Computational aspects of balanced fairness

Thomas Bonald, Alexandre Proutière, James Roberts, Jorma Virtamo

Computational aspects of balanced fairness

T. Bonald[a], A. Proutière[a], J. Roberts[a] and J. Virtamo[b]

[a]France Telecom R&D
38-40, rue du Général Leclerc, 92794 Issy-les-Moulineaux Cedex 9, France

[b]Helsinki University of Technology, Networking Laboratory
P.O. Box 3000, 02015 HUT, Finland

Flow level behaviour of data networks depends on the allocation of link capacities between competing flows. It has been recently shown that there exist allocations with the property that the stationary distribution of the number of flows in progress on different routes depends only on the traffic loads on these routes and is insensitive to any detailed traffic characteristics. Balanced fairness refers to the most efficient of such allocations. In this paper we develop a general recursive algorithm for efficiently calculating the corresponding performance metrics like flow throughput. Several examples are worked out using this algorithm including the practically interesting case of concentration tree networks.

1. INTRODUCTION

Most traffic in today's Internet is elastic in that sending rates are adapted depending on prevailing network congestion. Typically, this traffic results from the transfer of documents such as files or web pages using TCP. Each document transfer is referred to here as a flow. Elastic traffic is not time critical at the packet level, i.e., there are no stringent deadlines for the delivery of individual packets. Network performance for elastic traffic is mainly manifested at flow level and can be gauged by measures such as the average per flow data transfer rate or throughput, i.e., the ratio of mean flow size to mean flow duration.

To analyze flow level behaviour in a dynamic setting where flows of a certain size stochastically arrive and are transferred across the network until finished, idealized models are needed. An appropriate abstraction in this context entails entirely disregarding packet level phenomena and considering the flow content as a fluid which is transmitted as a continuous stream through the network. It is assumed in this fluid model that rate changes occur instantaneously and simultaneously on all links on every flow arrival or departure. We assume flows occur in sessions consisting of a succession of flows and separating "think times" and that sessions occur as a Poisson process.

As there are many flows with different routes being transmitted simultaneously, performance depends on how network resources are shared between concurrent flows. Most work has focussed on so-called utility based allocations, where bandwidth is shared so

as to maximise some utility function of the instantaneous flow rates [1]. Examples of such allocations are classical max-min fairness [2] and Kelly's proportional fairness [3]. Crucially, the optimality of utility based allocations is defined for a static composition of flows. If the true random nature of traffic were taken into account, it would be necessary to define utility in terms of the performance of individual finite duration flows. In this case, it is not obvious that max-min or proportional fairness are optimal in any real sense. In random traffic, performance and therefore utility depend in general on the precise statistics of offered traffic and are virtually impossible to evaluate analytically.

An alternative notion of fairness, called "balanced fairness", has been introduced by Bonald and Proutière [4]. When flows share bandwidth with balanced fairness, performance is largely insensitive to detailed traffic characteristics and can be expressed in relatively simple terms. The name derives from the set of detailed balance relations satisfied by the instantaneous rates allocated to individual flows. These relations constitute necessary and sufficient conditions for insensitivity in the underlying stochastic networks [5,6]. The insensitivity is such that the distribution of the number of flows in progress, and consequently the expected throughput, depend only on the average traffic offered on each route. Balanced fairness thus generalizes to a general network context the insensitivity of bandwidth sharing of an isolated bottleneck link introduced in [7].

Balanced fairness can be viewed as a bandwidth sharing objective to be realized by appropriate packet level mechanisms. Alternatively, one might consider the mere existence of an insensitive allocation as evidence that more readily realized allocations do not depend significantly on traffic characteristics beyond expected demand. In other words, the balanced fairness model is a good candidate to approximate the behaviour of elastic traffic in the Internet.

The main contribution of the present paper is to provide an efficient recursive algorithm to compute these metrics for a certain class of topologies, including the practically interesting case of concentration tree networks. This algorithm has been implemented in Mathematica (available from http://netlab.hut.fi/tutkimus/com2/Qlib/) and readily provides numerical and symbolic evaluations. The results derived are notably useful to investigate the accuracy of simpler approximations that are more likely to be used for practical dimensioning purposes.

