# CONVERGENCE OF ASYNCHRONOUS OPTIMIZATION FLOW CONTROL

Steven H. Low and David Lapsley *
*Department of Electrical & Electronic Engineering*
*University of Melbourne*
*Australia*
{slow,lapsley}@ee.mu.oz.au

**Abstract** We proposed earlier an optimization approach to reactive flow control where the objective of the control is to maximize the total source utility over their transmission rates. The source utility functions model their valuation of bandwidth and can be different for different sources. The control mechanism is derived as a gradient projection algorithm to solve the dual problem. In this paper we generalize the algorithm and the convergence result to an asynchronous setting where the computations at and the communications among the links and sources are uncoordinated and based on possibly outdated information.

**Keywords:** Optimization flow control, congestion control, congestion pricing, asynchronous algorithm.

## 1. INTRODUCTION

We have proposed previously an optimization approach to flow control where the control mechanism is derived as a gradient projection algorithm to solve (the dual of) a global optimization problem [10; 11]. The purpose of this paper is to show that the basic algorithm converges in both synchronous and asynchronous settings.

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 first author acknowledges the support of the Australian Research Council under grant S499705, the second author acknowledges the Australian Commonwealth Government for their Australian Postgraduate Awards, and both acknowledge the financial support of Melbourne IT, Australia.

D. H. K. Tsang et al. (eds.), *Broadband Communications*

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. The key to a distributed solution is to consider the dual problem that decomposes the task into simple local computations to be executed at individual links and sources.

The algorithm takes the familiar form of reactive flow control. Based on the local *aggregate* source rate each link $l \in L$ calculates a 'price' $p_l$ for a unit of bandwidth. A source $s$ is fed back the scalar price $p^s = \sum p_l$, where the sum is taken over all links that $s$ uses, and it chooses a transmission rate $x_s$ that maximizes its own benefit $U_s(x_s) - p^s x_s$, utility minus the bandwidth cost. These individually optimal rates $(x_s(p^s), s \in S)$ may not be socially optimal for a general price vector $(p_l, l \in L)$, i.e., they may not maximize the total utility. The algorithm iteratively approaches a price vector $(\hat{p}_l, l \in L)$ that aligns individual and social optimality such that $(x_s(\hat{p}_l), s \in S)$ indeed maximizes the total utility. In order words, the price $\hat{p}^s$ represents the complete congestion information source $s$ needs for its control decision.

The basic algorithm is presented in [10] and a preliminary prototype is briefly discussed in [11]. The basic algorithm requires communication of link prices to sources and source rates to links. This requirement is greatly simplified in [13,12], as follows. In [13] we prove that a link can simply set its price to a fraction of its buffer occupancy, thus eliminating the need for explicit communication from sources to links. This can be seen as have the links estiamte the gradient using local information in carrying out the gradient projection algorithm. In [12], we describe a marking scheme, inspired by [ 6], that achieves the communication from links to sources using only binary feedback. The result is a variant of Random Early Detection (RED) scheme [ 4], that not only stabilizes network queues, as RED does, but does so in a way that optimizes a global measure of performance.

Optimization based flow control have also been proposed in [5; 7; 3; 8; 9; 6]. All these works, as ours, motivate flow control by an optimization problem and derive their control mechanisms as a solution to the optimization problem. They differ in their choice of objective functions or their solution approaches, and result in rather different flow control mechanisms to be implemented at the sources and the network links. In particular both [8; 9] and our work solve the same optimization problem of maximizing aggregate utility over source transmission rates. The two works however differ in their solution approach, which lead to different algorithms and their implementation through marking [ 6; 12]. See [14] for a detailed comparison.

The present paper is structured as follows. In Section 2 we present the optimization problem and its dual that motivate our approach. In Section 3 we briefly derive a synchronous solution and present its convergence property.

In Section 4 we extend the synchronous algorithm and its convergence to an asynchronous setting. All proofs are omitted due to space limitation and can be found in [14].

