Priority Pricing in Utility Fair Networks

Tobias Harks[*]
Zuse-Institute-Berlin
Email: harks@zib.de

Tobias Poschwatta [†]
Technische Universität Berlin
Email: posch@tkn.tu-berlin.de

Abstract

This paper deals with a new pricing approach in utility fair networks, where the user's application is associated with a utility function. We allow users to have concave as well as non-concave utility functions. Bandwidth is allocated such that utility values of applications are shared fairly. In this work, we derive a fairness measure for utility functions that takes their specific shape into account. Based on this fairness measure, we present a simple pricing mechanism: the user announces his utility function and the network charges in accordance with the fairness measure. Then, we apply our pricing mechanism to a content provider's network. In our model, customers want to scale their utilities to achieve their goals (e.g. file download, multimedia streaming) in a cost optimal way. In this regard, we formulate a download problem with predefined deadline as an optimal control problem and account for dynamic changes of the state of congestion by using (online) model predictive control techniques. Finally, we develop online control strategies and implement them in a User Agent (UA) that automatically scales the utilities.

1. Introduction

In the last years, congestion control of communication networks has been interpreted as a distributed algorithm at sources and links in order to solve a global optimization problem [6, 9–11]. Each user is associated with an increasing, strictly concave bandwidth utility function representing elastic traffic. The congestion control algorithms aim at maximizing aggregate utility subject to capacity constraints on the links. The solution to this problem is derived by decomposing the overall problem into subproblems that can be solved by links and sources using only local information. The links communicate a price based on usage measurements; the source collects the aggregate price on its path and adapts its sending rate in order to maximize its surplus. Even though considerable progress has been made in this direction, the existing work focuses only on elastic traffic, such as file transfer (FTP, HTTP) or electronic mail (SMTP). As shown in [15], some applications, especially real-time applications, have non-concave bandwidth utility functions. Several works [2, 7, 14] argue that it is an application performance measure, i.e. the utility that should be shared fairly among users. To achieve this, we have constructed in [4] a special class of concave functions, i.e. *second order utility functions* and derived a utility fair operating point as the solution of an associated optimization problem. The objective is to maximize aggregate second order utility subject to capacity constraints at the links.

In general, the resulting bandwidth allocation of a utility fair operating point strongly depends on the bandwidth utility functions that are used. In the existing literature on utility fair networks, e.g. [2, 7, 14], a proper fairness metric for utility functions is missing. In this paper, we address this issue by defining a fairness metric for bandwidth utility functions. Based on this metric, we associate *priority costs* with bandwidth utility functions and derive a new pricing framework for the interaction between the network and the users. The users announce their utility function to the network and are in turn charged priority costs depending on the fairness index of their utility function. Then, we apply our pricing mechanism to a content provider's network. A network operator offers services such as video streaming, web browsing, or file download to customers in their network. Customers want to cost optimally scale their utilities to achieve their goals (e.g. file download, multimedia streaming). We formulate the download problem as an optimal control problem and account for dynamic changes of the state of congestion by using (online) model predictive control techniques. Using Pontrjagin's Minimum Principle, we develop online control strategies and implement them in a User Agent (UA) that automatically scales the utilities. Based on different prediction approaches, we compare the resulting control strategies with respect to cost efficiency by ns-2 simulations.

The rest of the paper is organized as follows. After re-

[*]Supported by the German research funding agency 'DFG' under the graduate program 'Graduiertenkolleg 621 (MAGSI/Berlin)'

[†]Partially supported by the Ambient Networks Project, partially funded by the European Commission under its Sixth Framework Programme.

viewing the related work, and giving some background in 2, we introduce the new fairness metric for utility functions in 3. Based on this metric, we outline the proposed pricing scheme and apply the scheme to a content provider's network. Using a file-download problem in a given time interval, we demonstrate the functionality of the UA in Section 4. We also give a brief introduction to the main principles of model predictive optimal control theory. In Section 5, we show simulation results that allows us to compare different control strategies in terms of cost efficiency.

2. Background and Related Work

We model a packet switched network by a set of nodes (routers) connected by a set L of unidirectional links (output ports) with finite capacities $c = (c_l, l \in L)$. The set of links is shared by a set S of sources indexed by s. A source s represents an end-to-end connection and its route involves a subset $L(s) \subset L$ of links. Equivalently, each link is used by a subset $S(l) \subset S$ of sources. A transmission rate $x_s(t) \in X_s = [x_s^{\min}, x_s^{\max}]$ in *packets per second* is associated with each source s. A rate vector $x(t) = (x_s(t), s \in S)$ is said to be *feasible* if it satisfies the conditions: $x_s(t) \in X_s \; \forall s \in S$ and $\sum_{s \in S(l)} x_s(t) \leq c_l \; \forall l \in L$. With each link l, a scalar positive congestion-measure $p_l(t)$, called *price*, is associated. Let $y_l(t) = \sum_{s \in S(l)} x_s(t - \tau_{ls}^f)$ be the aggregate transmission rate of link l, i.e. the sum of all rates using that link in which the forward delays τ_{ls}^f between sources and links are accounted for. Let $q_s(t) = \sum_{l \in L(s)} p_l(t - \tau_{ls}^b)$ be the end-to-end congestion measure of source s, where again τ_{ls}^b are the backward delays from links to sources. The total RTT is given by $\tau_s = \tau_{ls}^f + \tau_{ls}^b$. If the transmission rate of user s is x_s, user s receives a benefit measured by the bandwidth utility $U_s(x_s)$.

2.1. Elastic Traffic

The existing work on congestion control algorithms using the utility framework is focused on elastic traffic such as TCP. Congestion control mechanisms are regarded as a distributed algorithm carried out by sources and links in order to solve a global optimization problem [6, 9–11]. The objective is to maximize aggregate source utility over transmission rates subject to capacity constraints:

$$\max_{x_s \in X_s} \sum_{s \in S} U_s(x_s) \;\; s.t. \sum_{s \in S(l)} x_s \leq c_l, \; l \in L \qquad (1)$$

Source rates can be interpreted as primal variables, congestion measures as dual variables. Using the dual approach [10], a gradient projection method to generate optimal prices is applied to the dual objective function:

$$\dot{p}_l = \begin{cases} \frac{1}{c_l}(y_l - c_l) & \text{if } p_l > 0 \\ \frac{1}{c_l}[y_l - c_l]^+ & \text{if } p_l = 0. \end{cases} \qquad (2)$$

