Semi-Definite Programming for bandwidth allocation in communication networks

Corinne Touati[a], Eitan Altman[a] and Jérôme Galtier[ab]

[a]INRIA, 2004 route des Lucioles, BP93, 06902 Sophia Antipolis, France

[b]France Telecom R&D, 905 rue Albert Einstein, 06921 Sophia-Antipolis, France

We aim at assigning bandwidth fairly to calls that have minimum bandwidth requirements. Our goal is to provide a general computational tool based on Semi-Definite Programming (SDP). The advantage of our approach is that many appealing complex objective functions related to fairness criteria are special cases of our formalism. Therefore, such optimization problems can be solved in an automatic manner provided that a SDP package is available to the network manager. In particular, our approach gives the solutions of maxmin and proportional fairness assignments, and is suitable for both ATM and Internet contexts.

1. Introduction

We are interested in assigning fairly bandwidth between connections that have minimum requirements. Two major approaches have been developed in recent years for bandwidth assignment.

In ATM networks, the network is responsible for bandwidth assignment [1]. In particular, applications using the ABR (Available Bit Rate) service are informed by the network of the transmission rate assigned to them. This is done either through control cells that include explicit rate commands, or through a single congestion notification bit whose value determines whether the call is allowed to increase or has to decrease its transmission rate. In the Internet, an alternative approach, which leaves the congestion control to the sources, has evolved. In this approach, it is the transport control protocol (TCP) used by a connection that determines the amount of bandwidth (or throughput) of that connection.

Efficient (possibly centralized) numerical procedures for computing bandwidth allocation can be useful both for ATM and Internet networks. For ATM networks, they can be used directly in the control of transmission rates. In Internet networks they can be used for network dimensioning purposes. Indeed, it has been observed that the actual way that bandwidth is shared using Internet type congestion protocols can be described as a solution of an optimization problem [9,4,5]. By solving this problem, the network manager can thus predict the influence of network topology and link capacities on the resource utilization and on the resulting bandwidth sharing.

Our goal is to provide a general computational tool for bandwidth allocation in networks which is based on SDP. The advantage of our approach is that many appealing complex objective functions that appear in network optimization problems (especially those related to fair bandwidth allocation problems) are shown to be special cases of our formalism, and therefore such optimization problems can be solved in an automatic manner provided a standard SDP software package is available to the network manager. The alternative would be to use general nonlinear optimization software which are much more difficult to handle from the numerical point of view. In contrast, the inputs of the SDP program will be shown to be simple entries of some matrices. Our main contribution is to identify the way to transform objective functions related to fairness into the constraints of positiveness of matrices. Finally, we point out that the algorithm is polynomial.

After recalling the basics of bandwidth assignment and of fairness concepts in Section 2, we show in Section 3 how different network functionals can be described with SDP. In Section 4 we present some practical results obtained using our method and comment on its implementation and complexity.

2. Technical framework

2.1. Fairness and global optimization criteria

Several concepts of fairness can be found in the literature. The most commonly found are the max-min and the proportional fairness. The max-min fairness has been adopted by the ATM-forum [1] as the standard for throughput allocation for the best-effort traffic class ABR. An allocation of the excess throughput (beyond the minimum required) is max-min fair if and only if there is no possibility to allocate more excess throughput to a connection unless we allocate less excess throughput to another connection that has less excess throughput than the previous one.

Proportional fairness on the other hand can characterize TCP. Indeed, it has been shown that Vegas variant of TCP achieves proportional fairness [9]. A bandwidth assignment is proportionally fair if for any other assignment, the aggregate changes in the throughputs is zero or negative. It is shown in [6] that an assignment is proportionally fair if and only if it maximizes the product of the bandwidths.

We may also consider the criterion of maximizing the total used bandwidth although it can lead to an unfair solution in the sense that it may assign no bandwidth to a connection (examples can be found in [11]). It is called *Global Optimization*.

All these criteria produce Pareto optimal solutions ([3]). Let λ_n be the quantity allocated to connection n (typically, the throughput in excess with some given minimum MR_n). An allocation vector λ is said to be Pareto optimal if there is no other allocation that strictly dominates it. This means that there is no allocation where all connections get a greater or equal amount of resource than with λ, with at least one connection getting strictly more than with λ. The Pareto definition ensures that no bandwidth is wasted.

