Invited Paper

Flow control in networks with multiple paths

Wei-Hua Wang[a] and Marimuthu Palaniswami[a] and Steven H. Low[b]

[a]Department of Electrical and Electronic Engineering
The University of Melbourne
Victoria 3010, Australia

[b]Department of Computer Science and Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, USA

ABSTRACT

We propose two flow control algorithms for networks with multiple paths between each source-destination pair. Both are distributed algorithms over the network to maximize aggregate source utility. Algorithm 1 is a first order Lagrangian method applied to a modified objective function that has the same optimal solution as the original objective function but has a better convergence property. Algorithm 2 is based on the idea that, at optimality, only paths with the minimum price carry positive flows, and naturally decomposes the overall decision into flow control (determines total transmission rate based on minimum path price) and routing (determines how to split the flow among available paths). Both algorithms can be implemented as simply a source-based mechanism in which no link algorithm nor feedback is needed. We present numerical examples to illustrate their behavior.

Keywords: Optimal flow control, multiple paths network, optimization, Minimum-First-Derivative-Path

1. INTRODUCTION

Flow control can be regarded as a distributed computation over a network to solve an optimization problem.[1–7] In this formulation, each source is characterized by a utility function of its transmission rate and the goal is to maximize aggregate utility. Indeed, one can interpret major TCP congestion control protocols, such as Reno,[8] Vegas,[9] RED,[10] and REM,[11] within this framework where different protocols are merely different algorithms to solve the same prototype problem with different utility functions.[12]

Most of these papers assume that there is a unique path between source and destination, and the issue is to determine the source rate based on network congestion. On the other hand, multipath routing problem has received significant attention in recent literature, e.g. Ref. 13–15, where the issue is to determine efficient loop-free multipaths. In this paper, we propose algorithms that attempt to jointly optimize flow control and routing when multiple paths are available between source and destination.

This problem has been studied in Ref. 1 using a penalty function approach, in Ref. 16 using sliding mode control,[17] and in Ref. 18 using a subgradient method. The main obstacle in the multi-path case is that, even if the objective function is strictly concave in the total source rates, it is not *strictly* concave in path rates. Then the optimal path rates are nonunique and the dual problem becomes non-differentiable. Lagrangian multiplier method generally converges only when the objective function is strictly concave.

In this paper, we propose two new algorithms (Section 2) and present numerical results to illustrate their performance (Section 3). The first algorithm is derived as the first order Lagrangian method on a modified objective function that has the same optimal solution as the original objective function but apparently better convergence property. The second method is a subgradient-like method based on the idea that only paths that are least congested carry positive flows at optimality.

W. H. Wang: weihw@ee.mu.oz.au; M. Palaniswami: swami@ee.mu.oz.au; S. H. Low: slow@caltech.edu.
We acknowledge the support of the Australian Research Council through grant A49930405, CUBIN, the Caltech Lee Center for Advanced Networking and the Yuen Research Fund.

Internet Performance and Control of Network Systems II, Robert D. van der Mei, Frank Huebner-Szabo de Bucs, Editors, Proceedings of SPIE Vol. 4523 (2001)

2. MODEL AND ALGORITHMS

2.1. The optimization problem

Consider a network whose links are denoted by $L = \{1, 2, \ldots, L\}$. Let c_l be the capacity of link $l \in L$ and $c = [c_1, c_2, \ldots, c_L]^{\mathrm{T}}$. Let $S = \{1, 2, \ldots, S\}$ be the set of sources. Each source s has n_s available paths or routes from the source to the destination. Let the $L \times 1$ vector $R_{s,i}$ denote the set of links used by source $s \in S$ on its path $i \in \{1, 2, \ldots, n_s\}$, whose lth element equals 1 if the path contains link l and 0 otherwise. The set of all the available paths of user s is defined by

$$R_s = [R_{s,1}, R_{s,2}, \ldots, R_{s,n_s}]$$

and the total paths in the network are defined by a $L \times R$ routing matrix R,

$$R = [R_1, R_2, \ldots, R_S]$$

where $R = n_1 + n_2 + \ldots + n_S$ is abused here to indicate both the routing matrix and the total number of the paths.

For each source s, let $x_{s,i}$ be the rate of source s on path $R_{s,i}$, and $x_s = \sum_{i=1}^{n_s} x_{s,i}$ be the total source rate. Let $m_s \geq 0$ and $M_s \leq \infty$ be the minimum and maximum rate respectively, i.e., $m_s \leq x_s \leq M_s$. When source s transmits at a total rate of x_s, it attains a utility $U_s(x_s)$. We assume that $U_s : \Re_+ \rightarrow \Re$ is continuous, increasing and strictly concave. Let

$$x = [x_{1,1}, \ldots, x_{1,n_1}, x_{2,1}, \ldots, x_{2,n2}, \ldots, x_{n,1}, \ldots, x_{n,n_S}]^{\mathrm{T}} \in \Re_+^{R=n_1+n_2+\ldots+n_S}$$

be the vector of all path rates of all sources.

Our objective is to choose the rates x so as to maximize the total utility $\sum_{s \in S} U_s(x_s)$ subject to capacity constraints:

$$\max_{x} \quad \sum_{s \in S} U_s(x_s) \tag{1}$$

$$\text{subject to} \quad x_s = \sum_{i=1}^{n_s} x_{s,i}, \quad m_s \leq x_s \leq M_s, \quad s \in S \tag{2}$$

$$Rx \leq c, \quad x \geq 0. \tag{3}$$

The constraint (3) says that the total source rates at links l do not exceed the link capacities c_l. There exists a unique optimal solution for the source rates x_s since the objective function (1) is continuous and the feasible region (2) and (3) is compact. However, the set of path rates $x_{s,i}$ may not be unique since the objective function is not *strictly* concave in x. Solving (1–3) directly is impractical in a real network since the rates are coupled through shared links. The key to a distributed and decentralized solution is to look at the following Lagrangian form and find a *saddle-point* solution.

