

# An Optimization Approach to ABR Control

David Lapsley\*

Steven Low

Department of EEE, University of Melbourne, Australia

Email: {d.lapsley, s.low}@ee.mu.oz.au

## Abstract

ABR sources react to network feedback by adjusting their transmission rates. Most schemes fall into one of two types depending on what is fed back and where control decision is made. Explicit Congestion Notification schemes allow sources to make control decisions but only with incomplete information on congestion. Explicit Rate schemes use more accurate congestion information but make the control decision inside the network without regard to different desires of various sources. In this paper we propose an optimization approach that attempts to combine the advantage of both types of scheme. The objective is to maximize total utility of all sources over their transmission rates. The dual problem suggests treating network links and ABR sources as processors in a distributed computation system to solve the dual problem using gradient projection algorithm. In this system ABR sources select transmission rates that maximize their own benefits and network links adjust bandwidth prices to coordinate the sources' decisions. We show how to implement such a system using features defined in the ABR standard. We provide an asynchronous distributed algorithm for links and sources and illustrate their behavior with preliminary simulation results.

## 1 Introduction

It seems better to serve elastic traffics [15], those generated by applications that can tolerate a certain transfer delay, using Available Bit Rate (ABR) rather than Constant Bit Rate (CBR) service in an ATM network. Indeed this folklore is formally proved in [10] for a large class of elastic traffics. ABR sources react to network feedback by adjusting their transmission rates. There are two types of control differing in what information is fed back and where control decision is made. In the first type a congestion indication is fed back using one [7, 4] or two bits [9] and the sources decide how to react, but only with incomplete information on network status<sup>1</sup>. Based on more accurate congestion information, the second type of schemes feed back an explicit rate at which all sources sharing the same bottleneck links should transmit in order that network queues are stabilized around some strictly positive target levels [1, 13]. A drawback however is that the control deci-

sion is made inside the network without regard to different desires of various sources.

In this paper we propose a different approach to ABR control where the goal is not to stabilize network queues at target levels, but rather to maximize the total utility of all the (elastic) ABR sources. Specifically consider a network that consists of a set  $L$  of unidirectional links of capacity  $c_l$ ,  $l \in L$ . The network is shared by a set  $S$  of sources, where source  $s$  is characterized by a utility function  $U_s(x_s)$  that is concave increasing in its transmission rate  $x_s$ . The goal is to calculate source rates that maximize the sum of the utilities  $\sum_{s \in S} U_s(x_s)$  over  $x_s$  subject to capacity constraints. Solving this problem centrally would require not only the knowledge of all utility functions, but worse still, complex coordination among potentially all sources due to the coupling of sources through shared links. Instead we propose a decentralized scheme that eliminates this requirement and adapts naturally to changing network conditions. The key is to consider the dual problem whose structure suggests treating the network links and the sources as processors of a *distributed computation system* to solve the *dual* problem using gradient projection method. Each processor executes a simple algorithm using only local information, communicates its computation result to others, and the cycle repeats. In this paper we explain how the necessary communication among these processors can be implemented using features defined in the ABR standard and present preliminary simulation results to illustrate the behavior. See [11] for a proof of convergence of our scheme and other extensions. For related approaches see [5, 6, 3, 8].

The optimization approach has three advantages. First the overall goal of maximizing a social welfare seems more desirable than, say, stabilizing network queues around target levels, though under our approach stable queues are a by-product of the optimization. Second sources that share the same link do not necessarily equally share the available bandwidth. Rather their shares reflect how they value the resource as expressed by their utility functions and how their use of the resource implies a cost to others. Finally since sources are free to choose their transmission rates based on their own utilities and network feedbacks, our iterative algorithm may track (slowly) time-varying utility functions and available capaci-

\*The first author would like to thank the Australian Telecommunications and Electronics Board for their financial support.

<sup>1</sup>Though the ATM standard also specifies end system behavior this approach potentially allows users to make individual control decisions [14]

ties, as suggested by initial experimental measurements.

