frac{1}{1 - \sum_{i=1}^{L} \rho_i \prod_{j=1}^{L} (1 - \frac{\rho_j}{c_j})}.
\]
Then by using (8.13) we can obtain a closed-form expression for \( \mathbb{E} N_i^{BF}, \quad i = 0, \ldots, L, \)
The expression for \( \mathbb{E} N_i^{BF}, \quad i = 1, \ldots, L, \) is in general quite complicated, in contrast to the expression for the mean number of class-0 users, which is given by
\[
\mathbb{E} N_0^{BF} = \rho_0 \frac{\rho_0}{\sum_{i=1}^{L} c_i - \sum_{i=0}^{L} \rho_i}.
\]
From (8.14) it follows that \( \mathbb{E} N_i^{BF}, \quad i = 0, \ldots, L, \) is finite if the stability condition (8.11) holds.
8.3 Flow-level dynamics

8.3.2 AFS

As before, let \( N(t) = (N_0(t), \ldots, N_L(t)) \) denote the state of the network at time \( t \). In case of AFS \( N(t) \) 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 8.2, it follows from Theorem 1 in [23] that the process \( N(t) \) is stable if (8.11) holds.

Lemma 8.2.2 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 AFS. To gain some insight, we derive in this section approximations for the mean number of users of each class, i.e., \( EN_i^{AF} \), \( i = 0, \ldots, L \). 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 8.2.2 in Section 8.2.2, it follows that the network, depending on the state \( n \), has three possible behaviors: (i) if

\[
n_1 > \frac{c_1}{c_2} \left( \frac{\kappa_2}{\kappa_1} \right)^{1/\alpha} n_2 + \left( \frac{\kappa_0}{\kappa_1} \right)^{1/\alpha} n_0,
\]

then classes 0 and 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 \( \lambda_1 \) 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 \( \lambda_2 \) 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 [63] we would obtain

\[
EN_0^{(i)} = \frac{\rho_0}{c_2 - \rho_0 - \rho_2} \left( 1 + \frac{\rho_0 \rho_2 \left( \kappa_2^{1/\alpha} - \kappa_0^{1/\alpha} \right)}{\kappa_0^{1/\alpha} \mu_0 (c_2 - \rho_0) + \kappa_2^{1/\alpha} \mu_2 (c_2 - \rho_2)} \right);
\]

\[
EN_1^{(i)} = \frac{\rho_1}{c_1 - \rho_1};
\]
It can be verified that

\[ \mathbb{E}N_2^{(i)} = \frac{\rho_2}{c_2 - \rho_0 - \rho_2} \left( 1 + \frac{\mu_2 \rho_0 \left( \kappa_0^{1/\alpha} - \kappa_2^{1/\alpha} \right)}{\kappa_0^{1/\alpha} \mu_0 (c_2 - \rho_0) + \kappa_2^{1/\alpha} \mu_2 (c_2 - \rho_2)} \right). \]

Likewise, when the network behaves as \((i)\) 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 \(\mathbb{E}N_i^{(iii)}\), \(i = 0, 1, 2\):

\[
(c_1 + c_2)\mathbb{E}N_i^{(iii)} - \lambda \sum_{j=0}^2 \kappa_j^{1/\alpha} \frac{\lambda \mathbb{E}N_i^{(iii)} + \lambda \mathbb{E}N_j^{(iii)}}{\kappa_j^{1/\alpha} \mu_j + \kappa_i^{1/\alpha} \mu_i} = \rho_i, \quad i = 0, 1, 2,
\]

where \(\lambda := \lambda_0 + \lambda_1 + \lambda_2\), see [63]. 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

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

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, \(\mathbb{E}N_i^{AP}\), \(i = 0, 1, 2\), is bounded if (8.11) holds, i.e., if the process \(N(t)\) is stable.

In [24] 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 BFS. Therefore, we also propose the following approximation: \(\mathbb{E}N_i^{BF} \approx \mathbb{E}N_i^{AF}\), \(i = 0, 1, 2\).

