OPTIMIZATION FLOW CONTROL WITH NEWTON–LIKE ALGORITHM *

Sanjeewa Athuraliya and Steven Low
Department of EEE, University of Melbourne,
Parkville, Vic 3052, Australia

Abstract

We proposed earlier an optimization approach to reactive flow control where the objective of the control is to maximize the aggregate utility of all sources over their transmission rates. The control mechanism is derived as a gradient projection algorithm to solve the dual problem. In this paper we extend the algorithm to a scaled gradient projection. The diagonal scaling matrix approximates the diagonal terms of the Hessian and can be computed at individual links using the same information required by the unscaled algorithm. We prove the convergence of the scaled algorithm and present simulation results that illustrate its superiority to the unscaled 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 [14, 13, 17]. The solution is decomposed into simple algorithms that are executed at individual links and sources using 'local' information. It is well known that Newton method, where the gradient is scaled by the inverse of the second derivative matrix, typically enjoys a much faster convergence than gradient projection algorthim. For us, however, the exact Newton method will require non–local information and hence cannot be easily implemented in a large network. The purpose of this paper is to describe an approximate Newton method to solve the dual optimization problem using only diagonal scaling, and illustrate its behavior with preliminary simulation results.

Specifically consider a network that consists of a set L of unidirectional links of capacities 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 coupling of sources through shared links. The key is to solve the dual problem that decomposes the task into simple local computations 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 aggregate utility. The algorithm iteratively approaches a price vector $(p_l^*, l \in L)$ that aligns individual and social optimality.

The basic algorithm to solve the dual problem presented in [14] is a gradient projection method. A preliminary prototype based on this algorithm is discussed in [13]. Its convergence is proved in [18] in both synchronous and asynchronous settings. The basic algorithm requires communication of link prices to sources and source rates to links and is thus not implementatble in the current Internet. In [15], we describe a Random Early Marking (REM) scheme which can be regarded as a practical implementation of the basic algorithm in [14, 18] using binary feedback. It can be implemented, e.g., with the proposed explicit congestion notification (ECN) bit in the IP (Internet Protocol) header [6, 20].

In this paper, we generalize the basic algorithm of gradient projection to an approximate Newton algorithm that has a better convergence property.

There is a tremendous literature on flow control, including early schemes based on practical experiences, e.g., [10, 7], and recent schemes based on control theory, e.g., [1, 3, 4]. Optimization based flow control have been proposed in [9, 5, 11, 12, 8, 14, 16, 18] All these works motivate flow control by an optimization problem and derive their control mechanisms as solutions to the optimization problem. They differ in their choice of objective functions or their solution

*The first author acknowledges the Australian Commonwealth Government for their Australian Postgraduate Award, and the second author acknowledges the support of the Australian Research Council under grant S499705.

0-7803-5796-5/99/$10.00 © 1999 IEEE

approaches, and result in rather different flow control mechanisms to be implemented at the sources and the network links.

The present paper is structured as follows. In Section 2 we review our optimization framework and describe the Newton like algorithm. In Section 3 we first show that the algorithm converges and then show, through simulations, that it converges significantly faster than the gradient projection algortihm of [14, 16]. All proofs are omitted due to space limitation.

2 Model and algorithm

2.1 Model

Consider a network that consists of a set $L = \{1, \ldots, L\}$ of *unidirectional* links of capacities c_l, $l \in L$. [1] 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 $m_s \leq x_s \leq M_s$. 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 that U_s is increasing, *strictly* concave, and twice continuously differentiable on $I_s[m_s, M_s]$. For each link l let $S(l) = \{s \in S \mid l \in L(s)\}$ be the set of sources that use link l. 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:

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

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

The constraint (2) says that the aggregate 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], [19]:

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

where

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

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

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. 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)[2].

Indeed the unique maximizer $x(p)$ for (4) can be given explicitly, from the Kuhn–Tucker theorem, in terms of the marginal utility[3]:

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

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

[1] We abuse notation and use the same symbol to denote both a set and its cardinality when there is no danger of confusion.