We emphasize that though network feedbacks are discussed in terms of bandwidth ‘prices’ they may or may not form a component of the monetary charge a user pays. Our primary goal is not the pricing of services, but the steering of network towards an efficient operating point where the total source utility is maximized. The feedback a source receives is a measure of congestion specific to the source and is simply a control signal to guide its decision. If it further forms part of the service charge then it provides an incentive for the source to choose a socially optimal rate.

The rest of the paper is structured as follows. In §2 we present the optimization problem and its dual that motivate our optimization based flow control. In §3 we explain how the proposed control scheme can be implemented using features defined in the ABR standard, and present an asynchronous distributed algorithm for ABR control. In §4 we present preliminary simulation results to illustrate the behavior of these algorithms.

## 2 Our approach

### 2.1 The optimization problem

Consider a network that consists of a set  $L = \{1, \dots, L\}$  of unidirectional links of capacity  $c_l$ ,  $l \in L$ . The network is shared by a set  $S = \{1, \dots, S\}$  of sources. Source  $s$  is characterized by four parameters  $(L(s), U_s, m_s, M_s)$ . The path  $L(s) \subseteq L$  is a subset of links that source  $s$  uses,  $U_s : \mathbb{R}_+ \rightarrow \mathbb{R}$  is a utility function,  $m_s \geq 0$  and  $M_s \leq \infty$  are respectively the minimum and peak cell rate of source  $s$ . Source  $s$  attains a utility  $U_s(x_s)$  when it transmits at rate  $x_s$  that satisfies  $m_s \leq x_s \leq M_s$ . Let  $I_s = [m_s, M_s]$  denote the range in which source rate  $x_s$  must lie and  $I = (I_s, s \in S)$  be the vector. We assume  $U_s$  is increasing and strictly concave in its argument on  $I_s$ . For each link  $l$  let  $S(l) = \{s \in S \mid l \in L(s)\}$  be the set of sources that use link  $l$ .

Our objective is to choose source rates  $x = (x_s, s \in S)$  so as to:

$$\mathbf{P:} \quad \max_{x_s \in I_s} \quad \sum_s U_s(x_s) \quad (1)$$

$$\text{subject to} \quad \sum_{s \in S(l)} x_s \leq c_l, \quad l = 1, \dots, L. \quad (2)$$

The constraint (2) says that the total source rate at any link  $l$  is less than the capacity. Clearly a unique maximizer, called the primal optimal solution, exists since the objective function is strictly concave, and hence continuous, and the feasible solution set is compact.

<sup>2</sup>The capacity  $c_l$  in the model should be set to  $\rho_l$  times the real link capacity where  $\rho_l \in (0, 1)$  is a target utilization.

Though the objective function is separable in  $x_s$ , the source rates  $x_s$  are coupled by the constraint (2). Solving the primal problem (1–2) directly requires coordinating among possibly all sources and is impractical in real networks. The key to a distributed and decentralized solution is to look at its dual, e.g., [2, Section 3.4.2].

The objective function of the dual problem is [12]

$$D(p) = \max_{x_s \in I_s} L(x, p) = \sum_s B_s(p^s) + \sum_l p_l c_l$$

where

$$B_s(p^s) = \max_{x_s \in I_s} U_s(x_s) - x_s p^s \quad (3)$$

$$p^s = \sum_{l \in L(s)} p_l \quad (4)$$

and the dual problem is:

$$\mathbf{D:} \quad \min_{p \geq 0} D(p). \quad (5)$$

The first term of the dual objective function  $D(p)$  is decomposed into  $S$  separable subproblems (3–4). If we interpret  $p_l$  as the price per unit bandwidth at link  $l$  then  $p^s$  is the total price per unit bandwidth for all links in the path of  $s$ . Hence  $x_s p^s$  represents the bandwidth cost to source  $s$  when it transmits at rate  $x_s$ , and  $B_s(p^s)$  represents the maximum benefit  $s$  can achieve at the given price  $p^s$ . A source  $s$  can be induced to solve maximization (3) by bandwidth charging. For each  $p$ , a unique maximizer  $x_s(p)$  exists since  $U_s$  is strictly concave.