A general fairness criterion that covers the previous concepts is due to [6]. Let us denote λ_n the bandwidth allocated to connection n, with $1 \leq n \leq N$. Given a positive constant $\alpha \neq 1$, consider the optimization problem:

$$\text{Maximize } \frac{1}{1-\alpha} \sum_{n=1}^{N} \lambda_n^{1-\alpha} \tag{1}$$

subject to the problem's constraints (which we describe later). Since the utility function is concave and the constraints will be shown to be linear, this defines the unique so called α-allocation. It corresponds to the *globally optimal allocation* when $\alpha = 0$, the *proportional fairness* when $\alpha \to 1$, the so called *harmonic mean fairness* when $\alpha = 2$, and the *max-min* allocation when $\alpha \to \infty$.

Let f_n be the utlity functions of the assigned throughput x_n of connection n in excess to the minimum MR_n. They are known to be strictly concave increasing functions especially in real time applications [10].

We then consider the more general case in which λ_n in 1 represents the utility corresponding to the assigned bandwidth, i.e. $f_n(x_n)$ rather than the actual bandwidth x_n.

$$\text{Maximize } \frac{1}{1-\alpha}\sum_{n=1}^{N} f_n(x_n)^{1-\alpha}. \tag{2}$$

When $\alpha \to 1$, (2) represents the so called "generalized proportional fairness" (GPF) or the "Nash Bargaining solution" (NBS), see [11] and references therein. For that case, we can rewrite (2) in its equivalent form:

$$\text{Maximize } \prod_{n=1}^{N} f_n(x_n). \tag{3}$$

2.2. Problem formulation

We model a network with L links. C_l is the capacity of link $l, 1 \le l \le L$. Each of the N connections requires some bandwidth within the range $[MR_n, PR_n]$ (users constraints). We suppose that we have a fixed given $L \times N$ routing matrix A. The element $A_{i,j}$ is equal to 1 if connection j goes through link i, 0 otherwise. Then, if C is the capacity vector, the standard capacity constraints are $\forall l, 1 \le l \le L, (Ax)_l \le (C)_l$.

We assume that the network has sufficient bandwidth to satisfy all the connections' minimum requirements i.e. $\forall l, 1 \le l \le L, \sum_{n=1}^{N} a_{ln} MR_n < C_l$.

To handle a rich class of utilities functions with different degrees of concavity, we shall use a parabolic class of utilities (see [11]):

$$f_n(x) = c_n - a_n(x_n - b_n)^2. \tag{4}$$

The problem is then to find a vector $x \in X$:

$$\max \frac{1}{1-\alpha}\sum_{n=1}^{N} f_n(x_n)^{1-\alpha} \text{ where } X := \{x | \forall l, (Ax)_l \le (C)_l, \forall n, MR_n \le x_n \le PR_n\}. \tag{5}$$

3. A SDP formulation

In this section, we propose a centralized method for solving the general fairness problem (1). It uses a mathematical program called Semi-Definite Program, which can be solved in polynomial time in theory and is tractable in practice. SDP solves the minimization problem of a linear combination of variables (given by the scalar product of a vector L

and the vector of variables) subject to a constraint of positive semi-definiteness (psd) of some general symmetric matrix P whose entries are either variables or constants[1]. For more details on SDP, see [8]. One can use public domain programs[2] to solve SDP.

The basic idea, which we present below, is to transform the original maximization problem into a minimization problem of a new variable. We use the psd of the general matrix to (i) ensure the capacity and the users constraints (ii) replace the objective function by a single variable. The form of the original objective function is therefore expressed in the matrix.

The matrix is block diagonal, and is thus psd if and only if each block is psd. In our fairness computation, we will show in the following that the blocks will always be of size smaller than or equal to two.

We sketch in the next subsection some properties of psd matrices. We then give some ideas on the construction of the block matrices related to the different values of α. We finally present a detailed construction of a SDP.

3.1. Preliminary results on positive semi-definite matrices

First note that the positiveness of a matrix of size 1 (scalar values) expresses linear constraints. In particular, the capacity constraints require L matrices of size 1 and user constraints $2N$ matrices.