The value of \(\mathbb{E}N_i^{BF}\), \(i = 0, 1, 2\), can be obtained using (8.13), and is independent of the value of \(\alpha\).

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 \(\gamma\) and the AFS coefficient \(\alpha\) vary,
8.3 Flow-level dynamics

Table 8.1: Simulation results for scenario I.

<table>
<thead>
<tr>
<th>α</th>
<th>EN\textsuperscript{AF}\textsubscript{0}</th>
<th>EN\textsuperscript{AF}\textsubscript{1}</th>
<th>EN\textsuperscript{AF}\textsubscript{2}</th>
<th>EN\textsuperscript{AF}\textsubscript{0}</th>
<th>EN\textsuperscript{AP}\textsubscript{1}</th>
<th>EN\textsuperscript{AP}\textsubscript{2}</th>
<th>EN\textsuperscript{BF}\textsubscript{0}</th>
<th>EN\textsuperscript{BF}\textsubscript{1}</th>
<th>EN\textsuperscript{BF}\textsubscript{2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.06</td>
<td>0.11</td>
<td>0.11</td>
<td>0.06</td>
<td>0.11</td>
<td>0.11</td>
<td>0.06</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>0.2</td>
<td>0.15</td>
<td>0.26</td>
<td>0.26</td>
<td>0.14</td>
<td>0.25</td>
<td>0.25</td>
<td>0.14</td>
<td>0.27</td>
<td>0.27</td>
</tr>
<tr>
<td>0.3</td>
<td>0.30</td>
<td>0.46</td>
<td>0.46</td>
<td>0.27</td>
<td>0.43</td>
<td>0.43</td>
<td>0.27</td>
<td>0.49</td>
<td>0.49</td>
</tr>
<tr>
<td>0.4</td>
<td>0.55</td>
<td>0.77</td>
<td>0.77</td>
<td>0.50</td>
<td>0.67</td>
<td>0.67</td>
<td>0.50</td>
<td>0.83</td>
<td>0.83</td>
</tr>
<tr>
<td>0.5</td>
<td>1.10</td>
<td>1.39</td>
<td>1.39</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.10</td>
<td>1.50</td>
<td>1.50</td>
</tr>
<tr>
<td>0.6</td>
<td>3.17</td>
<td>3.48</td>
<td>3.48</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
<td>3.80</td>
<td>3.75</td>
<td>3.75</td>
</tr>
</tbody>
</table>

Table 8.2: Simulation results for scenario II.