In general  $(x_s(p), s \in S)$  may not be primal optimal, but by the duality theory, there exists a  $p^* \geq 0$  such that  $(x_s(p^*), s \in S)$  is indeed primal optimal. Moreover the source rates  $x^* \in I$  are primal optimal and the bandwidth prices  $p^* \geq 0$  dual optimal if and only if

$$\max_{x \in I} L(x, p^*) = L(x^*, p^*) = \min_{p \geq 0} L(x^*, p)$$

Hence we will focus on solving the dual problem (5). Once we have obtained the minimizing prices  $p^*$  the primal optimal source rates  $x^*$  can be obtained by individual sources  $s$  by solving (3), a simple maximization. The important point to note is that, given  $p^*$ , individual sources  $s$  can solve (3) *separately without the need to coordinate with other sources*. In a sense  $p^*$  serves as a coordination signal that aligns individual optimality of (3) with social optimality of (1).

We will solve the dual problem using gradient projection method (e.g., [12, 2]) where link prices are adjusted in opposite direction to the gradient  $\nabla D(p)$ :

$$p_l(t+1) = [p_l(t) - \gamma \frac{\partial D}{\partial p_l}(p(t))]^+ \quad (6)$$

Here  $\gamma > 0$  is a step size, and  $(z)^+ = \max\{z, 0\}$ . Let  $x_s(p)$  be the unique maximizer in (3). Then

$$\frac{\partial D}{\partial p_l}(p) = c_l - x^l(p) \quad (7)$$

where  $x^l(p) := \sum_{s \in S(l)} x_s(p)$  is the aggregate source rate at link  $l$ . Hence we obtain the following price adjustment rule for link  $l \in L$ :

$$p_l(t+1) = [p_l(t) + \gamma(x^l(p(t)) - c_l)]^+ \quad (8)$$

This indeed is consistent with the law of supply and demand: if the demand  $x^l(p(t)) = \sum_{s \in S(l)} x_s(p(t))$  for bandwidth at link  $l$  exceeds the supply  $c_l$ , raise price  $p_l(t)$ ; otherwise reduce price  $p_l(t)$ . As with (3) the decentralized nature of (8) is striking: though the dual problem is not separable in  $p$ , given aggregate source rate  $x^l(p(t))$  that goes through link  $l$ , the adjustment algorithm (8) is completely distributed and can be implemented by individual links using only local information.

### 3 ABR control as distributed computation

#### 3.1 Inter-processor communication

The above discussion leads to the useful view of treating the network links  $l$  and the sources  $s$  as processors in a distributed computation system to solve the *dual* problem (5). In each iteration, sources  $s$  individually solve (3) and communicate their results  $x_s(p)$  to links  $l \in L(s)$  on its path. Links  $l$  then update their prices  $p_l$  according to (8) and communicate the new prices to sources  $s$ , and the cycle repeats. We now explain how the communication among network links and ABR sources can be implemented using features defined in the ABR standards [14].

From time to time a source sends an RM cell which will be turned around at the destination and returned to the source. The explicit rate (ER) field of the RM cell is reset to zero when the RM cell leaves the source. As it passes through each link  $l$  on its forward path the current price  $p_l$  is added to the ER value. When it reaches the destination the ER field now contains the total bandwidth cost  $p^s = \sum_{l \in L(s)} p_l$  along the (forward) path. The RM cell is then returned to the source, where the ER field is not modified on return trip. Hence instead of being feedback an explicit rate at which a source should transmit, as in the conventional proposals, the source now receives current bandwidth cost along its path. Individual sources can then freely choose rates that maximize their benefits based on their own utility. In the reverse direction the sources can communicate their computation results  $x_s(p^s)$  explicitly to the links in their paths using the Current Cell Rate (CCR) field of a RM cell.

#### 3.2 Asynchronous environment

In reality the ABR sources may be located at different distances from the network links. Network state (prices in our case) may be probed by different sources at different rates, and feedbacks may reach different sources after different, and random, delays. These complications make the distributed computation system that consists of links and sources *totally asynchronous* [2, Chapter 6]. In such a system some processors may compute faster and execute more iterations than others, some processors may communicate more frequently than others, and the communication delays may be substantial and unpredictable.