The rest of this paper is organized as follows. In Section 2 a brief review is given of the notion of balanced fairness and some of its basic properties. The recursive algorithm is developed in Section 3. In Section 4 several examples of the application of the recursion on specific network topologies are provided. Section 5 presents an extension of the algorithm to cases where flows are subject to external rate limits. The results are summarized in Section 6.

2. BALANCED FAIRNESS

The following is a short summary of the notion of balanced fairness and the network model to which it pertains; for a full account readers are referred to [4,8].

The network consists of a set of links $\mathcal{L} = \{1, \ldots, L\}$ where link l has a capacity C_l. A random number of flows compete for the bandwidth of these links. There are N classes of flows where each class i is characterized by a route r_i consisting of a set of links. The

mean volume of information offered by flows in class i per unit time, i.e., the load of class i, is denoted ρ_i. The network state is defined by the vector $x = (x_1, \ldots, x_N)$, where x_i is the number of class-i flows in progress.

The total capacity $\phi_i(x)$ allocated to class-i flows is assumed to be shared equally between these flows and to depend on the network state x only. The capacity allocation must satisfy the capacity constraints,

$$\sum_{i:l\in r_i} \phi_i(x) \le C_l, \qquad \forall\, l \in \mathcal{L}. \tag{1}$$

The allocation is said to be balanced if

$$\frac{\phi_i(x - e_j)}{\phi_i(x)} = \frac{\phi_j(x - e_i)}{\phi_j(x)}, \qquad \forall\, i,j,\ x_i > 0,\ x_j > 0,$$

where e_i is an N-vector with 1 in component i and 0 elsewhere. The balance property implies that there is a balance function $\Phi(x)$ such that

$$\phi_i(x) = \frac{\Phi(x - e_i)}{\Phi(x)}, \qquad \forall\, i,\ x_i > 0.$$

Basically any positive function $\Phi(x)$ defines a balanced allocation. As shown in [4] there is a unique balanced allocation such that for any network state x all the capacity constraints (1) are satisfied and at least one of them is satisfied as an equality, i.e., at least one network link is saturated. For this allocation, the balance function is obtained recursively from

$$\Phi(x) = \max_l \left\{ \frac{1}{C_l} \sum_{i:l\in r_i} \Phi(x - e_i) \right\}. \tag{2}$$

This allocation is referred to as balanced fairness. Any link l that realizes the maximum in (2) is said to be *saturated* in state x.

Assuming Poisson flow arrivals and exponential flow size distributions, it may readily be verified from the balance property that the invariant measure is given by

$$\pi(x_1, \ldots, x_N) = \Phi(x_1, \ldots, x_N)\rho_1^{x_1} \cdots \rho_N^{x_N}. \tag{3}$$

This result, however, has a much wider validity. As shown in [4], the bandwidth sharing network can be identified with a so-called Whittle network of processor sharing servers (cf. [5]). The insensitivity properties of Whittle networks allow us to conclude that the invariant measure (3) is valid for much more general traffic characteristics. Flow sizes and think time durations can have quite general distributions and need not be independent. The number of flows per session can be generally distributed. The only requirement is that sessions arrive as a Poisson process [4,8].

An important role is played by the normalization constant,

$$G(\rho) = G(\rho_1, \ldots, \rho_N) = \sum_{x_1=0}^{\infty} \cdots \sum_{x_N=0}^{\infty} \Phi(x_1, \ldots, x_N)\rho_1^{x_1} \cdots \rho_N^{x_N},$$

where the traffic load vector is denoted $\rho = (\rho_1, \ldots, \rho_N)$. $G(\rho)$ may be identified as the generating function of the balance function $\Phi(x)$ and thus contains the same information.