<table>
<thead>
<tr>
<th>α</th>
<th>EN\textsuperscript{AF}\textsubscript{0}</th>
<th>EN\textsuperscript{AF}\textsubscript{1}</th>
<th>EN\textsuperscript{AF}\textsubscript{2}</th>
<th>EN\textsuperscript{AF}\textsubscript{0}</th>
<th>EN\textsuperscript{AP}\textsubscript{1}</th>
<th>EN\textsuperscript{AP}\textsubscript{2}</th>
<th>EN\textsuperscript{BF}\textsubscript{0}</th>
<th>EN\textsuperscript{BF}\textsubscript{1}</th>
<th>EN\textsuperscript{BF}\textsubscript{2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.06</td>
<td>0.11</td>
<td>0.11</td>
<td>0.06</td>
<td>0.11</td>
<td>0.11</td>
<td>0.06</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>0.2</td>
<td>0.13</td>
<td>0.28</td>
<td>0.27</td>
<td>0.12</td>
<td>0.25</td>
<td>0.25</td>
<td>0.14</td>
<td>0.27</td>
<td>0.27</td>
</tr>
<tr>
<td>0.3</td>
<td>0.23</td>
<td>0.54</td>
<td>0.49</td>
<td>0.22</td>
<td>0.43</td>
<td>0.43</td>
<td>0.27</td>
<td>0.49</td>
<td>0.49</td>
</tr>
<tr>
<td>0.4</td>
<td>0.39</td>
<td>0.97</td>
<td>0.83</td>
<td>0.35</td>
<td>0.67</td>
<td>0.67</td>
<td>0.50</td>
<td>0.83</td>
<td>0.83</td>
</tr>
<tr>
<td>0.5</td>
<td>1.68</td>
<td>1.95</td>
<td>1.46</td>
<td>0.79</td>
<td>1.43</td>
<td>1.00</td>
<td>1.00</td>
<td>1.50</td>
<td>1.50</td>
</tr>
<tr>
<td>0.6</td>
<td>1.55</td>
<td>5.93</td>
<td>3.47</td>
<td>1.38</td>
<td>4.82</td>
<td>2.80</td>
<td>3.80</td>
<td>3.75</td>
<td>3.75</td>
</tr>
<tr>
<td>0.1</td>
<td>0.06</td>
<td>0.11</td>
<td>0.11</td>
<td>0.06</td>
<td>0.11</td>
<td>0.11</td>
<td>0.06</td>
<td>0.11</td>
<td>0.11</td>
</tr>
<tr>
<td>0.2</td>
<td>0.14</td>
<td>0.27</td>
<td>0.26</td>
<td>0.13</td>
<td>0.25</td>
<td>0.25</td>
<td>0.14</td>
<td>0.27</td>
<td>0.27</td>
</tr>
<tr>
<td>0.3</td>
<td>0.26</td>
<td>0.50</td>
<td>0.48</td>
<td>0.24</td>
<td>0.43</td>
<td>0.43</td>
<td>0.27</td>
<td>0.49</td>
<td>0.49</td>
</tr>
<tr>
<td>0.4</td>
<td>0.47</td>
<td>0.88</td>
<td>0.81</td>
<td>0.42</td>
<td>0.67</td>
<td>0.67</td>
<td>0.50</td>
<td>0.83</td>
<td>0.83</td>
</tr>
<tr>
<td>0.5</td>
<td>0.87</td>
<td>1.71</td>
<td>1.44</td>
<td>0.77</td>
<td>1.31</td>
<td>1.00</td>
<td>1.00</td>
<td>1.50</td>
<td>1.50</td>
</tr>
<tr>
<td>0.6</td>
<td>2.35</td>
<td>4.81</td>
<td>3.66</td>
<td>2.06</td>
<td>3.55</td>
<td>3.08</td>
<td>3.08</td>
<td>3.75</td>
<td>3.75</td>
</tr>
</tbody>
</table>

whereas in scenario I we only let γ vary, as it can be verified that EN\textsuperscript{AF}\textsubscript{i}, i = 0, 1, 2, are independent of the value of α in scenario I. To ensure stability we assume that γ < 2/3. The results are reported in Tables 8.1 and 8.2. Each reported simulation value in these (and other) tables is measured over 4 · 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 chapter, 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 8.1, 8.2, 8.5 and 8.6, and less than 10% for all listed cases in Tables 8.3 and 8.4.

Table 8.1 compares the value of EN\textsuperscript{AF}\textsubscript{i} obtained by simulation with the approximations EN\textsuperscript{AP}\textsubscript{i} and EN\textsuperscript{BF}\textsubscript{i}, i = 0, 1, 2, are independent of the value of α in scenario I. The results show that EN\textsuperscript{AF}\textsubscript{i} ≥ EN\textsuperscript{AP}\textsubscript{i}, i = 0, 1, 2. Also, the table shows that EN\textsuperscript{AF}\textsubscript{0} ≥ EN\textsuperscript{BF}\textsubscript{0} and EN\textsuperscript{AF}\textsubscript{i} ≤ EN\textsuperscript{BF}\textsubscript{i}, i = 1, 2. Overall we see that both approximations are accurate in case of equal class weights, especially for low traffic load.

Table 8.2 reports the results corresponding to scenario II, i.e., in case of unequal class weights. In this case EN\textsuperscript{AF}\textsubscript{i} and EN\textsuperscript{AP}\textsubscript{i} do depend on the value of α, as is shown.
in the table. Again, we see that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AF}$, $i = 0, 1, 2$. For low traffic loads both approximations perform quite well, but for high traffic loads we see that the BFS approximation is less accurate than the other one.