## 2. MODEL

Consider a network that consists of a set $L = \{1, \ldots, L\}$ of *unidirectional* links of capacity $c_l$, $l \in L$. The network is shared by a set $S = \{1, \ldots, S\}$ 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, $U_s : \Re_+ \to \Re$ is a utility function, $m_s \geq 0$ and $M_s \leq \infty$ are the minimum and maximum transmission rates, respectively, required by source $s$. Source $s$ attains a utility $U_s(x_s)$ when it transmits at rate $x_s$ that satisfies $m_s \leq x_s \leq M_s$. We assume $U_s$ is increasing and strictly concave in its argument. 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. For each link $l$ let $S(l) = \{s \in S \mid l \in L(s)\}$ be the set of sources that use link $l$. Note that $l \in L(s)$ if and only if $s \in S(l)$.

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

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

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

The constraint (2) says that the total source rate at any link $l$ is less than the capacity. 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.

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 coordination 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, 15]:

$$\textbf{D:} \quad \min_{p \geq 0} \quad D(p) = \sum_s B_s(p^s) + \sum_l p_l c_l \tag{3}$$

where

$$B_s(p^s) = \max_{x_s \in I_s} U_s(x_s) - x_s p^s \tag{4}$$

$$p^s = \sum_{l \in L(s)} p_l. \tag{5}$$

The first term of the dual objective function $D(p)$ is decomposed into $S$ separable subproblems (4–5). 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 $p^s$. We shall see that this scalar $p^s$ summarizes all the congestion information source $s$ needs to know. A source $s$ can be induced to solve maximization (4) by bandwidth charging. For each $p$, a unique maximizer, denoted by $x_s(p)$, 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. Hence we will focus on solving the dual problem (3). Once we have obtained the minimizing prices $p^*$ the primal optimal source rates $x^* = x(p^*)$ can be obtained by individual sources $s$ by solving (4), a simple maximization (see below). The important point to note is that, given $p^*$, individual sources $s$ can solve (4) *separately without the need to coordinate with other sources.* In a sense $p^*$ serves as a coordination signal that aligns individual optimality of (4) with social optimality of (1). Note that despite the notation, a source $s$ does not require the vector price $p$, but only a scalar $p^s = \sum_{l \in L(s)} p_l$ that represents the sum of link prices on its path.

Indeed the unique maximizer $x(p)$ for (4) can be given explicitly, from the Kuhn–Tucker theorem, in terms of the marginal utility:

$$x_s(p) \quad = \quad [U_s'^{-1}(p)]_{m_s}^{M_s} \tag{6}$$

where $[z]_a^b = \max\{a, \min\{b, x\}\}$. Here $U_s'^{-1}$ is the inverse of $U_s'$, which exists over the range $[U_s'(M_s), U_s'(m_s)]$ since $U_s'$ is continuous and $U_s$ *strictly* concave. It is indeed the demand function in economics. It is illustrated in Figure 1. Let $x(p) = (x_s(p), s \in S)$.

In this paper, we abuse notation and use $x_s(\cdot)$ both as a function of scalar price $p \in \Re_+$ and of vector price $p \in \Re_+^{|L|}$. When $p$ is a scalar, $x_s(p)$ is given by (6). When $p$ is a vector, $x_s(p) = x_s(p^s) = x_s(\sum_{l \in L(s)} p_l)$. The meaning should be clear from the context.

## 3. SYNCHRONOUS DISTRIBUTED ALGORITHM

In [10; 14] we propose to solve the dual problem using the gradient projection algorithm where link prices are adjusted in opposite direction to the gradient $\nabla D(p)$ whose $l$-th component is given by:

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

where $x^l(p) := \sum_{s \in S(l)} x_s(p)$ is the aggregate source rate at link $l$. The synchronous algorithm there takes the following form. In each iteration, each

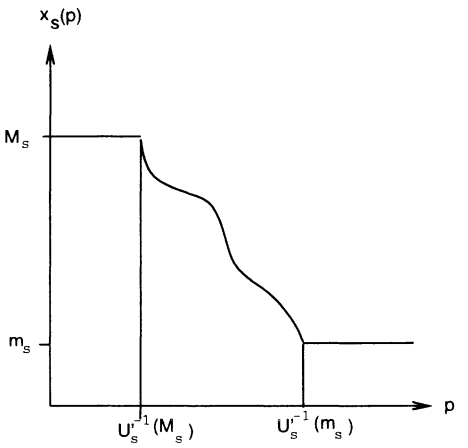


*Figure 1* Source rate $x_s(p)$ as a function of (scalar) price $p$.

