Optimization Flow Control with On–line Measurement or Multiple Paths *

Steven Low
Department of EEE, University of Melbourne, Australia
{slow@ee.mu.oz.au}

February 25, 1999

Abstract

We proposed earlier an optimization approach to reactive flow control where the objective of the control is to maximize the total 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 consider two extensions to the basic algorithm. First, the basic algorithm requires communication from sources of their rates to links in their paths in order to carry out the gradient projection algorithm. We prove that it is possible for the links to estimate the gradient using only local information, thus eliminating the need for explicit communication. Second, the basic algorithm assumes that each source is served by a single path. We generalize the model to the case where there are multiple paths between a source–destination pair. This allows flow control and routing to be jointly optimized.

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 [16, 15]. The purpose of this paper is to study two extensions to the basic algorithm.

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 consider the dual problem that decomposes the task into simple local computations to be executed at individual links and sources.

*Partial and preliminary results will be presented at the 16th International Teletraffic Congress, Edinburgh, UK, June 1999 [18]. This research is partly supported by the Australian Research Council under grant S499705.

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 $(p_l^*, l \in L)$ that aligns individual and social optimality such that $(x_s(p_l^*), s \in S)$ indeed maximizes the total utility. In other words, the price p^{*s} represents the complete congestion information source s needs for its control decision.

The basic algorithm is presented in [16] and a preliminary prototype is briefly discussed in [15]. Its convergence is proved in [19] in both synchronous and asynchronous settings. We now motivate and summarize two extensions considered in this paper.

First the basic algorithm requires communication of link prices to sources and source rates to links. In [17], we describe a marking scheme, inspired by the work of [9], that achieves the communication from links to sources using only binary feedback. In this paper we eliminate the need for explicit communication in the reverse direction: a link simply sets its price to a fraction of its buffer occupancy. This is equivalent to the links estimating the gradient using local information in carrying out the gradient projection algorithm. We prove that descent direction is maintained and hence the algorithm converges to yield optimal rate allocation. In [17] we combine these two simplifications to the basic scheme to obtain a variant of Random Early Detection (RED) scheme [7], that not only stabilizes network queues, as RED does, but does so in a way that optimizes a global measure of network performance.

Second the basic algorithm assumes that each source is served by a single path. However, source routing, where a source splits its traffic among multiple paths, offers an important way to load balance in a service provider network (ISP), e.g., in of MultiProtocol Label Switching (MPLS) networks [12]. In this paper we extend the basic algorithm to a network with multiple paths between a source and its destination, where flow control and routing can be jointly optimized. The problem turns out to be more difficult as the dual problem with multiple paths is nondifferentiable. We propose a direct extension of the basic algorithm. A source is fed back a price for each path available to it, and it chooses a total transmission rate based on the minimum price exactly as it would in the basic algorithm. Then it splits this rate *evenly* across all paths with the minimum price. The latter step is similar to the minimum–first–derivative–path routing of [3, 418–419]. Each link then adjusts its price in the same way as in the basic algorithm. We show that this is equivalent to solving the dual problem using a subgradient method [2, Chapter 6.3] where the gradient is replaced by *a* subgradient. We illustrate its behavior with a simple example.

We make three remarks. First under our scheme 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 on others. This could be a basis to provide differentiated services in terms of different rate allocations. Second 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. Finally though the optimization problem is formulated as a static problem, the flow control algorithm itself naturally extends to the time-varying case. The system tracks optimality if network or source conditions change slowly compared with convergence time [15].

There is a tremendous literature on flow control, including early schemes based on practical experience, e.g., [11, 7, 22], and recent schemes based on control theory, e.g., [1, 4, 23, 5]. Optimization based flow control have been proposed in [8, 10, 6, 13, 14, 16, 15, 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 approaches, and result in rather different flow control mechanisms to be implemented at the sources and the network links. In [8, 10], the goal is to minimize a cost function of source rates using gradient type algorithm. The cost function has two components, the first can be considered as the (minus of) total utility of our model and the second being the cost of link congestion. This second component serves a similar function as the capacity constraints in our model, but it can also be used to specify (model) the effect of link traffics on quality measures such as delay and loss. In contrast the cost of congestion in our model, as measured by bandwidth prices, is determined by the interaction of sources through the capacity constraints. Another important difference is our solution of the optimization through the dual problem and the resulting simple switch algorithm. Though in different form the social welfare function in [6] is actually a special case of our formulation here with a specific utility function. Their algorithm can also be considered as a *scaled* gradient projection algorithm for the dual problem. Our model is closest to that in [13, 14]. There however the overall objective of maximizing the total utility is decomposed into optimization subproblems for the network and the sources, and they propose a different mechanism for its solution where each source chooses a willingness to pay and the network allocates rates to these sources in a way that is proportionally fair. In [13] they prove the optimality of proportional fairness and in [14] they propose simple iterative source and switch algorithms to approach it. An interesting feature of the approach in [13] is that it allows the users to decide their payments and receive what the network allocates, whereas in our approach, the users decides their rates and pay what the network charges. See a more detailed comparison in [19].

The present paper is structured as follows. In Section 2 we review our optimization framework and the basic OFC (Optimization Flow Control) algorithm. In Section 3 we present OFC with on-line measurement and prove its convergence. In Section 4 we present OFC with multiple paths and justify it with a numerical example.

2 Model and basic 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, $U_s : \Re_+ \rightarrow \Re$ 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:

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

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

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., [3, Section 3.4.2], [20]:

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

where

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

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

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

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)[2].

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

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

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. It is illustrated in Figure 1. Let $x(p) = (x_s(p), s \in S)$.