Proposition 3.1 (psd of matrices of size 2) *Let a, b, and c be positive real numbers. Then*

$$\begin{pmatrix} a & c \\ c & b \end{pmatrix} \succeq 0 \textit{ if and only if } ab \geq c^2. \tag{6}$$

Using (6), we have:

Proposition 3.2 (quadratic utility functions) *Let x_n be a positive number and f_n its utility function as defined in (4). We have $P_n \succeq 0 \Leftrightarrow w_n \leq f_n(x_n)$ with:*

$$P_n = \begin{pmatrix} -\frac{w_n - c_n}{a_n} & x_n - b_n \\ x_n - b_n & 1 \end{pmatrix}.$$

We can then easily prove that $\frac{1}{1-\alpha}\sum w_n^{1-\alpha} \leq \frac{1}{1-\alpha}\sum f_n(x_n)^{1-\alpha}$ regardless of the value α. We then introduce intermediate variables y_n and ensure that $y_n \leq \frac{w_n^{1-\alpha}}{1-\alpha}$. Clearly,

Lemma 3.1 *Let y and $y_n, 1 \leq n \leq N$ be positive numbers and $\forall n, y_n \leq \frac{w_n^{1-\alpha}}{1-\alpha}$. Then $M \succeq 0 \Rightarrow y \leq \sum \frac{w_n^{1-\alpha}}{1-\alpha}$ with M the matrix $M = (\sum y_n - y)$.*

Then, maximizing the variable y will maximize $\frac{1}{1-\alpha}\sum f_n(x_n)^{1-\alpha}$. We therefore provide, in the following subsection variables y that verify (we omit the "n" subscript)

$$y \leq w^{1-\alpha} \quad \text{if } 1-\alpha > 0, \qquad y \geq w^{1-\alpha} \quad \text{otherwise.}$$

[1]A matrix A is said psd ($A \succeq 0$) if all its eigenvalues are non negative. Equivalently, for a symmetric matrix of size $n \times n$, $A \succeq 0 \Longleftrightarrow \forall x \in \mathbb{R}^n, x^T Ax \geq 0$

[2]see http://www.cs.nyu.edu/cs/faculty/overton/software/sdppack.html

3.2. Fairness computation for $\alpha \neq 1$

The following can be drawn from Proposition (3.1).

Proposition 3.3 (Case $\alpha = 2$, harmonic fairness) *The positiveness of matrix H assures that $w \leq 1/y$.*

$$H = \begin{pmatrix} y & 1 \\ 1 & w \end{pmatrix}$$

Thanks to an idea of Nemirovski [7], we can also integrate the following series of functions in our model.

Proposition 3.4 (case $0 < \alpha < 1$) *Let $w, y \in \mathbb{R}^+$. It is possible, using SDP constraints, to bound w and y by the relation $y \leq w^{k/2^p}$, with $p \in \mathbb{N}$ and $k \in \{0, \ldots, 2^p - 1\}$.*

Proof: Let $a_1, \ldots, a_p$ be a series of 0/1 integers, such that $k = \sum_{i=1}^{p} a_i 2^{i-1}$. We note $y_0 = 1$, and submit $y_1, \ldots, y_p$ to the following constraints:

$$\begin{cases} \begin{pmatrix} y_{i-1} & y_i \\ y_i & w \end{pmatrix} \succeq 0 & \text{if } a_i = 1 \\ \begin{pmatrix} y_{i-1} & y_i \\ y_i & 1 \end{pmatrix} \succeq 0 & \text{if } a_i = 0 \end{cases}$$

Then, obviously, $y_i^2 \leq y_{i-1} w^{a_i}$, and $y_p \leq w^{k/2^p}$. Hence the result, by setting $y_p = y$. □

In other words, if $\alpha \in (0, 1)$ is approximated by the value $1 - k/2^p$, with $k \in \{0, \ldots, 2^p - 1\}$, then one can generate constraints of the form $y \leq \sum_{i=1}^{n} {w_i}^{1-\alpha}$ and maximizing y solves our problem with very good precision.

We next present a simple solution for $\alpha \in (1, 2)$.

Proposition 3.5 (case $1 < \alpha < 2$) *Let $w, y \in \mathbb{R}^+$. It is possible, using SDP constraints, to bound w and y by the relation $y \geq w^{-\beta}$, where $\beta = k/2^p$, $p \in \mathbb{N}$ and $k \in \{0, \ldots, 2^p - 1\}$.*

Proof: Let z be an intermediate variable. Using proposition 3.4, one can set $z \leq w^\beta$. Also one can write

$$\begin{pmatrix} y & 1 \\ 1 & z \end{pmatrix} \succeq 0$$

which leads to $yz \geq 1$. Then $yw^\beta \geq 1$. □

The following proposition covers the cases $\alpha \in (2; +\infty)$.