In [12], it is shown that if the utility functions are strictly concave, (2) combined with the dynamic source law

$$\begin{aligned} \tau_s \dot{\xi}_s &= \beta_s \left(U_s'(x_s) - q_s \right) \\ x_s &= x_s^{\max} e^{\left(\xi_s - \frac{\alpha_s q_s}{M_s \tau_s} \right)} \end{aligned}$$

converges to the unique optimal solution $x_s^* = U_s'^{-1}(q_s^*)$ starting from any initial condition (M_s is an upper bound on the number of bottlenecks, α_s and β_s are positive parameters). Furthermore, this approach has the appealing property that the equilibrium is locally stable within given delay bounds [13].
Following [11], we can associate a class of concave utility functions with corresponding bandwidth fairness-criteria as follows:

$$U_s(x_s, \eta_s) = \begin{cases} -w_s \frac{x^{1-\eta_s}}{1-\eta_s}, \; \eta_s > 0, \; \eta_s \neq 1, \\ w_s \log(x_s), \; \eta_s = 1. \end{cases} \qquad (3)$$

Then, in the case $\eta_s = 1$, we have *proportional* fairness [6]. In the case $\eta_s = 2$, we have *minimum potential delay* fairness, and for $\eta_s \to \infty$, we have *max-min* fairness.

2.2. Utility Fairness

Another line of research focuses on utility fairness rather than bit-rate fairness [2, 4, 7, 14]. An equilibrium point should result in roughly equal utility values for different applications. To motivate this paradigm, let us consider a single link of capacity c shared by two users. One user transfers data according to an elastic application with strictly increasing, and concave bandwidth utility $U_1(\cdot)$. The other user transfers real-time video data with a non-concave bandwidth utility function $U_2(\cdot)$ (steps represent encoding layers). Figure 1 shows, how different bandwidth allocations

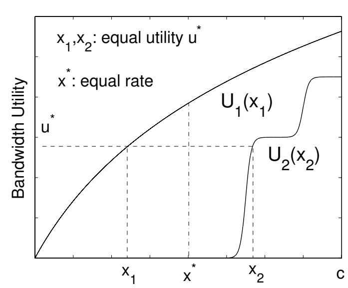


Figure 1: Utility functions with $U_1(x_1) < U_2(x_2), x \in X$

affect the received utility. If bandwidth is shared equally, what is referred to as *max-min bandwidth* allocation in this example, user 1 receives a much larger utility than user 2. Conversely, user 2 would not be satisfied since he receives a utility value zero (e.g. minimum encoding rate is

Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP'05)
0-7695-2437-8/05 $20.00 © 2005 IEEE

not achieved). If we want to share *utility* equally instead of bandwidth, we would like to have a resource allocation, where the received utilities are equal or *utility max-min fair*, i.e. $U_1(x_1) = U_2(x_2) = u^*$. Using this fairness approach, it is possible to include real-time applications with associated non-concave utility functions into a unified resource allocation framework. In [14], only mild assumptions on the feasible utility functions are required (non-decreasing, not necessarily continuous, min. bandwidth exists for a given utility value). The drawbacks of this approach are that the links have to maintain per-flow states in order to allocate bandwidth utility fair, and that there are no stability results given in the presence of communication delay.

In [4], we adopt a middle course by transforming non-concave bandwidth utility functions into strictly concave second order utility functions:

$$F_s(x_s) = \int f_s^{-1}(U_s(x_s))dx_s, \tag{4}$$

where $f_s(q_s)$ is a strictly decreasing function. Analogous to (1), we interpret an equilibrium as the solution of the optimization problem:

$$\max_{x_s \in X_s} \sum_{s \in S} F_s(x_s) \;\; s.t. \; \sum_{s \in S(l)} x_s \leq c_l, \; l \in L$$

In equilibrium, the following equation $U_s(x_s^*) = f_s(q_s^*)$ holds. The value $f_s(q_s)$ can be interpreted as the *available utility* the network can offer to source s. This approach enables us to use scalable, decentralized, and stable congestion control algorithms in the line of [9, 10, 13]. Yet, we relax the concavity assumption on the bandwidth utilities, and achieve utility fairness in equilibrium [4]. If not stated otherwise, we set in the following $f_s(q_s) := 1/q_s$. For a detailed discussion about different choices of $f_s(q_s)$ and corresponding fairness criteria we refer to [4].

We assume throughout the paper associated bandwidth utility functions for various applications, such as VoIP, video-streaming, file download or web browsing are available. As shown in [8], it is possible to generate bandwidth utility functions for various application types *online*. Thus, this assumption is by all means realistic.

2.3. Our Contribution

After we have outlined the different fairness notions of utility based networks, we want to point out some open issues. Fig. 1 shows that the actual bandwidth share of a user depends on the chosen utility function.

In recent works on utility fair networks, e.g. [2, 7, 14], a pricing mechanism that reflects the choice of utility functions is missing. In this paper, we tackle this issue by providing a regulating mechanism that takes the choice of utility functions into account. We define a fairness measure for utility functions that allows us to compare different utility functions based on the induced fairness metric. Using this metric, we outline a pricing mechanism that connects the specific course of bandwidth utility functions with priority costs. Unlike the proportional fair pricing framework of Kelly [6], in our pricing model users pay for utility functions rather than bandwdith. We believe, this is more appropriate to respect the inelastic behavior of real-time applications and serves as a framework for both traffic classes.

On the user side, we identify several scenarios where a strategic scaling of the utility functions is reasonable. In this regard, we explicitly address the issue of cost-optimally scaling the utility function in the context of a download problem, where the user wants to download a file in a given time interval. We model the download problem as an optimal control problem and analytically derive the optimal control strategy. In order to account for dynamic changes of the state of congestion, we further refine the optimal control model by model predictive control techniques. The derived control algorithms are implemented in a user agent that automatically scales the utility function. We further compare different controller types by simulations.