2.2 Algorithm A1

In [16, 19] 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 ([3, pp.669]):

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

where $x^l(p) := \sum_{s \in S(l)} x_s(p)$ is the aggregate source rate at link l. In the following $[z]^+ = \max\{z, 0\}$.

Algorithm A1: Basic OFC

Link l's algorithm:

[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.

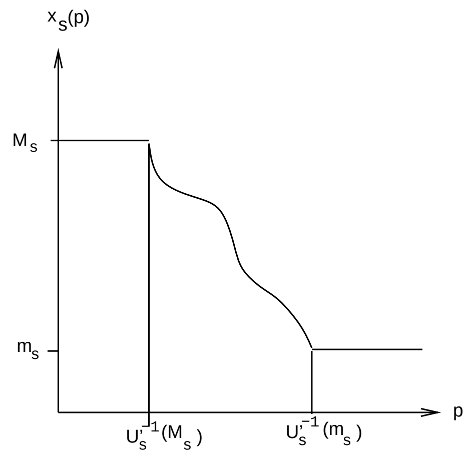

Figure 1: Source rate $x_s(p)$ as a function of (scalar) price p.

At times $t = 1, 2, \ldots$, link l:

1. *Receives rates $x_s(t)$ from all sources $s \in S(l)$ that go through link l.*
2. *Computes a new price*

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

where $x^l(t) = \sum_{s \in S(l)} x_s(t)$.

3. *Communicates new prices $p_l(t+1)$ to all sources $s \in S(l)$ that use link l.*

Source s's algorithm:

At times $t = 1, 2, \ldots$, source s:

1. *Receives from the network the sum $p^s(t) = \sum_{l \in L(s)} p_l(t)$ of link bandwidth prices in its path.*
2. *Chooses a new source rate $x_s(t+1)$ for the next period:*

$$x_s(t+1) \quad = \quad \arg \max_{x_s \in I_s} U_s(x_s) - p^s(t) x_s.$$

3. *Communicates new rate $x_s(t+1)$ to links $l \in L(s)$ in its path.*

As noted above though the optimization problem is formulated as a static problem the flow control algorithm naturally adapts to changing link capacities and set of sources at a link: simply use the current link capacity $c_l(t)$ and the current set $S(l;t)$ of sources at link l in the link algorithm of A1.

In [19] we prove that Algorithm A1 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 gradient of the dual objective function is Lipschitz which guarantees 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 step size* γ *satisfies* $0 < \gamma < 2/\overline{\alpha}\overline{L}\overline{S}$, *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 Algorithm A1 are primal–dual optimal. That is,* x^* *gives the source rates that maximize total utility and* p^* *the shadow bandwidth prices.*

In [19] we also present an asynchronous version of the algorithm where 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), where feedbacks may reach different sources after different, substantial and variable delays, and where the communication and computation at sources and links are uncoordinated and possibly based on outdated information. We prove there that the asynchronous algorithm converges provided the time interval between consecutive updates at any link or any source is bounded. For this paper however we will focus on the synchronous model.

3 OFC with on-line measurement

Algorithm A1 requires communication between sources and links: source s must obtain the sum $p^s(t)$ of link prices in its path, and link l must obtain the aggregate source rates $x^l(t)$ at the link. In [17] we describe a method to achieve the communication from links to sources using only binary feedback. In this section we describe a method to achieve the communication from sources to links *implicitly*: a link sets its price to a fraction of its current buffer occupancy. We shall see that this is equivalent to the link estimating the aggregate *source* rate $x^l(t)$ by the measured aggregate *input* rate $\hat{x}^l(t)$ at the link and using this estimate in the calculation of the gradient (in place of $x^l(p)$ in (7)).

We first extend our model to include buffer processes at the links. Then we describe precisely the algorithm and prove its optimality.

3.1 Model

We assume each link has a large buffer so that no packets are lost. We also assume that these links are work conserving, i.e., their output rates are nonzero as long as the buffer is nonempty.

In the last section the set $L(s)$ of links s visits is unrestricted and can be a tree. Here we restrict $L(s)$ to be a path for a point-to-point connection. We use $L(s)$ to denote, depending on the context, both the set of links in source s's path and the vector of these links with $L(s;i)$ as the ith link in the path.

Let $x_{ls}(t)$ be the input rate from source s at link l at time t. Then $x_{L(s;1)s}(t) = x_s(t)$ is the source rate of s at time t which is also the input rate from s at its first link. Note that $x_{L(s;i)s}(t)$ is both the input rate from s at its ith link and the output rate from s at its $i-1$st link. Hence we ignore the propagation delay on source s's ith link. Let $\hat{x}^l(t) = \sum_{s \in S(l)} x_{ls}(t)$ be the aggregate input rate at link l. Note that $\hat{x}^l(t)$ is generally different from the aggregate *source* rate $x^l(t) = \sum_{s \in S(l)} x_s(t)$ used in Algorithm A1.

Let $b_l(t)$ be the buffer backlog at link l at time t, and $b_{ls}(t)$ be the fraction of $b_l(t)$ that is from source s. Hence $b_l(t) = \sum_{s \in S(l)} b_{ls}(t)$. A useful fact that relates the buffer occupancy from individual sources to their input rates is:

$$b_{L(s;i)s}(t+1) = b_{L(s;i)s}(t) + x_{L(s;i)s}(t) - x_{L(s;i+1)s}(t). \tag{8}$$

The aggregate buffer occupancy evolves according to:

$$b_l(t+1) = [b_l(t) + \hat{x}^l(t) - c_l]^+. \tag{9}$$

Finally let $\eta_{ls}(t) = b_{ls}(t+1) - b_{ls}(t)$, $\eta_l(t) = b_l(t+1) - b_l(t)$, and $\pi(t) = p_l(t+1) - p_l(t)$.