In particular, $\Phi(x)$ itself and the performance measures can be derived from $G(\rho)$. A key performance measure for class-i flows is the flow throughput, $\gamma_i = \mathrm{E}\left[S_i\right]/\mathrm{E}\left[T_i\right]$, where $\mathrm{E}\left[S_i\right]$ and $\mathrm{E}\left[T_i\right]$ are the mean flow size and sojourn time, respectively. Expanding by the flow arrival rate λ_i and applying Little's result we have $\gamma_i = \rho_i/\mathrm{E}\left[x_i\right]$. The denominator can be obtained by derivation yielding

$$\gamma_i = \frac{\rho_i}{\mathrm{E}\left[x_i\right]} = \frac{G(\rho)}{\frac{\partial}{\partial \rho_i}G(\rho)}. \tag{4}$$

In the rest of the paper we concentrate on an efficient recursive method for calculating the normalization constant and flow throughputs for particular network configurations.

3. RECURSIVE ALGORITHM

Expression (2) suggests an efficient recursive method can indeed be applied to those network topologies for which it is possible to identify the saturated links.

3.1. Recursion for the normalization constant

Let $\mathcal{I}_k = \{i_1, \ldots, i_k\}$ be a k-tuple, $k \leq N$, of unequal indices, $1 \leq i_1 < \ldots < i_k \leq N$. Define a k-dimensional set of states $\Omega_{\mathcal{I}_k}$ as follows,

$$\Omega_{\mathcal{I}_k} = \{x : x_i > 0 \text{ if and only if } i \in \mathcal{I}_k\},$$

i.e., the set of states where there are active flows in each of the classes represented by the index set $\mathcal{I}_k$, and only in those. Specifically, we define $\mathcal{I}_0$ to mean the empty set $\emptyset$ and $\Omega_\emptyset$ to mean the set consisting solely of the zero state, $\Omega_\emptyset = \{(0, \ldots, 0)\}$. The whole state space is decomposed as

$$\Omega = \sum_{k=0}^{N} \sum_{\mathcal{I}_k} \Omega_{\mathcal{I}_k}.$$

The decomposition Ω for a 3-dimensional case, $\Omega = \Omega_\emptyset + \Omega_1 + \Omega_2 + \Omega_3 + \Omega_{1,2} + \Omega_{1,3} + \Omega_{2,3} + \Omega_{1,2,3}$, is illustrated in Figure 1 (the set $\Omega_{1,2,3}$ not shown).

Defining partial sums over the sets $\Omega_{\mathcal{I}_k}$,

$$G_{\mathcal{I}_k} = \sum_{x \in \Omega_{\mathcal{I}_k}} \Phi(x_1, \ldots, x_N)\rho_1^{x_1} \cdots \rho_N^{x_N},$$

the normalization constant $G(\rho)$ can be decomposed accordingly,

$$G(\rho) = \sum_{k=0}^{N} \sum_{\mathcal{I}_k} G_{\mathcal{I}_k}(\rho). \tag{5}$$

Now we introduce the central assumption that for each $\mathcal{I}_k$ it is possible to identify at least one link that is saturated in all the states of $\Omega_{\mathcal{I}_k}$ (the links that are saturated are always uniquely defined by (2)). Note that this assumption is not valid for any network topology so that the algorithms described below do not have general applicability. The assumption allows us to derive a recursion expressing $G_{\mathcal{I}_k}(\rho)$ in terms of the $G_{\mathcal{I}_{k-1}}(\rho)$, where $\mathcal{I}_{k-1} \subset \mathcal{I}_k$ (one index in the set $\mathcal{I}_k$ dropped). The number of the different $G_{\mathcal{I}_k}$ is
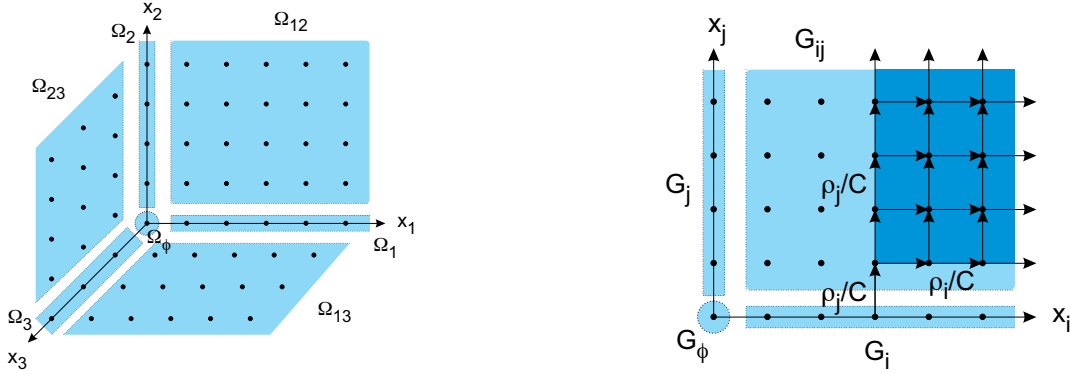
Figure 1. Decomposition of the state space (left) and illustration of the recursion for the normalization constant (right).