3. Pricing Utility Functions

In the preceding section we have presented various fairness criteria depending on the choice of utility functions. To achieve utility fairness in equilibrium, sources react to the path price $q_s = \sum_{l \in L(s)} p_l$ according to

$$\begin{aligned} \tau_s \dot{\xi}_s &= \beta_s \left(f_s^{-1}(U_s(x_s)) - q_s \right) \\ x_s &= \max \left\{ x_s^{\min}, x_s^{\max} e^{\left(\xi_s - \frac{\alpha_s q_s}{M_s \tau_s} \right)} \right\}. \end{aligned} \tag{5}$$

Using (5), in equilibrium the sending rates are given as:

$$x_s^* = U_s^{-1} \left([f_s(q_s)]_{u_s^{\min}}^{u_s^{\max}} \right), \tag{6}$$

where $[w]_a^b := \min\{\max\{w, a\}, b\}$. Since $f_s(\cdot)$ is strictly decreasing, the slower $U_s(x_s)$ is increasing the more bandwidth will user s get in equilibrium. Feasible utility functions are defined as:

Assumption 3.1. *A bandwidth utility function $U_s(x_s)$ is feasible, if it satisfies:*

(i) $U_s(x_s) \geq 0, \;\; U_s^{'}(x_s) \geq \gamma_1 > 0, \; x_s \in X_s$

(ii) $U_s(x_s^{\min}) =: u_s^{\min} \geq u_{\min}$,

(iii) $U_s(x_s^{\max}) =: u_s^{\max} \leq u_{\max}$, $u_{\min}, u_{\max} \in \mathbb{R}^+$.

Note that we do not rely on concavity.

3.1. Fair Utilities

Traditionally, bandwidth allocation is considered fair if flows get (approximately) equal shares of the available bandwidth, i.e. $x_i \approx x_j \forall i, j \in S$. In utility fair networks, this relation holds for utilities, i.e. $U_i(x_i) \approx U_j(x_j)$. However, since the state of congestion changes over time it is reasonable to consider long-term bandwidth fairness in utility fair networks. To assess long-term bandwidth fairness of utility functions we say that two utility functions $U_1(x_1)$ and $U_2(x_2)$ are bandwidth fair with respect to a probability distribution of the available utility $f_{1,2}(q)$, if the *expected* bandwidth allocations are equal, i.e. $E[x_1] = E[x_2]$. In the following, we assume that the available utility is equally distributed. Then, we can define the following fairness measure to compare different utility functions:

Definition 3.2. *The fairness measure $\delta_s(U_s, X)$ on the interval $X = [x_{\min}, x_{\max}]$ is defined as*

$$\delta_s(U_s, X) := \int_{X \cap X_s} U_s(x_s)\, dx_s + \int_{\min\{x_{\max}, x_s^{\max}\}}^{x_{\max}} u_{\max} dx_s \tag{7}$$

Then, $U_s(x_s)$ is said to be $\delta_s(U_s, X)$-fair in X.

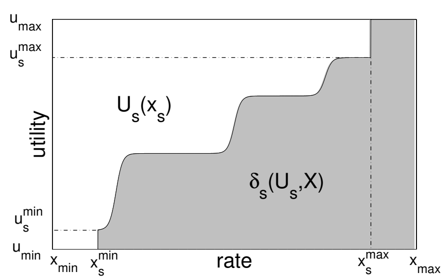


Figure 2: Fairness measure $\delta_s(U_s, X)$

See Figure 2 for a graphical depiction of the terms in this definition. This measure implies a fairness metric for utility functions: an application/user s' with fairness measure $\delta_{s'}(U_{s'}, X) < \delta_s(U_s, X)$ for its utility function will get on average more bandwidth on the interval X than user s. In the following Lemma, we formalize this relation.

Lemma 3.3. *Suppose, two users $s = 1, 2$ use the path L_p through the network. Their utility functions have fairness measures $\delta_1(U_1, X)$, and $\delta_2(U_2, X)$, $X = [x_{\min}, x_{\max}]$. Assume the path price q on L_p varies in the interval $[q_{\min}, q_{\max}]$, so that the available utility $f_s(q)$, $s = 1, 2$ is equally distributed on the interval $[u_{\min}, u_{\max}]$. Then, the following conditions hold:*

(i) If $\delta_1(U_1, X) = \delta_2(U_2, X)$, the expected bandwidth share $E[x_s], s = 1, 2$ for users $s = 1, 2$ is equal; i.e. $E[x_1] = E[x_2]$.

(ii) If $\delta_1(U_1, X) \leq \delta_2(U_2, X)$, we have $E[x_1] \geq E[x_2]$.

Proof. Using (6), and the assumption that $f_s(q_s)$ is equally distributed on $[u_{\min}, u_{\max}]$, the expected bandwidth share for user 1 on the interval $[u_{\min}, u_{\max}]$ is given as

$$E[x_1] = \int_{u_{\min}}^{u_{\max}} \frac{[U_1^{-1}(\tau)]_{x_s^{\min}}^{x_s^{\max}}}{u_{\max} - u_{\min}} d\tau.$$

Due to symmetry, we have:

$$\begin{aligned} E[x_1] &= \int_{u_{\min}}^{u_{\max}} \frac{[U_1^{-1}(\tau)]_{x_s^{\min}}^{x_s^{\max}}}{u_{\max} - u_{\min}} d\tau \\ &= (x_{\max} - x_{\min})(u_{\max} - u_{\min}) - \delta_1(U_1, X) \\ &= (x_{\max} - x_{\min})(u_{\max} - u_{\min}) - \delta_2(U_2, X) \\ &= E[x_2] \end{aligned}$$

Using $\delta_1(U_1, X) \leq \delta_2(U_2, X)$, we immediately get $E[x_1] \geq E[x_2]$. □

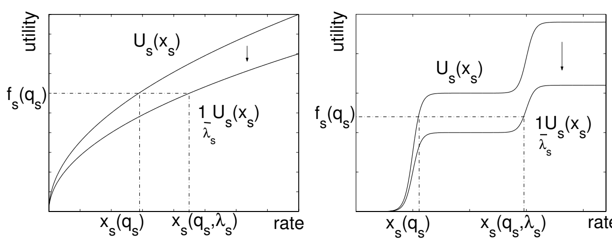
The above Lemma allows us to give an alternative definition of traditional *TCP-friendliness* in the context of utility fair networks. As shown in [9], it is possible to reverse engineer the underlying utility functions of TCP. In our alternative definition, we consider a real-time application to be TCP-friendly over a certain bandwidth interval, if the corresponding utility function has the same fairness measure as the underlying TCP utility function. The interpretation is different from the original TCP-friendliness paradigm though. Due to its inelasticity, a real-time flow may not be able to adapt the sending rate with arbitrary granularity (e.g. layered multimedia) and behave as aggressive as TCP would. The alternative definition rather indicates that an application with a TCP-fair utility function will get on *average* (light loaded network versus heavily loaded network) as much bandwidth as a TCP flow would.
A pricing approach connecting the utility function with some costs must consider that different applications operate on different bandwidth scales. A VoIP application operates on its encoding layers (8 kbit/s, 16 kbit/s), whereas the rate of an elastic file transfer is only bounded by the bottleneck link of the used path. To account for this diversity, we introduce a *reference utility function* $U_{\text{ref}}(x)$ on the interval $\mathcal{X} = [0, c_{\max}]$, where $c_{\max}$ is an upper bound of the fastest link in the network. For each interval $X \subset \mathcal{X}$ the reference fairness measure is defined as:

$$\delta_{\text{ref}}(U_{\text{ref}}, X) := \int_X U_{\text{ref}}(\tau) d\tau.$$

We assume that each user is associated with a normalized $\delta_{\text{ref}}(U_{\text{ref}}, X_s)$-fair feasible application specific *standard utility* $U_s(x_s)$ on its operating interval X_s; i.e. $\delta_s(U_s, X_s) = \delta_{\text{ref}}(U_{\text{ref}}, X_s)$.