Lagrangian multiplier

$$\begin{aligned} L(x,u) &= L(x, \overline{\lambda}, \underline{\lambda}, p, \mu) \\ &= \sum_{s \in S} (U_s(\sum_{i=1}^{n_s} x_{s,i}) + \overline{\lambda}_s (M_s - \sum_{i=1}^{n_s} x_{s,i}) - \underline{\lambda}_s (m_s - \sum_{i=1}^{n_s} x_{s,i})) - p^{\mathrm{T}}(Rx - c) + \mu^{\mathrm{T}} x \\ &= \sum_{s \in S} (U_s(\sum_{i=1}^{n_s} x_{s,i}) - \sum_{i=1}^{n_s} p_{s,i}^r x_{s,i} - \overline{\lambda}_s \sum_{i=1}^{n_s} x_{s,i} + \underline{\lambda}_s \sum_{i=1}^{n_s} x_{s,i}) \\ &\quad + p^{\mathrm{T}} c + \overline{\lambda}^{\mathrm{T}} M - \underline{\lambda}^{\mathrm{T}} m + \mu^{\mathrm{T}} x \end{aligned}$$

where $\overline{\lambda} = [\overline{\lambda}_1, \overline{\lambda}_2, \ldots, \overline{\lambda}_S]^{\mathrm{T}}$, $\underline{\lambda} = [\underline{\lambda}_1, \underline{\lambda}_2, \ldots, \underline{\lambda}_S]^{\mathrm{T}}$, $p = [p_1, p_2, \ldots, p_L]^{\mathrm{T}}$, $\mu = [\mu_{1,1}, \ldots, \mu_{1,n_1}, \ldots, \mu_{S,1}, \ldots, \mu_{S,n_S}]^{\mathrm{T}}$, $M = [M_1, M_2, \ldots, M_S]^{\mathrm{T}}$, $m = [m_1, m_2, \ldots, m_S]^{\mathrm{T}}$, and $u = (\overline{\lambda}, \underline{\lambda}, p, \mu)$ are all nonnegative, and $p_{s,i}^r = p^{\mathrm{T}} R_{s,i}$. The Lagrange multipliers $u = (\overline{\lambda}, \underline{\lambda}, p, \mu)$ have several simple interpretations. For example, we may view p_l as the price per unit bandwidth at link l, and $p_{s,i}^r$ as the path price, the sum of all link prices on path $R_{s,i}$. Kuhn-Tucker theorem directly provides the optimality condition for our problem:

Theorem 1. The optimal solution of the path rates $x_{s,i}$ in problem (1)–(3) must satisfy

$$\begin{aligned} U_s^{'}(x_s) - \overline{\lambda}_s + \underline{\lambda}_s &= p^r_{s,i} - \mu_{s,i} && (4)\\ \mu_{s,i} x_{s,i} &= 0 && (5)\\ \overline{\lambda}_s (M_s - x_s) &= 0 && (6)\\ \underline{\lambda}_s (m_s - x_s) &= 0, \quad s \in S, \quad i = 1, 2, \ldots, n_s && (7)\\ p_l (x^l - c^l) &= 0, \quad l \in L \end{aligned}$$

for some $\mu_{s,i} \geq 0$, $\overline{\lambda}_s \geq 0$, $\underline{\lambda}_s \geq 0$, and $p_l \geq 0$, where source rate $x_s = \sum_{i=1}^{n_s} x_{s,i}$, link flow $x^l = R_{l.} x$ in which $R_{l.}$ denotes the lth row of routing matrix R.

From (4) and (5) we see that, at optimality, the prices on paths $R_{s,i}$ that carry positive flows $x_{s,i} > 0$ must be minimum, and hence equal, among all the paths R_s of source s. Moreover, the optimal source rates are given by

$$x_s^* = \sum_{R^*_{s,i} \in R^*_s} x^*_{s,i} = [U_s^{'-1}(p_s^{r^*})]_{m_s}^{M_s}$$

$$\text{and} \;\; x_{s,i} = 0 \quad \text{if} \;\; p^r_{s,i} > p_s^{r^*}$$

where path $R^*_{s,i}$ has the minimum path price $p^{r^*}_{s,i} = p^{r^*}_s$, and R^*_s defines the set of all minimum price paths $R^*_{s,i}$ of source s.

2.2. Algorithm 1

In this subsection, we present a distributed Algorithm 1 using the Arrow-Hurwicz gradient method.[19] When the objective function is not strictly concave, such as ours, it is well known that a first order Lagrangian algorithm may oscillate. The algorithm presented below has converged in all our numerical experiments, even though we do not yet have a proof of its convergence.

The idea is to applied the first order Lagrangian method to the following modified objective function:

$$\max_{x,\overline{x}} \quad \sum_{s \in S} U_s\Big(\sum_{i=1}^{n_s} x_{s,i}\Big) - \sum_{s \in S} \sum_{i=1}^{n_s} \frac{1}{2}(x_{s,i} - \overline{x}_{s,i})^2 \qquad (8)$$

where $\overline{x}_{s,i}$ is an augmented variable. If $(x, \overline{x})$ is a maximizer of (8) subject to the constraints (2) and (3), then x must also be an optimal solution of the original problem (1–3). This is because at optimality, $x = \overline{x}$ so that the added non-positive term is zero. With the non-positive term $-\sum_{s\in S} \sum_{i=1}^{n_s} \frac{1}{2}(x_{s,i} - \overline{x}_{s,i})^2$, the modified objective function becomes strictly concave in x for a fixed $\overline{x}$, and strictly concave in $\bar{x}$ for a fixed x. It is however not *strictly* concave in $(x, \overline{x})$.