Proposition 3.6 (case $\alpha > 2$) *Let $w, y \in \mathbb{R}^+$. It is possible, using SDP constraints, to bound w and y by the relation $y \geq w^{-1/\beta}$, where $\beta = k/2^p$, $p \in \mathbb{N}$ and $k \in \{0, \ldots, 2^p - 1\}$.*

Proof: Similarly, we obtain $wy^\beta \geq 1$. □

3.3. Computing the NBS and GPF (case $\alpha \to 1$)

The result for the NBS or GPF relies on the following:

Proposition 3.7 *Let y, and $w_1, \dots, w_n$ be real positive numbers. Then using SDP constraints, it is possible to bound these numbers by the relation*

$$y^{2^{\lceil \log_2(n) \rceil}} \leq \prod_{i=1}^{n} w_i.$$

Thus maximizing y leads immediately to the solution of the problems NBS and GPF.

Proof: Let p be the smallest integer such that $2^p \geq n$. We construct a family of real positive variables $y_{i2^k+1,(i+1)2^k}$ with $1 \leq k \leq p$, and $i \in \{0, \dots, 2^{p-k} - 1\}$ satisfying the constraints expressed by the following $2^p - 1$ matrices:

$$\begin{pmatrix} y_{2i2^{k-1}+1,(2i+1)2^{k-1}} & y_{i2^k+1,(i+1)2^k} \\ y_{i2^k+1,(i+1)2^k} & y_{(2i+1)2^{k-1}+1,(2i+2)2^{k-1}} \end{pmatrix} \succeq 0,$$

where we denote $y_{j,j} = w_j$ for $j \in \{1, \dots, n\}$, $y_{j,j} = 1$ for $j \in \{n+1, \dots, 2^p\}$, and $y = y_{1,2^p}$.

Then we obtain constraints of the form $y_{1,2}^2 \leq w_1 w_2$, $y_{3,4}^2 \leq w_3 w_4$, $y_{1,4}^2 \leq y_{1,2} y_{3,4}$ and finally $y_{1,2^p}^{2^p} \leq \prod_i w_i$. □

For instance, in our SDP program, we can add artificial connections, in the case of the computation of the NBS so that the total number of connections will have the form 2^p with $p \in \mathbb{N}$ and then use the previous theorem to ensure that $y^{2^p} \leq \prod_{i=1}^{p} w_i$. We will suppose that the extra connections use their own links, and therefore do not modify the NBS.

3.4. A simple example of NBS

Consider $N = 3$ connections over $L = 4$ links. The connections are defined by the routing matrix:

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

Recall that element $A_{i,j}$ equals 1 if and only if connection j uses link i. In our SDP program, as $N = 3$, we need to add one extra artificial connection (as explained in Section 3.3). We suppose that this extra connection uses its own link, and therefore does not modify the NBS of our problem.

Step 1 As shown in Section 3.1, the first four blocks of the matrix link the variables x_i with their utility:

$$MAT_{1,i} = \begin{pmatrix} -\frac{w_i - c_i}{a_i} & x_i - b_i \\ x_i - b_i & 1 \end{pmatrix}$$

Step 2 The following matrices link the w_i variables together to obtain a single variable that SDP will maximize (from proposition 3.7).

$$\begin{pmatrix} w_1 & y_{1,2} \\ y_{1,2} & w_2 \end{pmatrix}, \begin{pmatrix} w_3 & y_{3,4} \\ y_{3,4} & w_4 \end{pmatrix}, \begin{pmatrix} y_{1,2} & y_{1,4} \\ y_{1,4} & y_{3,4} \end{pmatrix}.$$

The positiveness of these matrices implies that $(y_{1,4})^4 \leq (y_{1,2})^2 \cdot (y_{3,4})^2 \leq w_1 \cdot w_2 \cdot w_3 \cdot w_4$. Then, maximizing the single variable $y_{1,4}$ will lead to the required maximization of $\prod_i (c_i - a_i(x_i - b_i)^2)$.
Step 3 We incorporate the linear constraints of the problem. The constraints $(Ax)_l \leq C_l$ lead to the declaration of L matrices that are in our example:

$$(C_1 - (x_1 + x_3)), \quad (C_2 - x_2), \quad (C_3 - (x_2 + x_3)) \text{ and } (C_4 - x_1).$$

The constraints $x_i \leq PR_i$ and $x_i \geq MR_i$ are reflected by 8 matrices of size one that are:

$$(PR_i - x_i), \quad \textit{and} \quad (x_i - MR_i), \quad 1 \leq i \leq 4.$$

We can notice that the values PR_4 and MR_4 corresponding to the artificial connection are not important since the connection is independent of the others. Still, it is important to bound x_4 so that it will not grow without control which may cause an error. The solution of SDP will then be $x_4 = PR_4$.