link $l$ adjusts its price according to:

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

where $[z]^+ = \max\{z, 0\}$. This is consistent with the law of supply and demand: if the demand $x^l(p(t))$ for bandwidth at link $l$ exceeds the supply $c_l$, raise price $p_l(t)$; otherwise reduce price $p_l(t)$. Each source adjusts its rate to:

$$x_s(t+1) \quad = \quad [U_s'^{-1}(p^s(t))]_{m_s}^{M_s} \tag{9}$$

They then exchange their results: each link receives from all sources using that link their sources rates and each source receives a scalar price equal to the sum of the link prices along its path. The iteration is repeated until it converges.

We prove in [13; 14] that the algorithm generates a sequence that approaches the optimal rate allocation, provided the following conditions are satisfied:

C1: On the interval $I_s = [m_s, M_s]$, the utility functions $U_s$ are increasing, strictly concave, and twice continuously differentiable.

C2: The curvatures of $U_s$ are bounded away from zero on $I_s$: $-U_s''(x_s) \geq 1/\overline{\alpha}_s > 0$ for all $x_s \in I_s$.

These conditions imply that the dual objective function is Lipschitz which guarantees the convergence of gradient projection algorithms. Define $\overline{L} := \max_{s \in S} |L(s)|$, $\overline{S} := \max_{l \in L} |S(l)|$, and $\overline{\alpha} := \max\{\overline{\alpha}_s,\ s \in S\}$. In words $\overline{L}$ is the length of a longest path used by the sources, $\overline{S}$ is the number of sources sharing a most congested link, and $\overline{\alpha}$ is the upper bound on all $-U_s''(x_s)$.

**Theorem 1** *Suppose assumptions C1–C2 hold. Provided that the stepsize $\gamma$ satisfies $0 < \gamma < 2/\overline{\alpha}\overline{LS}$, starting from any initial rates $m \leq x(0) \leq M$ and prices $p(0) \geq 0$, every accumulation point $(x^*, p^*)$ of the sequence $(x(t), p(t))$ generated by algorithm (8–9) are primal–dual optimal. That is, $x^*$ gives the source rates that maximize total utility and $p^*$ the shadow bandwidth prices.*

# 4. ASYNCHRONOUS DISTRIBUTED ALGORITHM

The synchronous model of the last section assumes that updates at the sources and the links are synchronized to occur at times $t = 1, 2, \ldots$. In this section we will extend the model to an asynchronous setting which better resembles the reality of large networks. In such networks 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, e.g., the Resource Management (RM) cells in an ATM networks are sent at different rates by different sources. Feedbacks may reach different sources after different, and variable, delays. These complications make our distributed computation system consisting of links and sources asynchronous. 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 $T_l^1 \subseteq \{1, 2, \ldots, \}$ be a set of times at which link $l$ adjusts its price based on its current knowledge of source rates and $T_s^2 \subseteq \{1, 2, \ldots, \}$ a set of times at which source $s$ updates its rate. The asynchronous algorithm is similar to the synchronous algorithm, except that computations and communications by sources and links are not coordinated and the computations are carried out using possibly outdated information.

**Algorithm: Asynchronous Gradient Projection**

**Link $l$'s algorithm:**

1. *From time to time link $l$ receives source rates from sources that go through link $l$. Link $l$ replaces the oldest rates in its local memory with the newly received rates.*

2. *At each update time $t \in T_l^1$, link $l$ computes an estimate $\lambda_l(t)$ of $\partial D/\partial p_l(p(t))$ (see (11–14) below) and adjusts its price according to*

$$p_l(t+1) \quad = \quad [p_l(t) - \gamma\lambda_l(t)]_0^{\bar{p}_l}$$

*where $\bar{p}_l$ is a large constant satisfying $\bar{p}_l > \max_{s \in S(l)} U_s'(m_s)$ and $[a]_0^{\bar{a}} = \min\{\max\{0, a\}, \bar{a}\}$. At times $t \notin T_l^1$, $p_l(t+1) = p_l(t)$.*

