Chapter 3

Pricing for elastic services

3.1 Overview

In future communication networks there are expected to be applications that are able to modify their data transfer rates according to the available bandwidth within the network. Traffic from such applications is termed elastic [96]; a typical current example is TCP traffic over the Internet [69], and future examples may include the controlled-load service of the Internet Engineering Task Force [105] and the Available Bit Rate transfer capability of ATM (asynchronous transfer mode) networks [13].

The key issue we address in this chapter concerns how the available bandwidth within the network should be shared between competing streams of elastic traffic. Traditionally stability has been considered an engineering issue, requiring an analysis of randomness and feedback operating on fast time-scales, while fairness has been considered an economic issue, involving static comparisons of utility. In future networks the intelligence embedded in end-systems, acting on behalf of human users, is likely to lessen the distinction between engineering and economic issues and increase the importance of an interdisciplinary view.

There is a substantial literature on rate control algorithms, recently reviewed by Hernandez-Valencia et al. [67]. Key early papers of Jacobson [69] and Chiu and Jain [40] identified the advantages of adaptive schemes that either increase flows linearly or decrease flows multiplicatively, depending on the absence or presence of congestion. Important recent papers of Bolot and Shankar [31], Fendick et al. [52] and Bonomi et al. [34] have analysed the stability of networks with a single bottleneck resource, where congestion is signalled by the build-up of a queue at the bottleneck’s buffer, and where propagation delays are significant. (In wide-area networks propagation times may be significant in comparison with queueing times: for a transatlantic link of 600 Megabits per second, ten million bits may be in flight between queues.) The framework we adopt in this chapter is simpler than that analysed by these authors in that we directly model only rates and not queue lengths, but more complex in that we model a network with an arbitrary number of bottleneck resources. Theoretical work [48], [75] on queues serving the superposition of a large number of streams indicates circumstances when the busy period preceding a buffer overflow may be relatively short, and several authors have argued the advantages of preventing queue build-up through the bounding of rates (see, for example, Charny et al. [39]).

Any discussion of the performance of a rate control scheme must address the issue of fairness, since there exist situations where a given scheme might maximize network throughput, for example, while denying access to some users. The most commonly discussed fairness criterion is that of max-min fairness: loosely, a set of rates is max-min fair if no rate may be increased without simultaneously decreasing another rate which is already smaller. In a network with a single
bottleneck resource max-min fairness implies an equal share of the resource for each flow through it. Mazumdar et al. [85] have pointed out that from a game-theoretic standpoint such an allocation is not special, and have advocated instead the Nash bargaining solution, from cooperative game theory, as capturing natural assumptions as to what constitutes fairness.

The need for networks to operate in a public (and therefore potentially non-cooperative) environment has stimulated work on charging schemes for broadband networks: see MacKie-Mason and Varian [82] for a description of a ‘smart market’ based on a per-packet charge when the network is congested, and the collection edited by McKnight and Bailey [86] for several further papers and references. Kelly [76] describes a model for elastic traffic in which a user chooses the charge per unit time that the user is willing to pay; thereafter the user’s rate is determined by the network according to a proportional fairness criterion applied to the rate per unit charge. It was shown that a system optimum is achieved when users’ choices of charges and the network’s choice of allocated rates are in equilibrium. There remained the question of how the proportional fairness criterion could be implemented in a large-scale network. In section 2 of this chapter we show that simple rate control algorithms, using additive increase/multiplicative decrease rules or explicit rates based on resource shadow prices, can provide stable convergence to proportional fairness per unit charge, even in the presence of random effects and delays.

Mechanisms by which supply and demand reach equilibrium have, of course, long been a central concern of economists, and there exists a substantial body of theory on the stability of what are termed tatonnement processes [26], [66], [102]. From this viewpoint the rate control algorithms described in this chapter are particular embodiments of a ‘Walrasian auctioneer’, searching for market clearing prices. The ‘Walrasian auctioneer’ of tatonnement theory is usually considered a rather implausible construct; we show that the structure of a communication network provides a natural context within which to investigate the consequences for a tatonnement process of stochastic perturbations and time lags.

The organization of the chapter is as follows. In section 3.2 we review the work of Kelly et al [77] and present an approach to pricing based on a proportional fairness criterion. In section 3.3 we describe a model based on the work of Courcoubetis et al [47] and present two possible ABR implementations of proportionally fair pricing. In section 3.4 we present a pricing scheme based on the sharing of effective usage rather than the sharing of simple rates.

### 3.2 Proportionally fair pricing

**The basic model**

Consider a network with a set $J$ of resources, and let $C_j$ be the finite capacity of resource $j$, for $j \in J$. Let a route $r$ be a non-empty subset of $J$, and write $R$ for the set of possible routes. Set $A_{jr} = 1$ if $j \in r$, so that resource $j$ lies on route $r$, and set $A_{jr} = 0$ otherwise. This defines a $0-1$ matrix $A = (A_{jr}, j \in J, r \in R)$.

Associate a route $r$ with a user, and suppose that if a rate $x_r$ is allocated to user $r$ then this has utility $U_r(x_r)$ to the user.