2^N, which leads to a manageable recursion up to $N = 10, \ldots, 15$. The number is anyhow far less than the number of states, n^N, to be evaluated by the basic recursion (2) when the state space is truncated in each direction at n (typically n is of the order $10, \ldots, 100$).

The recursion is best illustrated by an example. Consider a 2-dimensional set $\Omega_{i,j}$, see Figure 1. By our assumption, in all states of this set a given link is saturated. Let the capacity of this link be C. One or both of flow classes i and j must go through the link. For the sake of discussion, we assume that both classes do so. In view of (2) and (3), we deduce:

$$\pi(x) = \frac{\rho_i}{C}\pi(x - e_i) + \frac{\rho_j}{C}\pi(x - e_j), \quad \forall x \in \Omega_{i,j}.$$

Each state x in set Ω_i contributes as a "source of recursion" to the state sum $G_{i,j}(\rho)$ an amount equal to its own measure $\pi(x)$ times the expression

$$\frac{\rho_j}{C} \cdot \frac{1}{1 - (\frac{\rho_i}{C} + \frac{\rho_j}{C})}.$$

The first factor comes from the "bridge" and the second factor represents the infinite sum, $S_{i,j}$, over the area indicated in dark grey in Figure 1, when the value the lower left corner of the dark area is fixed to 1. Then the values of the neighbouring points are ρ_i/C and ρ_j/C and we see that $S_{i,j}$ is obtained as the solution of the equation

$$S_{i,j} = 1 + \frac{\rho_i}{C}S_{i,j} + \frac{\rho_j}{C}S_{i,j}.$$

Thus the overall contribution of the set Ω_i to $G_{i,j}(\rho)$ is

$$\frac{\rho_j}{C} \cdot \frac{1}{1 - (\frac{\rho_i}{C} + \frac{\rho_j}{C})} \cdot G_i(\rho).$$

A similar contribution comes from the set Ω_j and the sought for recursion is

$$G_{i,j}(\rho) = \frac{\rho_j G_i(\rho) + \rho_i G_j(\rho)}{C - (\rho_i + \rho_j)}.$$

In general, we have the recursion

$$G_{\mathcal{I}}(\rho) = \frac{\sum\limits_{i \in \mathcal{I}'} \rho_i G_{\mathcal{I} \setminus \{i\}}(\rho)}{C_{\sigma(\mathcal{I})} - \sum\limits_{i \in \mathcal{I}'} \rho_i}, \tag{6}$$

where for any set of classes $\mathcal{I}$, $\sigma(\mathcal{I})$ denotes the link which is saturated in any state $x \in \Omega_{\mathcal{I}}$, $\mathcal{I}' \subset \mathcal{I}$ stands for those classes $i \in \mathcal{I}$ for which $\sigma(\mathcal{I}) \in r_i$. If $\sigma(\mathcal{I})$ is not unique any of the saturated links can be used as the basis for the recursion.

