Semi-Definite Programming approach for bandwidth allocation and routing in networks

Corinne Touati*, Eitan Altman*,† and Jérôme Galtier*,◇
*INRIA, 06902 Sophia Antipolis, France
†CESIMO, Facultad de Ingenieria, Universidad de Los Andes, Mérida, Venezuela
◇ France Telecom R&D, 06921 Sophia-Antipolis, France

Abstract—**We consider in this paper the problem of assigning bandwidth 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 that appear in network optimization problems are shown to be special cases of our formalism, and therefore such optimization problems can be solved in an authomatic manner whenever an SDP package is available to the network manager. In particular, our approach produces the solutions of maxmin fairness and proportional fairness assignments, and is suitable both for ATM as well as for Internet contexts. We finally extend our formalism to include also routing.**

I. INTRODUCTION

We consider in this paper the problem of assigning bandwidth to calls that have minimum bandwidth requirements. The problem is then of how to assign the excess capacity in the network, if available. Two major approaches have developed in recent years for bandwidth assignment. In ATM networks, the responsibility for bandwidth assignment lies on the network [1]. In particular, application using the ABR (Available Bit Rate) service are informed by the the network on the allowable transmission rate (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). An alternative approach for bandwidth assignment which has evolved in the Internet leaves the congestion control to the sources; the amount of bandwidth (or throughput) used by a connection is determined by the transport control protocol (TCP) version that is used by that connection as well as that used by other connections sharing the network.

Efficient (possibly centralized) numerical procedures for computing bandwidth allocation can be useful for both ATM as well as Internet networks. For ATM networks, they can be used directly in the control of transmission rates. In Internet networks they can be used for networking dimensionning 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 [5], [6]. By solving this optimizaton 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 in this paper is to provide a general computational tool for joint routing as well as bandwidth allocation in networks which is based on Semi-Definite Programming (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 authomatic manner whenever a standard SDP software package is available to the network manager. In particular, our approach produces the solutions to the two problems stated above, of bandwidth allocation for ATM and for Internet networks.

After recalling the basics of network flow and fairness problems in Section 2, we show in Section 3 how different network functionals can be described as a SDP program. Finally, in Section 4 we present some practical results obtained using our method.

II. TECHNICAL FRAMEWORK

A. Fairness and global optimization criteria

Several concepts of fairness can be found in the litterature. 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 (Available Bite Rate). An allocation is max-min fair if and only if there is no possibility to allocate more excess throughput to an operator unless we allocate less excess throughput to another operator that has less excess throughput than the previous one.

The Internet is an example where proportional fairness is used. Indeed, the TCP congestion control mechanism, and more generally, congestion controls mechanisms based on linear increase and multiplicative decrease, turn out to share network's bandwidth in a way approximated by the proportional fairness concept ([5]). A bandwidth assignment is proportionally fair if for any other asignment, the aggregate changes in the throughputs is zero or negative. It is shown in [7] that an assignment is proportionnally fair if and only if it maximizes the product of the excess of bandwidth.

We may also consider the criterion of maximizing the total used bandwidth although we saw that it would lead to an unfair solution. It is called *Global Optimization*.

All these criteria produce Pareto optimal solutions ([3]). Let λ_i correspond to a quantity that is to be allocated for connection i (typically, the throughput in excess with some given minimum MR_i). 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 operators get an allocation greater than or equal to λ with at least one operator getting strictly more than according to λ. The Pareto definition assures that no bandwidth is wasted.

A general fairness criterion that covers all the previous concepts is due to [7]. Given a positive constant $\alpha \neq 1$, consider the optimization problem:

$$\text{Maximize } \frac{1}{1-\alpha}\sum_{i=1}^{n}\lambda_i^{1-\alpha} \tag{1}$$

subject to the problem's constraints (which we shall describe later). Since the utility function is concave and the constraints are linear, this defines a unique allocation which is called the α-throughput allocation. It turns out that this allocation 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 $\alpha \to \infty$.

Similarily, we can define the weighted α-throughput allocation as the unique solution of:

$$\text{Max } \frac{1}{1-\alpha}\sum_{i=1}^{n}(w_i\lambda_i)^{1-\alpha} \tag{2}$$

We notice by computing the limit of this criterion as $\alpha \to 1$ that there is no "weighted proportionnal fairness". This is a remarkable property of proportionnal fairness: multiplying the bandwidth of a connection does not modify the allocation (see [10]).