Based on Arrow-Hurwicz gradient method, we have the following optimization algorithm

$$\begin{aligned} x_{s,i}(t+1) &= [(1-\gamma)x_{s,i}(t) + \gamma \overline{x}_{s,i}(t) \\ &\quad + \gamma(U_s^{'}(x_s(t)) - \overline{\lambda}_s(t) + \underline{\lambda}_s(t) - p^r_{s,i}(t))]^+ \\ \overline{x}_{s,i}(t+1) &= (1-\gamma)\overline{x}_{s,i}(t) + \gamma x_{s,i}(t) \\ x_s(t+1) &= \sum_{i=1}^{n_s} x_{s,i}(t+1) \\ \overline{\lambda}_s(t+1) &= [\overline{\lambda}_s(t) - \gamma(M_s - x_s(t))]^+ \\ \underline{\lambda}_s(t+1) &= [\underline{\lambda}_s(t) + \gamma(m_s - x_s(t))]^+ \\ p_l(t+1) &= [p_l(t) + \gamma(x^l(t) - c_l)]^+ \end{aligned}$$

where $\gamma > 0$ is a small step size, and $[z]^+ = \max\{0, z\}$, $x^l = R_{l.} x$ is the aggregate source rate at link l in which $R_{l.}$ is the lth row of routing matrix R, and $p^r_{s,i} = p^{\mathrm{T}} R_{s,i}$ is the path price for routing $R_{s,i}$. Then we have the following synchronous flow control algorithm for multiple paths.

Algorithm 1:

Source s's algorithm:

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

1. Receives from the network the path prices $p^r_{s,i}(t) = p^{\mathrm{T}}(t)R_{s,i}$ for all its paths $R_{s,i}$, $i = 1, 2, \ldots, n_s$.

2. Updates the path rate $x_{s,i}(t+1)$, its optimal estimation $\overline{x}_{s,i}(t+1)$, $i = 1, 2, \ldots, n_s$ and the source rate $x_s(t+1)$

$$\begin{aligned} x_{s,i}(t+1) &= [(1-\gamma)x_{s,i}(t) + \gamma\overline{x}_{s,i}(t) \\ &\quad + \gamma(U_s^{'}(x_s(t)) - \overline{\lambda}_s(t) + \underline{\lambda}_s(t) - p^r_{s,i}(t))]^+ \quad (9) \\ \overline{x}_{s,i}(t+1) &= (1-\gamma)\overline{x}_{s,i}(t) + \gamma x_{s,i}(t) \quad (10) \\ x_s(t+1) &= \sum_{i=1}^{n_s} x_{s,i}(t+1) \end{aligned}$$

3. Computes the new Lagrangian multipliers $\overline{\lambda}(t+1)$ and $\underline{\lambda}(t+1)$ for the next step

$$\begin{aligned} \overline{\lambda}_s(t+1) &= [\overline{\lambda}_s(t) - \gamma(M_s - x_s(t))]^+ \\ \underline{\lambda}_s(t+1) &= [\underline{\lambda}_s(t) + \gamma(m_s - x_s(t))]^+ \end{aligned}$$

4. Communicates the new flow rate $x_{s,i}(t+1)$ to all the links which contained in paths $R_{s,i}$.

Link l's algorithm:

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

1. Receives flow rates $x_{s,i}(t)$ for all paths $R_{s,i}$ that contain link l.

2. Computes a new price

$$p_l(t+1) = [p_l(t) + \gamma(x^l(t) - c_l)]^+ \quad (11)$$

   where $x^l = R_{l.}x$.

3. Communicates new prices $p_l(t+1)$ to all the sources whose path $R_{s,i}$ contains link l.

From (10), $\overline{x}_{s,i}(t)$ is a low-pass version of $x_{s,i}(t)$. If the algorithm converges, then $(x_{s,i}(t) - \overline{x}_{s,i}(t))$ will converge to zero. By subtracting (10) from (9), we see that either $x_{s,i}(t) \to 0$ or

$$U_s^{'}(x_s(t)) - \overline{\lambda}_s(t) + \underline{\lambda}_s(t) - p^r_{s,i}(t) \quad \longrightarrow \quad 0$$

as $t \longrightarrow \infty$. This is the Kuhn-Tucker condition, and hence the limit point must be optimal.

When source s has only a single path, then the source algorithm is simplified to:

$$x_s(t+1) = [x_s(t) + \gamma(U_s^{'}(x_s(t)) - p^r_s(t))]^{M_s}_{m_s} \quad (12)$$

which is a 'smoothed' version of the algorithm in Ref. 3. A popular utility function is $U_s(x_s) = a_s \log x_s$. Then (12) becomes:

$$x_s(t+1) = [x_s(t) + \gamma(\frac{a_s}{x_s(t)} - p^r_s(t))]^{M_s}_{m_s} \quad (13)$$

As observed in Ref. 20, since $U_s^{'}(x_s(t)) = a_s/x_s(t)$ is large when $x_s(t)$ is small, (13) can lead to severe rate and queue oscillation. To damp the oscillation, (13) can be modified to:

$$x_s(t+1) \quad = \quad [x_s(t) + \alpha(a_s - p^r_s(t)x_s(t))]^{M_s}_{m_s}$$

where α is the new step size. It is a discrete version of the primal algorithm in Ref. 1.