We summarize our notations as follows.

- $L(s)$: set or vector of links in the path of s.
  $L(s;i)$: ith link in the path of s.

- $x_{ls}(t)$: input rate at link l from source s at time t.
  $x_s(t) = x_{L(s;1)s}(t)$: source rate of s at time t.

- $x^l(t) = \sum_{s \in S(l)} x_s(t)$: aggregate source rate at link l at time t.
  $\hat{x}^l(t) = \sum_{s \in S(l)} x_{ls}(t)$: aggregate input rate at link l at time t.

- $b_{ls}(t)$: backlog at link l from s at time t.
  $b_l(t) = \sum_{s \in S(l)} b_{ls}(t)$: aggregate backlog at link l at time t.
  $\eta_{ls}(t) = b_{ls}(t+1) - b_{ls}(t)$; $\eta_l(t) = b_l(t+1) - b_l(t)$.

- $p_l(t)$: price at link l at time t.
  $\pi(t) = p_l(t+1) - p_l(t)$.

- $\gamma > 0$: step size in the OFC algorithm.

3.2 Algorithm A2

The new algorithm eliminates the need for explicit communication of the source rates to links. The source algorithm is identical to that in A1, except step 3 becomes unnecessary. The link algorithm is simplified to the following:

Link l's algorithm:

At times $t = 1, 2, \ldots$, link l:

1. *Updates price* $p_l(t) \;=\; \gamma b_l(t)$.
2. *Communicates price* $p_l(t)$ *to all sources* $s \in S(l)$ *that use link* l.

Algorithm A2 converges, possibly with a smaller step size, under the following additional assumption:

C3: For all links l and all sources $s \in S(l)$ at all t, we have $\eta_{ls}(t) = \theta_{ls}(t)\eta_l(t)$ for some $\theta_{ls}(t) \geq 0$ with $\sum_{s \in S(l)} \theta_{ls}(t) = 1$.

Note that C3 is consistent with $\sum_{s \in S(l)} \eta_{ls}(t) = \eta_l(t)$. The condition requires that the capacity of link l be distributed among the sources $s \in S(l)$ 'fairly' in that if the aggregate backlog $b_l(t)$ is increased (decreased) then no individual backlog $b_{ls}(t)$ will be *strictly* decreased (increased). It is fairly general as $\theta_{ls}(t)$ can depend on (l, s, t). Service disciplines such as round robin or generalized process sharing satisfy C3, but first–in–first–out does not.

Theorem 2 *Under conditions C1–C3, the conclusion of Theorem 1 holds for Algorithm A2 with a step size* γ *satisfying* $0 < \gamma < 1/\overline{\alpha}\overline{LS}$.

We prove Theorem 2 in the rest of this section. Define the sequence $\{\lambda(t) = (\lambda_l(t), l \in L)\}$ by

$$\lambda_l(t) \quad = \quad c_l - \hat{x}^l(t). \tag{10}$$

We can treat $\hat{x}^l(t)$ as an estimate of the aggregate source rate $x^l(t)$ and $\lambda_l(t)$ an estimate of the partial derivative $\partial D/\partial p_l$ in (7).

Lemma 1 *The price sequence* $\{p(t) = (p_l(t), l \in L)\}$ *evolves according to*

$$p_l(t+1) \quad = \quad [p_l(t) - \gamma\lambda_l(t)]^+$$

Proof. From (9) and (10), it follows that for any $\gamma > 0$, we have

$$\gamma b_l(t+1) \quad = \quad [\gamma b_l(t) - \gamma\lambda_l(t)]^+.$$

The claim follows by construction of $p_l(t)$ in Algorithm A2. ■

Given any source s, we say that *link l' precedes link l with respect to s*, denoted $l' \prec_s l$, if both l' and l are in $L(s)$ and s visits l' before l, i.e., if $L(s;i) = l'$ and $L(s;j) = l$ then $i < j$. The following lemma says that the aggregate input rate at link l is the aggregate source rate minus the aggregate changes in backlog from these sources at links preceding link l in their paths. It relates the aggregate source rate to its estimate.

Lemma 2 $\hat{x}^l(t) = x^l(t) - \sum_{s\in S(l)} \sum_{l' \prec_s l} \eta_{l's}(t)$.

Proof. Applying (8) repeatedly we have

$$x_{L(s;i)s}(t) \quad = \quad x_s(t) - \sum_{j=1}^{i-1} \eta_{L(s;j)s}(t)$$

or equivalently $x_{ls}(t) \; = \; x_s(t) - \sum_{l' \prec_s l} \eta_{l's}(t)$. Summing over $S(l)$ yields the result. ■

Let $S(l' \to l) \subseteq S(l') \cap S(l)$, $l' \neq l$, be the set of sources that visit link l' before link l. Note that $S(l \to l) = \phi$. Define $a_{l'l}(t) = \sum_{s\in S(l'\to l)} \eta_{l's}(t)$. Then Lemma 2 implies that

$$\hat{x}^l(t) - x^l(t) \quad = \quad -\sum_{l' \neq l} a_{l'l}(t)$$

Since at least one of $S(l' \to l)$ and $S(l \to l')$ is empty (links are *unidirectional*), at least one of $a_{l'l}(t)$ and $a_{ll'}(t)$ is zero. Under condition C3 we have

$$|a_{l'l}(t)| \;=\; |\sum_{s\in S(l'\to l)} \theta_{l's}(t)\; \eta_{l'}(t)| \;\leq\; |\eta_{l'}(t)| \;=\; \gamma^{-1}|\pi_{l'}(t)| \tag{11}$$

where the last equality holds since $p_l(t) = \gamma b_l(t)$.