The entries we should give to the SDP algorithm are the matrix we described, plus the vector L. As we want to maximize $y_{1,4}$, the vector should have a coefficient -1 and 0 otherwise.

Remark 3.1 *In case of N connections with L links, we stress we have at most $6N-7$ variables, $4N-5$ blocks of size 2, and $4N+L-4$ blocks of size 1. Therefore, solutions can be obtained even for very large systems.*

Indeed, let p be the smallest integer such that $2^p \geq N$ and $q = 2^p$. Then, one can easily show that, our method will lead to: $q + \sum_{i=0}^{p} q2^{-i} = 3 \times 2^p - 1$ variables, $q + \sum_{i=1}^{p} q2^{-i} = 2^{p+1} - 1$ blocks of size 2 and $L + 2 \cdot 2^p = 2^{p+1} + L$ blocks of size 1. In the worst case, we have $N = 2^{p-1} + 1$, hence the result.

4. Practical experiments

We implemented the SDP approach using a Matlab program on a SUN ULTRA 1 computer to obtain the GPF. We first tested our program on a linear network (for which we had explicit expressions), and the results completely agreed [11].

We then considered two more complex networks which we describe below. The computational time (including the display part) in both cases was less than a minute. In both networks, all links are assumed to have the same capacity C=100 (although the program allows to handle different capacities without increasing the complexity). For each network, we present a figure with the set of links and the amount of assigned bandwidth to each connection. All connections had the same quadratic utility 4 with parameters $MR = 10$, $PR = 80$, $a = 1/490$, $b = 745$ and $c = 1102.5$. Bandwidth parameters and assignments are given in percent of full link capacity.

Consider the first network. It has $L = 10$ links and $N = 11$ connections as defined in matrix A below.

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The solution given in Fig. 1 involved adding 36 intermediate variables, and the psd matrix was of size 104 (31 matrices of size 2, and 42 of size 1).