3.2. Recursion for the throughput

As noted before the throughput can be derived from the normalization constant, eq. (4). One possibility then is to use the recursion (6) to find an explicit expression for the normalization constant, as in the examples presented later, and to obtain the throughput by derivation. In practice, however, when the number of classes is large the expression for the normalization constant easily becomes too cumbersome to be handled by hand (or even by a symbolic program like Mathematica). For such cases it is desirable to have a more direct way of calculating the throughput numerically. The denominator of (4) is decomposed using (5) as

$$\frac{\partial}{\partial \rho_i} G(\rho) = \sum_{k=0}^{N} \sum_{\mathcal{I}_k} \frac{\partial}{\partial \rho_i} G_{\mathcal{I}_k}(\rho) \equiv \sum_{k=0}^{N} \sum_{\mathcal{I}_k} H_{\mathcal{I}_k}^{(i)}(\rho),$$

and each term is obtained by derivation of (6),

$$H_{\mathcal{I}_k}^{(i)}(\rho) \equiv \frac{\partial}{\partial \rho_i} G_{\mathcal{I}_k}(\rho) = \frac{1_{i \in \mathcal{I}_k'} (G_{\mathcal{I}_k}(\rho) + G_{\mathcal{I}_k \setminus \{i\}}(\rho)) + \sum\limits_{j \in \mathcal{I}_k'} \rho_j H_{\mathcal{I}_k \setminus \{j\}}^{(i)}(\rho)}{C_{\sigma(\mathcal{I}_k)} - \sum\limits_{j \in \mathcal{I}_k'} \rho_j},$$

where 1_A is the indicator function having value 1 when event A is true and 0 otherwise. This recursion for the $H_{\mathcal{I}_k}^{(i)}$ can be applied in parallel with the recursion (6) for the $G_{\mathcal{I}_k}$.

4. APPLICATION TO SPECIFIC NETWORK TOPOLOGIES

We now apply the above algorithm to a number of network topologies for which the saturation property described in §3.1 holds.

4.1. Line

This configuration consists of $L = n$ links with capacities C_i and $N = n + 1$ classes or routes. Route 0 goes through all links and each of routes $i = 1, \ldots, n$ only goes through link i. When only route 0 is active the link with minimum capacity, denoted by C, is saturated. Whenever a route $i = 1, \ldots, n$ is active link i is saturated (Lemma 1 of [8]).

In this special case it is advantageous to redefine the sets in order to handle even larger groups of states together. We define:

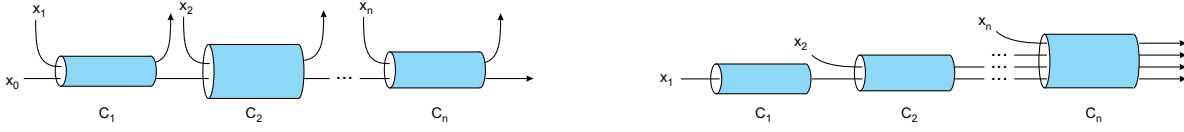$$\Omega_i = \{x : x_j = 0 \text{ for } j > i\},$$

Figure 2. Line and parking lot configurations.

and correspondingly $G_i(\rho)$ is the state sum over Ω_i. One easily sees that

$$\begin{cases} G_0(\rho) &= \frac{C}{C-\rho_0}, \\ G_i(\rho) &= \left(1 + \frac{\rho_i}{C_i-\rho_0-\rho_i}\right) \cdot G_{i-1}(\rho) = \frac{C_i-\rho_0}{C_i-\rho_0-\rho_i} \cdot G_{i-1}(\rho). \end{cases}$$

Thus we have the result

$$G(\rho) = \frac{C}{C-\rho_0} \cdot \frac{C_1-\rho_0}{C_1-\rho_0-\rho_1} \cdots \frac{C_n-\rho_0}{C_n-\rho_0-\rho_n}.$$

Using (4) the throughput of flow 0 is found to be

$$\gamma_0 = \left(\frac{1}{C-\rho_0} + \sum_{l=1}^{n} \left(\frac{1}{C_l-\rho_l-\rho_0} - \frac{1}{C_l-\rho_0}\right)\right)^{-1},$$

while the throughput of other flows is $\gamma_i = C_i - \rho_i - \rho_0$, for $i = 1, \ldots, n$ (cf. [8]).

4.2. Parking lot

In the parking lot configuration, $L = n$ links with capacities $C_1 \le C_2 \le \ldots \le C_n$ carry $N = n$ classes of flows. Class 1 flows go through all the links, class 2 flows go through links 2 to n, and, in general, class i flows go through links i to n (see Figure 2).