The next lemma proves that the prices converge by bounding the error $||\nabla D(p(t)) - \lambda(t)||^T \pi(t)$ in gradient estimate.

Lemma 3 *Under conditions C1–C3, $\pi(t) \longrightarrow 0$ and $\lambda(t) \longrightarrow \nabla D(p(t))$ as $t \longrightarrow \infty$ provided $\gamma < 1/\overline{\alpha}\overline{LS}$.*

Proof. We have proved in [19] that ∇D is Lipschitz (the proof is included in the Appendix for completeness). Hence by the descent lemma [2, pp. 553]

$$\begin{aligned} D(p(t+1)) &\leq D(p(t)) + \nabla D^T(p(t))\pi(t) + \zeta||\pi(t)||_2^2 \\ &= D(p(t)) + [\nabla D(p(t)) - \lambda(t)]^T\pi(t) + \lambda^T(t)\pi(t) + \zeta||\pi(t)||_2^2 \end{aligned} \tag{12}$$

where $\zeta = \overline{\alpha}\overline{LS}/2$.

Now, from (7), (10) and Lemma 2 we have

$$\begin{aligned} [\nabla D(p(t)) - \lambda(t)]^T \pi(t) &= \sum_l (\hat{x}^l(t) - x^l(t))\pi_l(t) \\ &= -\sum_l \sum_{l' \neq l} a_{l'l}(t)\pi_l(t) \\ &\leq \sum_l \sum_{l' \neq l} |a_{l'l}(t)| \; |\pi_l(t)| \\ &= \sum_{l<l'} (|a_{l'l}(t)| \; |\pi_l(t)| + |a_{ll'}(t)| \; |\pi_{l'}(t)|) \end{aligned}$$

Using (11) and the fact that at least one of $a_{ll'}(t)$ and $a_{l'l}(t)$ is zero, we have

$$[\nabla D(p(t)) - \lambda(t)]^T \pi(t) \leq \frac{1}{\gamma} \sum_{l<l'} |\pi_l(t)| \; |\pi_{l'}(t)| \tag{13}$$

Using Lemma 1 and the projection theorem [2, pp. 183] it can be shown that (e.g., see [19]) $\lambda^T(t)\pi(t) \leq -\gamma^{-1}||\pi(t)||_2^2$. Applying this inequality and (13) to (12) we have

$$\begin{aligned} D(p(t+1)) &\leq D(p(t)) - \frac{1-\gamma\zeta}{\gamma} \left(\sum_l \pi_l^2(t) - \frac{1}{1-\gamma\zeta} \sum_{l<l'} |\pi_l(t)| \; |\pi_{l'}(t)| \right) \\ &< D(p(t)) - \frac{1-\gamma\zeta}{\gamma} \left(\sum_l \pi_l^2(t) - 2 \sum_{l<l'} |\pi_l(t)| \; |\pi_{l'}(t)| \right) \\ &\leq D(p(t)) - \frac{1-\gamma\zeta}{\gamma} \; ||\pi(t)||_1^2 \end{aligned}$$

where the second inequality holds provided $\gamma < 1/2\zeta = 1/\overline{\alpha}\overline{LS}$. Summing over all t yields

$$D(p(t+1)) \leq D(p(0)) - \frac{1-\gamma\zeta}{\gamma} \sum_{t=1}^{\infty} ||\pi(t)||_1^2.$$

Since $D(p)$ is bounded below by the primal objective value, $\sum_t ||\pi(t)||_1^2$ must be bounded, and hence $||\pi(t)|| \longrightarrow 0$ as desired.

For the second half of the theorem note that from (7), (10), Lemma 2 and (11), we have

$$|[\nabla D(p(t)) - \lambda(t)]_l| = \sum_{l' \neq l} |a_{l'l}(t)| \leq \frac{1}{\gamma} \sum_{l'} |\pi_{l'}(t)|$$

which tends to zero as $t \to \infty$. This completes the proof. ■

Putting everything together yields the proof of Theorem 2.

Proof of Theorem 2. Let p^* be a limit point of the sequence $\{p(t)\}$ generated by A2. At least one exists because without loss of generality we can restrict $\{p(t)\}$ to a compact set (see [19]).

Consider any subsequence $\{p(t_k)\}$ that converges to p^*. We will show that p^* is dual optimal, and since $x(p(t))$ given by (6) is continuous, $\{x(p(t_k))\}$ converges to $x(p^*)$, the primal optimal rate.

By Lemmas 3, and the continuity of ∇D, we have

$$\lim_k \lambda(t_k) \;=\; \lim_k \nabla D(p(t_k)) \;=\; \nabla D(p^*)$$

Hence

$$[p^* - \gamma^* \nabla D(p^*)]^+ - p^* \;=\; \lim_k \, [p(t_k) - \gamma^* \lambda(t_k)]^+ - p(t_k) \;=\; \lim_k \pi(t_k) \;=\; 0$$

where the last equality follows from Lemma 3. Then the projection theorem implies that for all $p \geq 0$

$$[\nabla D(p^*)]^T (p - p^*) \;\geq\; 0$$

which, due to the concavity of D, implies that p^* is a minimizer of D. This completes the proof. ■

4 OFC with multiple paths

In this section we extend the model in Section 2 to include multiple paths between a source and its destination, following the formulation of [13], and generalize Algorithm A1 to the new model. This allows OFC to be applied to networks, such as MPLS, where flow control and routing can be jointly optimized.

