An Enhanced Random Early Marking Algorithm for Internet Flow Control

Sanjeewa Athuraliya David Lapsley Steven Low
Department of EEE, University of Melbourne, Australia
{sadsa,laplsey,slow}@ee.mu.oz.au

Abstract— **We propose earlier an optimization based flow control for the Internet called Random Early Marking (REM). In this paper we propose and evaluate an enhancement that attempts to speed up the convergence of REM in the face of large feedback delays. REM can be regarded as an implementation of an optimization algorithm in a distributed network. The basic idea is to treat the optimization algorithm as a discrete time system and apply linear control techniques to stabilize its transient. We show that the modified algorithm is stable globally and converges exponentially locally. This algorithm translates into an enhanced REM scheme and we illustrate the performance improvement through simulation.**

Keywords—**Internet flow control, pricing, optimization flow control, marking, REM, feedback delay**

I. INTRODUCTION

We proposed earlier a flow control scheme for the Internet called Random Early Marking (REM) [1]. It is derived from an optimization model where each source is characterized by a utility function that models its valuation of bandwidth and the goal is to maximize aggregate source utility over their transmission rates subject to capacity constraints [2], [3], [4]. The basic flow control algorithm can be regarded as a distributed computation performed by the sources and links to minimize the dual problem. The algorithm however requires communication between sources and links. This communication requirement is greatly simplified in [5], [1] and leads to REM, a binary feedback scheme similar to Random Early Detection (RED) [6]. The purpose of this paper is to propose an enhancement to REM that attempts to significantly speed up its convergence in the face of large feedback delays.

The value of the optimization model presented in [2], [4] is twofold. First, though it may not be possible, nor critical, that optimality is exactly attained in a real network, the optimization framework offers a means to explicitly steer the *entire* network towards a desirable operating point. Second it makes possible a systematic method to design and refine practical flow control schemes, which can be treated simply as implementations of a certain optimization algorithm, where modifications to the flow control mechanism is guided by modifications to the optimization algorithm. For instance, it is well known that Newton algorithm has much faster convergence than gradient projection algorithm. By replacing the gradient projection algorithm presented in [2], [4] by the Newton algorithm we derive in [7] a practical Newton–like flow control scheme that can be proved to maintain optimality, has the same communication requirement as the original scheme but enjoys a much better convergence property. This paper provides another example on how the optimization framework can be exploited to systematically refine REM.

There is a tremendous literature on flow control. The works closest to this paper are optimization based [8], [9], [10], [11], [12], [13], [2], [5], [1] where the problem is formulated as one of optimizing a social welfare and the flow control mechanisms are derived 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 particular both [11], [12] and our work solve the same optimization problem of maximizing aggregate utility over source transmission rates. The two works however differ in their solution approach, which lead to different algorithms and their implementation through marking [13], [1]. See [4] for a detailed comparison.

The paper is structured as follows. In Section II we summarize our optimization model and the REM algorithm. In Section III we extend the model to include feedback delay and derive the enhanced algorithm. In Section IV we present preliminary simulation results to illustrate the performance improvement. We conclude in Section V with future work. All proofs are omitted and can be found in a forthcoming full paper.

II. OPTIMIZATION MODEL AND REM

Consider a network that consists of a set $L = \{1, \ldots, L\}$ of *unidirectional* links of capacity c_l, $l \in L$. The network is shared by a set $S = \{1, \ldots, S\}$ of sources. Source s is characterized by four parameters $(L(s), U_s, m_s, M_s)$. The path $L(s) \subseteq L$ is a subset of links that source s uses, $U_s : \Re_+ \to \Re$ is a utility function, $m_s \geq 0$ and $M_s \leq \infty$ are the minimum and maximum transmission rates, respectively, required by source s. Source s attains a utility $U_s(x_s)$ when it transmits at rate x_s that satisfies $m_s \leq x_s \leq M_s$.

The first two authors acknowledge the Australian Commonwealth Government for their Australian Postgraduate Awards, and the last author acknowledges the support of the Australian Research Council under grants S499705 and A49930405.

