A survey of optimal network congestion control for unicast and multicast transmission

Nicolas Bonmariage [a,1], Guy Leduc [a,*]

[a] *University of Liège, EECS Department, Research Unit in Networking (RUN), Belgium*

Abstract

In the last few years, there has been a large body of literature on congestion control based on optimization and control theories. This paper provides an overview of optimization flow control starting from the first original papers, and traces the development in a unified framework, from unicast to multicast, from theory to algorithms to implementation issues.

The optimal congestion control problem is formulated, both for unicast and multicast. Decentralized theoretical solutions are derived by applying duality theory. Based on these results, actual generic algorithms and implementations are proposed for solving these problems in a distributed way. Some alternative methods not based on duality theory are also reviewed. Finally the complementary problem of choosing suitable utility functions in the optimisation problem is addressed.

Key words: Congestion control, Optimization theory, Utility function, Multicast

1 Introduction

This work presents a survey of the recent applications of optimization theory to network congestion control. A growing amount of works have been dedicated to this field, giving today a strong analytical basis to the previously empirical

* Corresponding author: Guy.Leduc@ulg.ac.be, Université de Liège, Institut Montefiore, B28, B-4000 Liège, Belgium

Email addresses: `Nicolas.Bonmariage@bsb.com` (Nicolas Bonmariage), `Guy.Leduc@ulg.ac.be` (Guy Leduc).

1 Work realized as Research Fellow of the Belgian National Fund for Scientific Research and partially supported by the Belgian Science Policy in the framework of the IAP program (MOTION P5/11 project).

research on congestion control. This basis is both a source of understanding of well known protocols such as TCP and of inspiration for original proposals.

We will adopt a constructive approach by first exposing the mathematical formulation of the network congestion control problem and then developing theoretical solutions and their practical implementations. As the ground work in the field has been developed in the unicast case, we will also articulate the survey around the unicast context first and in parallel develop the various extensions relating to multicast communications, either single-rate or multirate.

The related works that we consider to be the most relevant will also be mentioned in this survey, but care will be taken to integrate them directly in the text body, in order to give a self-contained presentation of optimal congestion control.

The paper is structured as follows. Section 2 formulates the optimal congestion control problem, both for unicast and multicast. Section 3 explains how decentralized theoretical solutions can be derived in both cases by applying duality theory. Based on these results, section 4 presents generic actual algorithms and implementations for solving these problems. Section 5 reviews some alternate methods not based on duality theory. Finally section 6 addresses the complementary problem of choosing suitable utility functions in the optimisation problem. A reminder of optimisation theory is provided in annex A.

2 Optimal congestion control problem formulation

2.1 The unicast case

The basic objective of congestion control is to best exploit the available network resources while preventing sustained overload of network nodes and links. This idea can by itself be easily formulated in an optimization theory framework.

We first introduce some notation. Let $L = \{1, \ldots, L\}$ be the set of unidirectional network links. Each link $l \in L$ is characterized by its capacity c_l. We consider a set $S = \{1, \ldots, s\}$ of sources using these links. Each of these sources is characterized by a strictly concave utility function U_s which is a function of the transmission rate x_s of that source. We consider that the rate of the source must lie within some interval I_s, i.e. $x_s \in I_s = [b_s, B_s]$. We denote by S_l the set of flows using link l.

In [1], Kelly argues that bandwidth should be shared so as to maximize an objective function representing the overall utility of the flows in progress, while keeping the sum of rates on a link below that link's capacity. This gives formally the following objective function and associated constraints [2]:

$$\begin{aligned} &\max_{x_s \in I_s} \sum_{s \in S} U_s(x_s) \\ \text{subject to} &\sum_{s \in S_l} x_s \leq c_l, \ l = 1, \ldots, L \end{aligned} \tag{1}$$

A unique solution to this problem exists, since the objective function is strictly concave and the feasible set is convex (see appendix A for some theoretical background). Concave utility functions are suitable and have been used extensively for traditional elastic data services in the Internet, turning problem (1) into a convex optimization problem. However, for delay and rate sensitive services and many services in wireless networks, non concave utility functions (e.g. sigmoïdal-like "S" functions) are more realistic [3], although requiring more complex algorithms to find the global optimum of problem (1).

We now give some comments on this particular formulation. The congestion control problem is not only considered for a particular flow between two end nodes but instead at the whole network level, all the receivers being simultaneously introduced in the sum of utilities. The solution to this optimization problem is thus meant to be a global optimum for all network users. Although the congestion constraints introduced for the links seem unavoidable in the problem formulation, one might question the use of utility functions or more generally the choice of the objective function. The objective function translates in mathematical terms the actual quantity to be optimized. Using utility functions in the objective function definition allows a great flexibility in terms of problem formulation : the choice of a particular function or class of functions enables us to focus on particular aspects of the congestion control problem, while retaining the solution developed for the more general formulation based on utility functions. These aspects include bare throughput maximization and fairness.

Besides this formulation considerations, an issue to be also considered for a given optimal solution is the impact of the choice of a particular class of functions on the properties of the resulting algorithm, such as convergence and stability.

Another important aspect at this point is that, whatever solution we develop for this problem, it will imply the nodes in its implementation, as the capacities are explicitly introduced in the problem formulation.

2.2 The multicast case

The above classical formulation (1) for the unicast case can readily be generalized to the single-rate multicast case, where the considered rates are now multicast session rates.

The work in [2] has been specifically extended to the multirate multicast case in [4]. Let M denote the set of all multicast groups in the network and, for any $m \in M$, let R_m denote the set of receivers for group m. To express the link capacity constraints in the layered case, we need an expression of the flow rate of a multicast group m on a given link l, based on the choice variable x_r representing the rate associated with receiver r. If we consider a hierarchically encoded layered stream, the rate on a link upstream to a subset of receiver is the maximum of all rates reaching this subset of receivers, so that the cumulated rate of multicast group m on link l is simply $\max\limits_{r \in S_l \cap R_m} x_r$. The problem (1) can now be formulated as

$$\begin{aligned} &\max_{x_r \in I_r} \sum_{r \in R} U_r(x_r) \\ \text{subject to} &\sum_{m \in M} \max_{r \in S_l \cap R_m} x_r \leq c_l, \ \forall l \in L \end{aligned} \tag{2}$$

where $R = \bigcup\limits_{m \in M} R_m$ is the set of all multicast receivers, considering — with no loss of generality — disjoint sets of receivers.

If we assume that there exists an interior point to the set of constraints, problem (2) is feasible. If we further assume that the utility functions are strictly concave, then this solution is provably unique (see section A).

3 Decentralized solution

The various optimization problems formulated consider the congestion control problem at the network level. Although the associated objective functions are separable in the choice variables, a coupling among these variables is systematically introduced by the congestion constraints on the link, so that any solution would in the end involve synchronization and communications between all users in the network, each of them controlling the evolution of one such variable. This of course would lead to an impractical solution in a real environment.

The common key in solving these optimization problem is to use *duality theory.*

This part of optimization theory is amenable to determine decentralized and scalable solution to optimization problems where the objective function and constraints are additive, i.e. are the sum of functions depending only on one choice variable. We will now see how this can be applied in our context.

3.1 The unicast case

The basic idea behind duality theory is to construct and solve a new optimization problem — called the *dual* — of the original problem — called the *primal*. As the optimal solutions for each of them are closely related, it is possible to deduce the solution of the primal from the solution of the dual. As solving the dual can also lead to decentralized solutions while this is not the case for the primal, it is therefore possible to obtain a decentralized algorithm to solve the primal, although it is basically not separable.

The basic optimization problem (1) is precisely one example of an additive problem for which no decentralized solution can be expected by direct solving of this problem. To build its dual, we first construct the Lagrangian $L(x, p)$ of the problem, where $x = (x_s, s \in S)$ represents the vector of choice variables and $p = (p_l, l \in L)$ a vector of newly defined dual variables called the Lagrange variables. One such variable is introduced per constraint, so that the Lagrangian has in our case arity $S + L$.

$$
\begin{aligned}
L(x,p) &= \sum_{s \in S} U_s(x_s) - \sum_{l \in L} p_l \left(\sum_{s \in S_l} x_s - c_l \right) \\
&= \sum_{s \in S} \left(U_s(x_s) - x_s \sum_{l \in L_s} p_l \right) + \sum_{l \in L} p_l c_l
\end{aligned}
$$

where L_s is the set of links used by flow s.