3.2. Dynamic Scaling

The course of a standard utility function is designed to reflect the specific properties of the corresponding application (e.g. multiple step function for layered video, linear or concave utility function for elastic traffic). In this section, we extend this framework by giving users the freedom to dynamically scale their standard utility. For instance, a user running a file transfer may specify a hard deadline for the job to be accomplished. In this context, at increasing congestion a user may choose to scale down its standard utility: maintaining the level of service required to meet the deadline, at the expense of higher priority costs. A user running a video application may want to keep a high encoding rate. Hence, an appropriate down-scaling of the standard video utility function can be applied. This is illustrated in the Figures 3(a) and 3(b). In the following, we consider *scaled*



(a) Scaled elastic utility (b) Scaled real-time utility

Figure 3: Scaled Utility Functions

standard utilities, i.e. users can choose a function out of:

$$\Omega_s := \left\{ \frac{1}{\lambda_s} U_s(x_s) |\; 0 < \epsilon \le \lambda_s \le \frac{1}{\epsilon} \right\}$$

The parameter ϵ bounds the feasible scaling space and can be chosen arbitrarily small. The sending rate of user s changes to:

$$x_s^* = U_s^{-1}\left([\lambda_s f_s(q_s)]_{u_s^{\min}}^{u_s^{\max}} \right) \quad (8)$$

As defined in (7), the fairness measure of the scaled utility function becomes

$$\delta_s(X_s, \lambda_s U_s) \;=\; \int\limits_{X_s} \frac{1}{\lambda_s} U_s(x_s) dx_s = \frac{1}{\lambda_s} \delta_{\text{ref}}(U_{\text{ref}}, X_s).$$

3.3. Priority Cost

In order to prevent users from using *arbitrary* utility functions, we introduce *priority costs* depending on the *current* fairness measure. To simplify the following pricing scheme, we redefine the fairness measure as

$$\tilde{\delta}_s(X_s, \lambda_s U_s) := \frac{1}{\delta_s(X_s, \lambda_s U_s)}, s \in S$$

and set w.l.o.g. $\delta_{\text{ref}}(U_{\text{ref}}, X_s) \equiv 1$. The larger $\tilde{\delta}_s(X_s, \lambda_s U_s)$ is, the higher will be the average bandwidth share of user s. We associate with every fairness measure $\tilde{\delta}_s(X_s, \lambda_s U_s)$ a non-negative, increasing, and strictly convex cost function $C_s(\tilde{\delta}_s(X_s, \lambda_s U_s))$.

Assumption 3.4. *The cost function $C_s(\cdot)$, is assumed to be non-negative, strictly increasing and strictly convex:*

(i) $C_s(\cdot) \ge 0$, $C_s^{'}(\cdot) \ge 0$, $C_s^{''}(\cdot) \ge \gamma_2 > 0$

(ii) for a network without price-differentiation, we set $C_s(\tilde{\delta}_s(X_s, \lambda_s U_s)) = C(\tilde{\delta}_s(X_s, \lambda_s U_s))\,, s \in S.$

We call this function *priority cost function*. The priority cost function $C_s(\tilde{\delta}_s(X_s, \lambda_s U_s))$ for the scaled standard utility becomes:

$$C_s(\tilde{\delta}_s(X_s, \lambda_s U_s)) = C_s(\lambda_s / \delta_{\text{ref}}(U_{\text{ref}}, X_s)) = C_s(\lambda_s)$$

Hence, priority costs only depend on the scaling factor λ_s.

3.4. Priority Pricing Scenario

We apply our pricing approach to a content provider network. A network operator offers services such as video streaming, web browsing, or file download to customers in his network. We consider a single server from which customers can receive data streams in real-time or just download files. In our scenario, the server offers two pricing variants: fixed prices for standard utilities and priority pricing for scaled utilities. In the first variant, a fixed price is charged depending on the type of service (e.g. streaming or download) and requested data object (e.g. music file, blockbuster movie, etc.). By paying the fixed price, data transfer will be controlled by a standard normalized utility function and the download duration or streaming quality will depend on the level of congestion in the network.

In some cases, customers might request enhanced services for which they are willing to pay higher prices. This leads to our dynamic pricing approach. The quality of a service (i.e. the data transfer rate) depends on the background load imposed on the network by other users. This is reflected in the path's congestion measure q_s. Customers can control and influence their service quality by dynamically scaling their bandwidth utility function ($\frac{1}{\lambda_s} U_s(x_s)$). Due to (8) this has an immediate effect on the customer's data rate. Based on the fairness measure for the scaled utility functions, *priority costs* $C_s(\lambda_s)$ are charged.[1]

[1] In our scenario, the priority costs are charged instead of the fixed price. However, this approach could also be modified such that priority cost are charged in addition to a base price.

0-7695-2437-8/05 $20.00 © 2005 IEEE

4. Strategic Scaling by User Agents

The previously presented pricing scenario allows the scaling of utilities where users are in turn charged priority costs. From a user perspective, a crucial question remains: what is a *strategic* scaling procedure? In the following, we answer this questions in terms of cost efficiency for a download problem with predefined deadline. We model the download problem as an optimal control problem and further refine this model by using Model Predictive Control (MPC) techniques [1] that take dynamic changes of the state of congestion into account. We present two priority controllers of different complexity that try to solve the download problem with minimal cost. To carry out the derived algorithms, we propose to place an intelligent user agent (UA) on top of the application.