The same modification can be applied to the multi-path case where (9) is modified to:

$$x_{s,i}(t+1) = [(1-\gamma)x_{s,i}(t) + \gamma\overline{x}_{s,i}(t) + \alpha(a_s - (\overline{\lambda}_s(t) - \underline{\lambda}_s(t) + p^r_{s,i}(t))x_s(t))]^+ \quad (14)$$

2.3. Algorithm 2

Algorithm 1 is derived by applying the Lagrange first order method to a modified objective function that has the same optimal solution as the original objective function. In this subsection, we present another algorithm based on a subgradient method. It decomposes the source algorithm into a flow control problem, which determines the total source rate, and a routing problem, which decides how to split the total rate among a set of least congested paths.

Recall that (4) and (5) in Theorem 1 imply that at optimality, only those paths that have the minimum price carry a positive flow. This is the idea of minimum first derivative path discussed in, e.g., Ref. 21. Indeed, (4)–(7) imply that if $p_s^{r^*}$ is minimum path price among R_s, then the optimal total source rate x_s^* is given by:

$$x_s^* = \sum_{R_{s,i}^* \in R_s^*} x_{s,i}^* = [U_s^{'-1}(p_s^{r^*})]_{m_s}^{M_s} \quad \text{and} \quad x_{s,i} = 0 \quad \text{if} \;\; p_{s,i}^r > p_s^{r^*} \tag{15}$$

where path $R_{s,i}^*$ has the minimum path price $p_{s,i}^{r^*} = p_s^{r^*}$.

The condition (15) suggests a way to adapt the total source rate to congestion, but it does not specify how the total rate should be split among the available paths. A naive approach is to simply split it *evenly only* along paths that have the least current price. This algorithm however does not converge, e.g., when multiple paths have different link capacity. We present a routing strategy, based on the idea of Bertsekas,[21] that has a better convergence property. Algorithm 2 below uses the same link algorithm as in Algorithm 1 to update the link prices. The source algorithm is decomposed into two decisions, flow control and routing.

Flow control at source s:

$$\begin{aligned} x_s(t+1) &= [U_s^{'-1}(p_s^{r^*}(t))]_{m_s}^{M_s} \\ p_s^{r^*}(t)) &= \min_{i=1,2,\ldots,n_s} p_{s,i}^r(t) \end{aligned}$$

Hence, at each step $t+1$, source s sends at a rate $x_s(t+1)$ determined by the minimum path price $p_s^{r^*}(t)$.

Routing at source s:

- $$x_{s,i}(t+1) = [x_{s,i}(t) - \gamma(p_{s,i}^r(t) - p_s^{r^*}(t))]^+ \quad \text{for all } i = 1, 2, \ldots, n_s \tag{16}$$
- Pick any $R_{s,j}$ that has the minimum price and set its rate to:

$$x_{s,j}(t+1) = [x_s(t+1) - \sum_{i=1,\ldots,j-1,j+1,\ldots,n_s} x_{s,i}(t+1)]^+ \tag{17}$$

Hence, at each step $t+1$, the rates on all paths that cost more than the minimum are reduced by an amount proportional to the excess price, and the rate on one of the minimally priced paths is increased, so that the new rates on all paths sum to the new total source rate determined in the flow control decision. When the algorithm converges, only paths with the minimum price will carry positive flows.

Algorithm 2.:

Link l's algorithm:

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

1. Receives flow rates $x_{s,i}(t)$ for all paths $R_{s,i}$ that contain link l.
2. Computes a new price
$$p_l(t+1) = [p_l(t) + \gamma(x^l(t) - c_l)]^+ \tag{18}$$
where $x^l = R_l.x$.
3. Communicates new prices $p_l(t+1)$ to sources s whose path $R_{s,i}$ contains link l.

Source s's algorithm:

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

1. Receives from the network the path prices $p^r_{s,i}(t) = p^{\mathrm{T}}(t)R_{s,i}$ for all its paths $R_{s,i}$, $i = 1, 2, \ldots, n_s$, and decides the minimum path price $p^{r^*}_s(t) = \min_{i=1,2,\ldots,n_s} p^r_{s,i}(t)$.
2. Updates the source rate $x_s(t+1)$
$$x_s(t+1) = (U_s^{'}(p_s^{r^*}(t)))_{m_s}^{M_s} \tag{19}$$
3. Updates the path rate $x_{s,i}(t+1)$ on path $R_{s,i}$
$$x_{s,i}(t+1) = [x_{s,i}(t) - \gamma(p^r_{s,i}(t) - p^{r^*}_s(t))]^+ \tag{20}$$
4. Picks any path $R_{s,j}$ that has the minimum price and set its rate to
$$x_{s,j}(t+1) = [x_{s,t+1} - \sum_{i=1,\ldots,j-1,j+1,\ldots,n_s} x_{s,i}(t+1)]^+ \tag{21}$$
5. Communicates all the new flow rate $x_{s,i}(t+1)$ to links l contained in paths $R_{s,i}$.