3. *From time to time link $l$ communicates the current price to sources that go through link $l$.*

**Source $s$'s algorithm:**

1. *From time to time source $s$ receives bandwidth prices fedback from links in its path. Source $s$ replaces the oldest prices in its local memory with the newly received ones.*

2. *At each update time $t \in T_s$ source $s$ chooses a new rate based on its current estimate $\hat{p}^s(t)$ of prices (see (16–18) below):*

$$x_s(t+1) \;=\; x_s(\hat{p}^s(t))$$

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

3. *From time to time source $s$ communicates the current source rate to links in its path.*

We now describe more precisely the update steps in the above algorithm. Link $l$ updates its price at times $t \in T_l^1$ according to

$$p_l(t+1) \;=\; [p_l(t) - \gamma\lambda_l(t)]_0^{\bar{p}_l} \tag{10}$$

where (cf. (7))

$$\lambda_l(t) \;=\; c_l - \hat{x}^l(t) \tag{11}$$

$$\hat{x}^l(t) \;=\; \sum_{s\in S(l)} \hat{x}_{ls}(t) \tag{12}$$

$$\hat{x}_{ls}(t) \;=\; \sum_{t'=t-t_0}^{t} a_{ls}(t',t)\; x_s(t'), \quad s \in S(l) \tag{13}$$

with

$$\sum_{t'=t-t_0}^{t} a_{ls}(t',t) \;=\; 1, \quad \forall t, \forall l, s \text{ with } s \in S(l). \tag{14}$$

In (10) the projection onto the range $[0, \bar{p}_l]$, instead of $[0, \infty)$, can be motivated by the fact that in practice a price must be represented by a finite number of bits. Moreover, since $\bar{p}_l > \max_{s\in S(l)} U_s'(m_s)$, the projection imposes no restriction on the source rates. In (11–12), $\hat{x}^l(t) = \sum_{s\in S(l)} \hat{x}_{ls}(t)$ is the aggregate estimated source rates. Note that the estimate $\hat{x}_{ls}(t)$ depends on $(l, s, t)$ and can be different for different link–source pairs and at different times. It is

obtained by 'averaging' (convex sum) over the past source rates (see (13–14)). This model is very general and includes in particular the following two popular types of policies:

- **Latest data only:** only the last received rate $x_s(\tau)$, for some (possibly *unknown*) $\tau \in \{t-t_0, ..., t\}$, is used to estimate $\hat{x}_{ls}(t)$, i.e., $a_{ls}(t', t) = 1$ if $t' = \tau$ and 0 otherwise.
- **Latest average:** only the average over the latest $k$ received rates is used in the estimate $\hat{x}_{ls}(t)$, i.e., $a_{ls}(t', t) > 0$ for $t' = \tau - k + 1, \ldots, \tau$ and 0 otherwise, for some (possibly *unknown*) $\tau \in \{t - t_0, ..., t\}$.

The interpretation in both cases is that rates $x_s(t')$ for $t' > \tau$ have not been received at link $l$ by time $t$, and rates $x_s(t')$ for $t' < \tau$ or for $t' \leq \tau - k$ have been discarded.

In the following, we abuse notation and use $x_s(\cdot)$ both as a function of time, to denote source rate at time $t$ under the algorithm, and as a function of price given by (6). The meaning should be clear from the context.

Source $s$ updates its rate at times $t \in T_s^2$ according to

$$x_s(t) \;=\; x_s(\hat{p}^s(t)) \tag{15}$$

where $x_s(\cdot)$ is given by (6), and

$$\hat{p}^s(t) \;=\; \sum_{l \in L(s)} \hat{p}_{ls}(t) \tag{16}$$

$$\hat{p}_{ls}(t) \;=\; \sum_{t'=t-t_0}^{t} b_{ls}(t', t)\, p_l(t'), \quad l \in L(s) \tag{17}$$

with

$$\sum_{t'=t-t_0}^{t} b_{ls}(t', t) \;=\; 1, \quad \forall t, \forall l, s \text{ with } l \in L(s). \tag{18}$$

In (15–16) the source computation is the same as in the synchronous case except that it is based on its current estimate $\hat{p}^s(t)$ of link prices. As in the link algrithm the estimated link price $\hat{p}_{ls}(t)$ is obtained by 'averaging' over the past available prices (see (17–18)), and can depend on $(l, s, t)$. Again the 'averaging' model is very general and include the policy of using only the last received price or the average over the last $k$ prices; see above.