The objective function D of the dual problem is defined by maximizing the Lagrangian on the original choice variables, resulting in a function of the only Lagrange variables:

$$D(p) = \max_{x_s \in I_s} L(x, p)$$

The dual problem consists then of minimizing the dual objective function on the dual variables:

$$\min_{p \geq 0} D(p) \tag{3}$$

In the case of the problem (1), the Lagrangian is separable in the choice variables x_s, so that the maximum of L can be expressed as a sum of maximum depending only on one choice variables. The dual objective function can thus in this case be expressed as

$$
\begin{aligned}
D(p) &= \max_{x_s \in I_s} \left(\sum_{s \in S} \left(U_s(x_s) - x_s \sum_{l \in L_s} p_l \right) + \sum_{l \in L} p_l c_l \right) \\
&= \sum_{s \in S} \underbrace{\max_{x_s \in I_s} \left(U_s(x_s) - x_s p^s \right)}_{B_s(p^s)} + \sum_{l \in L} p_l c_l, \quad \text{with } p^s = \sum_{l \in L_s} p_l
\end{aligned} \tag{4}
$$

The fundamental point to notice here is that the *evaluation* of the dual objective function D for a given vector p of dual variables can be distributed among the network users, as each of them can now perform on its own the maximization required in the calculation of each B_s term. As the only constraint for the dual problem is for the dual variables to be positive, the *solving* of the dual problem can thus be performed in a decentralized way, although this was not the case for the original primal problem. As we will see, the minimization (3) can then possibly be performed in a decentralized way too.

Suppose now the dual solution p^* is known. The most general duality result states that $F(x^*) \leq D(p^*)$, where F and x^* are the primal objective function and solution respectively. The basic problem of duality is to determine under which conditions this inequality turns out to be an equality, in which case we can say there is no duality gap. One particular case where there is no duality gap is when the objective function is concave and the constraints are linear (see appendix A), which is precisely the case for problem (1), the utility functions being supposed concave. When there is no duality gap, the solution of the primal problem are amongst the maximizers of the Lagrangian $L(x, p^*)$ where the dual variables are set to the dual solution. The utility functions appearing in the formulation of problem (1) being strictly concave, the function $L(x, p^*)$ admits a single maximizer, which is therefore the solution x^* to the primal problem. We thus have

$$
\begin{aligned}
x^* &= \arg\max_{x \in I} L(x, p^*) = \arg\max_{x \in I} \sum_{s \in S} U_s(x_s) - x_s p^{s*} && (5) \\
&= \arg \sum_{s \in S} \max_{x_s \in I_s} \left(U_s(x_s) - x_s p^{s*} \right) && (6) \\
&= \arg \sum_{s \in S} B_s(p^{s*}) && (7)
\end{aligned}
$$

where $I = (I_s, s \in S)$. The key point is that the maximization (7) can be done in a decentralized way, each source performing on its own the maximization of its associated function $B_s(p^{s*})$. This supposes of course that each source knows the dual solution p^*, which acts as synchronization signal between them.

The classical interpretation of Lagrange variables as prices in an economical framework gives some insight in the considered problem. If the dual variable

p_l is seen as the price per unit bandwidth used at link l, we see that aggregate price p^s represents then the total price per unit bandwidth source s is virtually charged for the path it uses. The resulting cost is $x_s p^s$ and we see that the local maximization performed by each source reduces to the maximization of its individual benefits. The aggregated price acts in this context as synchronization signal that aligns individual welfare with social welfare for all network sources.

Note also that at the optimal pair (x^*, p^*), the complementary slackness conditions state that $p_l > 0$ if and only if the associated constraint is active, i.e. $\sum_{s \in S_l} x_s = c_l$, meaning that the price of an uncongested link is zero.

3.2 The multicast case

Obtaining a distributed and scalable solution is of critical importance in the multicast case. Any derived solution must indeed scale not only at the multicast group level like in the single source-single receiver case, but also inside a given group.

Although the max functions appearing in the link constraints of problem (2) are non linear, the constraint set remains convex and there is thus no duality gap, the utility functions being strictly concave. Duality theory, if being applied directly, would however in this case lead to a more difficult solution precisely because of these non linearities in the constraints. This would result in a much more complex maximization of the Lagrangian. But more fundamentally, problem (2) is not separable anymore. The max functions indeed couple several variables together, making it impossible to reduce the global Lagrangian maximization to a set of local optimizations.

One way to circumvent this difficulty is to replace each max term by a set of linear constraints, which can always be carried out. A direct replacement would however lead to an exponential number of (linear) constraints. Obtaining a decentralized solution to problem (2) by means of duality theory requires in fact a reformulation of the problem. One simple way to achieve this is to replace each max term appearing in the link constraints by a separate variable representing the rate on the corresponding branch of the multicast tree. The choice variables are now the cumulated rates of the various multicast groups on each link. In the previous formulation, the receiver rates were considered. It is therefore necessary to introduce additional constraints on the feasible set of branch rates to ensure that the latter are coherent with a layered scheme, i.e. the rate on a branch cannot be greater than the rate on the parent branch but can possibly be lower if a layer is not subscribed anymore on that branch.

Before giving the alternate formulation, we introduce some more notation. We partition the set of nodes in the network in junction nodes and non junction nodes. A *junction node* is a node where one of the multicast trees branches off in two or more children. We denote $\hat{R}$ the set of all junction nodes over all multicast groups and by $\tilde{R} = R \cup \hat{R}$ the union of all junction and receiver nodes (which are assumed with no loss of generality to be logically different). We call a *branch* the set of links joining two junction nodes in a given tree and note $\hat{J}$ and J the set of branches ending respectively at a junction or receiver node. We also use $\tilde{J} = J \cup \hat{J}$. We associate a rate variable y_j with each branch $j \in \tilde{J}$ and denote by $r(j)$ the receiver or junction node associated with branch $j \in \tilde{J}$. The alternate problem formulation used to solve the optimization problem in the multilayer multicast case is then

$$\max_{y_j \in Y_j} \sum_{j \in J} U_j(y_j) \tag{8}$$

$$\text{subject to } \sum_{j \in K_l} y_j \leq c_l, \ \forall l \in L \tag{9}$$

$$y_j \leq y_{\pi(j)}, \ \forall j \in \tilde{J} \text{ s.t. } \pi(j) \neq \phi \tag{10}$$

where $K_l \subseteq \tilde{J}$ is the set of branches that share link $l \in L$ and $\pi(j)$ is the parent branch of branch j and where

$$Y_j = \begin{cases} I_{r(j)} = [b_{r(j)}, B_{r(j)}] & \text{if } j \in J \\ [0, B] & \text{if } j \in \hat{J} \end{cases}$$

with B being any number satisfying $B > \max_{r \in R} B_r$. This reformulated problem has grown in size, since we consider one choice variable per branch and not only per receiver and since we have added branch constraints to the usual link constraints.

The problem (8) is now separable, enabling us to use duality theory to obtain a decentralized solution. We note p_l, $l \in L$, the dual variables associated with the link constraints (9) and q_j, $j \in \{j' : \pi(j') \neq \emptyset\}$, the dual variables associated with the branch constraints (10), the latter variables being assumed to be identically zero for branches starting from source nodes, i.e. for branches j such that $\pi(j) = \emptyset$. We will directly give the expression of the objective function, to avoid the notational burden of its derivation, which is anyway similar to the unicast case. The dual objective function D is

$$D(p, q) = \max_{y \in Y} L(y, p, q) \tag{11}$$

$$= \sum_{j \in J} \tilde{D}_j(p, q) + \sum_{l \in L} p_l c_l \tag{12}$$

with

$$\tilde{D}_j(p,q) = \begin{cases} \max\limits_{y_j \in Y_j} \{U_j(y_j) - y_j(\tilde{p}_j + q_j)\} & \text{if } j \in J \\ \max\limits_{y_j \in Y_j} \left\{-y_j(\tilde{p}_j + q_j - \sum_{k \in C_j} q_k)\right\} & \text{if } j \in \hat{J} \end{cases} \tag{13}$$

where $\tilde{p}_j = \sum_{l \in L_j} p_l$, L_j is the set of links constituting branch j and $C_j = \{k \in \hat{J} \mid \pi(k) = j\}$ the set of children branches of branch j. As in the unicast case, we see that the evaluation of the dual objective function can be reduced to a set a distinct branch optimization problems for which the only knowledge required are the p and q prices for that branch and the q prices for the children branch. This will enable the derivation of decentralized algorithm solving indirectly the global optimization problem (2).

The interpretation of the link prices p_l is similar to the unicast case, as they are associated with the link capacity constraints: they represent the price to be paid per unit bandwidth when the associated link is congested, and remain zero while the constraint is inactive. The cumulated prices $\tilde{p}_j$ are then the corresponding branch prices.

The interpretation of the q prices gives us an important insight into problem (8). If we look at the first piece of expression (13) relative to branches ending at receiver nodes, we see that at optimality each receiver is again maximizing its individual profit, but this time the price per unit bandwidth is the sum of the price $\tilde{p}_j$ of the branch ending at that receiver and of the price q_j associated with that branch. This latter price can be seen as the price this receiver has to pay for its usage of branches located in the tree upwards branch j. In the unicast case, the price was related to the whole path to the source. This is no longer the case here, as the path has been subdivided into a set of branches describing the multicast tree.