2.4. Implementation

Both Algorithms 1 and 2 require communication between sources and links: source s must obtain the sum $p^r_{s,i}(t)$ of link prices in its paths $R_{s,i}$ for $i = 1, 2, \ldots, n_s$, and link l must obtain the aggregate *source* rate $x^l(t)$. To eliminate the need for source-to-link communication, link l can measure the local aggregate *input* rate $\hat{x}^l(t)$ and use that to approximate the aggregate source rate $x^l(t)$, as suggested in Ref. 22. Hence the link algorithm (11) is modified to

$$p_l(t+1) = [p_l(t) + \gamma(\hat{x}^l(t) - c_l)]^+$$

where $\hat{x}^l(t)$ is the aggregate input rate measured by link l at time t.

In the reversed direction, a marking scheme is proposed in Ref. 11,20 to communicate path prices to sources using a single bit. In this subsection, we propose a way to communicate path prices *implicitly*, i.e., a source deduces the aggregate price on each path from observed round trip time without the need for explicit feedback from links. The idea is to use a different step size $\frac{\beta}{c_l}$ at each link l:

$$p_l(t+1) = [p_l(t) + \frac{\beta}{c_l}(\hat{x}^l(t) - c_l)]^+ \tag{22}$$

Since queue length evolves according to:

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

multiplying both sides by $\frac{\beta}{c_l}$, we have

$$\frac{\beta}{c_l} b_l(t+1) = [\frac{\beta}{c_l} b_l(t) + \frac{\beta}{c_l}(\hat{x}^l(t) - c_l)]^+ \tag{23}$$

Comparing (23) with (22), we see that link price at time t is proportional to the current queuing delay $q_l(t) := b_l(t)/c_l$ at link l:

$$p_l(t) = \beta \frac{b_l(t)}{c_l} = \beta q_l(t)$$

The price on path $R_{s,i}$ is then proportional to the *end-to-end* queuing delay at time t:

$$p^r_{s,i}(t) \;=\; p^{\mathrm{T}}(t)R_{s,i} \;=\; \beta q^{\mathrm{T}}(t)R_{s,i}$$

Given end-to-end propagation delay $PD^r_{s,i}$, the end-to-end queue delay $q^{\mathrm{T}}(t)R_{s,i}$, and hence the aggregate price, can be deduced from the round trip time $RTT^r_{s,i}(t)$ observed at the source:

$$p^r_{s,i}(t) \;=\; q^{\mathrm{T}}(t)R_{s,i} \;=\; \beta(RTT^r_{s,i}(t) - PD^r_{s,i}(t))$$

In practice, the propagation delay $PD^r_{s,i}$ can be estimated by the minimum $RTT^r_{s,i}(t)$ observed so far. The idea of using queue delay as a congestion measure and extracting it from round trip time has been used in Ref. 6,7,9,23.

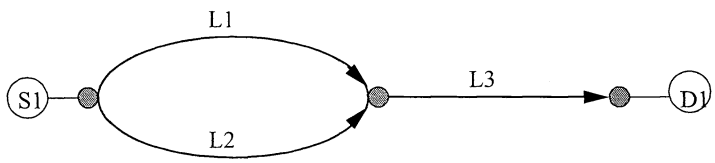

Figure 1. Network topology of Example 1.

3. EXAMPLES AND SIMULATIONS

In this section we present numerical results on two network topologies. In the first example both Algorithms 1 and 2 converge; we also show the behavior of Lagrangian method applied to the *original* objective function as opposed to the modified objective function used in Algorithm 1 and the behavior of the recent algorithm of Ref. 18.

Example 1:

Consider the following simple network which consists 3 unidirectional links labeled L1, L2 and L3 with capacities $c = (1, 2, 3)$ as shown in Fig. 1. There is a single source with the utility function $U_1(x_1) = \log x_1$. Its total rate x_1 is upper bounded by 5 and it routes its flow along two paths, $(L1 \to L3)$ with path rate $x_{1,1}$ and $(L2 \to L3)$ with path rate $x_{1,2}$.

We have run both Algorithms 1 and 2 on this network, with the modification (14) and step sizes $\alpha = 0.1$, $\beta = 0.02$ and $\gamma = 0.02$. The results are shown in Fig. 3 and Fig. 4 for Algorithm 1 and 2 respectively. For both algorithms, source rate x_1 and path rate $x_{1,1}$ and $x_{1,2}$ converge to the optimal point $(3, 1, 2)$, and the path prices converge to $U_1'(3) = \frac{1}{3}$.

Fig. 5 shows the behavior of Algorithm 1 when $\gamma = 0$. This corresponds to a first order Lagrangian method on the *original* objective function (1). As expected, the algorithm does not converge. This motivated the modification that leads to Algorithm 1.

Fig. 6 shows the behavior of the algorithm of Ref. 18, with a $\kappa = 4$ and step size $\lambda = 0.01$. As expected, it converges to a limit cycle in a small neighborhood of the optimal.

Example 2

The network consists 6 unidirectional links labeled L1, L2, ..., L6 as shown in Fig. 2. Link capacities are $c = (20, 25, 20, 60, 60, 60)$. It is shared by two sources 1 and 2. Source 1 with a total rate of x_1 uses the paths: $(L1 \to L4 \to L5)$ with rate $x_{1,1}$ and $(L2 \to L4 \to L5)$ with rate $x_{1,2}$. Source 2 with a total rate of x_1 uses the paths: $(L2 \to L4 \to L6)$ with rate $x_{2,1}$ and $(L3 \to L4 \to L6)$ with rate $x_{2,2}$. Sources s have utility functions $U_s(x_s) = a_s \log(x_s)$ and minimum and maximum source constraints m_s and M_s.

The simulation proceeds in three stages

- Stage 1: $t = 0 \to 1000$

  $a_1 = 10$, $a_2 = 20$, neither source has rate constraints, i.e., $m_s = 0$ and $M_s = \infty$.