4.1 Model

A path, or route, $r \subseteq L$ is a subset of links. Let R denote the set of paths. Let $r(l)$, $l \in L$, denote the set of paths that contain l. Let the $L \times R$ matrix A be defined by $A_{lr} = 1$ if $l \in r$ and 0 otherwise. A source s is characterized by four parameters $(R(s), U_s, m_s, M_s)$, where $R(s) \subseteq R$ is the set of paths that source s uses, and U_s, m_s and M_s are, as before, the utility function, minimum and maximum rates, respectively. We assume that for each path r, there is a unique source, denoted by $s(r)$, that uses r. Hence $\{R(s), s \in S\}$ is a (disjoint) partition of R. We assume U_s is increasing and strictly concave in its argument. Source s splits its rate x_s into flows y_r along paths $r \in R(s)$. Let $y(s) = (y_r, r \in R(s))$ be the vector of flow rates for source s, and let $y = (y(s), s \in S) = (y_r, r \in R)$. We say *a flow y supports rate x* if $\sum_{r \in R(s)} y_r = x_s$ for all s. Note that a flow rate y_r always represents the rate at the *source* of path r, which is not necessarily equal to the *input* rate at a link in the path.

For each path r let $L(r) = \{l \in L \mid l \in r\}$ be the set of links in r. Note that $l \in L(r)$ if and only if $r \in r(l)$. Let $Z(s) = \{(x_s, y(s)) \mid m_s \leq x_s \leq M_s,\ 0 \leq y_r \leq M_{s(r)}, r \in R(s), \sum_{r \in R(s)} y_r = x_s\}$, and let Z be the product set $(Z(s), s \in S)$. Hence Z is the set of all valid source rates $x = (x_s, s \in S)$

and flows $y = (y_r, r \in R)$ that support them. Source s's task is to decide its source rate x_s and how to split its traffic y_r along paths r, $r \in R(s)$, available to it such that $(x_s, y(s)) \in Z(s)$.

Our objective is to choose source rates x and supporting flows y so as to:

$$\mathbf{P:} \quad \max_{(x,y)\in Z} \quad \sum_s U_s(x_s) \tag{14}$$

$$\text{subject to} \quad Ay \; \leq \; c \tag{15}$$

A maximizer exists since the objective function is concave, and hence continuous, and the feasible solution set is compact. Define the Lagrangian

$$\begin{aligned} L(x,p) &= \sum_s U_s(x_s) + p^T(c - Ay) \\ &= \sum_s (U_s(x_s) - \sum_{r\in R(s)} p^r y_r) + p^T c. \end{aligned}$$

The dual problem is thus (e.g., [3, Section 3.4.2], [20])

$$\mathbf{D:} \quad \min_{p\geq 0} \quad D(p) \; = \; \sum_s B_s(p(s)) + \sum_l p_l c_l \tag{16}$$

where the vector $p(s) = (p_l, l \in R(s))$ represents the link prices along *all* paths of source s, and

$$B_s(p(s)) \;=\; \max_{(x_s,y(s))\in Z(s)} \quad U_s(x_s) - \sum_{r\in R(s)} p^r(s) y_r \tag{17}$$

$$p^r(s) \;=\; \sum_{l\in L(r)} p_l, \quad r \in R(s) \tag{18}$$

4.2 Algorithm A3

A natural strategy is, as in the single–path case, to solve the dual problem (16–18) using a descent method. The situation however is more complex in the multipath case because the primal objective function $\sum_s U_s(x_s)$ is strictly concave in x but not *strictly* concave in (x, y), and hence the dual problem, unlike in the single–path case, is not differentiable. Indeed $D(p)$, being convex, is subdifferentiable at all price vectors p with subgradient ξ defined by [24]:

$$D(p + \pi) - D(p) \;\; \geq \;\; \xi^T \pi, \quad \text{for all } \pi$$

Let $\partial D(p)$ denote the subdifferential of D at p, the set of all subgradients at p. Let $Z(p) = \{(x, y)\}$ be the set of maximizers (x, y) of (17) given the price vector p. Then the set of all subgradients is completely determined by the set of maximizers, $\partial D(p) = \{(c_l - y^l, l \in L) \mid (x, y) \in Z(p)\}$, e.g., [2, Chapter 6.1], [21, Theorem 5.4.7]. $D(p)$ is differentiable only at price vectors p that have *unique minimum* path prices for each source s, i.e., $\arg\min_r \{p^r(s), r \in R(s)\}$ is unique for all s. At such

p, the subdifferential of D reduces to the usual gradient, $\partial D(p) = \{\nabla D\}$, whose l–th component is given by

$$\frac{\partial D}{\partial p_l}(p) \quad = \quad c_l - y^l(p) \tag{19}$$

where $y^l(p) := \sum_{r \in r(l)} y_r(p)$ is the aggregate flow rate through link l at price p.

In the following we present a distributed algorithm to solve the dual problem, that is a direct extension of the basic OFC algorithm. Comparing the dual problem (3–5) in Section 2 and that (16–18) of the multipath model, sources will solve (17) and links will carry out a subgradient method to solve (16) [2, Chapter 6.3].