We shall consider in particular the more general case in which λ_i represents some conacve utility f_i of the assigned throughput x_i of connection i in excess with the minimum MR_i: $\lambda_i = f_i(x_i) - f_i(MR_i)$. This creates a fair assignment in which it is the utility of the throughput rather than the througput itself which is assigned fairly. When $\alpha \to 1$, (1) then represents the so called "generalized proportional fairness" (GPF) or the "Nash Bargaigning solution" (NBS), see [10] and references therein. For that case, we can rewrite (1) in the equivalent form:

$$\text{Maximize } \prod_{i=1}^{n}(f_i(x_i) - f_i(MR_i)). \tag{3}$$

The fact that utility functions are often strictly concave in the throughput is well known especially in real time applications, see [9].

B. Problem formulation

We model a network with a set V of vertices, a set E of edges on V. Consider a set I of n connections each of which requires some bandwidth within the range $[MR_i, PR_i]$. Let $s(i)$ and $d(i)$ denote the source and destination nodes of connection i.

The first general problem we focus on is of bandwidth allocation for a fixed given routing matrix:

$$\begin{gathered}\max_{x\in X}\prod_{i=1}^{n}(f_i(x_i) - f_i(MR_i)) \text{ where } X_i := \\ \{x|\forall l = 1,...,L, (Ax)_l \le (C)_l, MR_i \le x_i \le PR_i\},\end{gathered} \tag{4}$$

with L the number of links, A the routing matrix (the element $A_{i,j}$ being equal to 1 if connection j goes through link i, 0 otherwise), and C the capacity vector (C_l is the capacity of link l). $(Ax)_l \le (C)_l$ are the standard capacity constraints. We assume that the network has sufficient bandwidth to satisfy all the connections' minimum requirements i.e. $\forall \in 1..L$ we have $\sum_{i=1}^{n} a_{li}MR_i < C_l$. f_i represents a general utility related to the excess throughput assigned to connection i beyond its minimum rate. To handle a rich class of concave utilities with different degrees of concavity, we shall use a parabolic class of utilities i.e. $f_i(x) = c_i - a_i(x_i - b_i)^2$ *see* RR An alternative class originates from that of [11], given by the maximization of

$$\sum_{i=1}^{n}\frac{1}{\tau_i}\log\left(\frac{x_i}{r_i+\eta_i x_i}\right),$$

where τ_i, r_i and η_i are positive constants. This optimization function is equivalent to the maximization of

$$\prod_{i=1}^{n}\left(\frac{x_i}{r_i+\eta_i x_i}\right)^{\frac{1}{\tau_i}}.$$

Note that at this point we can suppose $\tau_i > 1$ by elevating if necessary the equation to the requested power. Obviously, this is a special case of our framework with $\alpha = 0$ and $f_i(x_i) = \left(\frac{x_i}{r_i+\eta_i x_i}\right)^{\frac{1}{\tau_i}}$. Finally, we could take it to have the form $f_i(x_i) = {x_i}^{\alpha}$ in order to obtain fair assignment allocation. This problem will be treated in details in Sections III and IV.

We shall then consider another extended framework that has more flexibility in the decisions and constraints: it includes also routing decisions assuming that the traffic of a connection can be split. Again we shall assume that the minimum demands can be satisfied and that even some extra margin is available on each link to allocate throughput in excess to the minimum requirements. We shall further introduce a criterion related to free available residual throughput: it may be desirable to provide a throughput and routing strategy assignment that maximizes some leftover capacity over links. This leftover bandwidth could then be use by some other type of best-effort traffic that do not have minimum guarantees, or it can constitute free resources available for possible arrivals of future connections. (This is in particular important in the networks in which once a throughput allocation and a route of a connection is fixed, one cannot dynamically redefine them.)

Define $\phi_i^e, e \in E$ the flow of connection i through link e, and $\kappa_e, e \in E$ the leftoever capacity on link e. We can fix the desirable leftovers κ and search for an optimal (or fair) allocation and routing strategy related to connections' throughputs. We call this the user-aware optimization:

$$\text{Maximize } \frac{1}{1-\alpha}\sum_{i\in I}{\delta_i}^{1-\alpha}, \quad \alpha \ge 0, \alpha \ne 1.$$

Alternatively, we may wish to maximize the leftover capacities in some sense, fixing some constraint on the minimum assignment for each connection. We call this the network-aware optimization:

$$\text{Maximize } \frac{1}{1-\alpha}\sum_{e\in E}\kappa_e^{1-\alpha}, \quad \alpha \ge 0, \alpha \ne 1.$$