Note however that complementary slackness conditions imply at optimality that the price q_j paid for the usage of upwards branches is zero when $y_j < y_{\pi(j)}$, i.e. when the considered branch doesn't carry all the layers which are carried by its parent. A branch has to pay for its usage of the path from the source to itself only when it actually wants to receive the cumulated rate of all layers carried by its parent. A receiver is thus not charged for an upper branch if it doesn't use the maximum bandwidth carried.

Equation (13) relative to branches ending at junction nodes can also be viewed as profit maximization, or more precisely, a cost minimization, as these nodes don't have a utility function and therefore any profit. A junction node can be thought of as being in charge of conveying to its children the layers they have subscribed to. It therefore has to pay for the resulting usage of the branch ending locally but also above in the tree. The price thus still includes the price $\tilde{p}_j$ for the branch ending at that node and the price q_j of the upper branches in the tree. But the children nodes are also charged for their usage of the tree from the source down to them, so that the price in (13) can be diminished by the corresponding amount. This is again only true for children using all the layers conveyed by their parent node, since otherwise the price for the use of the tree is zero, as a result of the slackness conditions.

We can also calculate the total profit P_m realized by a given multicast group m by summing for the receiver and junction branches the profit terms appearing in (13) inside the max terms. We have

$$P_m = \sum_{j \in J_m} U_j(y_j) - \sum_{j \in \tilde{J}_m} \tilde{p}_j y_j - \sum_{j \in \tilde{J}_m} y_j q_j + \sum_{j \in \hat{J}_m} y_j \sum_{k \in C_j} q_k \tag{14}$$

or, the variables q_j being identically zero for branches starting at the source,

$$= \sum_{j \in J_m} U_j(y_j) - \sum_{j \in \tilde{J}_m} \tilde{p}_j y_j - \underbrace{\sum_{j \in \tilde{J}_m} q_j \left(y_j - y_{\pi(j)} \right)}_{=0 \text{ (complementary slackness conditions)}} \tag{15}$$

At the optimality point, each group maximizes its profit, which is the sum of its receiver utilities diminished by the amount that the group has to pay for its branch usage in the multicast tree. Again, the price of a branch is zero when all the links constituting that branch are not saturated.

4 Duality theory based algorithms and implementations

The theoretical developments of the previous sections have pointed out the possibility to obtain decentralized solutions for global network optimization problem. We will now review actual algorithms solving these problems and how they can be implemented in a distributed way in a real network. This section focuses on the duality theory derived algorithms.

4.1 A generic algorithm for the unicast case

We saw in section 3.1 that it was possible to solve problem (1) by having each receiver calculating the maximizer of its associated B_s function — see (4)–(7) — provided the solution p^* to the dual problem (3) is known. The dual problem can be easily solved using classical gradient projection method, where the gradient of the dual objective function D is used as an update direction in the iterative calculation of each of the dual variables p_l

$$p_l(t+1) = \left[p_l(t) - \gamma \frac{\partial D}{\partial p_l}(p(t))\right]^+$$

where $\gamma > 0$ is a step size and $[z]^+ = \max(z, 0)$. If we denote by $x_s(p)$ the unique maximizer of the B_s function, i.e.

$$x_s(p) = \arg\max_{x_s \in I_s} (U_s(x_s) - x_s p) \tag{16}$$

and use it in (4), we have

$$\frac{\partial D}{\partial p_l}(p) = c_l - x^l(p)$$

where $x^l(p) = \sum_{s \in S_l} x_s(p)$ denotes the aggregate source rate at link l. We thus have finally

$$p_l(t+1) = \left[p_l(t) + \gamma(x^l(p(t)) - c_l)\right]^+ \tag{17}$$

Two important observations can be made on the update rule (17):

(1) the rule admits an immediate economical interpretation: if at time t the demand $x^l(p)$ for bandwidth at link l exceeds that link's capacity, then raise the price p_l at that link, and otherwise, decrease it
(2) the rule allows again an immediate decentralized implementation at each link, as the only information needed for the update (17) is the aggregate bandwidth at the link. And this even if the dual problem (3) was not separable in p.

This last observation suggests to involve the links in the network, together with the receivers, in the development of a completely decentralized solution. Indeed, each receiver needs to know the aggregate price p^s on its path to the source s (see (17)) in order to find the bandwidth x_s^* maximizing the primal, and each link needs to know its aggregate bandwidth x^l (in order to compute its price). The following iterative algorithm could definitely be used to solve the primal problem (1):

- for source s:
(1) receive from the network the path price $p^s(t)$ to its receiver
(2) update its transmission rate $x_s(t+1)$ by solving (16), now that the current price is known
- for link l:
(1) receive the rates from all sources going through it
(2) use the aggregate bandwidth $x^l(t)$ to update the link price p_l using (17)

It can be proved that the above algorithm generates a sequence of (x, p) pairs converging to the primal-dual optimal pair (x^*, p^*) and this under mild assumptions, i.e.

(1) the utility functions U_s are increasing, strictly concave and twice continuously differentiable on the I_s interval
(2) the curvature of U_s satisfies $0 < \frac{1}{\bar{\alpha}_s} \leq -U_s''(x_s), \forall x_s \in I_s$ and some $\bar{\alpha}_s$

The original proof and an amendment to it can be found respectively in [2] and [5].

A variation of the price update rule (17) is proposed in [6]. The authors propose there to use the Newton's method, where the gradient in the update rule is scaled by the inverse of the Hessian matrix, leading to

$$p_l(t+1) = \left[p_l(t) - \gamma\left[\nabla^2 D(p(t))\right]^{-1} \nabla D(p(t))\right]^+$$

This scaled algorithm is known to give much faster convergence, but the Hessian computation cannot be distributed to individual links, as it can be calculated to depend on rates or utilities second derivatives at other links. The authors propose to compute an approximation of the Hessian using only local information. With this approximation, the modified update rule still converges under the same assumptions and also to give much faster convergence, although at the price of larger fluctuations in the converging rate-price sequence.

Note also that this basic algorithm still converge in the asynchronous case, i.e. when the update time t differs for all the entities involved in the various calculations. Furthermore, experiments tend to prove that convergence properties remain unchanged in a slowly time varying environment.

4.2 A possible implementation

This section consider the problem of implementing the above algorithms in a real network. Each of them indeed requires explicit communication between sources and links : the sources have to communicate their rates to links on their path and the links have to communicate their prices to the sources flowing through them. These explicit communications in both directions — from sources to links and from links to sources — prevent these algorithms from being implemented directly in a real network.

Regarding the communication from sources to links, one can easily imagine that it can be in some way suppressed and the price calculation based only on local link information: the measured aggregate rate on link is a local parameter that is an obvious candidate to estimate the aggregate rate x^l at the link, which is the information needed by the algorithm.

This idea has been investigated in [7]. Let $x_{ls}(t)$ denote the input rate of source s at link l at time t and $\hat{x}^l(t) = \sum_{s \in S_l} x_{ls}(t)$ the aggregate input rate at link l. Due to packet losses and delays between the source and the link, the aggregate input rate $\hat{x}^l(t)$ is in general different from the aggregate source rate $x^l(t) = \sum_{s \in S_l} x_s(t)$ used in the link algorithm of section 4.1. We could however try to use $\hat{x}^l(t)$ in the price update rule and, as the (aggregate) buffer backlog $b_l(t)$ at link l at time t evolves according to

$$b_l(t+1) = \left[b_l(t) + \hat{x}^l(t) - c_l\right]^+$$

we would then obtain the following modified price update rule at link l

$$p_l(t) = \gamma b_l(t) \tag{18}$$

where the price is simply set as a fraction of the buffer occupancy at that link. The price is derived directly from the local buffer process without involving any explicit rate communication from sources to the link. Provided the step size γ is smaller, the modified algorithm using the update rule (18) can be shown to converge under the same assumptions about utility functions. Links can thus indirectly estimate their gradient using only local on-line measurements.

The update rule (18) does however not scale with the number of sources: the equilibrium price vector will increase and then the (proportional) buffer occupancy, leading to large feedback delays. This motivates in [8] and [9] the use of the following price update rule

$$p_l(t+1) = \left[p_l(t) + \gamma(\alpha_l b_l(t) + \hat{x}^l(t) - c_l)\right]^+ \tag{19}$$

where $\alpha_l > 0$ is a small constant possibly different at each link. At equilibrium, when $p_l(t+1) = p_l(t)$, a non bottleneck link will have $p_l^* = 0$, $b_l^* = 0$ and $\hat{x}^{l*} \leq c_l$. A bottleneck link will have its price $p_l^* > 0$ and then $\alpha_l b_l^* + \hat{x}^{l*} = c_l$ must hold. If $b_l^* > 0$, then we must have $\hat{x}^{l*} < c_l$, so that the buffer could not have been in equilibrium. This contradiction allows us to conclude that the update rule (19) will lead at equilibrium to empty buffers and full link utilization. The convergence of (19) to the optimal prices and rates is proved in [10] under mild assumptions. There, the algorithm is reformulated in terms of a continuous time system, whose dynamics are studied on the basis of a Lyapunov function, and for which global stability is proved.