- Stage 2: $t = 1001 \to 2000$

  Source 2 increases a_2 to 50.

- Stage 3: $t = 2001 \to 3000$

  Source 1 adds minimum source rate requirement $m_1 = 30$.

Fig. 7 shows the behavior of Algorithm 1, with the step sizes $\alpha = 0.1$, $\beta = 0.1$ and $\gamma = 0.1$.

- Stage 1: Source rates $x_1(t)$ and $x_2(t)$ climb rapidly and converge to the optimal rates $(20, 40)$, and their path rates converge to an equilibrium. Also the path prices converge 0.5, the value of $U_1'(20)$ and $U_2'(40)$.

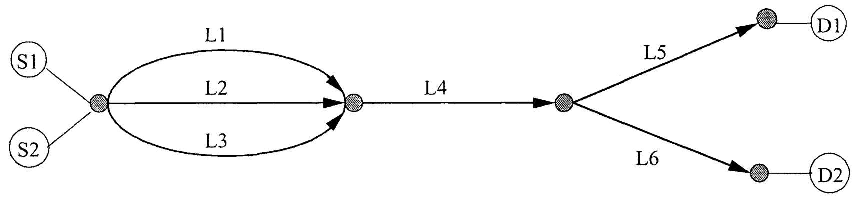

Figure 2. Network topology of Example 2.

- Stage 2: When source 2 increases its a_2 to 50, source rate $x_2(t)$ increases and exhausts the bandwidth of both links L2 and L3 to achieve a maximal rate 45. Source 1 transmits its flow $x_1(t)$ only across L1 and has a rate 15. Its path rate $x_{1,2}(t)$ drops to 0. The path price $p^r_{1,1}(t)$ converges to $U_1'(15) = \frac{2}{3}$ and the other path price converges to $U_2'(45) = \frac{10}{9}$.
- Stage 3: Since a minimal rate requirement of 30 is added to source 1, $x_1(t)$ increases from 15 to 30, and $x_2(t)$ drops to 30. All path prices converge to a value $U_2'(30) = \frac{5}{3}$, and path rates converge to their equilibria.

Fig. 8 shows the simulation results with Algorithm 2 using $\beta = 0.1$ and $\gamma = 0.2$. The source and path prices converge to the same values as in Fig. 7 for Algorithm 1. The path rates also converge, but the path rates in stages 1 and 3 are slightly different from those of Fig. 7. This confirms that optimal path rates are generally nonunique.

4. CONCLUSION AND FUTURE WORKS

In this paper, we propose two flow control algorithms for networks with multiple paths between source-destination pairs. Algorithm 1 is a first order Lagrangian method on a modified objective function that has the same optimal solution as the original objective function but has a better convergence property. Algorithm 2 is based on the idea that, at optimality, only paths with the minimum price carry positive flows, and naturally decomposes the overall decision into flow control (determines total transmission rate based on minimum path price) and routing (determines how to split the flow among available paths). Both algorithms can be implemented as simply a source-based mechanism in which no link algorithm nor feedback is needed. We have presented numerical examples to illustrate their behavior. It would be interesting to prove their convergence analytically.

REFERENCES

1. F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society* **49**, pp. 237–252, March 1998.
2. R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica* **35**, pp. 1969–1985, December 1999.
3. S. H. Low and D. E. Lapsley, "Optimal flow control, I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking* **7**, pp. 861–874, December 1999.
4. J. Golestani and S. Bhattacharyya, "End-to-end congestion control for the Internet: A global optimization framework," in *Proceedings of International Conference on Network Protocols (ICNP)*, October 1998.
5. S. Kunniyur and R. Srikant. End–to–end congestion control schemes: utility functions, random losses and ECN marks. In *Proceedings of IEEE Infocom*, March 2000.
6. J. Mo and J. Walrand, "Fair end-to end window-based congestion control," *IEEE/ACM Transactions on Networking* **8**, pp. 556–567, October 2000.
7. R. La and V. Anantharam. Charge-sensitive TCP and rate control in the Internet. In *Proceedings of IEEE Infocom*, March 2000.
8. V. Jacobson, "Congestion avoidance and control," in *Proceedings of SIGCOMM 1988*, August 1988. An updated version is available via ftp://ftp.ee.lbl.gov/paper/congavoid.ps.Z.
9. L. S. Brakmo and L. L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet," *IEEE Journal of Selected Areas in Communications* **13**, pp. 1465–1480, October 1995.

10. S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397–413, August 1993. ftp://ftp.ee.lbl.gov/papers/early.ps.gz.
11. S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin. REM: active queue management. *IEEE Network*, 2001. http://netlab.caltech.edu.
12. S. H. Low. A duality model of TCP flow controls. In *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, September 18-20 2000. http://netlab.caltech.edu.
13. J. Chen, P. Drushel, and D. Subramanian, "An efficient multi-path forwarding method," in *Proceedings of IEEE INFOCOM 1998*, March 1998.
14. J. Chen, P. Drushel, and D. Subramanian, "A simple, pratical, distributed multi-path routing algorithm," Tech. Rep. 98-320, Rice University, 1998.
15. S. Vutukury and J. J. Garcia-Luna-Aceves, "MPATH: a loop-free multipath routing algorithm," *Elsevier Journal of Microprocessors and Microsystems* **24**, pp. 319–327, 2000.
16. A. Lagoa and H. Che, "Decentralized optimal traffic engineering in the Internet," 2000. Submitted for publication.
17. V. I. Utkin, *Sliding Modes in Control Optimization*, Springer-Verlag, Berlin, 1992.
18. K. Kar, S. Sarkar, and L. Tassiulas, "Optimization based rate control for multipath sessions," Tech. Rep. 2001-1, Institute for Systems Research, University of Maryland, 2001.
19. K. J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Nonlinear Programming*, Stanford University Press, 1958.
20. S. Athuraliya and S. H. Low, "Optimization flow control, II: Implementation", 2000. Submitted for publication.
21. D. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*, Prentice Hall, 1989.
22. S. H. Low, "Optimization flow control with on-line measurement," in *Proceedings of the 16th International Teletraffic Congress*, (Edinburgh, UK), June 1999.
23. S. H. Low, L. Peterson, and L. Wang. Understanding Vegas: a duality model. In *Proceedings of ACM Sigmetrics*, June 2001.