Assume that the utility $U_r(x_r)$ is an increasing, strictly concave and continuously differentiable function of $x_r$ over the range $x_r \geq 0$ (following Shenker [96], we call traffic that leads to such a utility function elastic traffic). Assume further that utilities are additive, so that the aggregate utility of rates $x = (x_r, r \in R)$ is $\sum_{r \in R} U_r(x_r)$. Let $U = (U_r(\cdot), r \in R)$ and $C = (C_j, j \in J)$. Under this model the system optimal rates solve the following problem.
3.2. PROPORTIONALLY FAIR PRICING

\[ \text{SYSTEM}(U, A, C): \]

\[
\begin{align*}
\text{maximize} & \quad \sum_{r \in R} U_r(x_r) \\
\text{subject to} & \quad Ax \leq C \\
\text{over} & \quad x \geq 0.
\end{align*}
\]

While this optimization problem is mathematically fairly tractable (with a strictly concave objective function and a convex feasible region), it involves utilities \( U \) that are unlikely to be known by the network. We are thus led to consider two simpler problems.

Suppose that user \( r \) may choose an amount to pay per unit time, \( w_r \), and receives in return a flow \( x_r \) proportional to \( w_r \), say \( x_r = w_r/\lambda_r \), where \( \lambda_r \) could be regarded as a charge per unit flow for user \( r \). Then the utility maximization problem for user \( r \) is as follows.

\[ \text{USER}_r(U_r; \lambda_r): \]

\[
\begin{align*}
\text{maximize} & \quad U_r \left( \frac{w_r}{\lambda_r} \right) - w_r \\
\text{over} & \quad w_r \geq 0.
\end{align*}
\]

Suppose next that the network knows the vector \( w = (w_r, r \in R) \), and attempts to maximize the function \( \sum_r w_r \log x_r \). The network's optimization problem is then as follows.

\[ \text{NETWORK}(A, C; w): \]

\[
\begin{align*}
\text{maximize} & \quad \sum_{r \in R} w_r \log x_r \\
\text{subject to} & \quad Ax < C \\
\text{over} & \quad x \geq 0.
\end{align*}
\]

It is known [76] that there always exist vectors \( \lambda = (\lambda_r, r \in R) \), \( w = (w_r, r \in R) \) and \( x = (x_r, r \in R) \), satisfying \( w_r = \lambda_r x_r \) for \( r \in R \), such that \( w_r \) solves \( \text{USER}_r(U_r; \lambda_r) \) for \( r \in R \) and \( x \) solves \( \text{NETWORK}(A, C; w) \); further, the vector \( x \) is the unique solution to \( \text{SYSTEM}(U, A, C) \).

A vector of rates \( x = (x_r, r \in R) \) is proportionally fair if it is feasible, that is \( x \geq 0 \) and \( Ax \leq C \), and if for any other feasible vector \( x^* \), the aggregate of proportional changes is negative

\[
\sum_{r \in R} \frac{x^*_r - x_r}{x_r} < 0. \tag{3.1}
\]

If \( w_r = 1, r \in R \), then a vector of rates \( x \) solves \( \text{NETWORK}(A, C; w) \) if and only if it is proportionally fair. Such a vector is also the Nash bargaining solution (satisfying certain axioms of fairness [56]), and, as such, has been advocated in the context of telecommunications by Mazumdar et al. [85].

A vector \( x \) is such that the rates per unit charge are proportionally fair if \( x \) is feasible, and if for any other feasible vector \( x^* \)

\[
\sum_{r \in R} w_r \frac{x^*_r - x_r}{x_r} < 0. \tag{3.2}
\]

The relationship between the conditions (3.1) and (3.2) is well illustrated when \( w_r, r \in R \), are all integral. For each \( r \in R \), replace the single user \( r \) by \( w_r \) identical sub-users, construct the proportionally fair allocation over the resulting \( \sum_r w_r \) users, and provide to user \( r \) the aggregate rate allocated to its \( w_r \) sub-users; then the resulting rates per unit charge are proportionally fair.
This construction also illustrates the need to adapt the notion of fairness to a non-cooperative context, where it is possible for a single user to represent itself as several distinct users. It is straightforward to check [76] that a vector of rates $x$ solves $\text{NETWORK}(A, C; w)$ if and only if the rates per unit charge are proportionally fair.

We note in passing that if, for a fixed set of users and arbitrary parameters $w = (w_r, r \in R)$, the network solves $\text{NETWORK}(A, C; w)$, then the resulting rates $x = (x_r, r \in R)$ solve a variant of the problem $\text{SYSTEM}(U, A, C)$, with a weighted objective function $\sum_r \alpha_r U_r(x_r)$ where $\alpha_r = w_r / (x_r U'_r(x_r))$ for $r \in R$. Thus a choice of the parameters $w = (w_r, r \in R)$ by the network (rather than by users) corresponds to an implicit weighting by the network of the relative utilities of different users, with weights related to the users’ various marginal utilities.