We have seen it was possible to eliminate the need for explicit communications from sources to links. Communications in the reverse direction can also be suppressed by using a probabilistic packet marking mechanism at the links. Together with the update rule (19), this will lead to the Random Early Marking algorithm (REM). The probabilistic marking acts as follows. At period t, a link will mark a packet using the ECN bit in the IP header with a probability

$m_l(t)$ reflecting its current link price, i.e.

$$m_l(t) = 1 - \phi^{-p_l(t)}$$

where ϕ is a constant. This mark is then carried out to the source through the acknowledgment of the packet. As the marking probability at a link is based on a exponential function, the overall probability $m^s(t)$ for a whole path in the network will keep the same analytical form, as it is expressed by a product of probabilities

$$m^s(t) = 1 - \prod_{l \in L_s} (1 - m_l(t)) = 1 - \prod_{l \in L_s} \phi^{-p_l(t)} = 1 - \phi^{-p^s(t)}$$

An estimate $\hat{p}^s(t)$ of the aggregate price $p^s(t)$ for that path can then easily be obtained by a source using the fraction $\hat{m}^s(t)$ of marked packets

$$\hat{p}^s(t) = -\log_\phi(1 - \hat{m}^s(t)) \tag{20}$$

Experiments in [9] show that the REM algorithm quickly converges to a neighborhood of the optimum and then fluctuates around it, due to various approximations used.

Note that the price and rate update rules used respectively in the links and sources algorithms both assume zero feedback delay, in the sense that for instance a price at time $t+1$ is updated with rates values that are supposed to be the values at time t, although these are in fact values at time $t - \tau$, where τ is a propagation delay. This issue has been investigated in [11] using control system theory. The considered state equations for the (discrete) system are equations (16) and (17), the system states being the link prices. Constant propagation delays are introduced in these equations. An approximate controller is then designed in order to place the poles of the system close to the origin, therefore ensuring faster convergence to the equilibrium operating point. The mathematical developments result in a modified price update rule that averages over past prices in order to stabilize the transient behavior of the system. But it requires links to know the round trip delays of sources.

4.3 Extension to the multicast case

We will consider in this section the derivation of algorithms solving the optimization problem (2) for the multirate multicast case. Further implementation considerations will not be covered, as the techniques of the unicast case for aggregate price estimation on a path or aggregate bandwidth calculation on a link can be used immediately.

Gradient-based methods cannot be applied to the minimization of dual objective function (11) in the layered multicast case. The function is indeed non differentiable, so that its gradient may not always exists. Two approaches are then possible ([4]): either use subgradient techniques, or obtain an approximation of the original problem where the classical gradient based methods used in the unicast case can also be applied.

A subgradient is the generalization of gradient, details can be found in [12]. The gradient projection algorithm used in the unicast case can also be used if we now use the subgradient of the dual objective function as update directions for the iterative price calculation. The subgradients $\partial D(p, q)$ of function (11) are

$$\begin{aligned} \partial D \mid_{p_l} &= c_l - \sum_{j \in K_l} y_j(p, q) \\ \partial D \mid_{q_j} &= y_{\pi(j)}(p, q) - y_j(p, q) \end{aligned}$$

The price update rules in this case become

$$p_l(t+1) = \left[p_l(t) + \gamma(t) \left(\sum_{j \in K_l} \bar{y}_j(t) - c_l \right) \right]^+ \tag{21}$$

$$q_j(t+1) = \left[q_j(t) + \gamma(t) \left(\bar{y}_j(t) - \bar{y}_{\pi(j)}(t) \right) \right]^+ \tag{22}$$

where $\bar{y}(t)$ attains the maximum in (13) with prices $p_l(t)$ and $q_j(t)$ and where the step size $\gamma(t)$ has to satisfy

$$\lim_{t \to \infty} \gamma(t) = 0 \qquad \sum_{t=1}^{\infty} \gamma(t) = \infty \tag{23}$$

Note again that the $\bar{y}(t)$ can be straight calculated by each individual node through a simple one variable function maximization.

The utility functions being strictly concave, under assumption (23) and the feasibility of the initial prices, the subgradient algorithm will converge to the unique optimal solution of the primal problem.

Another way to solve the dual problem is to use a proximal approximation algorithm. A gradient projection algorithm cannot be used to solve (8) because the primal objective function is not strictly concave in the variable $y_j, j \in \hat{J}$, as the function does not even depend on these variables. So we could add a strictly concave term for each variable to the primal objective function, solve the problem by dual minimization using a gradient algorithm, and iterate in some way in order to make the modified primal function closer to the original.

In practice, for all $j \in \hat{J}$, we add to the objective function in (8) a strictly concave term $U_j(y_j) = -\frac{1}{2\kappa}(y_j - z_j)^2$, where $\kappa > 0$ is a constant and the z_j

are additional variables. The new primal problem considered has the same constraints but the objective is to solve

$$\max_{y_j} \sum_{j \in \tilde{J}} U_j(y_j) = \max_{y_j} \left(\sum_{j \in J} U_j(y_j) - \frac{1}{2\kappa} \sum_{j \in \hat{J}} (y_j - z_j)^2 \right) \tag{24}$$

The proximal approximation algorithm consists of a converging sequence of iterations. Each iteration starts by solving problem (24) using a gradient projection algorithm for the dual minimization. Then, for each $j \in \hat{J}$, z_j is set to y_j and a new iteration is started. Intuitively, the modification of the variables z_j will progressively allow to reach a greater value in (24), so that the solution will get closer to the solution of the non modified primal problem.

In terms now of algorithms, if we consider for instance the subgradient method, the following iterative calculations are performed in the multirate multicast case:

- at a link l : local update of the link price p_l according to (21)
- at a junction node j: update of the pseudo rate y_j by solving the bottom maximization in (13) and of the branch price q_j using (22)
- at a receiver : calculation of the receiver rate by solving the top maximization in (13)

As the various calculations are similar to the unicast case, we will just point out the main difference, which is the need to consider specific updates for the junction nodes between branches, and not only the links.

5 Alternate derivations of decentralized algorithms

The key challenge for the optimization-based formulations reviewed so far is the ability to derive decentralized algorithms. As we have seen, the classical way to fulfill this requirement in the optimization framework is to use duality theory. This section focuses on alternate methods which have been presented in the litterature.

5.1 Direct solving of the primal problem

A decentralized solution to the optimal congestion control problem can be directly derived from the primal problem. But as we will see, the resulting algorithm is a close variation on the corresponding solution obtained by applying duality.

Consider again the primal problem (1) for the unicast case. The following algorithm solves this optimization problem without considering the dual problem [13]:

$$x_s(t+1) = \begin{cases} [x_s(t) + \alpha_t U'(x_s(t))]_{I_s} & \text{if } \tilde{e}_s(t) = 0 \\ [x_s(t) - \beta_t \tilde{e}_s(t)]_{I_s} & \text{if } \tilde{e}_s(t) > 0 \end{cases} \tag{25}$$

where $[z]_I = \min(B, \max(b, z))$ denotes the projection of scalar z on the interval $I = [b, B]$ and where

$$\begin{aligned} \tilde{e}_s(t) &= \sum_{l \in L_s} \tilde{e}_l(t) \\ \tilde{e}_l(t) &= \mathbf{1}\left(\sum_{s \in S_l} x_s(t) > c_l\right) \\ &= \begin{cases} 0 & \text{if } \sum_{s \in S_l} x_s(t) \leq c_l \\ 1 & \text{if } \sum_{s \in S_l} x_s(t) > c_l \end{cases} \end{aligned} \tag{26}$$

The function $\tilde{e}_l(t)$ is thus a congestion indicator for link l and $\tilde{e}_s(t)$ represents the number of congested links on a particular source-receiver path. The update procedure (25) acts as before, the rate x_s of a receiver being increased in the absence of congestion and decreased otherwise, the rate of decrease being proportional to the number of congested links on the path. But this time, as the dual problem has not been considered, no dual price variables are present in the problem. The update procedure can however still be implemented in a decentralized way, each receiver updating its own rate x_s. Local optimality is still aligned with global optimality through the $\tilde{e}_s(t)$ functions, that act as synchronization signal between the receivers, replacing the dual variables of the previous approach. In this sense, the update procedure (25) remains similar to the previous one, especially in the sense that both of them use some form of synchronization signal to be able to solve the global optimization problem in a decentralized way.

The convergence of (25) is guaranteed ([13]) under mild assumptions provided the step size sequences satisfy

$$\lim_{t \to \infty} \alpha_t = 0 \qquad \sum_{t=1}^{\infty} \alpha_t = \infty$$
$$\lim_{t \to \infty} \beta_t = 0 \qquad \sum_{t=1}^{\infty} \beta_t = \infty$$
$$\lim_{t \to \infty} \frac{\alpha_t}{\beta_t} = 0$$

The increment rate in the absence of congestion has to be asymptotically smaller than the decrement rate in the presence of congestion. If the step sizes are chosen constant, it is still possible to guarantee convergence to a given neighborhood of the optimum by appropriately choosing the step sizes ratio.