Tables 8.1 and 8.2 show that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AF}$, $i = 0, 1, 2$, 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 AFS, whereas rate $c_1$ is continuously available to class 1 in (i). Clearly, this implies that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{(i)}$. With similar reasoning, we find that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{(ii)}$. Since class-$i$ users cannot be allocated more than $c_i$, $i = 1, 2$, under AFS, 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 $n \in \mathbb{N}_0^3 \setminus \{0\}$ it can be verified that the AFS allocation to class 0 is larger or equal than the one obtained in the three-class DPS model, so one would expect $\mathbb{E}N_0^{AF} \geq \mathbb{E}N_0^{(ii)}$ at first sight. However, recall that we argued that the number of users of classes 1 and 2 in the model operating under AFS 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 AFS is less than or equal to that in the three-class DPS model, i.e., we may also expect $\mathbb{E}N_0^{AF} \geq \mathbb{E}N_0^{(ii)}$. The above reasoning indeed suggests that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AF}$, $i = 0, 1, 2$.

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

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

**Proof:** In [94] 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. \qed
8.3 Flow-level dynamics

Let us return to the setting with $L = 2$ nodes and $L + 1 = 3$ classes of users. Proposition 8.3.1 allows us to obtain simple closed-form expressions for $\rho_i$, $i = 0, 1, 2$, when $r \to \infty$. Assuming that higher indexed classes operate on faster time scales and that the stability condition (8.11) holds, we find that

$$E\rho_0^{AP(\infty)} := \frac{\rho_0}{\bar{c}_1 + \bar{c}_2 - \rho_0};$$

$$E\rho_1^{AP(\infty)} := \max \left\{ \frac{\rho_1}{\bar{c}_1}, \frac{\rho_1}{\bar{c}_1 + \bar{c}_2} + \frac{\kappa_1^{1/\alpha} \rho_0 \rho_1}{\kappa_1^{1/\alpha} (\bar{c}_1 + \bar{c}_2 - \rho_0)(\bar{c}_1 + \bar{c}_2)} \right\},$$

and $E\rho_2^{AP(\infty)}$ equals

$$\max \left\{ \frac{\rho_2}{\bar{c}_2}, \frac{\rho_2}{\bar{c}_1 + \bar{c}_2} + \frac{\kappa_2^{1/\alpha} \rho_0 \rho_2}{\kappa_2^{1/\alpha} (\bar{c}_1 + \bar{c}_2 - \rho_0)(\bar{c}_1 + \bar{c}_2)} + \frac{\kappa_2^{1/\alpha} \rho_1 \rho_2}{\kappa_2^{1/\alpha} (\bar{c}_1 + \bar{c}_2)(\bar{c}_1 + \bar{c}_2)} \right\},$$

where $\bar{c}_i := c_i - \rho_i$, $i = 1, 2$. In case of equal class weights, $\kappa_i = \kappa$, $i = 0, 1, 2$, it is not hard to see that

$$E\rho_0^{AP(\infty)} = \frac{\rho_0}{\bar{c}_1 + \bar{c}_2 - \rho_0};$$

$$E\rho_1^{AP(\infty)} = \max \left\{ \frac{\rho_1}{\bar{c}_1}, \frac{\rho_1}{\bar{c}_1 + \bar{c}_2 - \rho_0} \right\}; \quad E\rho_2^{AP(\infty)} = \max \left\{ \frac{\rho_2}{\bar{c}_2}, \frac{\rho_2}{\bar{c}_1 + \bar{c}_2 - \rho_0} \right\}.$$

Clearly, $E\rho_i^{AP(\infty)}$, $i = 0, 1, 2$, strongly depends 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 thus assume that higher indexed classes operate on faster time scales.

Tables 8.3 and 8.4 report the results for scenarios I and II, respectively. Recall that $E\rho_i^{AP}$ and $E\rho_i^{AP(\infty)}$, $i = 0, 1, 2$, are independent of the value of $\alpha$ in scenario I, whereas they are sensitive to the value of $\alpha$ in scenario II. The tables show that in the fluid and quasi-stationary regimes the approximations are appropriate as well.
Flow-level performance of traffic-splitting networks

Table 8.4: Results for the fluid and quasi-stationary regimes (scenario II).