Under the decomposition of the problem $\text{SYSTEM}(U, A, C)$ into the problems $\text{NETWORK}(A, C; w)$ and $\text{USER}_r(U_r; \lambda_r), r \in R$, the utility function $U_r(x_r)$ is not required by the network, and only appears in the optimization problem faced by user $r$. The Lagrangian [104] for the problem $\text{NETWORK}(A, C; w)$ is

$$L(x, z; \mu) = \sum_{r \in R} w_r \log x_r + \mu^T (C - Ax - z)$$

where $z \geq 0$ is a vector of slack variables and $\mu$ is a vector of Lagrange multipliers (or shadow prices). Then

$$\frac{\partial L}{\partial x_r} = \frac{w_r}{x_r} - \sum_{i \in r} \mu_j,$$

and so the unique optimum to the primal problem is given by

$$x_r = \frac{w_r}{\sum_{j \in r} \mu_j} \quad \text{(3.3)}$$

where $(x_r, r \in R), (\mu_j, j \in J)$ solve

$$\mu \geq 0, \quad Ax \leq C, \quad \mu^T (C - Ax) = 0 \quad \text{(3.4)}$$

and relation (3.3). Furthermore the associated dual problem quickly reduces, after elision of terms not dependent on the shadow prices $\mu$, to the following problem.

$\text{DUAL}(A, C; w)$:

$$\text{maximize} \quad \sum_{r \in R} w_r \log \left( \sum_{j \in r} \mu_j \right) - \sum_{j \in J} \mu_j C_j$$

$$\text{over} \quad \mu \geq 0.$$

While the problems $\text{NETWORK}(A, C; w)$ and $\text{DUAL}(A, C; w)$ are mathematically tractable, it would be difficult to implement a solution in any centralized manner. A centralized processor, even if it were itself completely reliable and could cope with the complexity of the computational task involved, would have its lines of communication through the network vulnerable to delays and failures. Rather, interest focuses on algorithms which are decentralized and of a simple form: the challenge is to understand how such algorithms can be designed so that the network as a whole reacts intelligently to perturbations. Next we describe two simple classes of decentralized algorithm, designed to implement solutions to relaxations of the problems $\text{NETWORK}(A, C; w)$ and $\text{DUAL}(A, C; w)$. 
3.2. PROPORTIONALLY FAIR PRICING

A primal algorithm

Consider the system of differential equations

$$\frac{d}{dt} x_r(t) = \kappa \left( w_r - x_r(t) \sum_{j \in R} \mu_j(t) \right)$$  \hspace{1cm} (3.5)

where

$$\mu_j(t) = p_j \left( \sum_{s, j \in S} x_s(t) \right).$$  \hspace{1cm} (3.6)

(Here and throughout we assume that, unless otherwise specified, \( r \) ranges over the set \( R \) and \( j \) ranges over the set \( J \).) We may motivate the relations (3.5)–(3.6) in several ways. For example, suppose that \( p_j(y) \) is a price charged by resource \( j \), per unit flow through resource \( j \), when the total flow through resource \( j \) is \( y \). Then by adjusting the flow on route \( r \), \( x_r(t) \), in accordance with equations (3.5)–(3.6), the network attempts to equalize the aggregate cost of this flow. \( x_r(t) \sum_{j \in R} \mu_j(t) \), with a target value \( w_r \), for every \( r \in R \). (For an enlightening description of the technological implementation of such algorithms in an ATM network, see Courcoubetis et al. [47].)

For an alternative motivation, suppose that resource \( j \) generates a continuous stream of feedback signals at rate \( p_j(y) \) when the total flow through resource \( j \) is \( y \). Suppose further that when resource \( j \) generates a feedback signal, a copy is sent to each user \( r \) whose route passes through resource \( j \), where it is interpreted as a congestion indicator requiring some reduction in the flow \( x_r \). Then equation (3.5) corresponds to a response by user \( r \) that comprises two components: a steady increase at rate proportional to \( w_r \), and a multiplicative decrease at rate proportional to the stream of feedback signals received. (For early discussions of algorithms with additive increase and multiplicative decrease see Chiu and Jain [40] and Jacobson [69]; Hernandez-Valencia et al. [67] review several algorithms based on congestion indication feedback.)

We establish that under mild regularity conditions on the functions \( p_j, j \in J \), the expression

$$\mathcal{U}(x) = \sum_{r \in R} w_r \log x_r - \sum_{j \in J} \int_0^{\sum_{s, j \in S} x_s} p_j(y)dy$$  \hspace{1cm} (3.7)

provides a Lyapunov function for the system of differential equations (3.5)–(3.6), and we deduce that the vector \( x \) maximizing \( \mathcal{U}(x) \) is a stable point of the system, to which all trajectories converge.

The functions \( p_j, j \in J \), may be chosen so that the maximization of the Lyapunov function \( \mathcal{U}(x) \) arbitrarily closely approximates the optimization problem \( \text{NETWORK}(A, C; w) \), and, in this sense, is a relaxation of the network problem. We show that certain relaxations correspond naturally to a system objective which takes into account loss or delays, as well as flow rates.

The Lyapunov function (3.7) thus provides an enlightening analysis of the global stability of the system (3.5)–(3.6), and of the relationship between this system and the problem \( \text{NETWORK}(A, C; w) \). However, the system (3.5)–(3.6) has omitted to model two important aspects of decentralized systems, namely stochastic perturbations, and time lags. We analyze these aspects by considering small perturbations to the stable point \( x \).