As for the REM algorithm, the links have to participate in the decentralized implementation through the evaluation of the $\tilde{e}_l(t)$ function. Although this would require for a link to know the actual rates of the flows going through, it is again possible to estimate this information locally based on measurements, though no proof of convergence exists. Practical experiments ([13]) show that convergence is however not affected. Based on the value of $\tilde{e}_l(t)$, the links also have to increment some field — for instance in acknowledgment packets — in order to communicate to the source the value of $\tilde{e}_s(t)$, so that it can update the transmission rate according to (25). The algorithm for the direction of communication between links and sources cannot be thus directly implemented using a single bit. A randomized version is however still possible and the ECN bit used with REM can still be used to convey this time the number of congested links on a path.

Compared to the solution of section 4.1 that uses the dual approach, we see that links need in each case to communicate some information to sources. In the dual approach, this information, the link price, is a real-valued variable, while here the information is simply a natural number indicating the number of congested links on a path. The latter is of course easier to encode in a header field, but this is no true advantage if probabilistic mechanism are used through packet marking. The maximization involved in (7) might however be more costly than the simple derivative evaluation of (25).

Note finally that this algorithm has also been extended in [14] to the multirate multicast problem (2). The resulting algorithm is similar to the one presented in section 4.3, with update rules based on a development similar to the one presented in this section.

5.2 Use of an approximate problem

The primal problem (1) in the unicast case has also been solved in [15] using a decomposition of the problem into two simpler optimization subproblems, one for each receiver and one for the network.

Consider that each receiver (or equivalently *source* in the unicast case) s accepts to pay an amount w_s per unit time, receiving in return a proportional flow $x_s = \frac{w_s}{\lambda_s}$, where λ_s is a charge per unit flow. The receiver optimization problem is then a profit maximization problem

$$\max_{w_s \geq 0} \left(U_s \left(\frac{w_s}{\lambda_s} \right) - w_s \right) \tag{27}$$

The network optimization problem is then formulated as

$$\begin{aligned} \max_{x_s \geq 0} \sum_{s \in S} w_s \log x_s \\ \text{s.t.} \sum_{s \in S_l} x_s \leq c_l,\ l = 1, \ldots, L \end{aligned} \tag{28}$$

where the w_s solving the receiver problem are supposed to be known. Solving the network problem does not require knowledge of the users utility functions. The choice of logarithm functions gives to the optimal solution some appealing properties in terms of fairness and stability conditions.

If the utility functions are increasing, strictly concave, and continuously differentiable, then there always exists vectors $\lambda = (\lambda_s, s \in S)$, $w = (w_s, s \in S)$ and $x = (x_s, s \in S)$ solving the above problems (27) and (28) and such that the x vector is the unique solution of (1). Moreover, a vector x solves the network problem (28) if and only if the rates x_s are proportionally fair, i.e. if for any other feasible vector $\bar{x}$, we have

$$\sum_{s \in S} w_s \frac{\bar{x}_s - x_s}{x_s} \leq 0$$

In other words, if we deviate from the optimal solution x_s to another feasible solution $\bar{x}_s$, then the weighted sum of proportional changes in each user's rate is less than or equal to zero. Hence the solution is called (weighted) proportionally fair. This property is a direct consequence of the use of logarithm functions in the problem formulation.

Two different algorithms are proposed in [15] to solve the network and user optimization problems, the first by solving the primal network problem and the second by solving the dual network problem. In each case, the algorithms are presented in terms of differential equations which are considered as the description of a dynamical system. A Lyapunov function is then constructed for the system in such a way that the vector x maximizing the function is a stable point of the system to which all trajectories converge. The Lyapunov function itself can then be constructed such that maximizing the Lyapunov function can closely approximate the solution of the original network problem, and so give an approximate solution of the global optimization problem (1). The w vector is supposed to be known and at most slowly varying at the time scale considered in the solving of (28)

6 Utility functions

The optimization framework described so far is based on general utility functions for which certain properties are required in order to be able to derive convergence proofs. A complementary problem that arises when one wishes to apply this framework is the choice of particular utility functions. We will present in this section utility functions that could be used in order to mimic stationary TCP behavior, for instance to ensure fairness towards TCP flows inside the optimization framework. We will also give an example of the derivation of a general utility function based only on general requirements for the resulting congestion control algorithm.

Note that choosing different utility functions for the receivers is also a way to implement some form of traffic discrimination and to favour some receivers, provided all of them are compliant to the general algorithm.

6.1 *Utility functions and resulting resource allocation*

We have already seen in 5.2 that if the utility function is $w_s \log x_s$, leading to the objective function $\sum_{s \in S} w_s \log x_s$, then the rate allocation is proportionally fair ([1]). This objective maximizes rate allocations assuming each flow has a weighted logarithmic utility function (law of diminishing return).

If we choose $-\frac{w_s}{x_s}$ as utility function, it is shown in [16,17] that the rate allocation is minimum potential-delay fair. This objective can indeed be thought of as minimizing the sum of the file transfer delays $\frac{w_s}{x_s}$, where the file sizes are w_s.

In [18], a general class of utility functions which subsumes proportional fairness, minimum potential-delay fairness and max-min fairness is given by

$$U_s(x_s) = w_s \frac{x_s^{1-\alpha_s}}{1-\alpha_s}$$

When $\alpha_s = 2$ for all s, we get minimum potential-delay fairness. When α_s tends to 1, we get proportional fairness. Finally, when all α_s are equal and tend to ∞ with $w_s = 1$, we get max-min fairness.

6.2 *TCP-like utility functions*

In the unicast case, we know from section 4.1 that each source (or receiver) calculates an optimal rate by solving

$$\arg \max_{x_s \in I_s} \left(U_s(x_s) - x_s p^s \right) \tag{29}$$

Consider now a TCP Reno congestion algorithm in congestion avoidance phase for a given source s. If the round trip time τ_s is constant, we can approximate the source rate x_s by

$$x_s(t) = \frac{w_s(t)}{\tau_s} \tag{30}$$

where the window $w_s(t)$ is expressed in Maximum Segment Size units. If we take the link loss probabilities as price p_l for link l, we have for the path loss probability q_s of source s

$$q_s = 1 - \prod_{l \in L_s} (1 - p_l(t)) \approx \sum_{l \in L_s} p_l(t) = p^s \text{ for small } p_l\text{'s}$$

We can model the AIMD algorithm of TCP as follow ([19]). At time t, a fraction $(1 - q_s(t))$ of acknowledgments are positive, so that the window $w_s(t)$ is each time incremented by an amount of $\frac{1}{w_s(t)}$, giving a window increase rate of $x_s(t)\frac{(1-q_s(t))}{w_s(t)}$. The other acknowledgments each half the window, giving a window decrease rate of $x_s(t)q_s(t)\frac{w_s(t)}{2}$. Using (30), we can model the Reno dynamics with the following equation

$$\dot{x}_s = \frac{1 - q_s(t)}{\tau_s^2} - \frac{1}{2} q_s(t) x_s^2(t)$$

which at equilibrium reduces to

$$q_s^* = \frac{2}{2 + \tau_s^2 (x_s^*)^2} \tag{31}$$

But, in the context of our optimization framework, the equilibrium rate x_s^* also satisfies (29), so that we have

$$\begin{aligned} \frac{\partial}{\partial x} \left(U_s(x) - xp^s\right)\Big|_{x_s^*} &= 0 \\ &= \frac{\partial}{\partial x} \left(U_s(x) - xq_s\right)\Big|_{x_s^*} \end{aligned}$$

and thus we have $U_s'(x_s^*) = q_s^*$. Using (31), we can infer the following utility function for a TCP Reno source

$$U_s(x_s) = \frac{\sqrt{2}}{\tau_s} \arctan\left(\frac{\tau_s x_s}{\sqrt{2}}\right) \tag{32}$$

Note that it is also possible to model the dynamics of a RED queue in a form similar to the link price update rule (17).

Analog developments can be found in [20] for the case of the TCP Vegas algorithm. The resulting utility function is

$$U_s(x_s) = \alpha_s \tau_s \log x_s$$

where α_s is a protocol parameter.

The possible use of such a utility function in our optimization framework is straightforward, i.e. it allows us to give fairness guarantees to competing TCP flows.

6.3 Derivation of a general utility function

A very appealing process can be found in [21] where the authors specify some generic requirements for a congestion control law in terms of control theory, derive system equations and finally a utility function for the optimization framework presented here.