Let  $T_s \subseteq \{1, 2, \dots\}$  be a set of times at which source  $s$  updates its rates based on its current knowledge of bandwidth prices along its path. At time  $t \in T_s$  the bandwidth prices  $(p_l(\tau_l^s(t)))$ ,  $l \in L(s)$  available at source  $s$  are the prices computed by the links  $l \in L(s)$  at earlier times  $\tau_l^s(t)$ , where  $0 \leq \tau_l^s(t) \leq t$  for all  $t \in T_s$ . The difference  $t - \tau_l^s(t)$  represents the communication delay from link  $l$  to source  $s$ . Note that this delay depends on  $(l, s, t)$  and can be different for different link-source pairs and at different times. At an update time  $t \in T_s$ , source  $s$  solves (3) with bandwidth cost  $\sum_{l \in L(s)} p_l(\tau_l^s(t))$ , and assign the unique maximizer to be the source rate in the next time period. At times  $t \notin T_s$  between updates source rates are unchanged.

Similarly let  $\Theta_l \subseteq \{1, 2, \dots\}$  be a set of times at which link  $l$  adjusts its price. At an update time  $t \in \Theta_l$  link  $l$  has available source rates  $(x_s(\theta_s^l(t)))$ ,  $s \in S(l)$  computed by  $s \in S(l)$  at earlier times  $\theta_s^l(t)$ , where  $0 \leq \theta_s^l(t) \leq t$  for all  $t \in \Theta_l$ . Again the difference  $t - \theta_s^l(t)$  represents the communication delay from source  $s$  to link  $l$ , and can be different for different  $s$  and  $l$  and at different times  $t$ .

We now present an asynchronous distributed algorithm for ABR control.

#### 3.3 Algorithm: Asynchronous Gradient Projection

Source  $s$ 's algorithm:

1. At each update time  $t \in T_s$  source  $s$  chooses a new rate based on its current knowledge of prices:

$$x_s(t+1) = \arg \max_{m_s \leq x_s \leq M_s} U_s(x_s) - x_s \sum_{l \in L(s)} p_l(\tau_l^s(t))$$

It then transmits at this rate until the next update, i.e.,  $x_s(t+1) = x_s(t)$  for  $t \notin T_s$ .

2. When a RM cell returns source  $s$  replaces the price in its local memory with the value in the ER field.

- From time to time source  $i$  transmits a RM cell with the CCR field set to the current source rate  $x_s(t)$  and ER field to zero.

#### Link $l$ 's algorithm:

- At each update time  $t \in \Theta_l$  link  $l$  computes a new price

$$p_l(t+1) = [p_l(t) + \gamma(\sum_{s \in S(l)} x_s(\theta_s^l(t)) - c_l)]^+.$$

At times  $t \notin \Theta_l$ ,  $p_l(t+1) = x_l(t)$ .

- When a RM cell from source  $s$  comes by (in the forward path), link  $l$ :

- increments the ER field by the current price  $p_l(t)$ .
- replaces the current copy of rate  $x_s$  by the value of the CCR field.

## 4 Experimental results

We present in this section some preliminary experiments to illustrate the behavior of our algorithm.

### 4.1 Simulation model

The network we consider is adapted from [3] and consists of three switching nodes and three ABR sources, as depicted in figure 1. The three ABR sources transmit data cells and RM cells across the network to their respective destinations.

Source 1 turns on at time 0, source 2 at 20 ms, source 3 at 40ms, and each remains on for 60 ms. Each source has a minimum cell rate (MCR) of 0.150 Mbps and a peak cell rate (PCR) equal to the link rate of 150 Mbps. While on a source sends an RM cell after every 31 data cells (i.e.,  $N_{rm} = 32$ ).

The utility functions of sources  $i$  are characterized by the parameter  $a_i$  and are given by  $U_i(x_i) = a_i \ln(1 + x_i)$ ,  $i = 1, 2, 3$ . The corresponding demand function, i.e., the maximizer of  $\max_{x_i \geq m_i} U_i(x_i) - qx_i$  as a function of price  $q$ , is  $D_i(q) = \frac{a_i}{q} - 1$ .