Stochastic perturbations within the network may well arise from a resource’s method of sensing its load. Equation (3.6) represents the response \( \mu_j(t) \) of resource \( j \) as a continuous function of a load, \( y = \sum_{s, j \in S} x_s \), which is assumed known. In practice a resource may assess its load by an error-prone measurement mechanism, and then choose a feedback signal from a small set of possible signals. (See Hernandez-Valencia et al. [67] and Bonomi et al. [34] for more detailed descriptions of binary feedback and congestion indication rate control algorithms.) We describe how such mechanisms motivate various stochastic models of the network. One particular model
Chapter 3. Pricing for Elastic Services

takes the form

$$dx_r(t) = \kappa \left( w_r dt - x_r(t) \sum_{j \in r} \left( \mu_j(t) dt + \mu_j(t)^{1/2} \varepsilon_j^{1/2} dB_j(t) \right) \right)$$

(3.8)

where $B_j(t)$ is a standard Brownian motion, representing stochastic effects at resource $j$, and $\varepsilon_j$ is a scaling parameter for these effects. If the scaling parameters $\varepsilon_j, j \in J$, are small then the stochastic differential equation (3.8) has, as solution, a multidimensional Ornstein–Uhlenbeck process, centred on the stable point $x$ of the differential equations (3.5)–(3.6). The stationary distribution for $(x_r(t), r \in R)$ is a multivariate normal distribution, with covariance matrix that can be explicitly calculated in terms of the parameters of the network.

Similarly we describe a model incorporating time lags that generalizes equations (3.5)–(3.6), and shall analyse its behaviour close to the stable point $x$. Our models of stochastic effects and of time-lags provide important insights into the behaviour of the network, and allow us to quantify the various relationships and trade-offs between speed of convergence, the magnitude of fluctuations about the equilibrium point, and the stability of the network.

A dual algorithm

The equations (3.5)–(3.6) represent a system where rates vary gradually, and shadow prices are given as functions of the rates. Next we consider a system where shadow prices vary gradually, with rates given as functions of the shadow prices. Let

$$\frac{d}{dt} \mu_j(t) = \kappa \left( \sum_{r, j \in r} x_r(t) - q_j \left( \mu_j(t) \right) \right)$$

(3.9)

where

$$x_r(t) = \frac{w_r}{\sum_{k \in r} \mu_k(t)}.$$

(3.10)

The relationship between the algorithm (3.9)–(3.10) and the problem $DUAL(A, C; w)$ parallels that between the primal algorithm (3.5)–(3.6) and the problem $NETWORK(A, C; w)$, and, again, we may motivate the algorithm in several ways. For example, suppose that $q_j(\eta)$ is the flow through resource $j$ which generates a price at resource $j$ of $\eta$. Then an economist would describe the right hand side of equation (3.9) as the vector of excess demand at prices $(\mu_j(t), j \in J)$, and would recognize equations (3.9)–(3.10) as a tatonnement process by which prices adjust according to supply and demand (Varian [102], Chapter 21).

We establish that under mild regularity conditions on the functions $q_i, j \in J$, the expression

$$\mathcal{V}(\mu) = \sum_{r \in R} w_r \log \left( \sum_{j \in r} \mu_j \right) - \sum_{j \in J} \int_0^{\mu_j} q_j(\eta) d\eta$$

(3.11)

provides a Lyapunov function for the system of differential equations (3.9)–(3.10), and we deduce that the vector $\mu$ maximizing $\mathcal{V}(\mu)$ is a stable point of the system, to which all trajectories converge. Further, by appropriate choice of the functions $q_i, j \in J$, the maximization of the function $\mathcal{V}(\mu)$ can arbitrarily approximate the problem $DUAL(A, C; w)$.

We consider stochastic perturbations of system (3.9)–(3.10), with a typical example taking the form

$$d\mu_j(t) = \kappa \left( \sum_{r, j \in r} \left( x_r(t) dt + x_r(t)^{1/2} \varepsilon_r^{1/2} dB_r(t) \right) - q_j \left( \mu_j(t) \right) dt \right)$$

(3.12)
where \( B_r(t) \) is a standard Brownian motion, representing stochastic effects associated with the flow on route \( r \). If the scaling parameters \( \varepsilon_r, r \in R \), are small then the stationary distribution for \( (\mu_j(t), j \in J) \) is centred on the stable point \( \mu \) of the differential equations (3.9)–(3.10), with a covariance matrix that can be explicitly calculated in terms of the parameters of the network. Also it is possible to analyse the stability of the model (3.9)–(3.10) when time-lags are introduced.

**User adaptation**

Our analyses of the primal algorithm (3.5)–(3.6) and the dual algorithm (3.9)–(3.10) assume that the parameters \( (w_r, r \in R) \) chosen by the users are fixed, at least on the time scales concerned in the analyses. With increasing intelligence embedded in end-systems, users may in the future be able to vary the parameters \( (w_r, r \in R) \) even within these short time scales. Both the algorithms may be extended to this situation.

Suppose that user \( r \) is able to monitor its rate \( x_r(t) \) continuously, and to vary smoothly the parameter \( w_r(t) \) so as to track accurately the optimum to \( \text{USER}_r(U_r; \lambda_r(t)) \), where \( \lambda_r(t) = w_r(t)/x_r(t) \) is the charge per unit flow to user \( r \) at time \( t \). Then, using revised Lyapunov functions, stability of both the primal and dual algorithms may again be established.

In [77] we provide detailed proofs of the various results outlined above, together with some numerical illustrations. We also extend the discussion to include routing control.