For both problems we can write the set of constraints as follows. We constrain the network to route for each connection i, a rate of $(1+\delta_i)MR_i$ and restrain the capacity used on link e to $(1-\kappa_e)C_e$. Let $N(u)$ denote the set nodes connected to node u, and denote $r_i^{s(i)} = -r_i^{d(i)} = 1$, and $r_i^u = 0$ for u other than $s(i)$ and $d(i)$. The set of constraints fot both problems is given by

$$\begin{array}{ll} \phi_i^e \geq 0, & \forall i \in I, e \in E, \\ \delta_i \geq 0, \quad (1+\delta_i)MR_i \leq x_i \leq PR_i, & \forall i \in I, \\ x_i r_i^u = \sum_{w \in N(u)} \phi_i^{u,w}, & \forall u \in V, i \in I, \\ \sum_{i \in I} \phi_i^e \leq (1-\kappa_e)C_e, \quad \kappa_e \geq 0, & \forall e \in E. \end{array}$$

III. AN SDP FORMULATION

In this section, we propose a centralized method for solving the general fairness problem (1). It uses a unique mathematical program called semi-definite program, which can be solved in polynomial time in theory and is tractable in practice. The basic idea of SDP is to transform the original maximization problem into a minimization problem of some new variable (or more generally of a linear combination of of variables) subject to a constraint of positive semi-definiteness (psd) [1] of some general matrix P. This matrix is block diagonal, and is thus psd if and only if each block is psd. In our fairness computation, the blocks will always be of size smaller than or equal to two. The psd of the general matrix will (i) imply the capacity constraints as well as those corresponding to the min and max throughputs, (ii) allow us to replace the objective function by a single variable. It thus contains information on the structure of the objective function of the original maximization problem.

SDP involves defining new intermediate variables which are necessary for expressing the required constraints. We sketch in the next subsection some ideas in the construction of the block matrices that are related to different α's (defining the general fairness criterion). We then present a detailed construction of the SDP and in particular, the matrix P and the objective function that will be defined through a scalar product between a vector L and the variables of the SDP. For more details on SDP, see [4]. One can use public domain programs to solve SDP[2].

A. Properties of positive semi-definite matrices

Proposition III.1: Let w, y and z be three positive real numbers. Then

$$\begin{pmatrix} w & z \\ z & y \end{pmatrix} \succeq 0 \text{ if and only if } wy \geq z^2. \tag{5}$$

In particular, if one sets $z = 1$, then the relation $y \geq 1/w$ allows us to obtain constraints of the form $y \geq \sum_{i=1}^n {w_i}^{-1}$. This explains how the minimization of $\sum_{i=1}^n {w_i}^{-1}$, that appears in the general fairness problem with $\alpha = 2$ is obtained[3] through the minimization of a single variable y subject to the psd matrix constraint in (5).

Thanks to an idea of Nemirovski[8], we can also integrate the following series of functions in our model.

Proposition III.2: Let w and y be two real positive number. 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 $k \in \{0, \ldots, 2^p - 1\}$.

[1] A matrix is psd iff its eigenvalues are non-negative. In the case of a matrix of size 2 of the form : $M = \begin{pmatrix} p & r \\ r & q \end{pmatrix}$ with $p \geq 0$ or $q \geq 0$ then M is psd iff $|M| \geq 0$, i.e. $p \cdot q \geq r^2$

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

[3] $\alpha = 2$ corresponds to the harmonic fairness that is characterized by the maximizer of $(\sum_i 1/w_i)^{-1}$, or equivalently, the minimizer of $(\sum_i 1/w_i)$, where $w_i = f_i(x_i)$ (another block in the big matrix will take care of the latter equality).

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

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:

$$\left\{ \begin{array}{ll} \begin{pmatrix} y_{i-1} & y_i \\ y_i & w \end{pmatrix} \succeq 0 & \text{if } a_i = 1 \\ \begin{pmatrix} y_{i-1} & y_i \\ y_i & 1 \end{pmatrix} \succeq 0 & \text{if } a_i = 0 \end{array} \right.$$

Then, obviously, $y_i^2 \leq y_{i-1} w^{a_i}$, and if $y_1, \ldots, y_{p-1}$ are submitted to no other constraints, we have:

$$y_p \leq w^\rho w^{k/2^p}, \text{ where } \rho := \sum_{i=1}^p \frac{a_i}{2^{p+1-i}}$$

Hence the result, by setting $y_p = y$.

Next we present a simple solution for $\alpha > 1$.

Proposition III.3: Let w and y be two real positive numbers. 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 $k \in \{0, \ldots, 2^p - 1\}$.

This is used to solve the case $\alpha \in (1, 2)$.

Proof: Let z be an intermediate variable. Using proposition III.2, 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 w and y are bounded by the unique relation: $yw^\beta \geq 1$, hence the result.

Proposition III.4: Let w and y be two real positive numbers. 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 $k \in \{0, \ldots, 2^p - 1\}$.