Consider the network as closed loop system with the following attributes:

- the reference input signal is for each source the optimal rate x_s^* solving (1)
- the feedback signal is the aggregate link price p_s^* fed back by the links with a round trip delay of τ_s

Our purpose here is to use this simple model to build a control law for this system using standard control theory objectives and then to relate the resulting law to our optimization framework in order to derive a utility function.

The main requirements for our control law are the following:

(1) at equilibirum, the (sum of) input rate signals should match the link capacities
(2) system stability should be ensured for arbitrary network delays

The first objective can be achieved introducing an integrator component in our control system: such a component indeed ensures a zero static error for a stable system. Thinking in terms of the Laplace domain, we thus introduce a $\frac{1}{s}$ factor in the loop transfer function.

Delays introduce exponential terms in the system transfer function and thus can easily lead to instabilities as network delay increases. Invariance to delay can be obtained by scaling the loop gain by the round trip delay, leading thus to a $\frac{1}{\tau}$ factor in the transfer function, where τ is the round trip delay.

The resulting loop transfer function has thus the general form

$$K\frac{e^{-\tau s}}{\tau s}$$

where K is a proportional constant part of the loop gain. This function is scale invariant and an appropriate choice for K can ensure stability for all τ.

Now the overall loop again, taking all links on a path into account, must be ensured to scale down as the number of links increase. To ensure this and still be able to derive a decentralized implementation of the control law, we must restrict ourselves to local information such as the capacity c_l of each link and the equilibrium rate x_{0s} at each source. If we sum the gains at link l, denoting by $\tau^f_{s,l}$ the forward delay between source s and link l, we obtain the following bound

$$\left| \frac{1}{c_l} \sum_{s \in S_l} x_{0s} e^{-\tau^f_{s,l} s} \right| \le \frac{1}{c_l} \sum_{s \in S_l} x_{0s} = 1$$

at equilibrium. Finally, the loop gain for source s should also obviously be scaled down by the number M_s of bottleneck on the source's path.

In summary, based on simple control requirements, we have derived dependencies for the loop gain, which can be distributed as follows between source and links:

- for source s: the gain should be $\frac{\alpha_s x_{0s}}{M_s \tau_s}$, where $\alpha_s \in (0,1)$ is a parameter
- for link l: the gain should be $\frac{1}{c_l}$

The closed loop system described above is provably linearly stable ([21]).

The actual implementation of this control scheme is straightforward for links, which just need to integrate their excess capacity for their price calculation

$$\dot{p}_l = \begin{cases} \frac{\sum_{s \in S_l} x_s - c_l}{c_l} & \text{if } p_l > 0 \\ \max\left(0, \frac{\sum_{s \in S_l} x_s - c_l}{c_l}\right) & \text{if } p_l = 0 \end{cases}$$

At equilibrium, links with non zero price will be saturated while non saturated links will have zero price.

For the source, we need to find a relation between the source rate x_s and the feedback signal, which is the aggregate price p^s. Consider the general relation $x_s = f_s(p^s)$, where f_s is a decreasing function of p^s. Linearizing this equation around an equilibrium point (x_{0s}, p_{0s}), we obtain

$$x_s = \left. \frac{\partial f_s}{\partial p^s} \right|_{p_{0s}} p^s$$

if we denote variations around the equilibrium point by the variables themselves. Identifying the above proportional constant with the expression of the source gain, we obtain

$$\frac{\partial f_s}{\partial p^s} = -\frac{\alpha_s f_s(p^s)}{M_s \tau_s}$$

where the minus sign ensures we have a negative first derivative. Solving this differential equation gives the source law

$$x_s = f_s(p^s) = x_{\max,s} e^{-\frac{\alpha_s p^s}{M_s \tau_s}} \tag{33}$$

where $x_{\max,s}$ is the maximum source rate. Thus we obtain the desired control objectives using an exponential backoff for the source as a function of the aggregate price, together with the link integrators.

We are now able to establish a bridge between the derived control law and the utility functions of our generic optimization problem. We know that we have $U_s'(x_s^*) = p_s^*$ at equilibrium. We also know from (33) that

$$p^s = -\frac{M_s \tau_s}{\alpha_s} \ln\left(\frac{x_s}{x_{\max,s}}\right)$$

giving

$$U_s(x_s) = -\frac{M_s \tau_s}{\alpha_s} \int \ln\left(\frac{x_s}{x_{\max,s}}\right) dx_s$$

and finally

$$U_s(x_s) = \frac{M_s \tau_s}{\alpha_s} x_s \left[1 - \ln\left(\frac{x_s}{x_{\max,s}}\right)\right] \tag{34}$$

Used in conjunction with the appropriate integrator equation at the links, we have an example of utility function derived only on the basis of high level requirements for the congestion control law.

6.4 Multicast utility functions

Our discussion of utility functions so far concerns unicast utility functions. Any utility function can of course be used in a multicast context, but fairness between unicast and multicast flows can be impacted by the utility functions used, as pointed out in [22]. Consider the single-rate multicast case, for which the formulation (1) applies. A typical multicast session will use many more links than would a unicast flow between the (unique) source and any given receiver. The aggregated link price for the multicast session will thus typically be higher and the resulting session rate lower at the optimum. Unicast flows will tend to be unfair to multicast sessions inside the optimization framework, so that it is reasonable to contemplate the use of a bias in the utility function in order to compensate for this.

Contrarily to the unicast case, the problem is then to define a convenient multicast session utility function, because it is not obvious to associate the session utility either with the source or the receivers. In the former case, the session function U_s in (1) can be chosen as it would be for a unicast flow. In the latter, if we suppose for simplicity that each receiver in the multicast session has the same utility function $u(x)$, then the resulting session utility function U_s will naturally be expressed in terms of the sum of utilities of receivers and then will depend on the group size R, $U_s(x_s) = Ru(x_s)$.

Having a dependency of the utility function on the group size offers a possibility to bias the optimal rate repartition in favour of the multicast flows. Indeed, the optimal rate x_s^* for session s maximizes the profit $U_s(x_s) - x_s p^{s*}$ and thus satisfies

$$\left.\frac{\partial U_s}{\partial x_s}\right|_{x_s*} = R \left.\frac{\partial u}{\partial x_s}\right|_{x_s*} = p^{s*}$$

and so we have $x_s^* = u'^{-1}(\frac{p^{s*}}{R})$. As u is concave, u' is a strictly decreasing function and so is its inverse. Thus, for a fixed session price, a larger number of receivers in the multicast group leads to a higher equilibrium session rate.

The point now is to compare in this case the two opposite bias, which are the natural bias in favour of unicast flows due to higher session prices for multicast sessions and the bias in favour of multicast sessions resulting from the use of session utility function depending on the group size. Stated otherwise, we must compare the effects of the bias associated with the higher price faced by a bigger multicast tree and of the opposite bias associated with a bigger multicast group. The unfairness will always favour multicast sessions with a larger number of receivers, and so in particular, those sessions will be unfair to unicast flows. It is however pointed out in [22] that the choice of appropriate utility functions can bound this unfairness.

7 Conclusions

We have presented an overview of current state of the art optimal congestion control in communication networks. We followed a constructive approach starting with the introduction and formal definition of the basic problem. A rigorous mathematical framework was then used to derive adequate solutions, for which a strong accent was put on the practical implementation possibilities, among others through decentralization. This reasoning was concluded with the presentation of actual algorithms calculating the optimal solutions to the original problems. At each step the unicast case was first considered and then used as a starting point for the more complex and general multilayered multicast case.

It should be emphasized that the developments presented in this article result from continuous iterations between abstract mathematical theories and concrete networking considerations. This also led us to present complementary — but essential though — works aiming at the construction of specific utility functions, in order to be able to instantiate and use in a motivated way the optimization algorithms in a real environment.

The result of these processes is a complete and rigorous framework which can be used in two complementary ways. The first one is the specification of congestion control algorithms and protocols, which differ from existing solutions in that they are essentially self contained and rely on no empirical rules for their rate regulation. The second way one can use the presented framework is to analyze existing protocols in order to get a better insight in their performance and dynamics. We illustrated this approach with the derivation of TCP-like utility functions.

Most works referred to in this paper include simulation results for the various presented algorithms. Although theoretical refinements could be investigated, the most immediate extension to these works should now be real network implementations of some optimal solutions. Measurements could then be performed in order to investigate and validate issues such as network utilization, protocol dynamics, inter and extra fairness and also traffic discrimination on the basis of specific utility functions. These real world experiments would be the ideal complement to the theoretical material which has been exposed in this article.

References

[1] F. Kelly, Charging and rate control for elastic traffic, Eur. Transactions on Telecommunications 8 (1997) 33–37.

[2] S. Low, D. Lapsley, Optimization flow control, I: Basic algorithm and convergence, IEEE/ACM Transactions on Networking 7 (6) (1999) 861–874.

[3] J.-W. Lee, R. R. Mazumdar, N. B. Shroff, Non-convexity issues for internet rate control with multi-class services: Stability and optimality, in: Proceedings of IEEE INFOCOM'2004, Vol. 23, Hong Kong, 2004, pp. 24–34.