**Concluding remarks**

In this section we have addressed the issue of how available bandwidth within a large-scale broadband network should be shared between competing streams of elastic traffic. An optimization framework leads to a decomposition of the overall system problem into a separate problem for each user, in which the user chooses a rate per unit time that the user is willing to pay, and one for the network; we have shown that two classes of rate control algorithm are naturally associated with the objective functions appearing in, respectively, the primal and dual formulation of the network’s problem. In consequence the algorithms provide natural implementations of proportionally fair pricing. In [77] we have studied the stability of the algorithms in the presence of stochastic perturbations and time lags, and have illustrated our results with a study of a random network with a hundred resources and a thousand routes. Interesting and challenging questions remain concerning the stability of the entire system under more general assumptions on users’ reactions to the rates allocated to them by the network, and when the numbers of users and the amounts of capacity available for elastic traffic vary randomly. An outstanding practical issue concerns how protocols, such as TCP in the Internet or the Available Bit Rate transfer capability of an ATM network, can be adapted to be charge sensitive.

### 3.3 Dynamic pricing for ABR services

In this section we discuss how the pricing schemes discussed in Section 3.2 can be implemented using the closed loop rate-based congestion control framework of the Available Bit Rate (ABR) service category. The ABR service is one of the five service categories identified by the ATM Forum [54] and is intended for “elastic” traffic which has no strict requirements on delay or delay variance. Applications using this service are assumed to respond appropriately to congestion control messages they receive from the network. In addition, they may specify some Minimum Cell Rate (MCR), below which they will not be asked to fall.

Because our approach utilizes mechanisms provided by ABR’s rate-based flow control, it imposes no additional communication overhead, while the added complexity at the switches and end-
systems is small. According to the approach, prices depend on the demand for bandwidth and are adjusted in a decentralized and iterative manner. When the system (network and users) reaches equilibrium, the demand will equal the supply of bandwidth. In the implementation based on the distributed and iterative solution of the System and User Problems of Section 3.2, the network posts the price per unit of bandwidth to each user. Based on this price, the user selects the bandwidth he wishes to send with. In the implementation based on the dual algorithm of Section 3.2, the explicit rate that the network sends back to the user, which is the maximum rate the user is allowed to send with, is a function of the user’s willingness to pay (price per unit of time) and the current price per unit of bandwidth. The latter is adjusted, as before, in a decentralized and iterative manner.

**Implementation of dynamic prices**

The solution to the System and User Problems can be computed in a decentralized manner if at each link \( l \), the price per unit of rate \( \mu_l \) is increased or decreased if \( C_l < \sum_{i \in R_l} x_i \) or \( C_l > \sum_{i \in R_l} x_i \), respectively, where \( C_l \) is the bandwidth for ABR connections, \( x_i \) is the rate of connection \( i \), and \( R_l \) is the set of links connection \( i \) traverses. The update of \( \mu_l \) can be written as

\[
\frac{d}{dt} \mu_l(t) = \kappa \left( \sum_{i \in R_l} x_i(t) - C_l \right),
\]

where \( \kappa \) is some small constant. Prices are updated in time intervals (we will assume that they all have the same length) whose duration depends on how fast the aggregate demand \( \sum_i x_i(t) \) changes. This interval (price update interval) should not be less than the interval from the time new prices are posted until the time the users’ responses to these prices are received by the network, i.e., one round-trip delay.

Next we discuss how to implement dynamic prices within the framework of ABR congestion control.

According to ABR rate-based congestion control [54], every source sends special control cells, called Resource Management (RM) cells, either periodically or after a specific number of data cells carrying user information (Figure 3.1). When the destination receives an RM cell (called forward RM cell), it copies its information to a backward RM cell which it sends back to the source. When an intermediate switch detects congestion on one of its links, it sets a congestion indication bit and places a rate in the Explicit Rate (ER) field of all backward RM cells traversing that link.\(^1\) While the source does not receive RM cells, or while it receives RM cells with the congestion bit set, it decreases its cell rate by some percentage (multiplicative decrease). Furthermore, its cell rate must always be smaller than the explicit rate in the received RM cell. If the source receives an RM cell with the congestion indication bit cleared, it is allowed to increase its cell rate by some additive quantity (additive increase).

To implement the dynamic pricing scheme, switches adjust the price of bandwidth for each of their output links using a discrete version of equation (3.13). This equation requires that the total demand is known to the switches. The latter can be achieved by having every source place their demand in the ER field of the forward RM cells.

Finally, (3.13) requires the price \( \lambda_i = \sum_{l \in R_i} \mu_l \) for each connection \( i \). This sum can be created by adding a new price field \( P \) in the RM cell. When a backward RM cell is sent by the destination, the value of \( P \) is zero. Each switch adds the price \( \mu_l \) to the amount contained in the price field \( P \) of the backward RM cells that traverse link \( l \). When a backward RM cell reaches the source, \( P \) will contain the value \( \lambda_i = \sum_{l \in R_i} \mu_l \).