Whenever $x_i > 0$ but $x_j = 0$ for $j > i$, link $i = 1$ is saturated (this follows again from Lemma 1 of [8]). It is again advantageous to use the aggregate sets Ω_i, $i = 1, \ldots, n$, of the previous section. Now one finds that

$$\begin{cases} G_1(\rho) &= \frac{C_1}{C_1-\rho_1}, \\ G_i(\rho) &= \left(1 + \frac{\rho_i}{C_i-(\rho_1+\cdots+\rho_i)}\right) \cdot G_{i-1}(\rho) = \frac{C_i-(\rho_1+\cdots+\rho_{i-1})}{C_i-(\rho_1+\cdots+\rho_i)} \cdot G_{i-1}(\rho), \end{cases}$$

from which it follows on denoting the link i load by $R_i = \sum_{j=1}^{i} \rho_j$,

$$G(\rho) = \frac{C_1}{C_1-R_1} \cdot \frac{C_2-R_1}{C_2-R_2} \cdots \frac{C_n-R_{n-1}}{C_n-R_n}.$$

Throughputs are again obtained by eq. (4)

$$\gamma_i = \left(\frac{1}{C_i-R_i} + \sum_{l=i+1}^{n} \left(\frac{1}{C_l-R_l} - \frac{1}{C_l-R_{l-1}}\right)\right)^{-1}, \qquad i = 1, \ldots, n.$$

4.3. Concentration trees

We now consider tree topologies representing access networks. Routes are assumed to attain the network core via a common link constituting the root of the access network. The routes converge progressively over a number of levels. It can been shown that such trees satisfy the property that a given link or given set of links is saturated in all states corresponding a given set $\mathcal{I}$ of active flow classes. Limited space does not allow us to give the proof here but it will published in a future paper.

A saturated link for a given set $\mathcal{I}$ of active flow classes can be found as follows: First give each flow class the capacity of its access link. Proceeding from the leaves towards the root, always apply the capacity constraint of the links. The uppermost link (closest to the root) that is constraining is saturated in $\mathcal{I}$.

Tree with two branches

The 2-branch tree has a common link with capacity C_0 and two branches with capacities C_1 and C_2. Here we assume that the capacities of both branches are less than or equal to that of the common link, $C_1 \le C_0$ and $C_2 \le C_0$, but that their sum is greater than the capacity of the common link (otherwise the system reduces to a simpler configuration). This provides a straightforward application of the general recursion,
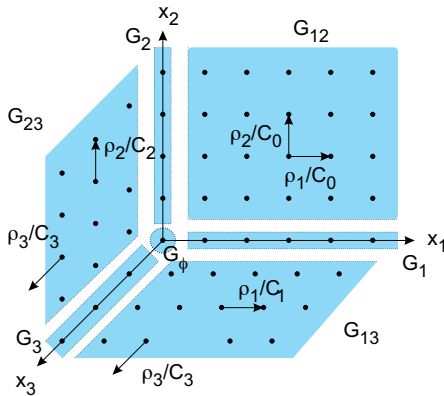
$$
\begin{cases}
G_\emptyset(\rho) & = \quad 1, \\[4pt]
G_1(\rho) & = \quad \frac{\rho_1 G_\emptyset(\rho)}{C_1 - \rho_1}, \\[4pt]
G_2(\rho) & = \quad \frac{\rho_2 G_\emptyset(\rho)}{C_2 - \rho_2}, \\[4pt]
G_{1,2}(\rho) & = \quad \frac{\rho_1 G_2(\rho) + \rho_2 G_1(\rho)}{C_0 - (\rho_1 + \rho_2)}.
\end{cases}
$$

Collecting the terms together gives $G(\rho) = G_\emptyset(\rho) + G_1(\rho) + G_2(\rho) + G_{1,2}(\rho)$,

$$
G(\rho) = \frac{1}{1 - \frac{\rho_1 + \rho_2}{C_0}} \cdot \left(\frac{1 - \frac{\rho_1}{C_0}}{1 - \frac{\rho_1}{C_1}} + \frac{1 - \frac{\rho_2}{C_0}}{1 - \frac{\rho_2}{C_2}} - 1 \right).
$$