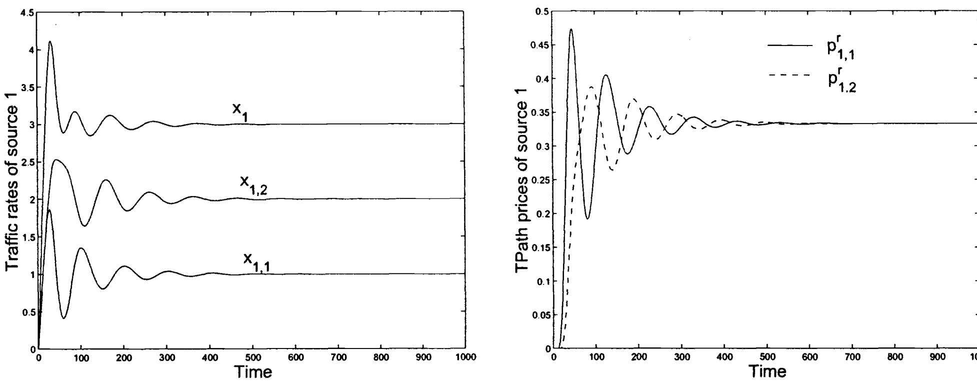

Figure 3. Simulation results of Example 1 using Algorithm 1, $\alpha = 0.1$, $\beta = 0.02$ and $\gamma = 0.02$.

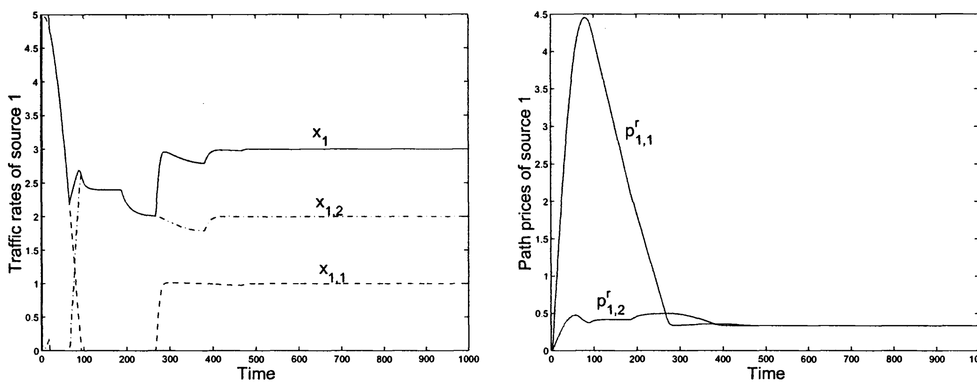

Figure 4. Simulation results of Example 1 using Algorithm 2, $\beta = 0.02$ and $\gamma = 0.02$.

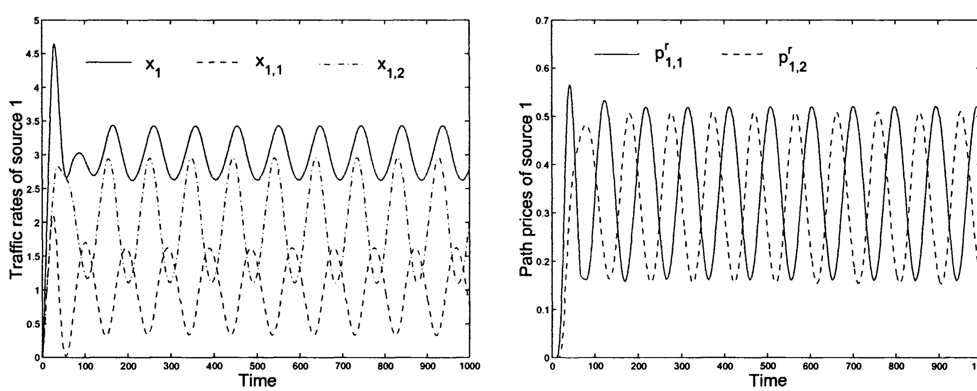

Figure 5. Simulation results of Example 1 using Algorithm 1, $\alpha = 0.02$, $\beta = 0.1$ and $\gamma = 0$.

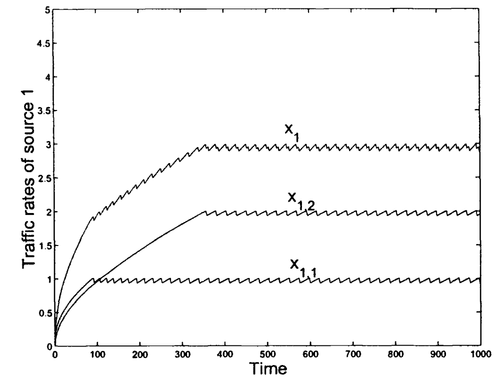

Figure 6. Simulation results of Example 1 using Kar's algorithm, $\lambda = 0.01$, and $\kappa = 1$.