We first derive the source algorithm. Writing $x_s = \sum_{r \in R(s)} y_r$, (17) is equivalent to

$$\begin{array}{rl} \max\limits_{y(s)} & U_s(\sum\limits_{r \in R(s)} y_r) - \sum\limits_{r \in R(s)} p^r(s) y_r \\ \text{subject to} & m_s \le \sum\limits_{r \in R(s)} y_r \le M_{s(r)} \\ & y_r \ge 0, \quad r \in R(s) \end{array}$$

By the Karush–Kuhn–Tucker theorem, the optimal rates $y(s)$ satisfy

$$U_s'(\sum_{r \in R(s)} y_r) - \overline{\lambda} + \underline{\lambda} \quad = \quad p^r(s) - \mu_r, \quad r \in R(s) \tag{20}$$

$$\mu_r y_r \quad = \quad 0, \quad r \in R(s) \tag{21}$$

$$\overline{\lambda}(M_s - \sum_{r \in R(s)} y_r) \quad = \quad 0 \tag{22}$$

$$\underline{\lambda}(m_s - \sum_{r \in R(s)} y_r) \quad = \quad 0 \tag{23}$$

for some $\mu_r \ge 0$, $\overline{\lambda} \ge 0$ and $\underline{\lambda} \ge 0$. Hence (20–21) imply that the prices $p^r(s)$ for paths r of positive flow, $y_r > 0$, must be minimum (and hence equal). Moreover, (20–23) imply that if $p^{r^*}(s)$ is minimum among $\{p^r(s), r \in R(s)\}$ then the maximizer $y^*(s)$ is such that

$$\sum_{r \in R(s)} y_r^* \quad = \quad [U_s'^{-1}(p^{r^*}(s))]_{m_s}^{M_s}$$

Hence given path prices p^r, $r \in R(s)$, source s's decision breaks down into two simple steps: it chooses a source rate x_s based on the minimum path price, as in the basic algorithm, and then splits its traffic across minimum priced paths. The second step is consistent with the minimum–first–derivative–path routing of [3, 418–419]. We summarize.

Lemma 4 *Given a price vector $p(s)$, suppose $p^{r^*}(s) = \min \{p^r(s), r \in R(s)\}$ is the minimum path price. Then a maximizer of (17) is obtained by setting the source rate to $x_s^* = [U_s'^{-1}(p^{r^*}(s))]_{m_s}^{M_s}$ and arbitrarily splitting the flow only among paths r with minimum price $p^r(s) = p^{r^*}(s)$ such that $\sum_{r \in R(s)} y_r^* = x_s^*$.*

We now explain the link algorithm. A direct extension of the basic algorithm is to use a subgradient $c_l - y^l(t)$ in place of the gradient $c_l - x^l(t)$ in the single–path case, where $y^l(t) = \sum_{r \in r(l)} y_r(t)$ and $y_r(t)$ denotes the flow rate chosen by source $s(r)$ in period t according to Lemma 4.

Algorithm A3: multipath OFC

Link l's algorithm:

At times $t = 1, 2, \ldots$, link l:

1. *Receives flow rates $y_r(t)$ for all paths $r \in r(l)$ that contain link l.*
2. *Computes a new price*

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

   where $y^l(t) = \sum_{r \in r(l)} y_r(t)$.

3. *Communicates new prices $p_l(t+1)$ to sources $s(r)$ of all paths $r \in S(l)$ that contain link l.*

Source s's algorithm:

At times $t = 1, 2, \ldots$, source s:

1. *Receives from the network the path prices $p^r(t) = \sum_{l \in L(r)} p_l(t)$ for all paths $r \in R(s)$.*
2. *Chooses a new source rate $x_s(t+1)$ for the next period:*

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

   where $p^{r^*}(t) = \min\{p^r(t), r \in R(s)\}$, and splits its traffic $y_r(t+1)$ **evenly** *along all paths r with the minimum price.*

3. *Communicates new flow rate $y_r(t+1)$ to links $l \in L(r)$ in paths $r \in R(s)$.*

For sufficiently small stepsize γ it can be shown that the distance of the iterate $(y_r(t), p_l(t), r \in R, l \in L)$ from the optimal solution set is reduced in each step [2, Proposition 6.3.1]. We illustrate its behavior with a simple example whose equilibrium can be easily obtained analytically.

Example: multiple paths

The network consists of 5 unidirectional links labeled $1, 2, \ldots, 5$ as shown in Figure 2. Link capacities are $c = (1, 1, 1, 2, 2)$. It is shared by two sources 1 and 2. Source 1 uses two paths $(1, 5)$ and $(2, 5)$,

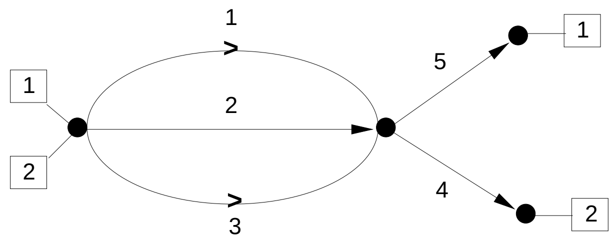

Figure 2: Network topology of Example 1

while source 2 uses $(2,4)$ and $(3,4)$. Sources s have utility functions $U_s(x_s) = a_s \log(1 + x_s)$ where $a_1 = 1$ and $a_2 = 2$. Both sources have a minimum rate of 0 and maximum rate of 3. Source 1 is active throughout from time $= 1$, while source 2 is inactive initially and becomes active at time $=$ 51. Step size $\gamma = 0.1$.