4.1. File Download Problem

Consider a user who wants to download a file of fixed size L in a given fixed time T. Clearly, users are assumed to be rational; they want to minimize the costs for achieving their goals. In a first step, we assume a large network scenario, i.e. the impact of the users rate x_s is negligible on the path price q_s. We introduce the state variable

$$l_s(t) = \int_0^t x_s(\tau)d\tau, \ \ 0 \leq t \leq T$$

describing the received amount of data at time t. Under the assumption that the path-price q_s lies in the interval $[q_s^{\min}, q_s^{\max}]$, the user's optimal control problem can be stated as follows:

$$\begin{aligned} &\min_{\lambda_s \in \Lambda_s} \int_0^T C_s(\lambda_s(t))dt \\ &\quad s.t.: \dot{l}_s = U_s^{-1}\left(\lambda_s f_s(q_s)\right) \\ &\quad l_s(T) = L, \ l_s(0) = 0, \ \Lambda_s = [\epsilon, \frac{1}{\epsilon}]. \end{aligned} \tag{9}$$

Solving this kind of problems with the use of Pontrjagin's Minimum Principle, the Hamiltonian and adjoint variable ρ_s have to be introduced. An introduction to optimal control theory can be found in [5]. For optimal control problems of the type considered here (Bolza type), the Hamiltonian denoted by $\mathcal{H}_s$ may be written as

$$\mathcal{H}_s := \mathcal{H}_s(l_s, \lambda_s, \rho_s) = C_s(\lambda_s) + \rho_s U_s^{-1}\left(\lambda_s f_s(q_s)\right).$$

The first order necessary conditions for a control λ_s to be optimal can be found by solving:

$$\frac{\partial \mathcal{H}_s}{\partial \lambda_s} = C_s^{'}(\lambda_s) + \rho_s \frac{\partial\left(U_s^{-1}\left(\lambda_s f_s(q_s)\right)\right)}{\partial \lambda_s} = 0. \tag{10}$$

$$\dot{\rho}_s = \frac{\partial \mathcal{H}_s}{\partial l_s} = 0 \Rightarrow \rho_s = \text{const.} \tag{11}$$

Assuming the strict Legendre-Clebsch Condition[2] $\frac{\partial^2 \mathcal{H}_s}{\partial^2 \lambda_s}(t) > 0$, a unique optimal control $\lambda_s^*(\rho_s; q_s)$ can be derived. The inequality is a short formulation for $\mathcal{H}_s$ to be positive definite. Since a closed form solution to (10) is difficult to obtain, we use the fact (11) that the adjoint function ρ_s is constant. Hence, the optimal control $\lambda_s^*(\rho_s; q_s) = \lambda_s^*$ is also constant and can be calculated by solving:

$$l_s(T) = L \Rightarrow \lambda_s^* = \frac{U_s\left(\frac{L}{T}\right)}{f_s(q_s)}$$

The optimal controller is not surprising: since q_s is assumed to be constant, and the cost function is convex, it is optimal to choose the average sending rate $x_s = \frac{L}{T}$ over the control horizon $[0, T]$. This is exactly, what the control λ_s^* does. In the following, we call this type of controller the *mean rate controller* (MRC).

4.2. Model Predictive Control Approach

In a realistic network, the congestion price q_s for user s will vary with the level of usage. Unfortunately, it is unlikely that the user has detailed information about the evolution of q_s on *arbitrary* long horizons. Nevertheless, it is reasonable to predict the evolution of q_s in a short time interval. In this context, we will use model predictive control techniques to cope with model uncertainties.

A model predictive control problem is in general formulated as solving a finite horizon open-loop optimal control problem *online* subject to system dynamics and constraints on the state and control. Based on measurements at time t, a prediction on the future dynamic behavior of the system is made. Using this prediction, a controller is determined such that a predetermined performance objective, i.e. cost functional, is optimized [1]. If there were no disturbances and no model-plant mismatches (constant q_s), assuming that the optimal control has a solution, the control can be determined *offline* and applied in the whole (in)finite horizon interval. However, this is not realistic. Due to disturbances, the real system will not behave as the predicted system. In order to incorporate some feedback mechanism, the open loop optimal control will only be implemented until new measurements of the real system are available. We assume that new measurements are updated in fixed time intervals. Using this new measurements, the process of predicting and solving the open loop optimal control problem is repeated and a new controller is applied.

The crucial assumption of problem (9) is that the congestion level q_s does not vary over time. To relax this unrealistic assumption, the user can convey the evolution of q_s and adapt

[2] This condition is fulfilled for many combinations of realistic utilities and convex cost functions, but does not hold in general.

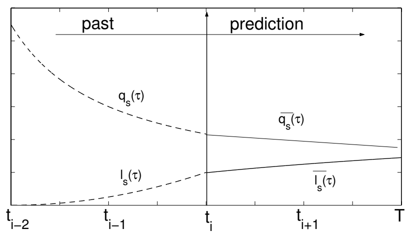


Figure 4: Principle of model predictive control

its strategy at every predefined control point $t_i = t_0 + hi$, $i = 1, ..., N$, where $h = \frac{T}{N}$ is a chosen step size. At each control point t_i, the information about the real system $l_s(t), q_s$ is updated and the next open loop optimal control problem is solved. In order to distinguish between the real system and the predicted model, we use the bar notation for the predicted internal variables $\bar{x}_s, \bar{q}_s, \bar{l}_s, \bar{\lambda}_s$. See Figure 4 for an illustration of the prediction principle. In general, the prediction $\bar{q}_s$ gives accurate approximations only locally. Hence, we restrict the optimal control problem to the horizon $[t_i, t_{i+1}]$, where we pose the terminal condition $l_s(t_{i+1}) = h\frac{L-l_s(t_i)}{T-t_i}$: the residual file length at time t_i is dived by the residual time left and multiplied by the interval length; this gives the average file length to be downloaded in the interval $[t_i, t_{i+1}]$.[3] Using the constrained state variable

$$\bar{l}_s(t) = \int_{t_i}^{t} \bar{x}_s(\tau)d\tau, \;\; t_i \leq t \leq t_{i+1},$$

we get for every time interval $[t_i, t_{i+1}],\ i = 0, ..N-1$ the open loop optimal control problem:

$$\begin{aligned} MRC_i: \; & \min_{\bar{\lambda}_s \in \Lambda_s} \int_{t_i}^{t_{i+1}} C_s(\bar{\lambda}_s(t))dt \\ s.t.: \; & \dot{\bar{l}}_s = U_s^{-1}\left(\bar{\lambda}_s f_s(\bar{q}_s)\right), \\ & \bar{l}_s(t_i) = 0, \;\; \bar{l}_s(t_{i+1}) = h\frac{L - l_s(t_i)}{T - t_i} \\ & \bar{q}_s(t_i) = \bar{q}_s, \; \Lambda_s = [\epsilon, \frac{1}{\epsilon}]. \end{aligned} \tag{12}$$

Problem (12) has the same structure as (9) and can be solved analytically. This enables us to implement the optimal control (MRC) in real-time. Simulation results for the MRC controller are given in the last section.