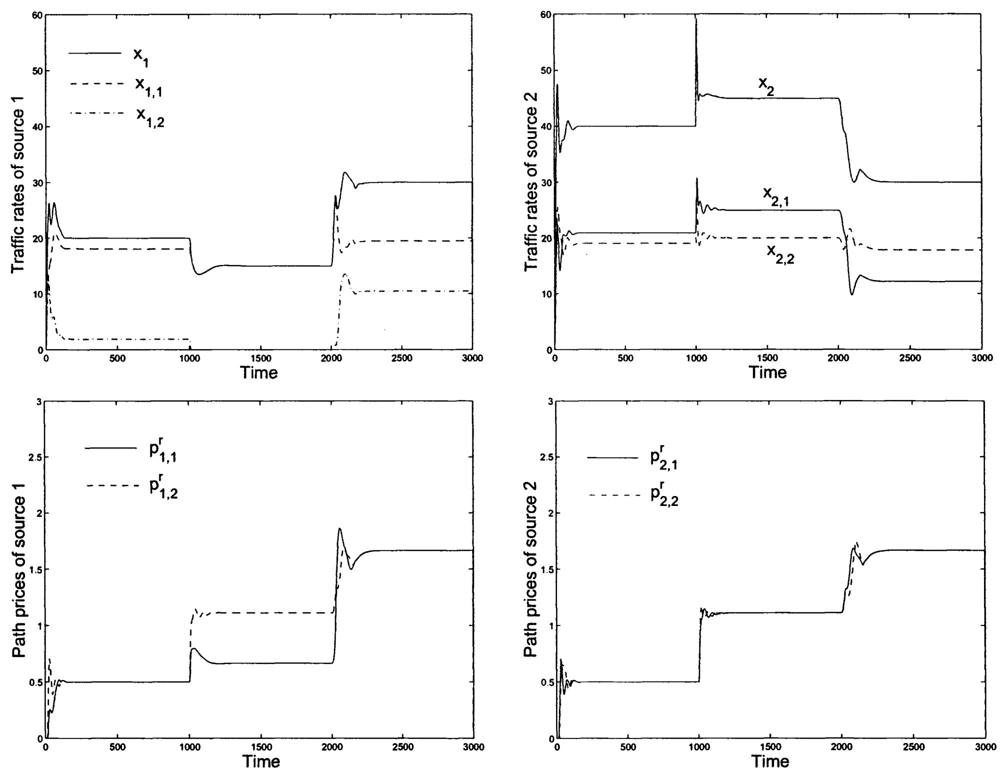

Figure 7. Simulation results of Example 2 using Algorithm 1, $\alpha = 0.1$, $\beta = 0.1$ and $\gamma = 0.1$.

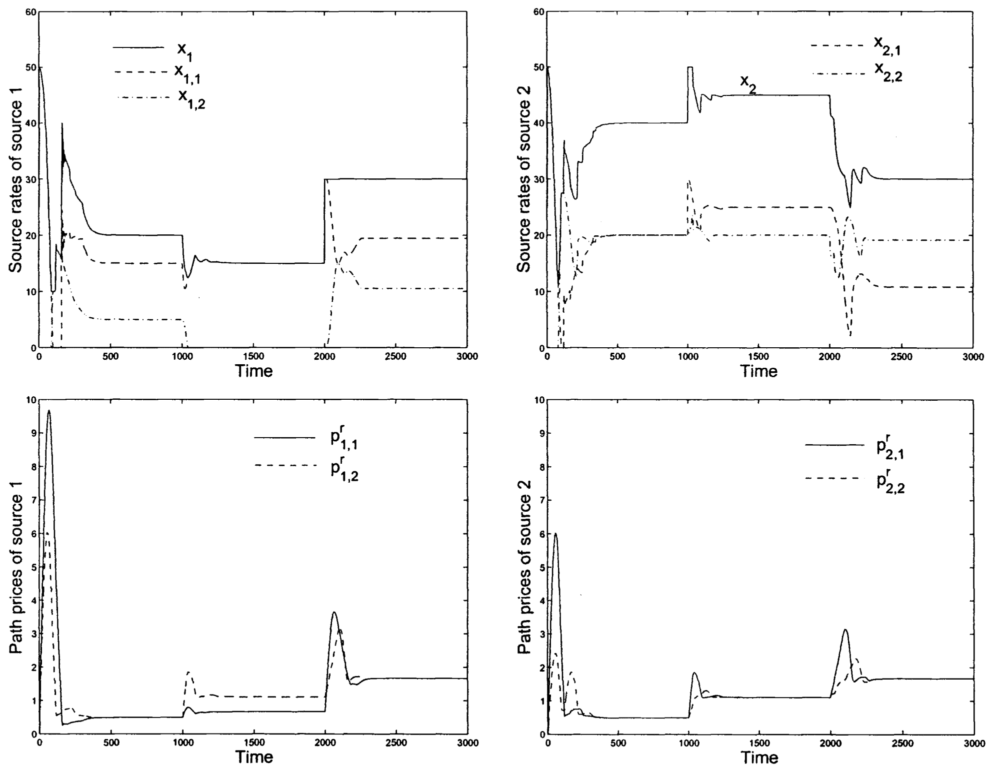

Figure 8. Simulation results of Example 2 using Algorithm 2, $\beta = 0.1$ and $\gamma = 0.2$.