[4] K. Kar, S. Sarkar, L. Tassiulas, Optimization based rate control for multirate multicast sessions, in: Proceedings of IEEE INFOCOM'2001, 2001, pp. 123–132.

[5] A. Karbowski, Comments on "Optimization flow control, I: Basic algorithm and convergence", IEEE/ACM Transactions on Networking 11 (2) (2003) 338–339.

[6] S. Athuraliya, S. Low, Optimization flow control with Newton-like algorithm, Journal of Telecommunication Systems 15 (3/4) (2000) 345–358.

[7] S. Low, Optimization flow control with on-line measurement, in: Proceedings of the 16th International Teletraffic Congress, Edinburgh, U.K., 1999, pp. 237–249.

[8] S. Athuraliya, S. Low, Optimization flow control, II : Implementation, preprint (May 2000).

[9] S. Athuraliya, S. Low, D. Lapsley, Random early marking, in: M. I. S. J. Crowcroft, J. Roberts (Ed.), Proceedings of the First International Workshop on Quality of future Internet Services (QofIS 2000), no. 1922 in LNCS, Springer, Berlin, Germany, 2000, pp. 43–54.

[10] F. Paganini, A global stability result in network flow control, Systems and Control Letters 46 (3) (2002) 165–72.

[11] S. Athuraliya, D. Lapsley, S. Low, An enhanced random early marking algorithm for internet flow control, in: Proceedings of IEEE INFOCOM'2000, Israel, 2000, pp. 1425–1434.

[12] D. P. Bertsekas, Nonlinear Programming, 2nd Edition, Athena Scientific, 1999.

[13] K. Kar, S. Sarkar, L. Tassiulas, A simple rate control algorithm for maximizing total user utility, in: Proceedings of IEEE INFOCOM'2001, 2001, pp. 133–141.

[14] K. Kar, S. Sarkar, L. Tassiulas, A low-overhead rate control algorithm for maximizing aggregate receiver utility for multirate multicast sessions, Tech. rep., Institute for Systems Research and University of Maryland (2000).

[15] F. Kelly, A. Maullo, D. Tan, Rate control in communication networks: shadow prices, proportional fairness and stability, Journal of the Operational Research Society 49 (1998) 237–252.

[16] L. Massoulié, J. Roberts, Bandwidth sharing: Objectives and algorithms, IEEE/ACM Transactions on Networking 10 (3) (2002) 320–328.

[17] R. Srikant, The Mathematics of Internet Congestion Control, Birkhäuser, Boston, 2004.

[18] J. Mo, J. Walrand, Fair end-to-end window-based congestion control, IEEE/ACM Transactions on Networking 8 (5) (2000) 556–567.

[19] S. Low, F. Paganini, J. Doyle, Internet congestion control: An analytical perspective, IEEE Control Systems Magazine 22 (1) (2002) 28–43.

[20] S. Low, L. Peterson, L. Wang, Understanding Vegas: a duality model, in: Proceedings of ACM Sigmetrics, Boston, MA, 2001, pp. 226–235.

[21] F. Paganini, J. Doyle, S. Low, Scalable laws for stable network congestion control, in: Proceedings of IEEE Conference on Decision and Control, Orlando, FL, 2001, pp. 185–190.

[22] J. Shapiro, D. Towsley, J. Kurose, Optimization-based congestion control for multicast communications, IEEE Communications Magazine 40 (9) (2002) 90–95.

[23] S. Kunniyur, R. Srikant, End-to-end congestion control schemes : Utility functions, random losses and ECN marks, in: Proceedings of IEEE INFOCOM'2000, 2000, pp. 1323–1332.

[24] K. Malinowski, Optimization network flow control and price coordination with with feedback : proposal of a new distributed algorithm, Computer Communications 25 (2002) 1028–1036.

[25] P. Varaiya, Lecture notes on optimization, downloadable re-issue of *Notes on Optimization*, Van Nostrand Reinhold, 1972.

A A reminder of optimization theory

This appendix gives some theoretical background on optimization theory. It is not meant to be exhaustive but rather to be a reminder for the most important results used throughout this article. The interested reader is refered to [12], on which this appendix is based.

A.1 Convex sets and functions

Many interesting theorems of optimization theory are derived in the context of convex sets and functions, that are the topic of this section. We first begin with a reminder on closed and open sets.

Definition A.1 (Closed and open sets) *A set $A \subset \Re^n$ is called* closed *if it contains all of its limit points. It is called* open *if its complement is closed. It is called* bounded *if there exists some $c \in \Re$ such that the magnitude of any coordinate of any element of A is less than c. The subset A is called* compact *if every sequence of elements of A has a subsequence that converges to an element of A. A* neighbourhood *of a vector x is an open set containing x. If $A \subset \Re^n$ and $x \in A$, we say that x is an* interior *point of A if there exists a neighbourhood of x that is contained in A. A vector $x \in A$ which is not an interior point of A is said to be a* boundary *point of A.* □

Definition A.2 (Convex set) *Let C be a subset of $\Re^n$. We say that C is* convex *if*

$$\alpha x + (1-\alpha)y \in C, \forall x, y \in C, \forall \alpha \in [0,1]$$

□

So, for a set to be convex, the linear interpolation between any two points in the set must yield a point within the set. Graphically, a line joining any two points must be enclosed in the set.

Here are some characterizations of convex sets:

Proposition A.1

(i) The intersection of convex sets is convex
(ii) The vector sum of two convex sets is a convex set
(iii) The image of a convex set under a linear transformation is convex

$\square$

Convex functions can now be defined:

Definition A.3 (Convex function) *Let C be a convex subset of $\Re^n$. A function $f : C \to \Re$ is called* convex *if*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y), \forall x, y \in C, \forall \alpha \in [0,1]$$

The function f is called concave if $-f$ is convex. $\square$

Convex functions can easily be recognized in the following cases:

Proposition A.2

(i) A linear function is convex
(ii) Any vector norm is convex
(iii) The weighted sum of convex functions (with positive weights) is convex

$\square$

The Weierstrass theorem states that a function f has at least one global minimum if it is continuous over a compact set. Stronger existence and unicity theorems exist for optimization problems involving convex sets and functions:

Proposition A.3 (Characterization of convex function extrema) *Let $f : X \to \Re$ be a convex function over the convex set X.*

(i) A local minimum of f over X is also a global minimum over X. If f is strictly convex, then there exists at most one global minimum of f.
(ii) If f is convex and the set X is open, then $\nabla f(x^*) = 0$ is a necessary and sufficient condition for a vector $x^* \in X$ to be a global minimum of f over X.

$\square$

The above theorem of course also guarantees the existence and unicity of a global *maximizer* for a *concave* function over a convex set.

A.2 Constrained optimization

A.2.1 Equality constraints

The basic problem of constrained optimization is the minimization of a function under equality constraints:

$$\begin{array}{c} \text{minimize } f(x) \\ \text{subject to } h_i(x) = 0, i = 1, \ldots, m \end{array} \tag{A.1}$$

The function f is called the *objective (or cost) function* and the functions h_i are called the *constraint functions*. We assume that $f : \Re^n \to \Re$ and $h_i : \Re^n \to \Re$ are continuously differentiable functions. The set of points verifying the constraint is called the *feasible set*.

The most fundamental result on the above problem is the Lagrange multiplier theorem:

Proposition A.4 (Lagrange theorem - Necessary conditions)
Let x^* be a local minimum of f subject to $h_i(x) = 0, i = 1, \ldots, m$, and assume that the constraint gradients $\nabla h_i(x^*)$ are linearly independent. Then there exists a unique vector $\lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)$ such that

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i^*(x^*) = 0 \tag{A.2}$$

□

The scalars λ_i are called the *Lagrange multipliers*, while the function $L : \Re^{n+m} \to \Re$ defined by

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x)$$

is called the *Lagrangian function*. The Lagrange multipliers can be shown to be the rate of change of the optimal cost as the level of constraint changes.

Equation (A.2) can be written as

$$\nabla_x L(x^*, \lambda^*) = 0, \nabla_\lambda L(x^*, \lambda^*) = 0$$

which represents a system of $n + m$ equations with $n + m$ unknowns. Every local minimum x^*, together with its associated Lagrange multiplier vector, will be a solution of this system. However, a solution of the system need not correspond to a local minimum; the following theorem gives sufficient conditions for a solution of the system to be a solution of the associated optimization problem.