[2] 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; see below.

[3] 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.

2.2 Algorithm

In [14, 16] we propose to solve the dual problem using the gradient projection algorithm where link prices are adjusted in opposite direction to the gradient $\nabla D(p(t))$:

$$p(t+1) = [p(t) - \gamma \nabla D(p(t))]^+ \qquad (7)$$

where $\gamma > 0$ is a step size and $[z]^+ \doteq \max\{z, 0\}$. By assumption, the utility functions are *strictly* concave and hence $\nabla D(p)$ indeed exists with its l-th component given by:

$$\frac{\partial D}{\partial p_l}(p) = c_l - x^l(p) \qquad (8)$$

where $x^l(p) := \sum_{s \in S(l)} x_s(p)$ is the aggregate source rate at link l. Hence the algorithm is decentralized: each link l can individually carry out the price adjustment given the aggregate source rates $x^l(t)$ at its link, and each source can individually compute its rate using (6) given the scalar price $p^s(t) = \sum_{l \in L(s)} p_l(t)$.

It is well known that Newton method, where the gradient is scaled by the inverse of the Hessian,

$$p(t+1) = [p(t) - \gamma[\nabla^2 D(p(t))]^{-1} \nabla D(p(t))]^+ \qquad (9)$$

typically converges much faster than the gradient projection algorithm (7). This price adjustment however is difficult to implement in a large network since the Hessian $\nabla^2 D(p)$ computation cannot be distributed to individual links, as a link may require the rates or the second derivatives of utilities of sources at other links [16, Lemma 2]. This is clearly not scalable. We propose instead an approximating positive definite diagonal scaling matrix $H(t)$ that can be computed at individual links using the same information available under the gradient projection algorithm.

First $H(t)$ retains only the diagonal terms of the Hessian and has zero off–diagonal terms. Second the diagonal terms are approximated by finite differences. From (8) the diagonal terms are

$$\begin{aligned}
[\nabla^2 D(p(t))]_{ll} &= -\sum_{s \in S(l)} \frac{\partial x_s}{\partial p_l}(p(t)) \\
&= -\sum_{s \in S(l)} x_s'(p^s(t)) \\
&\simeq -\sum_{s \in S(l)} \frac{x_s(t) - x_s(t-1)}{p^s(t) - p^s(t-1)}
\end{aligned}$$

where $x_s'(p^s(t))$ is the *total* derivative of the scalar function $x_s(\cdot)$ evaluated at the path price $p^s(t)$ at time t, and $x_s(t) = [U_s'^{-1}(p^s(t-1))]_{m_s}^{M_s}$ is the source rate at time t. This approximation is however difficult to implement as each link l requires the *path* price $p^s(t)$ of all sources $s \in S(l)$ going through l. Instead we use link price $p_l(t)$ as a substitute:

$$\begin{aligned}
[\nabla^2 D(p(t))]_{ll} &\simeq -\sum_{s \in S(l)} \frac{x_s(t) - x_s(t-1)}{p_l(t) - p_l(t-1)} \\
&= -\frac{x^l(t) - x^l(t-1)}{p_l(t) - p_l(t-1)}
\end{aligned}$$

Simulation results suggest that using $p_l(t)$ in place of $p^s(t)$ has a similar behavior. Finally we ensure that $H(t)$ is strictly positive definite by making its diagonal terms at least as large as an $\epsilon > 0$:

$$H_{kl}(t) = \begin{cases} \max\{\epsilon, -\frac{x^l(t) - x^l(t-1)}{p_l(t) - p_l(t-1)}\} & \text{if } k = l \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

We summarize.

Algorithm: Scaled gradient projection

At update times $t = 1, 2, \ldots$:

Link l's algorithm:

Given source rates $x_s(t), s \in S(l)$, at link l, compute a new price

$$p_l(t+1) = \left[p_l(t) + \gamma H_{ll}^{-1}(x^l(t) - c_l)\right]^+.$$

where $x^l(t) = \sum_{s \in S(l)} x_s(t)$ and H_{ll} is given by (10).