\(^1\)In ATM networks, the forward and backward RM cells follow the same route.
Based on the price $\lambda_i$ in the received $RM$ cell, user $i$ selects an amount of bandwidth based on some demand function $D_i(\lambda_i)$. The relationship between the demand function and the utility functions $U_i$ of Section 3.2 is the following. Let $D_i(\lambda) = x_i$, where $x_i$ is the solution to $\lambda = U_i'(x_i)$. With $D_i(\lambda) = 0$ if $\lambda \geq U_i'(0)$ and $D_i(\lambda) = \infty$ if $\lambda \leq U_i'(\infty)$. Hence, $D_i(\lambda)$ can be interpreted as the demand of user $i$ when confronted with a price per unit flow of $\lambda$.

### Simulation results

The goal of our simulation experiments was to study the convergence properties and transient behaviour of the dynamic prices. Rather than simulating the network at the cell level, we have modeled the propagation of rate changes and the propagation of resource management cells, thus achieving considerably smaller simulation times. The simulated network is shown in Figure 3.2. All link rates are 155 Mbps. In the three experiments we present here, interswitch distances were 1 km, 100 km, and 1000 km, respectively. Hence we can observe the transient behaviour in both a local and wide area environment.

We assume that switches update the prices $\mu_l(t)$ in discrete time intervals using the following discrete time version of (3.13):

$$\mu_l(t) = \mu_l(t - 1) + \kappa \left( \sum_{i : i \in R_i} x_i(t - 1) - C_l \right),$$

(3.14)

where $C_l$ is the capacity of link $l$ used by ABR services, $x_i(t - 1)$ is the demand for bandwidth by connection $i$ during the price update interval $t - 1$. We assume that $\mu_l(0) = 0$ for all $l \in L$. In [47] we have experimented with a price update function where $\kappa$ is a function of $\mu_l(t - 1)$.

For simplicity, we assume that sources send $RM$ cells in fixed time intervals, equal to 200\mu s (interval-based behavior), rather than after a specific number of data cells (counter-based behav-
ior). Furthermore, we assume that all sources \( i \) have a demand for bandwidth \( D_i \) which decreases exponentially with the price per unit of rate \( \lambda_i \), i.e., \( D_i(\lambda_i) = v_i e^{-\lambda_i} \), where \( v_i = 155 \text{ Mbps} \).

In the first experiment, the distance between switches was 1 km and the price update interval was 200 \( \mu s \). The value of parameter \( \kappa \) in (3.14) was set to 0.0005 \( \text{Mbps}^{-1} \). This parameter determines the number of round-trip times needed for prices to converge (on the contrary, the number of round-trips is independent of the round-trip delay). Larger values of \( \kappa \) could lead to oscillations. In general, the selection of \( \kappa \) depends on the number of multiplexed Virtual Connections (VCs), the magnitude of changes relative to link capacity, the network topology, and the sources’ demands. Figure 3.3 shows that prices converge quickly after changes of the input traffic.

In the second experiment, the distance between switches was 100 km and the price update interval was 2.5 msec. Parameter \( \kappa \) in equation (3.14) was 0.001 \( \text{Mbps}^{-1} \). From Figure 3.4(a), we see that convergence times (≈ 30 msec) are greater than the case of 1 km links (≈ 5-10 msec). In practice, this will not be a problem since in wide area networks, due to the high aggregation of traffic, changes are expected to be smaller and less frequent.

Finally, Figure 3.4(b) shows the dynamic behavior of prices when the distance between switches was 1000 km. The price update interval in this case was 21 msec. In the experiments of Figures 3.4(a) and 3.4(b) the network reached the equilibrium state after the same number of iterations (approximately 15). However, the convergence time is longer when the distance between switches is larger. This is due to the larger pricing interval (21 msec for 1000 km links compared to 2.5 msec for 100 km links) which is required because of the longer round-trip delay.

### An alternative user-network interaction

In order to implement the willingness to pay scheme given by the dual algorithm in Section 3.2, we introduce a new field \( W \) in the \( RM \) cells which stores the value of the willingness to pay \( w_i \) of each the connection \( i \). Furthermore, as discussed above, we introduce a price field \( P \) which stores the sum of the prices per unit of bandwidth on all links a connection traverses, i.e., the value \( \sum_{l \in R_i} \mu_l \). This sum is created by having each switch add the price \( \mu_l \) to the price field \( P \) in the forward \( RM \) cells (refer to Figure 3.1). When the first switch connected to the source receives the backward \( RM \) cell, it computes \( x_i = w_i / \sum_{l \in R_i} \mu_l \) by dividing the two fields \( W \) and \( P \), and sets the \( ER \) field of the backward \( RM \) cells of connection \( i \) equal to \( x_i \).

From the received value of \( ER = x_i \), the source can compute the current price \( \lambda_i = w_i / x_i \). Based on the value of \( \lambda_i \), the source can select a new willingness to pay which it inserts in the forward \( RM \) cells.

### 3.4 Sharing effective usage

It would be advantageous for a network operator that offers elastic traffic to have a simple way to reason about resource usage and fairness, in order to charge his users while achieving a stable and economically efficient operation of his network. Such an operator would greatly benefit from the existence of a simple but accurate unifying measure of the information transfer capability that the network provides to its users, which could also be used as a basis for charging. Defining such a measure is by far not obvious since different users value the quality of a network service differently. For example, for some users the delay might be the important aspect, whereas for others the average rate they send traffic or the probability of not losing cells.