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

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

B. Computing the NBS and GPF

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

Proposition III.5: Let y, and $w_1, \ldots, w_n$ 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 $i \in \{0, \ldots, 2^{p-k} - 1\}$ satisfying the following constraints:

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

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

C. Utility functions

Parabolic utility. We aim at generating a variable defined by $f_i(x) = c_i - a_i(x_i - b_i)^2$. Since the optimization criteria will naturally tend to increase the value of the utility, it suffices to generate some variable f_i governed by $f_i \leq c_i - a_i(x_i - b_i)^2$, which is obviously done by adding the constraint:

$$\begin{pmatrix} c_i - f_i & x_i - b_i \\ x_i - b_i & 1/a_i \end{pmatrix} \succeq 0.$$

Vojnovic/Le Boutec/Boutremans utility. Similarly, we generate some variable f_i governed by the constraint:

$$f_i \leq \left(\frac{x_i}{r_i + \eta_i x_i} \right)^{\frac{1}{\tau_i}}$$

.
Suppose $\tau_i = 1$, then one can see that:

$$\frac{x_i}{r_i + \eta_i x_i} = 1/\eta_i - r_i^2/(\eta_i(r_i + \eta_i x_i))$$

and we generate our constraint by setting:

$$\begin{pmatrix} 1/\eta_i - f_i & r_i/\sqrt{\eta_i} \\ r_i/\sqrt{\eta_i} & r_i + \eta_i x_i \end{pmatrix} \succeq 0.$$

Now when $\tau_i > 1$, we use the ideas of the previous sections using the binary decomposition of $1/\tau_i$.

D. A simple example

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

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

Element A_{ij} equals 1 if and only if connection j uses link i. In our SDP program, one adds artificial connections so that the total number of connections has the form 2^p with $p \in \mathbb{N}$. (The reason for that will follow from Step 2). As n is not of this form we need to add one extra artificial connection, so that the number of connections is now $n' = 3 + 1 = 4 = 2^2$. We suppose that this extra connection uses its own link, and therefore does not modify the NBS of our problem.

Step 1 The first four blocks of the matrix link the variables x_i with their utility. Therefore, they are of the form:

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

We have $MAT_{1,i} \succeq 0 \Leftrightarrow -\frac{g_i - c_i}{a_i} \geq (x_i - b_i)^2 \Leftrightarrow g_i \leq c_i - a_i.(x_i - b_i)^2$. Therefore, maximizing $\prod_i g_i$ will lead to maximizing $\prod_i (c_i - a_i(x_i - b_i)^2)$.

Step 2 The $n' - 1 = 3$ following matrices link the g_i variables together to obtain a single variable that SDP will maximize. These are:

$$\begin{pmatrix} g_1 & g_{12} \\ g_{12} & g_2 \end{pmatrix}, \begin{pmatrix} g_3 & g_{34} \\ g_{34} & g_4 \end{pmatrix}, \begin{pmatrix} g_{12} & g_{1234} \\ g_{1234} & g_{34} \end{pmatrix}$$

The positiveness of these matrices implies that:

$$g_1 \cdot g_2 \geq (g_{12})^2, \; g_3 \cdot g_4 \geq (g_{34})^2$$
$$g_{12} \cdot g_{34} \geq (g_{1234})^2$$

so that $(g_{1234})^4 \leq (g_{12})^2 \cdot (g_{34})^2 \leq g_1 \cdot g_2 \cdot g_3 \cdot g_4$. Then, maximizing the single variable g_{1234} will lead to the required maximization of $\prod_i (c_i - a_i(x_i - b_i)^2)$.

Step 3 We now have to incorporate the linear constraints of the problem: $(Ax)_l \leq cap_l, \; x_i \leq PR_i, \; x_i \geq MR_i$. For this purpose, we add matrices of size 1 (scalar values). SDP will assure us that they are positive (or null) values. Therefore, the constraints $(Ax)_l \leq C_l$ lead to the declaration of L matrices that are in our example:

$$cap_1 - (x_1 + x_3), \quad cap_2 - x_2,$$
$$cap_3 - (x_2 + x_3), \quad cap_4 - x_1.$$

The constraints $x_i \leq PR_i$ and $x_i \geq MR_i$ become in the SDP program 8 matrices of size one that are:

$$\begin{cases} PR_1 - x_1 \\ PR_2 - x_2 \\ PR_3 - x_3 \\ PR_4 - x_4 \end{cases} \quad \begin{cases} x_1 - MR_1 \\ x_2 - MR_2 \\ x_3 - MR_3 \\ x_4 - MR_4 \end{cases}$$

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. Whatever the value of PR_4, the solution of SDP will be PR_4. Still, it is important that we bound x_4 otherwize in the programming part it will grow without bound which may cause an error.