[3] If more accurate knowledge of the evolution of q_s is available, a more efficient scheduling of the amount of data per interval that is to be downloaded can be designed.

4.3. Linear Predicting Control

To account for model uncertainty, we relax the assumption that the congestion price q_s is constant over the control horizon. We approximate the local evolution of q_s by a linear function that leads to the differential equation:

$$\dot{\bar{q}}_s(t) = \zeta_s^i, \; t \in [t_i, t_{i+1}], \text{ for some } \zeta_s^i \in \mathbb{R}.$$

This equation is aimed to predict the behavior of q_s over the interval $[t_i, t_{i+1}]$. We get for every time interval $[t_i, t_{i+1}],\ i = 0, .., N-1$ the open loop optimal control problem:

$$LPC_i \qquad \min_{\bar{\lambda}_s \in \Lambda_s} \int_{t_i}^{t_{i+1}} C_s(\bar{\lambda}_s(t))dt \tag{13}$$

$$s.t.: \qquad \dot{\bar{l}}_s = U_s^{-1}\left(\bar{\lambda}_s f_s(\bar{q}_s)\right), \tag{14}$$

$$\dot{\bar{q}}_s = \zeta_s^i, \zeta_s^i \in \mathbb{R} \tag{15}$$

$$\bar{l}_s(t_i) = 0, \;\; \bar{l}_s(t_{i+1}) = h\frac{L-l_s(t_i)}{T-t_i} \tag{16}$$

$$\bar{q}_s(t_i) = q_s(t_i), \; \Lambda_s = [\epsilon, \tfrac{1}{\epsilon}]. \tag{17}$$

Using the Minimum Principle, we solve problem instance LPC_i analytically. Let ρ_s, ϱ_s be the adjoint variables to the differential equations (14),(15). Then, the Hamiltonian is given by $\mathcal{H}_s = C_s(\bar{\lambda}_s) + \rho_s U_s^{-1}\left(\bar{\lambda}_s f_s(\bar{q}_s)\right) + \varrho_s \zeta_s^i$. The Minimum Principle gives the necessary optimality conditions:

$$\begin{aligned} \frac{\partial \mathcal{H}_s}{\partial \bar{\lambda}_s} &= C_s'(\bar{\lambda}_s) + \rho_s \frac{\partial \left(U_s^{-1}\left(\bar{\lambda}_s f_s(\bar{q}_s)\right)\right)}{\partial \bar{\lambda}_s} = 0. \\ \dot{\rho_s} &= -\frac{\partial \mathcal{H}_s}{\partial \bar{l}_s} = 0 \Rightarrow \rho_s = \bar{\rho}_s = \text{const.} \end{aligned}$$

The boundary value problem (15),(17) can be solved in $[t_i, t_{i+1}]$:

$$\bar{q}_s(t) = q_s(t_i) + (t - t_i)\zeta_s^i, \; t \in [t_i, t_{i+1}]. \tag{18}$$

If $\mathcal{H}_s(t)$ is positive definite, i.e. $\frac{\partial^2 \mathcal{H}_s}{\partial^2 \bar{\lambda}_s}(t) > 0$, then a unique optimal control $\bar{\lambda}_s^*$ can be derived. We call this controller *linear predicting control* (LPC).

4.4. Example

We consider a data application s with a linear bandwidth utility function $U_s(x_s) = k_s x_s,\ k_s > 0$. The transformation function is given by $f_s(q_s) = \frac{1}{q_s}$, and the priority cost function is given by $C_s(\lambda_s) = \frac{1}{2}\lambda_s^2$. Using (4), the second order utility has the form

$$F_s(x_s) = \int \frac{1}{k_s x_s} dx_s = \frac{\log(x_s)}{k_s}.$$

The final time is T, and the file length L. The user predicts the path price by (15). The dynamics in (14) is given by:

$\dot{\bar{l}}_s(t) = \frac{\bar{\lambda}_s(t)}{k_s \bar{q}_s(t)}$. Thus, at every control point t_i measurements are updated and the optimal control problem (13)–(17) must be solved. Using the Minimum Principle, we obtain:

$$\frac{\partial \mathcal{H}_s}{\partial \bar{\lambda}_s} = \bar{\lambda}_s + \frac{\rho_s}{\bar{q}_s} = 0 \Rightarrow \bar{\lambda}_s^* = -\frac{\rho_s}{\bar{q}_s}$$

The sufficient Legendre-Clebsch condition for a unique optimum is fulfilled: $\frac{\partial^2 \mathcal{H}_s}{\partial^2 \lambda_s}(t) = 1 > 0$. $\bar{q}_s(t)$ is given by (18) and the optimal controller is then given by

$$\bar{\lambda}_s^*(\rho_s) = -\frac{\rho_s}{k_s(q_s(t_i) + (t - t_i)\zeta_s^i)}.$$

Since ρ_s is constant, we can eliminate ρ_s using the terminal constraint $\bar{l}_s(t_{i+1}) = h\frac{L - l_s(t_i)}{T - t_i}$.

5. Simulations

In this section we present simulation results of the mean-rate (MRC) and linear-predicting controllers (LPC). The simulation setup models the content provider scenario described earlier in Section 4 in which the provider offers download and streaming services to the customers in his network. In our setup, up to 60 customers perform downloads, one of them using either the MRC or the LPC in order to download a 215MB file in 30 minutes with minimum cost. We compare the two deadline controller variants with respect to to a cost function. Further, we compare the controllers with an optimal control strategy that has full a priori knowledge of the congestion measure.