The switching nodes are non-blocking and output buffered. Their buffers are sufficiently large to avoid cell loss. The switches are connected to the sources and each other via 150 Mbps links. They have an update interval of 100 cell transmission time slots or  $282\mu s$ . At the end of an update interval the switches calculate new bandwidth price for each link out of the switch.

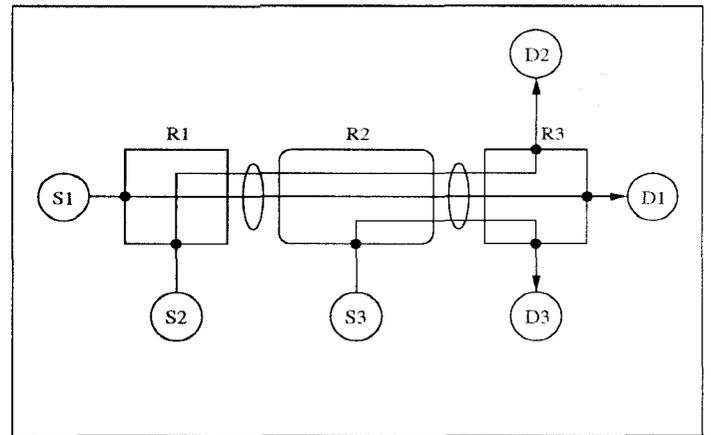


Figure 1: Network Topology

### 4.2 Results

Each network link uses the difference between aggregate CCRs and 90% of link capacity to calculate price according to (8). The step size  $\gamma$  in the price adjustment rule (8) is set to  $1 \times 10^{-6}$ . All sources have same utility function with  $a_i = 1 \times 10^5$ ,  $i = 1, 2, 3$ .

Figure 2 shows the Allowable Cell Rate v. time for the system simulated. There are three interesting features of this graph. First the source ACRs change quickly in response to sources activating/leaving the system. Second when the steady state is reached after each brief disturbance each source gets an equal share of the available capacity of 135 Mbps (90% of the link capacity), as they should since they have the same utility function. The allocation is also max-min fair. Third source 1's ACR exhibits a relatively large fluctuation during the first 20 ms of the simulation before source 2 turns on. This is because the demand is very sensitive (large derivative) to price at low prices when the network is uncongested for the  $D_i$  we use. We have found empirically that if the sources use the *maximum* bandwidth price along their forward path to calculate their transmission rate, instead of the *sum* of the bandwidth prices of all links in the forward path, then this oscillation can be eliminated.

Figure 3 shows that the prices of the link bandwidths increases as the demand increases. For the first 40 ms the bandwidth prices for link 1-2 and link 2-3 are approximately equal because during this period the two links both act as bottlenecks for sources 1 and 2. When source 3 becomes active, link 2-3 becomes the bottleneck link and so the price of bandwidth on link 1-2 becomes 0. Note that the prices converge quickly to the minimizer of the dual problem after a disturbance.