Tree with three branches

The tree has a common link of capacity C_0 and three branches of capacities C_1, C_2, and C_3. Again we assume that the capacities of all branches are less than that of the common link but that their sum is greater. It is necessary to distinguish between different cases according to how the sums of the capacities of different pairs of links relate to the capacity of the common link. Because of lack of space we deal with only one of the four cases, namely the case where only the sum $C_1 + C_2$ is greater than C_0, demonstrating the application of the general recursion:

$$
\begin{cases}
G_\emptyset(\rho) & = \quad 1, \\[4pt]
G_1(\rho) & = \quad \frac{\rho_1 G_\emptyset(\rho)}{C_1 - \rho_1}, \\[4pt]
G_2(\rho) & = \quad \frac{\rho_2 G_\emptyset(\rho)}{C_2 - \rho_2}, \\[4pt]
G_3(\rho) & = \quad \frac{\rho_3 G_\emptyset(\rho)}{C_3 - \rho_3}, \\[4pt]
G_{1,2}(\rho) & = \quad \frac{\rho_1 G_2(\rho) + \rho_2 G_1(\rho)}{C_0 - (\rho_1 + \rho_2)}, \\[4pt]
G_{1,3}(\rho) & = \quad \frac{\rho_1 G_3(\rho)}{C_1 - \rho_1} = \frac{\rho_3 G_1(\rho)}{C_3 - \rho_3}, \\[4pt]
G_{2,3}(\rho) & = \quad \frac{\rho_2 G_3(\rho)}{C_2 - \rho_2} = \frac{\rho_3 G_2(\rho)}{C_3 - \rho_3}, \\[4pt]
G_{1,2,3}(\rho) & = \quad \frac{\rho_1 G_{2,3}(\rho) + \rho_2 G_{1,3}(\rho) + \rho_3 G_{1,2}(\rho)}{C_0 - (\rho_1 + \rho_2 + \rho_3)}.
\end{cases}
$$

By simplification one obtains

$$
G(\rho) = \frac{1}{1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0}} \cdot \frac{1}{1 - \frac{\rho_3}{C_3}} \left(\frac{1 - \frac{\rho_1 + \rho_3}{C_0}}{1 - \frac{\rho_1}{C_1}} + \frac{1 - \frac{\rho_2 + \rho_3}{C_0}}{1 - \frac{\rho_2}{C_2}} - \left(1 - \frac{\rho_3}{C_0}\right) \right).
$$

5. EXTENSION TO LIMITED ACCESS RATES

We have so far considered a model where flow rates are constrained by network links only. In practice, the rate of a flow may additionally be constrained by a fixed maximum limit representing the user's access line for instance. Balanced fairness is then defined by adding the corresponding constraints in the basic recursion (2) [4]. For sake of clarity, we restrict the presentation of the extended algorithm to the particular case where the network reduces to a single link of capacity C.

All flows in each class i are limited by an access rate a_i, i.e., $\phi_i(x) \leq x_i a_i$. As long as the sum of the access rates of the active flows is less than C, each flow is allocated a rate equal to its access rate. Denote by $\Omega_\emptyset = \{x : \sum_i x_i a_i \leq C\}$ the set of the corresponding states: For the present discussion it is most convenient to think in terms of integer valued rates a_i and capacity C. The Kaufman-Roberts recursion (cf., e.g., [9]) can be used to obtain the partial sum over $\Omega_\emptyset$:

$$G_\emptyset = \sum_{c=0}^{C} g(c), \qquad g(c) = \sum_{i=1}^{N} \frac{\rho_i}{c} g(c - a_i), \qquad c = 1, 2, \ldots$$