5.1. Scenario

We consider a single network, where the network operator is both access and content provider. In our setup there is a single server S from which up to 60 users download files. Figure 5 shows the network topology. Downloads of customers start and end at random times (Figure 7(a)). The customer that requests a deadline from the server starts the download after 100 seconds with a deadline of 30 minutes (1800 seconds). Depending on the type of download, the server assigns one of three δ-fair utility functions to each connection.[4] For real-time video streaming, we define a multi-layer bandwidth utility function $U_r(x)$ with three discrete data rates. A linear bandwidth utility function $U_l(x)$ is used for downloads that want to use a priority controller.Six customers will use the real-time, another six the linear utility functions, one of them using a controller. The remaining downloads are assigned a concave bandwidth utility function $U_e(x)$. We set the packet size to 1500 bytes and measure sending rates x_s and link capacities c_l in packets per second (p/s). Each link in the network provides feedback

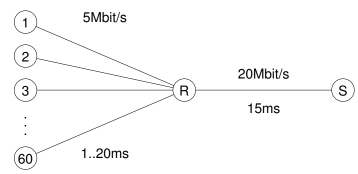


Figure 5: Simulated network topology with one server and up to 60 users

according to the link algorithm (2), where the link capacity c_l is pre-configured. The load y_l is measured by counting the number of packet arrivals in a constant time interval. The individual prices of each link on a path between server and user are summed in a double precision floating point header field in each packet. When a packet arrives at the receiver, it contains the path's congestion measure $q_s = \sum_{l \in L(s)} p_l$ which the receiver returns to the sender in acknowledgment packets. Note that real-world implementations have to make a trade-off between feedback precision (i.e. number of bits), per-packet overhead and computational complexity (floating point vs integer).

5.2. Utility Functions

We construct three δ-fair utility functions with different shapes: (i) a concave shaped function $U_e(x)$ for elastic traffic and arbitrarily high bandwidths; (ii) a linear function $U_l(x)$ for limited bandwidth but better performance (compared to the elastic function) in lower bandwidth regions; and (iii) a multiple step function $U_r(x)$ for layered video streaming up to 1.5Mbit/s.[5] We chose $U_e(x) = k_e\sqrt{x}$ and set $x_e^{\max}$ to $20\text{Mbit/s} \approx 1666\text{p/s}$ which is the capacity of the fastest link in our setup. Using $U_e(x)$ as reference, we also define

$$\delta_{\text{ref}}(U_e, X_e) = \int_0^{x_e^{\max}} U_e(x)dx = \frac{2}{3}k_e x_e^{\max\frac{3}{2}}$$

For the linear utility function $U_l(x) = k_l x$, we set $x_l^{\max}$ to $5\text{Mbit/s} \approx 416\text{p/s}$ which is the speed of the users' access links. To achieve $\delta_{\text{ref}}(U_e, X_l)$-fairness, we chose $k_l = 0.027$ which satisfies the fairness criterion:

$$\delta_{\text{ref}}(U_e, X_l) \leq \int_0^{x_l^{\max}} k_l x dx = \frac{1}{2}k_l x_l^{\max 2}$$

The real-time utility function represents a layered-video coding scheme in which a base layer at 0.5Mbit/s is the

[4] We assume that every sender uses the utility fair congestion control framework, i.e. every user is collaborative. For a detailed discussion about equilibrium properties of heterogeneous protocols (if users react to different congestion signals, e.g. packet loss, delay, or explicit prices), we refer to [16].

[5] We violate the minimum slope assumption for feasible utilities but due to the presence of elastic flows the possible damage to convergence, and stability is negligible.

minimum encoding rate supported. In addition to the base layer, we define two enhancement layers at $c_{r,2} = 1.0\text{Mbit/s} \approx 84\text{p/s}$, and $c_{r,3} = 1.5\text{Mbit/s} \approx 125\text{p/s}$ that is also the maximum data rate $x_r^{\max}$. Before we construct the $\delta_{\text{ref}}(U_e, X_r)$-fair real-time utility function, we specify the relative step-sizes for the different encoding rates. We set

$$U_r(x) = \begin{cases} 0 & \text{if } 0 \le x < c_{r,1} \\ 2k_r & \text{if } c_{r,1} \le x < c_{r,2} \\ 3.5k_r & \text{if } c_{r,2} \le x < c_{r,3} \\ \infty & \text{if } c_{r,3} = x \end{cases}$$

The area under $U_r(x)$ for $0 \le x < c_{r,3}$ is simply $\delta_r = 2k_r(c_{r,2} - c_{r,1}) + 3.5k_r(c_{r,3} - c_{r,2}) = 225.5k_r$. With $\delta_{\text{ref}}(U_e, X_r) = 372.68$, we get the coefficient $k_r = \delta_{\text{ref}}(U_e, X_r)/\delta_r(U_r, X_r) = 372.68/225.5 \approx 1.66$ so that $U_r(x)$ becomes $\delta_{\text{ref}}(U_e, X_r)$-fair. The resulting three δ_{ref}-

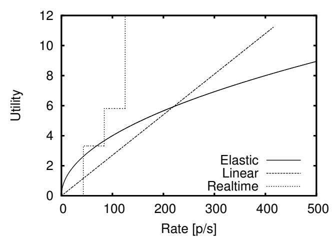


Figure 6: Utility functions

fair utility functions are shown in Figure 6. Alternatively, we could have used the linear utility function as reference. In this case, k_e and k_l as given would remain δ-fair when calculated for $x_e^{\max} = x_l^{\max} = 5\text{Mbit/s}$. For the real-time utility function, the coefficient $k_r = \delta_{\text{ref}}(x_r^{\max})/\delta_r \approx 1.66$ would have become $k_r \approx 0.94$, which is more aggressive.

5.3. Priority Costs

The total priority cost of a download is defined as:

$$\text{Total cost} = \int_0^T \frac{1}{2}\lambda_s^2(t)dt$$

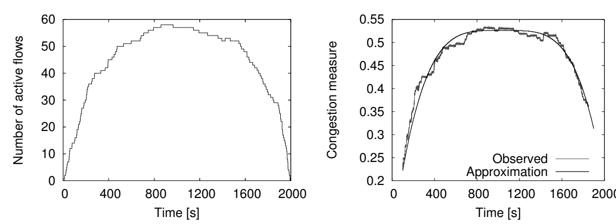
We performed simulations of MRC and LPC with control intervals between 1 and 600 seconds. Table 1 shows the used priority costs for increasing control intervals. Since MRC is a simplification of LPC, it requires higher costs due to larger prediction errors. As the table indicates, the savings of LPC depend on the chosen control interval. In our setting, the savings increase with the interval length. With longer intervals, the long-term trends in the congestion state are better captured by LPC. With shorter intervals, MRC can follow q_s more closely, so the advantage of LPC decreases.