8.4 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 the gain that can be achieved by using packet-level load balancing instead of static or flow-level load balancing. 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.

8.4.1 BFS

When static or flow-level load balancing is used, which is based on BFS, we 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-0 users at node $i$, $i = 1, 2$. Then the balance function is given by [22]

$$
\Phi(n) = \frac{(n_{01} + n_1)}{c_1 + n_{01}} \frac{(n_{02} + n_2)}{c_2 + n_{02}},
$$

and we obtain

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

Hence, at both nodes capacity is shared according to egalitarian PS.

Considering the symmetric parameter setting of the previous section, the optimal static load balancing policy is to route class-0 arrivals to node $i$, $i = 1, 2$, with probability $1/2$. Using the parameter values of the previous section, we thus find that
8.4 Comparison with static and flow-level load balancing

class-i (class-0) users arrive according to a Poisson process of rate $\gamma$ ($\gamma/2$) 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 using that capacity is shared according to PS at both nodes, it is a straightforward exercise to show that

$$E N BF_{st}^i := \frac{\gamma}{1 - \frac{3}{2} \gamma}, \quad i = 0, 1, 2.$$ 

In Table 8.5 we report $E N BF_{st}^i$, $i = 0, 1, 2$, for different values of the load $\gamma$.

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 $1/2$, $i = 1, 2$. In other words, an arriving class-0 user should join the shortest queue, see [164]. Since no explicit expressions are known for the mean number of users $E N BF_{fl}^i$ 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 8.5.

Table 8.5 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., $E N BF^i \leq E N BF_{fl}^i \leq E N BF_{st}^i$, $i = 0, 1, 2$. For low values of $\gamma$ (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 [109].

8.4.2 AFS

In case static or flow-level load balancing is executed through AFS, 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

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

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 the 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 $1/2$. 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 \(\gamma (\gamma/2)\) 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 that capacity is shared according to DPS at both nodes, the results of [63] imply that

\[
E_N^{AFst} := \frac{1}{2}\frac{\gamma}{1 - \frac{\gamma}{2}} \left( 2 + \frac{\gamma (\kappa^{1/\alpha} - \kappa_0^{1/\alpha})}{\kappa_0^{1/\alpha} (1 - 2\gamma) + \kappa_1^{1/\alpha} (1 - \gamma)} + \frac{\gamma (\kappa_2^{1/\alpha} - \kappa_0^{1/\alpha})}{\kappa_0^{1/\alpha} (1 - \frac{\gamma}{2}) + \kappa_1^{1/\alpha} (1 - \gamma)} \right)
\]

\[
E_N^{AFfl} := \frac{\gamma}{1 - \frac{\gamma}{2}} \left( 1 + \frac{\frac{1}{2} \gamma (\kappa_0^{1/\alpha} - \kappa_1^{1/\alpha})}{\kappa_0^{1/\alpha} (1 - \frac{\gamma}{2}) + \kappa_1^{1/\alpha} (1 - \gamma)} \right), \quad i =1, 2.
\]

Note that \(E_N^{AFst} = E_N^{BFst}, \quad i = 0, 1, 2\), in case of equal class weights. Therefore, we only focus on scenario II, and these results are shown in Table 8.6.

The optimal flow-level load balancing policy is as before to join the shortest queue, see [164]. As no explicit expressions for the mean number of users \(E_N^{AFfl} \) 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 \(E_N^{AFfl} = E_N^{BFfl}, \quad i =0, 1, 2\), in case of equal class weights, so we only report the results corresponding to scenario II, see Table 8.6.

Table 8.6 shows that packet-level load balancing performs better than both static and flow-level load balancing: \(E_N^{AP} \leq E_N^{AFfl} \leq E_N^{AFst}, \quad i =0, 1, 2\). Again, the results seem to vary more for high values of \(\gamma\).