with $g(0) = 1$ and $g(c) = 0$ for $c < 0$. This also gives the measures of the "blocking sets" (referring to a virtual multirate system, where flows are inelastic having fixed bandwidth requirements of a_i, cf. [9]),
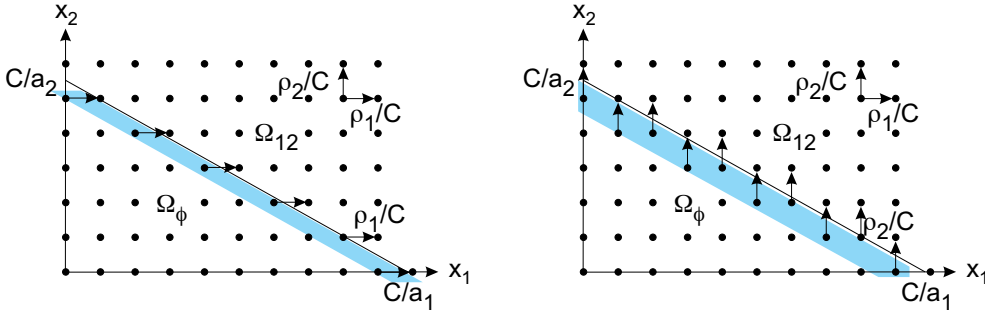
$$B_i = \sum_{c=C-a_i+1}^{C} g(c).$$

Figure 3. Boundary between inelastic and elastic capacity sharing between two classes.

Elastic capacity sharing only becomes effective when the sum of the access rates of the active flows exceeds C. The boundary between inelastic, $\Omega_\emptyset$, and elastic, $\Omega_{1,2}$, capacity sharing is illustrated in Figure 3 for the case of two access rate classes. The blocking sets of the respective classes are shown by the shaded areas. As in Section 3, each state x in the blocking set of class i contributes as a "source of recursion" to the normalization constant an amount equal to its own measure $\pi(x)$ times the expression $(\rho_i/C)/(1-\rho/C)$, where $\rho = \sum_i \rho_i$ is the total load. We deduce:

$$G = G_\emptyset + \sum_{i=1}^{N} \frac{\rho_i B_i}{C - \rho}.$$

6. SUMMARY

Balanced fairness is a new notion of bandwidth allocation with the very gratifying property that flow level performance metrics are insensitive to detailed traffic characteristics. This is particularly important for data network engineering since performance can be predicted from an estimate of overall traffic volume alone and is independent of changes in the mix of user applications.

The balanced fair allocation for any network is uniquely determined by the basic recursion (2). For larger networks, however, straightforward application of the recursion is hampered by the usual state space explosion problem. Our main contribution is the derivation of a recursive algorithm for directly calculating the normalization constant and flow throughputs for any network satisfying the saturation property discussed in Section 3, viz. that in each set of states $\Omega_{\mathcal{T}}$ it is possible to identify a link which is always saturated. While this crucial property is not generally valid, it is provably so for the line, the parking lot and the concentration tree considered here. Simple examples were provided; more complicated cases of concentration trees can easily be evaluated in the same way.

We have outlined an extension of the algorithm to take account of external limits on flow rate due, for example, to the speed of user access lines. We consider the particular case of an isolated bottleneck link with heterogeneous rate limits. The development of a recursive algorithm for more general network topologies is the subject of current research.

REFERENCES

1. J. Mo and J. Walrand, Fair end-to-end window-based congestion control, IEEE/ACM Transactions on Networking, 8 (2000) 556-567.
2. D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987.
3. F.P. Kelly, A. Maulloo and D. Tan, Rate control in communication networks: shadow prices, proportional fairness and stability, Journal of the Operational Research Society, 49 (1998) 237-252.
4. T. Bonald, A. Proutière, Insensitive bandwidth sharing in data networks, Queueing Systems, 44 (2003) 69-100.
5. R. Serfozo, *Introduction to Stochastic Networks*, Springer Verlag, 1999.
6. T. Bonald, A. Proutière, Insensitivity in processor-sharing networks, Performance Evaluation, 49 (2002) 193-209.
7. T. Bonald, A. Proutière, G. Régnié, J. Roberts, Insensitivity results for statistical bandwidth sharing, in Proceedings of ITC 17, Elsevier, 2001, 125-136.
8. T. Bonald, A. Proutière, On performance bounds for balanced fairness, to appear in Performance Evaluation, 2003.
9. J. Roberts, U. Mocci, and J. Virtamo (eds.), *Broadband Network Teletraffic*, Springer Verlag, 1996.