$t_{i+1} - t_i$	Cost MRC	Cost LPC	Saving with LPC [%]
1	955.3	954.9	0.1
5	958.3	955.7	0.3
15	964.8	957.6	0.7
30	973.9	960.4	1.4
60	990.9	965.7	2.5
120	1021.5	977.8	4.3
180	1048.4	989.4	5.6
240	1062.1	993.0	6.5
300	1096.7	1013.3	7.6
360	1113.6	1021.7	8.3
420	1103.5	1009.9	8.5
480	1140.2	1028.9	9.8
600	1113.3	1047.9	5.8

Table 1: Costs of mean-rate and linear-predictive control

5.4. Comparison with an Optimal Control

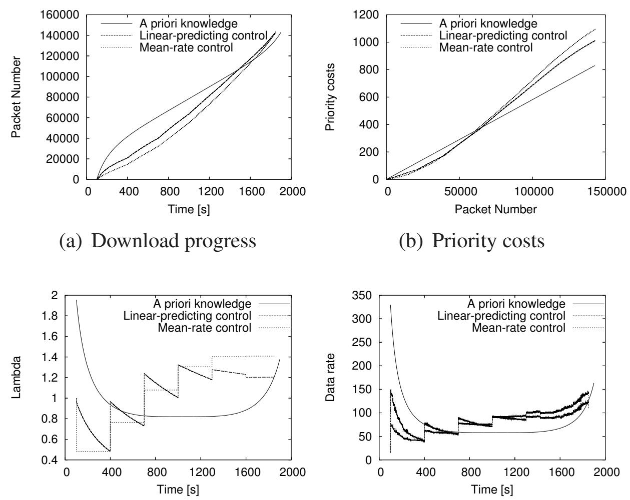
Now, we compare the two controllers with an almost optimal controller that has perfect a priori knowledge of $q_s(t)$. We approximate the evolution of $q(t)$ by a polynomial of 4th order (Figure 7(b)). Due to the complexity of an analytic solution, we solved the optimal control problem numerically using an interior point method based on the solver LOQO [3, 17].



(a) Number of active flows (b) Congestion measure

Figure 7: Number of active flows and observed congestion measure (LPC with 300 seconds control interval) and approximation used for optimal control

As can be seen in Figure 8(a), the optimal controller uses its knowledge to buy priority and download data aggressively at the beginning and at the end of the download, when congestion is low. Figure 8(b) compares the cost evolution with respect to the download progress. Interestingly, the optimal control distributes priority costs equally among packets, as can be seen from the linear increase of cost. Figure 8(c) shows the priority allocation of each controller. It can be seen that priority allocation of LPC in sections approximates the behavior of the optimal controller. MRC repeatedly underestimates q in the first half and thus needs



(a) Download progress (b) Priority costs

(c) λ_s (d) Data rates

Figure 8: Comparison of MRC, LPC and optimal control (300 s control interval)

higher priority in the second half. Finally, the data rates resulting from MRC, LPC and the optimal control are shown in Figure 8(d). Again, the data rate of LPC approximates the optimal behavior, while MRC has to catch up an increasing lag due to the under-prediction of the congestion state.

6. Conclusion

The shape of bandwidth utility functions in utility fair networks has an immediate impact on the resulting bandwidth allocation. In this work, we have provided a fairness measure that allows on the one hand to assess the aggressiveness of different application and gives on the other hand a useful tool to design utility functions.

To connect the choice of utility functions with a pricing mechanism, we associated priority costs with the utility function's fairness measure. The customer announces its utility function (fairness measure) and the network provider charges priority costs accordingly. We embedded our pricing mechanism in a content provider scenario and presented a sample customer strategy, where the customer wants to cost-optimally download a file in a given time interval. We derived different strategies by modeling the download problem as an optimal control problem. By taking system uncertainties into account, the optimal control problem becomes an *online* optimization problem. We addressed this *online* aspect by using model predictive control techniques. Using ns-2 simulations, we compared different control strategies with an adversary optimal control that has full knowledge about the system. The results of our simulation scenario (prediction horizon below 600s) show, that our linear predictive control achieves the goal while maintaining efficiency loss below 26% with respect to the optimal solution. An open problem is to evaluate the performance of our controller in *arbitrary* scenarios, e.g. when the offered load to the network changes rapidly. We currently work on dynamic redesigns of the prediction horizon depending on the state of congestion in the network. Another issue is to address the stability of the system, when many users dynamically scale their utilities to achieve certain goals.

References

[1] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, London, 1999.

[2] Z. Cao and E. Zegura. Utility max-min: An application-oriented bandwidth allocation scheme. In *Proceedings of IEEE INFOCOM'99*, pages 793–801, 1999.

[3] J. Czyzyk, M. Mesnier, and J. More. The neos server. *IEEE Journal on Computational Science and Engineering*, 5:68–75, 1998.

[4] T. Harks. Utility Proportional Fair Resource Allocation: An Optimization Oriented Approach. In *Proceedings of QoS in Multiservice IP Networks*, pages 61–74, Catania, Italy, February 2005. Springer.

[5] M. Hestenes. *Calculus of Variations and Optimal Control Theory*. John Wiley, New York, 1966.

[6] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness, and Stability. *Journal of the Operational Research Society*, 49:237–52, 1998.

[7] R. F. Liao and T. Campbell. A utility-based approach for quantitative adaption in wireless packet networks. *Wireless Networks*, 7(5):541–557, 2001.

[8] R. R.-F. Liao, P. Bouklee, and A. Campbell. Online Generation of Bandwidth Utility Function for Digital Video. In *Proceedings of Packet Video'99*, New York City, 1999.

[9] S. Low, F. Paganini, and J. C. Doyle. Internet Congestion Control. *IEEE Control Systems Magazine*, 22, 2002.

[10] S. H. Low and D. E. Lapsley. Optimization Flow Control I. *IEEE/ACM Trans. on Networking*, 7(6):861–874, 1999.

[11] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. on Networking*, 8(5):556–567, Oktober 2000.

[12] F. Paganini. A global stability result in network flow control. *Systems and Control Letters*, 46:165–172, 2002.

[13] F. Paganini, Z. Wang, J. Doyle, and S. Low. Congestion control for high performance, stability and fairness in general networks. *IEEE/ACM Trans. on Networking*, 13(1):43–56, February 2005.

[14] S. Sarkar and L. Tassiulas. Fair allocation of utilities in multirate multicast networks: A framework for unifying diverse fairness objectives. *IEEE Trans. on Automatic Control*, 47(6):931–944, June 2002.

[15] S. Shenker. Fundamental Design Issues for the Future Internet. *IEEE JSAC*, 13:1176–88, 1995.

[16] A. Tang, J. Wang, S. Low, and M. Chiang. Network Equilibrium of heterogeneous congestion control protocols. In *Proceedings of IEEE Infocom*, Miami, FL, March 2005.

[17] R. Vanderbei. Loqo user's manual - version 4.05. Princeton University, 2000.