Buffer processes, not shown here, are empty for almost all of the

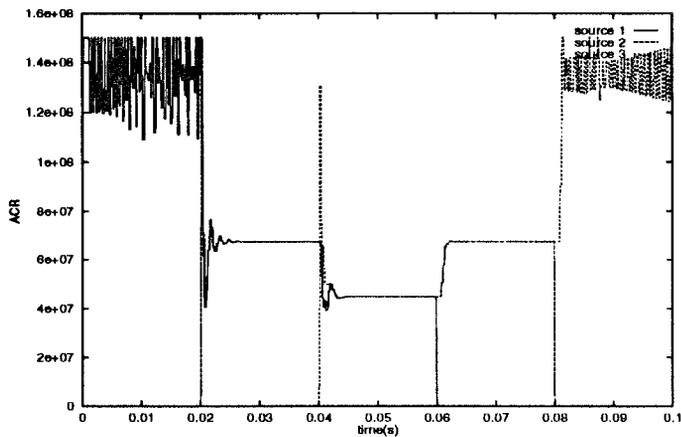


Figure 2: Source ACRs

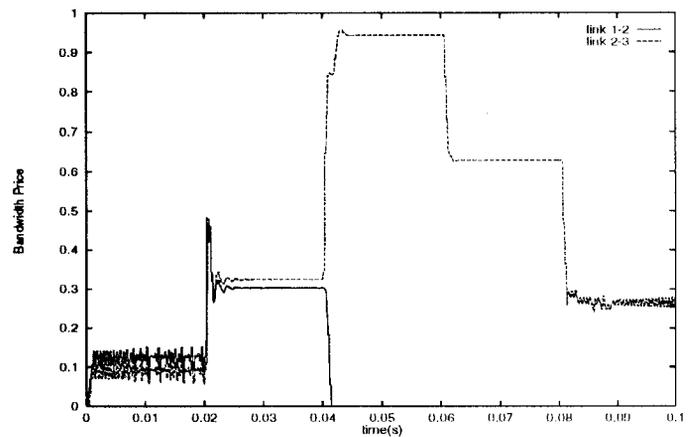


Figure 3: Link Bandwidth Prices

time. It becomes non-empty, exhibiting a very brief spike of approximately 80 cells, only during the transient overload that accompany a new source turning on.

## 5 Conclusion

We have presented an optimization approach to ABR rate control where the objective is to maximize the sum of source utilities. The approach is motivated by the dual problem which suggests naturally to use network links and ABR sources as processors in a distributed computation system to solve the dual problem by gradient projection method. We have shown how the necessary communication among these processors can be implemented by resource management cells in the ABR standard. Our approach necessitates a charging mechanism to provide cost incentive for sources to participate in the distributed computation, or put in a more positive tone, it integrates pricing and flow control. Alternatively the network feedbacks, called 'prices' in our discussion, can simply be treated as a control signal to coordinate sources decisions and may not be a component of the tariff a user faces. Preliminary simulation results of the proposed scheme are promising though more extensive measures are necessary.

## References

- [1] L. Benmohamed and S. M. Meerkov. Feedback control of congestion in store-and-forward networks: the case of a single congested node. *IEEE/ACM Transactions on Networking*, 1(6):693–707, December 1993.
- [2] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation*. Prentice-Hall, 1989.
- [3] Costas Courcoubetis, Vasilios A. Siris, and George D. Stamoulis. Integration of pricing and flow control for ABR services in ATM networks. *Proceedings of Globecom'96*, November 1996.
- [4] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397–413, August 1993.
- [5] R. G. Gallager and S. J. Golestani. Flow control and routing algorithms for data networks. In *Proceedings of the 5th International Conf. Comp. Comm.*, pages 779–784, 1980.
- [6] Jamal Golestani and Supratik Bhattacharyya. End-to-end congestion control for the Internet: A global optimization framework. Preprint, 1997.
- [7] V. Jacobson. Congestion avoidance and control. *Proceedings of SIGCOMM'88, ACM*, August 1988.
- [8] F. P. Kelly. Charging and rate control for elastic traffic. Preprint, 1997.
- [9] David E. Lapsley and Michael Rumsewicz. Improved buffer efficiency via the No Increase flag in EFCI flow control. In *Proceedings of the IEEE ATM '96 Workshop*, August 1996.
- [10] Steven H. Low. Equilibrium allocation of variable resources for elastic traffics. In *Proceedings of INFOCOM'98*, San Francisco, CA, USA, March 1998.
- [11] Steven H. Low and David E. Lapsley. An optimization approach to reactive congestion control. Submitted for publication, 1998.
- [12] David G. Luenberger. *Linear and Nonlinear Programming, 2nd Ed.* Addison-Wesley Publishing Company, 1984.
- [13] C. E. Rohrs, R. A. Berry, and S. J. O'Halek. A control engineer's look at atm congestion avoidance. In *Proceedings IEEE Globecom '95*, pages 1089–1094, Singapore, November 1995.
- [14] S. Sathaye. *Traffic Management Specification v 4.0*. ATM Forum Traffic Management Group, October 1996.
- [15] Scott Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communications*, 13(7):1176–1188, 1995.