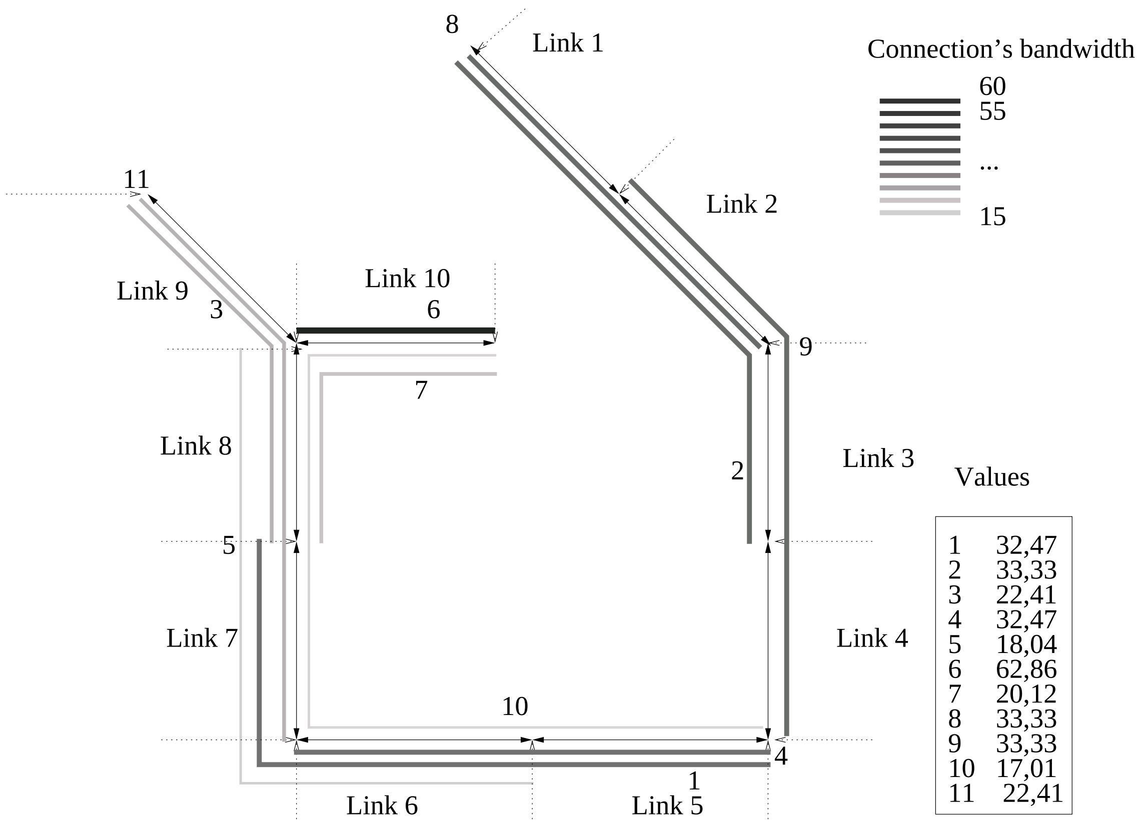

Figure 1. First network

We considered next the COST experimental network [2]. It contains 11 nodes, representing the main European capitals. We have considered the 30 connections with the highest forecast demand (We did not include more connections whose forecast demand, based on experiments dating from 1993, were less than 2.5 Gb/s). The solution, depicted in Fig. 2 involved adding 65 intermediate variables, and the psd matrix was of size 215 (63 matrices of size 2, and 89 matrices of size 1).