The values we should give to the SDP algorithm are the matrix we have just described, plus the vector of variables to minimize. As we want to maximize g_{1234}, we give a negative value to its corresponding coefficient and set $L = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1)$.

Remark III.1: In case of n connections with l links, we stress we have at most $6n - 5$ variables, $4n - 3$ blocks of size 2, and $4n + l - 4$ blocks of size 1. Therefore, although our problem is convex, it can be expressed in a simple and short way as a semi-definite program. This is a clear improvement compared to many other specific methods, since the solution can be obtained using any general SDP software[4]. Furthermore, additional conditions (for instance those linked to integer programming, or a weighted optimization on both the bandwidth assignment to connections and the link usage, or other telecommunication-specific requests, such as regulation ones) can now be introduced without further research on convex solving instability and other issues, such as convergence tests for the iterative method, or simply the maintenance of a numerical software.

IV. Practical experiments

We implemented the SDP approach using a Matlab program run on a SUN ULTRA 1 computer to obtain the NBS fair share which coincided with GPF (as we took $f_i(MR_i) = 0$). We first tested our program on the same linear network example for which we had explicit expressions for the NBS , and the results completely agreed.

We then considered the more complex network which we describe below. The computation time (including the display part) in both cases was less than twenty minutes. In both networks, all links are assumed to be bidirectional and have the same capacity (although the program allows to handle different capacities without increasing the complexity). We present two figures. The first with the set of links and nodes and the second with the set of connections and the cumulated amount of assigned bandwidth given by all paths determined by the connection. All connections had the same quadratic utility with the parameters $MR = 10$ and $PR = 80$, $T = 3$, $fPR = 200$. We took $cap = 100$ for all links. Bandwidth parameters and assignments are given in percent of full link capacity.

We considered the COST experimental network [2], depicted in Fig. 1. 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 inferior to 2.5 Gb/s). The solution, depicted in Fig. 2 involved adding extra 65 intermediate variables, and the matrix involved in the psd constraint was of size 215 (63 matrices of size 2, and 89 matrices of size 1).

Conclusion

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] S. Boyd, L. El Ghaoui, E. Feeron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. Springer, 1994.

[5] A. Maulloo F.P. Kelly and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *J. Oper. Res. Society*, 49:237–252, 1998.

[6] 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.

[7] 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.

[8] A. Nemirovski. What can be expressed via conic quadratic and semidefinite programming. *Lecture notes*, 1998.

[9] Scott Shenker. Fundammental design issues for the future internet. *IEEE JSAC*, 13(7):1176–1188, 1995.

[10] C. Touati, E. Altman, and J. Galtier. On fairness in bandwidth allocation. *INRIA Research Report*, 2001.

[11] M. Vojnovic, J. Y. Le Boudec, and C. Boutremans. Global fairness of additive-increase and multiplicative-decrease with heterogeneous round-trip times. In *Proceedings of IEEE INFOCOM'2000*, pages 1303–1312, Tel Aviv, Israel, March 2000.

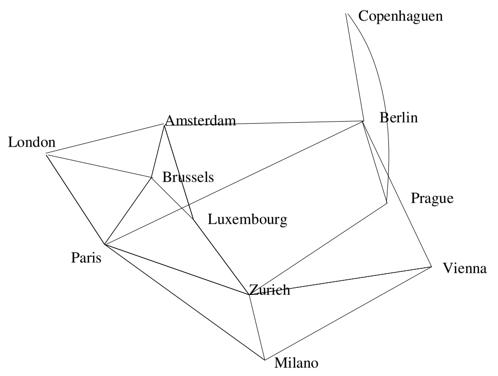

Fig. 1. COST network: links.

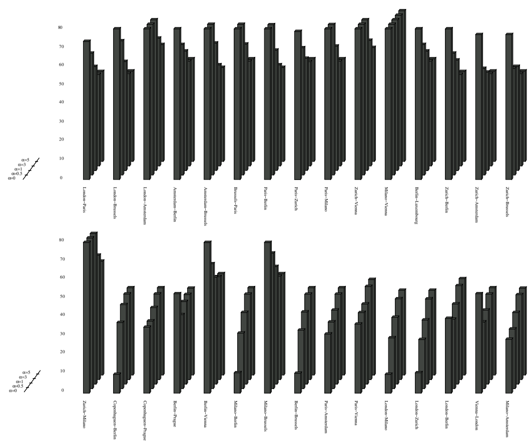

Fig. 2. COST network: solution.