Source s's algorithm:

Given path price $p^s(t) = \sum_{l \in L(s)} p_l(t)$, choose a new source rate $x_s(t+1)$:

$$\begin{aligned}
x_s(t+1) &= \arg\max_{x_s \in I_s} U_s(x_s) - p^s(t) x_s \\
&= [U_s'^{-1}(p^s(t))]_{m_s}^{M_s}
\end{aligned}$$

3 Performance

In this section we first prove that the scaled gradient projection algorithm given in the last section converges. Then we illustrate through simulation studies that its convergence is superior to the unscaled algorithm.

3.1 Convergence

The scaled 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 the ∇D is Lipschitz which leads to the convergence of the algorithm. 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 and the step size satisfies* $0 < \gamma < 2\epsilon/\overline{\alpha}\overline{L}\overline{S}$. *Then starting from any initial rates* $m \leq x(0) \leq M$ *and prices* $p(0) \geq 0$, *every limit point* (x^*, p^*) *of the sequence* $(x(t), p(t))$ *generated by the algorithm are primal–dual optimal. That is,* x^* *gives the source rates that maximize aggregate utility and* p^* *the shadow bandwidth prices.*

Note that x^* is unique but p^* may not be unique. We have found from our simulation experience that, in practice, a step size γ much larger than the bound in the theorem can be used, e.g., in the simulation reported below, $\gamma = 1$. Moreover the scaled algorithm seems much less sensitive to γ than the unscaled algorithm.

3.2 Simulation results

We now present simulation study carried out for the network in Figure 1 shared by five connections, with sources Si and destinations Di, $i = 1, \ldots, 5$. Connection S1–D1

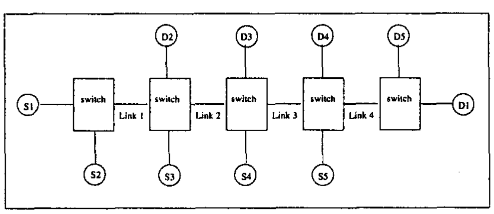

Figure 1: Network topology.

spanned all links 1, 2, 3, 4; connection S2–D2 spanned link 1; connection S3–D3 spanned link 2; connection S4–D4 spanned link 3; connection S5–D5 spanned link 4. Source S1 transmitted data from time 0s to time 300s. The start times of the other sources are staggered with S2 starting at time 40s, S3 at time 80s, S4 at time 120s, S5 at time 160s. Once turned on, sources S2, S3, and S4 remained active until time 240s, and S5 turned off earliest at time 200s. This enabled us to observe the dynamic behaviour of the algorithm as demand for bandwidth varies. The utility functions of the sources were set to $a_s \log(1 + x_s)$, with a_s equal to $1 \times 10^4, 5 \times 10^4, 7 \times 10^4, 6 \times 10^4, 2 \times 10^4$ for sources S1,S2,S3,S4 and S5 respectively. Notice that the longest connection S1–D1 was set to have the smallest marginal utility. The step size γ used to adjust the link prices was set to 1. A new link bandwidth price was calculated every 1s. The target bandwidth was set at 200 packets per 1s measuring interval.

Figure 2 shows the source rates for each source under the unscaled gradient projection algorithm. From time 0–40s, only source S1 was active. Its rate climbed steadily to the target bandwidth of 200 packets/s. From time 40s, source S2 became active whose rate, after an initial overshoot, stabilized to about 167 packets/s. This squeezed source S1's rate to about 33 packets/s. At these rates sources S1 and S2 had the same marginal utility. At times 80s, 120s, and 160s when sources S3, S4, S5 became active, similar dynamics were observed. S1's rate bounced back to 200 packets/s after all other sources had turned off.