Figure 3 shows the source rate $x_s(t)$. In the first phase when only source 1 is active, the source

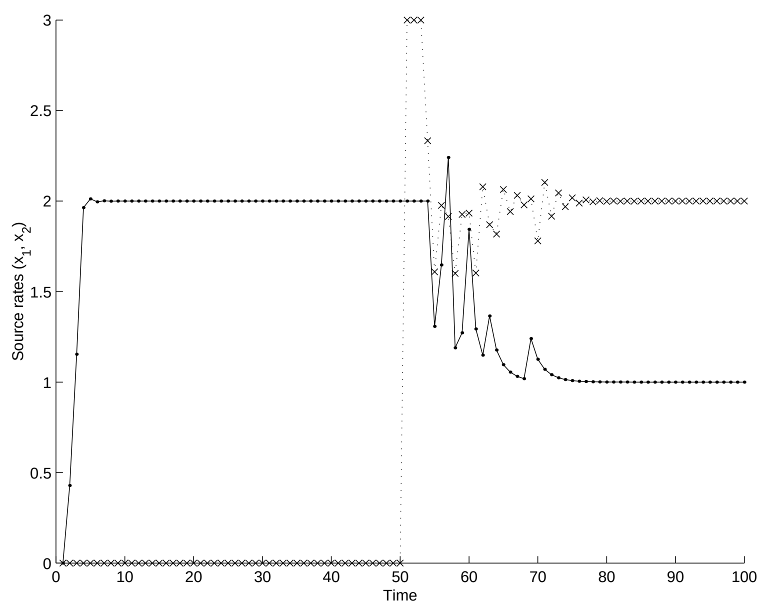

Figure 3: Source rate $x_s(t)$. Solid line is $x_1(t)$; dotted line is $x_2(t)$.

rate $x_1(t)$ (solid line) climbs rapidly to its optimal rate of 2. At time $= 51$, source 2 (dotted line) becomes active and immediately transmits at the maximum rate because it sees a path $(3,4)$ with zero price. After a brief transient, $x_1(t)$ converges to its optimal rate of 1 and $x_2(t)$ of 2. The different bandwidth allocation in equilibrium reflects, as desired, the different marginal utility of the sources.

Figure 4 shows the traffic split y_r of source 1 along its two paths $(1,5)$ and $(2,5)$. In the first

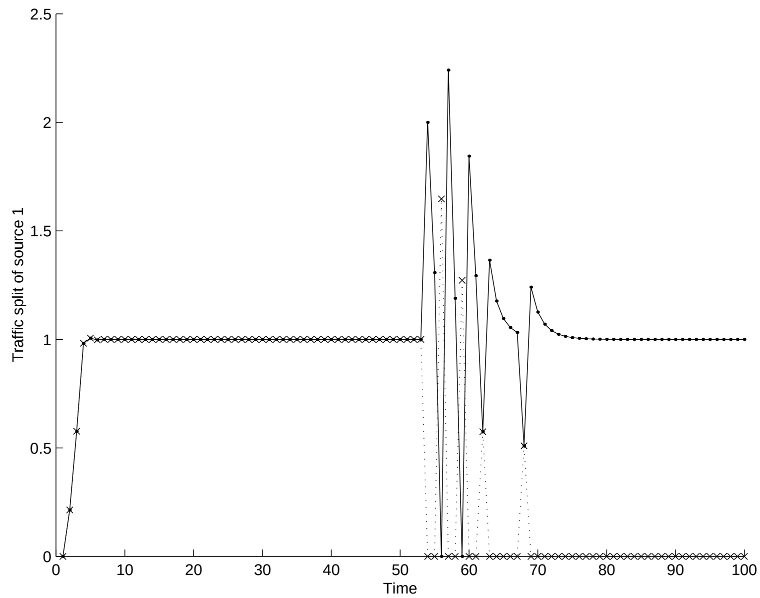

Figure 4: Traffic split of source 1. Solid line is along path $(1, 5)$; dotted line along $(2, 5)$.

phase, both paths have the same price and source 1 equally splits its traffic between them. In phase 2, path $(2, 5)$ has a high price than path $(1, 5)$ since the former path shares a link with source 2. Hence source 1 routes all its traffic on path $(1, 5)$.

Figure 5 shows the evolution of path prices. The oscillation in phase 2 is due to that some link prices do not converge, though any limit point is dual optimal. ■

5 Conclusion

We have presented two extensions to the basic optimization flow control (OFC) proposed in [16, 15]. The first extension eliminates the need for explicit communication from sources to links and makes OFC much more practical. Moreover under the on–line algorithm links need not perform any computation but simply set their prices to a fraction of their buffer occupancies. We have proved that it converges as the basic algorithm does. This simplification has been combined with a marking scheme to obtain a variant of RED scheme [7] that can be applied to Internet using the proposed explicit congestion notification bit in IP (Internet Protocol) header; see [17]. The advantage over RED is that the behavior of the *entire* network is under *explicit* control – to track a global optimality.

The second extension is motivated by networks, such as MPLS networks, where flow control and routing can be jointly optimized. The problem with multiple paths is however trickier due to nondifferentiability. We have proposed a simple subgradient algorithm adapted directly from the basic algorithm and illustrated its behavior with a simple example.

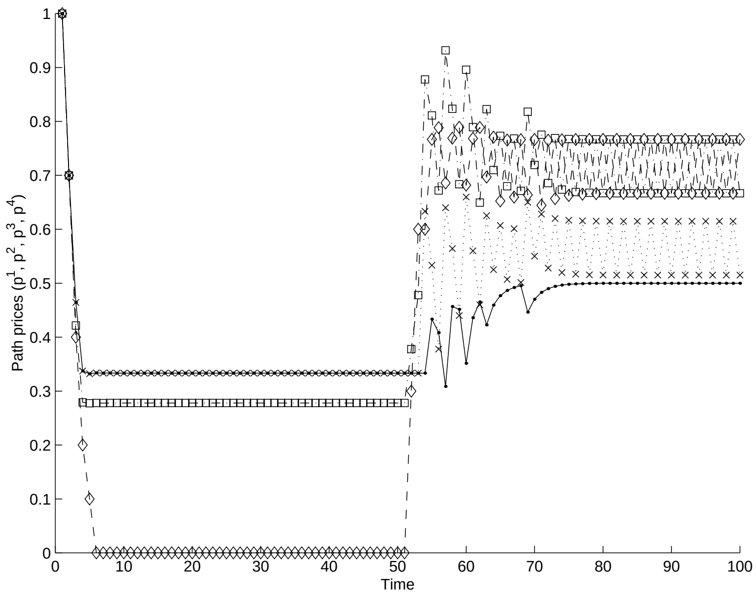

Figure 5: Path prices. Solid line is for path (1, 5), dotted line for (2, 5), dashed line with square for (2,4), and dashed line with diamond for (3,4).