Note that (13) and (17) above tacitly assume that the one-way delay between any $(l, s)$ pair is no more than $t_0$.

Our main result states that the difference between the various estimates and their true values converges to zero and that the algorithm yields the optimal rate allocation, provided the following additional assumption is satisfied:

C3: For all links $l$ and sources $s$, the time between consecutive updates (i.e., the difference between consecutive elements of $T_l^1$ or $T_s^2$) is bounded.

**Theorem 2** *The conclusion of Theorem 1 holds provided assumptions C1–C3 are satisfied and the stepsize $\gamma$ is sufficiently small.*

# 5. CONCLUSION

We have presented an asychronnous model for optimization flow control proposed in [10], where the objective of the control is to maximize total user utilities. We have shown that both the synchronouns and the asynchronous algorithms converge under very mild conditions on the utility functions.

We close with extensions on the work presented in this paper. The basic algorithm presented here requires communication between links and sources. In [13] we prove that it is possible to eliminate the need for explicit communication from sources to links. In [12], we describe a marking scheme that simplifies the communication in the reverse direction to binary feedback. These simplifications combine to yield a variant of RED scheme, called REM (Early Random Marking), that is applicable to Interent using the proposed explicit congestion notification bit in IP header. The optimization framework in which REM is derived has two advantages. First, though it may not be possible, nor critical, that optimality is exactly attained in a real network, the optimization framework offers a means to explicitly steer the *entire* network towards a desirable operating point. Second it makes possible a systematic method to design and refine practical flow control schemes, which can be treated simply as implementations of a certain optimization algorithm, where modifications to the flow control mechanism is guided by modifications to the optimization algorithm. For instance, it is well known that Newton algorithm has much faster convergence than gradient projection algorithm. By replacing the gradient projection algorithm presented in [10; 14] by the Newton algorithm we derive in [1] a practical Newton–like flow control scheme that can be proved to maintain optimality, has the same communication requirement as the original scheme but enjoys a much better convergence property.

# References

[1] Sanjeewa Athuraliya and Steven Low, "Newton–like algorithm for optimization flow control," Submitted for publication, 1999.

[2] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation.* Prentice-Hall.

[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] R. J. Gibbens and F. P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35, 1999.

[7] Jamal Golestani and Supratik Bhattacharyya. End-to-end congestion control for the Internet: A global optimization framework. In *Proceedings of International Conf. on Network Protocols (ICNP)*, October 1998.

[8] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[9] Frank P. Kelly, Aman Maulloo, and David Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of Operations Research Society*, 49(3):237–252, March 1998.

[10] David E. Lapsley and Steven H. Low. An optimization approach to ABR control. In *Proceedings of the ICC*, June 1998.

[11] David E. Lapsley and Steven H. Low. An IP Implementation of Optimization Flow Control. In *Proceedings of the Globecom'98*, November 1998.

[12] David Lapsley and Steven Low, "Random early marking: An optimisation approach to internet congestion control," in *Proceedings of IEEE ICON '99*, September 1999.

[13] Steven H. Low. Optimization flow control with on-line measurement. In *Proceedings of the ITC*, volume 16, June 1999.

[14] Steven H. Low and David E. Lapsley. Optimization flow control, I: basic algorithm and convergence. IEEE/ACM Transactions on Networking, to appear 1999.

[15] David G. Luenberger. *Linear and Nonlinear Programming, 2nd Ed.* Addison-Wesley Publishing Company, 1984.

[16] John N. Tsitsiklis and Dimitri P. Bertsekas. Distributed asynchronous optimal routing in data networks. *IEEE Transactions on Automatic Control*, 31(4):325–332, April 1986.