Figure 3 shows the source rates under the scaled gradient projection algorithm. While the same kind of interaction among the sources occurred as under the unscaled algorithm, we see that the convergence to optimal rates was achieved much faster under the scaled algorithm, though the magnitude of rate fluctuation was also much larger. The faster convergence rate implied less overloading of, and hence much less buffer requirement at, the links. Figure 4 shows the buffer occupancy at each link under the two schemes. Figure 3 shows the source rates under the scaled gradient projection algorithm.

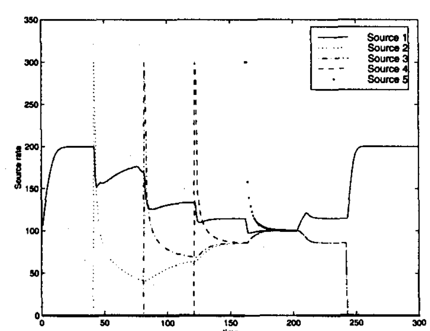

Figure 2: Source rates under unscaled gradient projection.

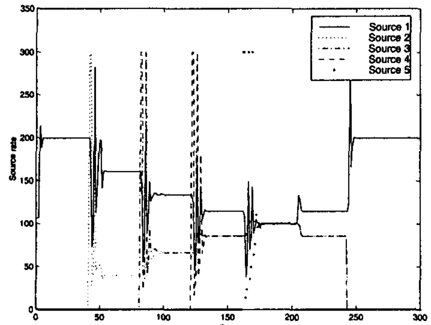

Figure 3: Source rates under scaled gradient projection.

4 Conclusion

The flow control mechanism of [14, 16] is derived as a gradient projection algorithm to solve a dual optimization problem. In this paper we have extended the algorithm to a scaled gradient projection, using a diagonal scaling that can be implemented with the same information as that is available under the basic algorithm of [14, 16]. We have proved the convergence of the algorithm and have presented simulation results that illustrate its superior performance compared to the unscaled algorithm.

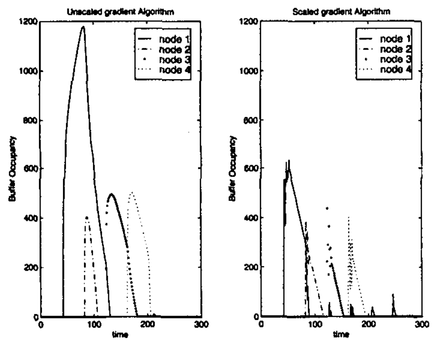

Figure 4: Buffer occupancy.

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] Flavio Bonomi, Debasis Mitra, and Judith B. Seery. Adaptive algorithms for feedback-based flow control in high-speed wide-area ATM networks. *IEEE Journal on Selected Areas in Communications*, 13(7):1267–1283, September 1995.

[4] S. Chong, R. Nagarajan, and Y.-T. Wang. Designing stable ABR flow control with rate feedback and open loop control: first order control case. *Performance Evaluation*, 34(4):189–206, December 1998.

[5] 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.

[6] S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communication Review*, 24(5), October 1994.

[7] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397–413, August 1993.

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

[9] 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.

[10] V. Jacobson. Congestion avoidance and control. *Proceedings of SIGCOMM'88, ACM*, August 1988. An updated version is available via ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z.

[11] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997. http://www.statslab.cam.ac.uk/̃frank/elastic.html.

[12] 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.

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

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

[15] David E. Lapsley and Steven H. Low. Optimization flow control, II: REM (Random Early Marking) and enhancements. To be submitted for publication, 1999.

[16] David E. Lapsley and Steven H. Low. Random Early Marking for Internet Congestion Control. In *Proceedings of IEEE Globecom'99*, December 1999.

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

[18] Steven H. Low and David E. Lapsley. Optimization flow control, I: basic algorithm and convergence. *IEEE/ACM Transcations on Networking*, 1999. To appear.

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

[20] K. K. Ramakrishnan and S. Floyd. A Proposal to add Explicit Congestion Notification (ECN) to IP. Internet draft draft-kksjf-ecn-01.txt, July 1998.