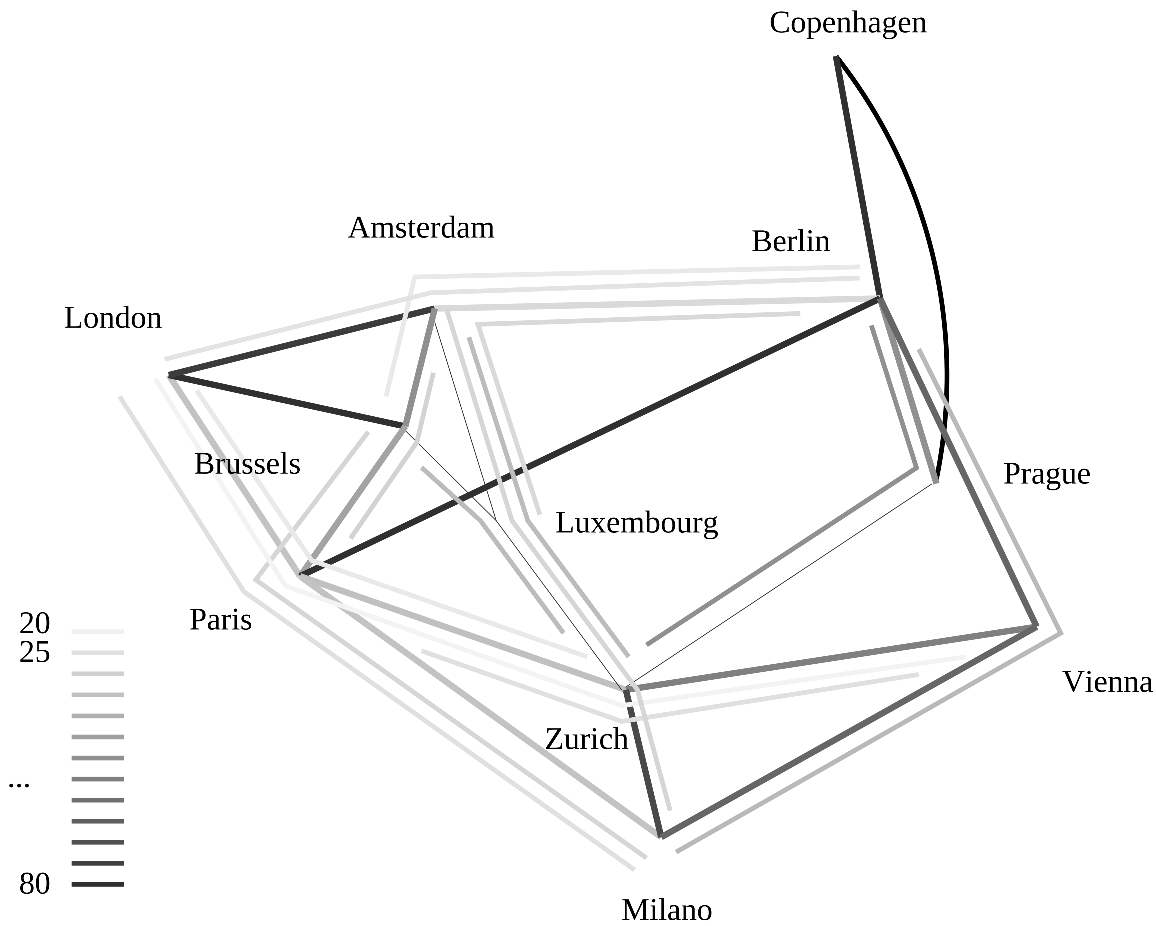

Figure 2. COST network

Conclusion

We have provided in this paper a centralized computational approach for fair bandwidth allocation, based on the SDP approach. This method has the advantage that it is simple to implement using any general SDP software. Furthermore additional conditions (for instance those linked to integer programming, or other telecommunication requests) can be introduced without any study on the stability or the convergence of the algorithm. This is a clear improvement compared to many other specific methods, and in particular to the iterative ones that have been proposed so far for solving fairness problems.

Also, we considered a fairness approach that takes into account not just the assigned throughput of each connection but instead the utility that the assigned throughput rep-

Connection	Bandwith	Connection	Bandwith	Connection	Bandwith
London-Paris	33.93	Zurich-Milano	71.58	Milano-Vienna-Berlin	37.00
London-Brussels	80.00	Copenhaguen-Berlin	80.00	Milano-Paris-Brussels	27.93
London-Amsterdam	76.27	Copenhaguen-Prague	80.00	Berlin-Amsterdam-Brussels	22.04
Amsterdam-Berlin	27.11	Berlin-Prague	50.00	Paris-Brussels-Amsterdam	28.42
Amsterdam-Brussels	49.54	Berlin-Vienna	63.00	Paris-Zurich-Vienna	25.48
Brussels-Paris	43.66	Milano-Vienna	63.00	London-Paris-Milano	24.74
Paris-Berlin	80.00	Berlin-Amsterdam-Luxembourg	27.11	London-Paris-Zurich	21.87
Paris-Zurich	33.19	Zurich-Prague-Berlin	50.00	London-Amsterdam-Berlin	23.73
Paris-Milano	47.34	Zurich-Luxembourg-Amsterdam	35.79	Vienna-Zurich-Paris-London	19.46
Zurich-Vienna	55.06	Zurich-Luxembourg-Brussels	35.79	Milano-Zurich-Luxembourg-Amsterdam	28.42

Figure 3. Bandwidth allocation for COST network.

resents. This type of fairness concept agrees with the game theoretic definition of fairness given by NBS and is an interesting generalization of the fairness criteria that have been used so far in the telecommunication context.

REFERENCES

1. Traffic management specification, version 4.0. Technical Report AF-TM-0056.000, ATM Forum Traffic Management Working Group, April 1996.
2. Cost 239. Ultra high capacity optical transmission networks, final report of Cost project 239. *ISBN 953-184-013-X*, 1999.
3. T. Bonald and L. Massoulié. Impact of fairness on internet performance. Sigmetrics, 2001.
4. F. Kelly. Mathematical modelling of the Internet. In B. Engquist and W. Schmid, editors, *Mathematics Unlimited - 2001 and Beyond*, pages 685–702. Springer-Verlag, Berlin, 2001.
5. S. Kunniyur and R. Srikant. End-to-end congestion control schemes: Utility functions, random losses and ECN marks. In *INFOCOM (3)*, pages 1323–1332, 2000.
6. J. Mo and J. Walrand. Fair end-to-end window-based congestion control. In *SPIE '98, International Symposium on Voice, Video and Data Communications*, 1998.
7. A. Nemirovski. What can be expressed via conic quadratic and semidefinite programming. *Lecture notes*, 1998.
8. E. Feeron S. Boyd, L. El Ghaoui and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. Springer, 1994.
9. L. Peterson S. Low and L. Wang. Understanding TCP Vegas: A Duality Model. In *Proceedings of the ACM SIGMETRICS 2001 Conference*, pages 226–235, June 2001.
10. S. Shenker. Fundammental design issues for the future internet. *IEEE JSAC*, 13(7):1176–1188, 1995.
11. C. Touati, E. Altman, and J. Galtier. On fairness in bandwidth allocation. *INRIA Research Report, RR 4269*, 2001.