Traditional measures of resource usage are the average bit rate, and minimum or maximum throughput; these measures do not capture the requirements on network resources by bursty users, which value combinations of properties, such as the average bit rate and the amount of distortion the network will apply to their traffic by delaying its entry into the network. To understand why
3.4. **SHARING EFFECTIVE USAGE**

![Graphs showing traffic flow over time for different link distances.](image)

(a) Price at link $l_{1 \rightarrow 2}$  
(b) Price at link $l_{2 \rightarrow 3}$

**Figure 3.3:** Results for 1 km switch distances. VC 1 started at 0 msec and terminated at 25 msec, VC 2 started at 0 msec, and VC 3 at 10 msec. Convergence time is approximately 5-10 msec.

![Graphs showing traffic flow over time for different link distances.](image)

(a) Results for 100 km switch distances. Convergence time is approximately 30 msec.  
(b) Results for 1000 km switch distances. Convergence time is approximately 300 msec.

**Figure 3.4:** Price at link $l_{1 \rightarrow 2}$ when the distance between switches is 100 km and 1000 km, respectively. VC 1 and VC 2 started at 0 msec, and VC 3 at 40 msec (for 100 km switch distances) or 300 msec (for 1000 km switch distances).
this is important, consider a charging scheme that charges solely based on the maximum rate, or *explicit rate* (ER) in ABR terminology, that a user is allowed to send traffic. Also consider two users, one bursty and one smooth. If both users are charged the same amount, both are entitled to the same explicit rate. However, the bursty user, who does not use his explicit rate at all times, is actually using less resources compared to the non-bursty user. Hence, the pricing scheme is unfair to him. Facing such a pricing scheme, the bursty user does not have the incentive to use less than his explicit rate at all times, even though he actually needs less. As a result, network resources will not be used according to the actual needs of users; such incentive incompatibility will induce a non-economically optimal operation of the network.

We propose that a user’s information transfer capability, or “effective rate”, is measured in terms of the effective bandwidth of the traffic that the user is allowed to send through the network. Users bid for effective bandwidth and the network controls the effective bandwidth of the users’ traffic streams by adjusting their explicit rates (ER) in order to achieve the economically fair sharing of resources. Since the effective bandwidth is an adequate measure of resource usage and burstiness, bidding for effective bandwidth allows users to reveal their true preference for network usage, and results in charges that reflect actual resource usage. The approach relies on the heavy multiplexing due to the large capacity of the links, which provides for the accuracy of the effective bandwidth definitions, and on the bursty nature of the traffic, which allows simple effective bandwidth approximations to be used. Since effective bandwidths are naturally defined over long time scales compared to round trip delays, the above control loop is a natural candidate to be implemented on long WAN links.

**Sharing bandwidth according to effective usage**

Next we discuss how the effective usage resource sharing scheme whose mathematical underpinnings are based on the price mechanism described by equations (3.9-3.10), and the notion of effective bandwidths, can be implemented using ABR’s flow control loop. The procedures that we describe refer solely to the operation of switches. The source behavior is the same as that of a normal ABR source [54]. The user is only required to declare at connection setup (or during renegotiation) his willingness to pay $w_r$ (price per unit of time).

We first define the following notation. Let $\tilde{a}_j(m, h)$ be the effective bandwidth using the on-off approximation (3.15) of a connection with mean $m$ and peak $h$ at link $j$ (hence $j$ captures the link’s operating point parameters $s,t$).

$$\tilde{a}_j(s,t) = \frac{1}{st} \log \left[ 1 + \frac{m}{h} (e^{sh} - 1) \right].$$

(3.15)

Let $\tilde{a}_j^{-1}(m,a)$ be the value of the peak rate $h$ for which $\tilde{a}_j(m, h) = a$, for some positive value $a$.

The network can control a bursty source so that its effective rate through link $j$ becomes equal to $a$ by measuring the source’s mean rate $m$ and limiting the source’s peak rate to the value $\tilde{a}_j^{-1}(m,a)$. This can be achieved with ABR rate-based flow control by setting the explicit rate ER of the source equal to $\tilde{a}_j^{-1}(m,a)$. To allow for efficient implementation, simple tables can be used for computing $\tilde{a}_j^{-1}(m,a)$. An accurate estimate of the short term mean for each source is not necessary, since ABR users are more interested in their average explicit rate over a large time interval. Furthermore, the mean rate can be measured once at the first switch of the network and circulated to the intermediate switches via $RM$ cells.

We start by defining the simplest version of our effective flow control implementation where all links of the network have similar operating characteristics, and hence the values of $s,t$ are identical throughout the network\(^2\). Under the above simplification, we can drop the link subscript from the effective bandwidth formula, which becomes $\tilde{a}(m, h)$ for all links.