Proposition A.5 (Second order sufficiency conditions) *Assume that f and h_i are twice continuously differentiable and let $x^* \in \Re^n$ and $\lambda^* \in \Re^m$ satisfy*

$$\nabla_x L(x^*, \lambda^*) = 0, \nabla_\lambda L(x^*, \lambda^*) = 0$$
$$\textit{and } y' \, \nabla^2_{xx} L(x^*, \lambda^*) \, y > 0, \forall y \neq 0 \textit{ with } \nabla h(x^*)' \, y = 0.$$

Then $x*$ is a strict local minimum of f subject to $h_i(x) = 0, i = 1, \ldots, m$ □

The resolution of the traditional unconstrained optimization problem — the simple maximization of a function — by the annulation of its first derivative can thus be transposed for the equality constrained case in the annulation of its Lagrangian. Equation (A.2) can be interpreted in the following ways:

(1) the cost gradient $\nabla f(x^*)$ belongs to the subspace spanned by the constraint gradients at x^*
(2) the cost gradient $\nabla f(x^*)$ is orthogonal to the subspace of first order feasible variations

$$V(x^*) = \{\Delta x | \nabla h_i(x^*)' \Delta x = 0, i = 1, \ldots, m\}$$

This is the subspace of variations Δx for which the vector $x = x^* + \Delta x$ satisfies the constraints $h_i(x) = 0$ up to the first order.

A.2.2 Inequality constraints

The problem (A.1) can be generalized to include inequality constraints, as stated below:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } h_i(x) = 0, \; i = 1, \ldots, m \\ &\qquad\qquad g_j(x) \leq 0, \; j = 1, \ldots r \end{aligned} \tag{A.3}$$

where again f, h_i, g_j are continuously differentiable functions from $\Re^n$ to $\Re$. The above inequality constraints can be further categorized as follows:

Definition A.4 (Active inequality constraints) *For any feasible point x, the* set of active inequality constraints *is denoted by*

$$A(x) = \{j | g_j(x) = 0\}$$

If $j \notin A(x)$, we say that the jth constraint is inactive *at x.* □

One approach to solve problem (A.3) is to reduce it to problem (A.1). Indeed, if x^* is a local minimum of (A.3), then x^* is also a local minimum for an identical problem where the inactive constraints at x^* have been discarded.

Treating active constraints as equalities, x^* is also a local minimum for the following equality constrained problem:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } h_i(x) = 0,\ i = 1, \ldots, m \\ &\qquad\qquad g_j(x) = 0,\ \forall j \in A(x^*) \end{aligned}$$

Applying the Lagrange theorem to the above problem enables the derivation of necessary conditions for problem (A.3):

Proposition A.6 (Karush-Kuhn-Tucker necessary conditions) *Let x^* be a local minimum of problem (A.3). Assume that the equality and active inequality constraint gradients $\nabla h_i(x^*)$ and $\nabla g_j(x^*)$, $j \in A(x^*)$, are linearly independent at x^*. Then there exist unique Lagrange multiplier vectors $\lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)$, $\mu^* = (\mu_1^*, \ldots, \mu_r^*)$ such that*

$$\begin{aligned} \nabla_x L(x^*, \lambda^*, \mu^*) = 0 \\ \mu_j^* \geq 0,\ j = 1, \ldots, r \\ \mu_j^* = 0\ \forall j \notin A(x^*) \end{aligned}$$

where $A(x^*)$ is the set of active constraints at x^* and the Lagrangian $L(x, \lambda, \mu)$ is

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{r} \mu_j g_j(x)$$

□

The above conditions on Lagrange multipliers for inactive constraints can also be written as follows:

Proposition A.7 (Complementary slackness condition)
Under the conditions of proposition (A.6), the optimal point x^* satisifies

$$\mu_j^* g_j(x^*) = 0,\ j = i, \ldots, r$$

□

As in the equality constrained case, sufficiency conditions can also be stated as follows:

Proposition A.8 (Second order sufficiency conditions)
Assume that f, h_i and g_j are twice continuously differentiable and let $x^* \in \Re^n$,

$\lambda^* \in \Re^m$ and $\mu^* \in \Re^r$ satisfy

$$\begin{gathered}\nabla_x L(x^*, \lambda^*, \mu^*) = 0,\ h_i(x^*) = 0,\ g_j(x^*) \le 0,\\ \mu_j^* \ge 0,\ j = 1, \dots, r,\\ \mu_j^* = 0\ \forall j \notin A(x^*),\\ y'\,\nabla^2_{xx} L(x^*, \lambda^*, \mu^*)\, y > 0,\end{gathered}$$

for all $y \neq 0$ such that

$$\nabla h_i(x^*)'\, y = 0\ \forall i = 1, \dots, m\ \nabla g_j(x^*)'\, y = 0\ \forall j \in A(x^*).$$

Also assume that

$$\mu_j^* > 0\ \forall j \in A(x^*)$$

Then $x*$ is a strict local minimum of f subject to $h_i(x) = 0$, $i = 1, \dots, m$, $g_j(x) \le 0$, $j = 1, \dots, r$ □

A.3 Duality theory

We define the *primal problem* as the following constrained optimization problem:

$$\begin{aligned}&\text{minimize } f(x)\\ &\text{subject to } x \in X,\\ &\qquad\qquad g_j(x) \le 0,\ j = 1, \dots r\end{aligned} \tag{A.4}$$

where $f : \Re^n \to \Re$, $g_j : \Re^n \to \Re$ and X is a subset of $\Re^n$. We denote by f^* the optimal value of this problem

$$f^* = \inf_{\substack{x \in X\\ g_j(x) \le 0,\ j=1,\dots,r}} f(x)$$

Note that an equality constraint can easily be formulated as a pair of inequality constraints.

We now define the *dual function* $q(\mu)$ for $\mu \in \Re^r$ as

$$q(\mu) = \inf_{x \in X} L(x, \mu)$$

where $L(x, \mu)$ is the Lagrangian function $L(x, \mu) = f(x) + \sum_{j=1}^r \mu_j g_j(x)$. The *dual problem* is

$$\begin{aligned}&\text{maximize } q(\mu)\\ &\text{subject to } \mu \ge 0\end{aligned} \tag{A.5}$$

The dual optimal value is denoted by

$$q^* = \sup_{\mu \geq 0} q(\mu)$$

The object of duality theory is the study of the relations between the primal and dual solutions. The following fundamental result shows the optimal dual value is always an underestimate of the primal value.

Proposition A.9 (Weak duality theorem) *We have*

$$q^* \leq f^*$$

□

The above relation between primal and dual optimal value leads to the following definition of the *duality gap*

Definition A.5 (Duality gap) *If $q^* = f^*$, we say there is no duality gap and if $q^* < f^*$ we say that there is a duality gap.* □

For the reminder of this section, we will use the following formal definition of the Lagrange multipliers:

Definition A.6 *A vector $\mu^* = (\mu_1^*, \ldots, \mu_r^*)$ is said to be a Lagrange multiplier vector for the primal problem if*

$$\mu_j^* \geq 0, \; j = 1, \ldots, r$$

and

$$f^* = \inf_{x \in X} L(x, \mu^*)$$

□

The following proposition demonstrate a strong connection between the Lagrange multipliers and the dual solution:

Proposition A.10

(i) If there is no duality gap, the set of Lagrange multipliers is equal to the set of optimal dual solutions
(ii) If there is a duality gap, the set of Lagrange multipliers is empty

□

Note that the dual problem can admit an optimal solution even in the presence of a duality gap.

As already stated, duality theory is concerned with the relation between primal and dual solutions. The most important results are stated below:

Proposition A.11 (Primal-dual characterization)
(x^*, μ^*) is an optimal solution-Lagrange multiplier pair if and only if

$$\begin{aligned} &x^* \in X, g_j(x^*) \leq 0,\ j = 1, \dots, r && \textit{(Primal feasability)} \\ &\mu^* \geq 0 && \textit{(Dual feasability)} \\ &x^* = \arg\min_{x \in X} L(x, \mu^*) && \textit{(Lagrangian optimality)} \\ &\mu_j^* g_j(x^*) = 0,\ j = 1, \dots, r && \textit{(Complementary slackness)} \end{aligned}$$

□

Proposition A.12 (Saddle point theorem)
(x^*, μ^*) is an optimal solution-Lagrange multiplier pair if and only if $x^* \in X$, $\mu^* \geq 0$ and (x^*, μ^*) is a saddle point of the Lagrangian, i.e.

$$L(x^*, \mu) \leq L(x^*, \mu^*) \leq L(x, \mu^*)\ \forall x \in X, \mu \geq 0$$

□

We conclude this section demonstrating two important cases where existence of a dual solution:

Proposition A.13 (Strong duality theorem) *There is no duality gap and there exists at least one Lagrange multiplier in the following cases:*

(i) the primal problem is feasible, the cost function f is convex over $\Re^n$, the constraint functions are linear, the optimal value f^* is finite and X is polyhedral

(ii) the primal problem is feasible, the cost function f and constraint function g_j are convex over X, X being a convex subset of $\Re^n$, the optimal value f^* is finite and there exists $\bar{x} \in X$ such that $g_j(\bar{x}) < 0\ \forall j = 1, \dots, r$ (interior point)

□

If the constraint set is bounded, the Weierstrass theorem shows that the primal problem also admits an optimal solution, since a convex function is also continuous.