Appendix: Proof of Lipschitz continuity of ∇D

We show in Lemma 6 below that ∇D is Lipschitz. We will often use vector notation when it is more convenient.

For any price vector p in $\Re_+^L$ define $\beta_s(p)$ by

$$\beta_s(p) \quad = \quad \begin{cases} \frac{1}{-U_s''(x_s(p))} & \text{if } U_s'(M_s) \leq p^s \leq U_s'(m_s) \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

where p^s is defined in (5) and $x(p)$ is the unique maximizer of (4). Let $B(p) = Diag\ (\beta_s(p), s \in S)$ be the $S \times S$ diagonal matrix with diagonal elements $\beta_s(p)$.

Lemma 5 *Under condition C1, the Hessian of D is given by $\nabla^2 D(p) = RB(p)R^T$, where it exists.*

Proof. From (7) we have $\nabla D(p) = c - Rx(p)$ where the sth component of $x(p)$ is given by (6), and hence

$$\nabla^2 D(p) \quad = \quad -R\left[\frac{\partial x}{\partial p}(p)\right]^T \tag{25}$$

where $\frac{\partial x}{\partial p}(p)$ is a $L \times S$ matrix whose (l, s) element is $\frac{\partial x_s}{\partial p_l}(p)$. Hence, when it exists,

$$\frac{\partial x_s}{\partial p_l}(p) \quad = \quad \begin{cases} \frac{R_{ls}}{U_s''(x_s(p))} & \text{if } U_s'(M_s) \leq p^s \leq U_s'(m_s) \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Using (24) we have

$$\left[\frac{\partial x}{\partial p}(p)\right]^T = -B(p)R^T \tag{27}$$

which together with (25) yields the result. ∎

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 $-U_s''(x_s)$.

Lemma 6 *Under conditions C1–C2, D is Lipschitz with*

$$||\nabla D(q) - \nabla D(p)||_2 \ \leq\ \overline{\alpha}\overline{L}\overline{S}\ ||q-p||_2$$

for all $p, q \geq 0$.

Proof. Given any $p, q \geq 0$, using Taylor theorem and Lemma 5 we have

$$\nabla D(q) - \nabla D(p) = \nabla^2 D(w)(q-p) = RB(w)R^T(q-p)$$

for some $w = tp + (1-t)q \geq 0$, $t \in [0,1]$[4]. Hence $||\nabla D(q) - \nabla D(p)||_2 \leq ||RB(w)R^T||_2 \cdot ||q-p||_2$. We now show that $||RB(w)R^T||_2 \leq \overline{\alpha}\overline{L}\overline{S}$ which yields the desired result.

Now (see e.g. [3, pp. 635])

$$||RB(w)R^T||_2^2 \leq ||RB(w)R^T||_\infty \cdot ||RB(w)R^T||_1 \tag{28}$$

i.e., $||RB(w)R^T||_2$ is upper bounded by the product of the maximum row sum and the maximum column sum of the $L \times L$ matrix $RB(w)R^T$. For $||RB(w)R^T||_\infty$ we have

$$\begin{aligned} ||RB(w)R^T||_\infty &= \max_l \sum_{l'} [RB(w)R^T]_{ll'} \\ &= \max_l \sum_{l'} \sum_s \beta_s(w) R_{ls} R_{l's} \\ &= \max_l \sum_s \beta_s(w) R_{ls} |L(s)| \end{aligned}$$

where $|L(s)|$ is the number of links in the path of source s. By definition $|L(s)| \leq \overline{L}$, $\beta_s(w) \leq \overline{\alpha}$, and hence $||RB(w)R^T||_\infty \leq \overline{\alpha}\overline{L}\ \max_l\ |S(l)| \leq \overline{\alpha}\overline{L}\overline{S}$. Similarly we can show $||RB(w)R^T||_1 \leq \overline{\alpha}\overline{L}\overline{S}$. With (28) we have $||RB(w)R^T||_2 \leq \overline{\alpha}\overline{L}\overline{S}$ and the proof is complete. ∎

[4]Where $\nabla^2 D(w)$ may not exist, at points where $w^s = U_s'(m_s)$ or $w^s = U_s'(M_s)$ for some s, derivatives should be replaced by convex subgradients in the proof. Then Lemma 6 and Theorem 1 hold. For simplicity we will ignore these issues in this paper.

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] D. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1995.

[3] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation.* Prentice-Hall, 1989.

[4] 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.

[5] 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.

[6] 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.

[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. 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.

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

[10] 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.

[11] 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.

[12] Cheng Jin, Indra Widjaja, Anwar Elwalid, and Steven Low. Multiple adaptive traffic engineering. Submitted for publication, 1999.

[13] 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.

[14] 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.

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

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

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

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

[19] Steven H. Low and David E. Lapsley. Optimization flow control, I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6), December 1999.

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

[21] Elijah Polak. *Optimization: Algorithms and Consistent Approximations.* Applied Mathematical Sciences 124. Springer, 1997.

[22] K. K. Ramakrishnan and R. Jain. A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer. *Proceedings of SIGCOMM'88, ACM*, August 1988.

[23] G. Ramamurthy and A. Kolarov. Application of control theory for the design of closed loop rate control for abr service. In *Proceedings of ITC 15*, pages 751–760, 1997.

[24] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, N.J., 1970.