\(^2\)There is an economic argument based on marginal costs of buffer and bandwidth which supports this argument [46].
The implementation of the scheme is now straightforward. In the \(RM\) cells we introduce a new field \(W\) storing the value of the willingness to pay \(w_r^3\), and a new field \(P\) storing the sum of the unit prices for effective bandwidth as posted by the links the connection traverses, i.e., the value \(\sum_{k \in r} \mu_k(t)\). This sum is created by having each link add the price \(\mu_j(t)\) to the price field \(P\) in the backward \(RM\) cells. (We assume that the price field \(P\) has an initial value zero when the backward \(RM\) cell is sent by the destination.) The prices \(\mu_j(t)\) are computed using a discrete version of (3.9), with the right hand side replaced by the expression

\[
\kappa \left( \sum_{r \in r_t} \frac{w_r}{\sum_{k \in r} \mu_k(t)} - C_j \right),
\]

(3.16)

where \(C_j\) is the effective capacity of link \(j\). When the first switch connected to the source receives the backward \(RM\) cell, it computes the value of the effective rate \(a_r = w_r / \sum_{k \in r} \mu_k\) (equation (3.10)) by dividing the two fields \(W, P\), and sets the \(ER\) field of the \(RM\) cells of connection \(r\) equal to

\[
ER = \tilde{\alpha}^{-1}(m_r, a_r),
\]

(3.17)

where \(m_r\) is the mean rate of connection \(r\). Such a procedure will provably converge to a fair allocation of the effective rates (as defined by formula (3.15)) which solve the optimization problem \(\text{NETWORK}(A, C; w)\).

Now we discuss the general situation where the operating conditions of the various links are different, and hence \(\tilde{\alpha}_j(m, h)\) possibly differs for different \(j\)’s. In this case, the prices at each link are computed as before, but every link \(j\) must compute for each connection \(r\) the value of its explicit rate \(ER'' = \tilde{\alpha}_j^{-1}(m_r, a_r)\), where \(a_r = w_r / \sum_{k \in r} \mu_k\). The latter requires that all switches know the sum \(\sum_{k \in r} \mu_k\). This can be achieved by having the first switch connected to the source copy the value of the price field \(P\) of the backward \(RM\) cells to the price field \(P\) of the forward \(RM\) cells, thus making the sum available to all the switches along the route of connection \(r\). After computing \(ER''\), the switch sets the value of the \(ER\) field of the backward \(RM\) cells belonging to connection \(r\) equal to the minimum of its current value and \(ER''\).

The achievable utilization using the above procedures relies on one hand on the degree of multiplexing that takes place, and on the other hand on the accuracy of the effective bandwidth approximation (3.15). However, this formula tends to be conservative [45]. A simple way to remedy this inefficiency, while keeping the essential merits of the approach, is to couple it with direct measurements of the cell loss probability (or equivalently, of the probability that cells are delayed more than some target value). Hence, while the operating point (link prices) is defined as above, we slowly perturb it in order to increase the efficiency. This can be done by having each link \(j\) measure the buffer overflow probability (or some “proxy” of it), and compute a load factor \(z_j\) reflecting the underutilization or overutilization of the link resources. The effective capacity \(C_j\) in (3.16) is replaced by the product \(z_j C_j\). While the cell loss probability is higher or lower than the prespecified maximum value, the value of \(z_j\) is decreased or increased, respectively. One anticipates that changing the load factor \(z_j\) at a slower time scale compared to the time it takes for prices to converge will eventually stabilize the system at an operating point where all links have the desired maximum utilization.

The procedures that we have described in this section for implementing the effective usage resource sharing scheme are summarized in Table 3.1. Note that these procedures refer solely to the operation of switches. The source behavior is the same as that of a normal ABR source [54]. The user is only required to declare at connection setup (or during renegotiation) the price per unit of time \(w_r\).

---

3 Either the source can insert his willingness to pay \(w_r\) in the \(W\) field of every \(RM\) cell it sends, or declare it at connection setup and have the first switch to which it is connected to insert it in the \(W\) field of the \(RM\) cells belonging to that connection.
/* Link parameters */
\( \kappa \) : price update parameter
\( T_p \) : price update interval
\( T_m \) : mean rate measurement interval
\( C_j \) : link capacity available for ABR connections

/* Link variables */
\( \mu_j \) : price per unit of bandwidth
\( z_j \) : link load factor

/* Connection variables */
\( w_r \) : price per unit of time for connection \( r \)
\( \lambda_r \) : price per unit of bandwidth for connection \( r \)
\( V_r \) : total number of cells of connection \( r \) since last mean rate measurement

/* RM cell fields */
\( ER \) : Explicit Rate field
\( P \) : Price field
\( W \) : Willingness to pay field

/* functions performed for each link \( j \) */

if (forward RM cell received for connection \( r \))
\[ \lambda_r = P \]
\[ w_r = W \]

if (backward RM cell received for connection \( r \))
\[ P = P + \mu_j \]
\[ x_r = w_r / \lambda_r \]
\[ ER' = \alpha^{-1}(m_r, x_r) \]
\[ ER = \min(ER', ER) \]

if (time to perform price update) /* performed once every \( T_p \) */
\[ \mu_i = \mu_i + \kappa \left( \sum_{r \in \mathcal{R}} w_r / \lambda_r - z_i C_i \right) \]

if (time to measure mean rate) /* performed once every \( T_m \) */
\[ m_r = V_r / T_p \]

Table 3.1: Switch operation for the effective usage resource sharing scheme. The first switch to which a source is connected to, in addition to the above, copies the value of the price field \( P \) of the backward RM cells (i.e., the value \( \lambda_r = \sum_{k \in \mathcal{R}} \mu_k \)) to the price field \( P \) of the forward RM cells. The link load factor \( z_i \) changes in a smaller time scale (compared to the price updates), based on whether the loss probability (or probability that the queue length becomes greater than some threshold) is higher or lower than some target value.