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:

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

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

The constraint (2) says that the 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., [14, Section 3.4.2], [15]:

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

where

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

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

The first term of the dual objective function $D(p)$ is decomposed into S separable subproblems (4–5). If we interpret p_l as the price per unit bandwidth at link l then p^s is the total price per unit bandwidth for all links in the path of s. Hence $x_s p^s$ represents the bandwidth cost to source s when it transmits at rate x_s, and $B_s(p^s)$ represents the maximum benefit s can achieve at the given price p^s. We shall see that this scalar p^s summarizes all the congestion information source s needs to know. A source s can be induced to solve maximization (4) by bandwidth charging. For each p, a unique maximizer, denoted by $x_s(p)$, exists since U_s is strictly concave.

In general $(x_s(p), s \in S)$ may not be primal optimal, but by the duality theory, there exists a $p^* \geq 0$ such that $(x_s(p^*), s \in S)$ is indeed primal optimal. Hence we will solve the dual problem (3). Given minimizing prices p^* the primal optimal source rates $x^* = x(p^*)$ can be obtained by individual sources s:[1]

$$x_s(p) = \begin{cases} U_s'^{-1}(p) & \text{if } U_s'(M_s) \leq p \leq U_s'(m_s) \\ m_s & \text{if } U_s'(m_s) < p \\ M_s & \text{if } U_s'(M_s) > p \end{cases} \tag{6}$$

Here $U_s'^{-1}$ is the inverse of U_s', which exists over the range $[U_s'(M_s), U_s'(m_s)]$ since U_s' is continuous and U_s *strictly* concave. It is illustrated in Figure 1. Let $x(p) = (x_s(p), s \in S)$.

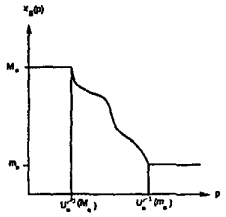


Fig. 1. Source rate $x_s(p)$ as a function of (scalar) price p.

In [2], [4] we propose to solve the dual problem using the gradient projection algorithm that leads to the following optimization flow control algorithm:

A1: Basic OFC

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

Here $x^l(t) := \sum_{s \in S(l)} x_s(t)$ is the aggregate source rate at link l at time t, and $[z]^+ = \max\{z, 0\}$. Hence a link raises or reduces its price accordingly as the demand $x^l(t)$ is greater or less than the supply c_l of bandwidth. A source raises or reduces its rate accordingly as the path price $p^s(t)$ is low or high (see (6)).

It is shown in [4] that provided all utility functions are strictly concave increasing and their second derivatives are bounded away from zero, the basic OFC (optimization flow control) algorithm A1 converges to yield the optimal rates for sufficiently small stepsize γ. As discussed there, 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.

Algorithm A1 requires communication of link prices to sources and source rates to links, and hence cannot be implemented on the Internet. In [5] we show that a link can

[1] 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.

simply set its price to a fraction of the buffer occupancy. 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. We prove there that this approximate gradient projection algorithm also converges to yield the optimal rates. This simplification eliminates the need for explicit communication from sources to links. In the reversed direction we propose a method in [1] that communicates link prices to sources using only binary feedback. The basic idea is for a link to mark a packet with a probability that is exponential to its link price $p_l(t)$ so that the end to end marking probability of a packet is exponential to the path price $p^s(t)$. A source can then estimate the end to end marking probability and hence $p^s(t)$. This can be implemented using the proposed ECN (Explicit Congestion Notification) bit in the IP header [16], [17]. Combining these two simplifications yields the REM algorithm of [1], where the marking probability is exponential in buffer occupancy.

A2: REM

Link l's algorithm:

1. Set price $p_l(t) = \gamma b_l(t)$ where $b_l(t)$ is the (average) buffer occupancy in period t.

2. For each packet that is not marked, mark it with the probability

$$m_l(t) = 1 - 2^{-p_l(t)}$$

Source s's algorithm:

1. Count the fraction $\hat{m}^s(t)$ of packets received in period t that are marked, and estimate the path price by:

$$\hat{p}^s(t) = -\log_2(1 - \hat{m}^s(t))$$

2. Choose a new transmission rate $x_s(t+1)$ for the next period: $x_s(t+1) = x_s(\hat{p}^s(t))$.

III. Enhanced algorithm

The model and algorithms in the last section assume zero feedback delay. In this section we relax this assumption and propose an enhanced algorithm.

Let τ_{ls} and τ_{sl} be the (constant) delay from link l to source s and from s to l, respectively. Let $\bar{d} = \max_{l',l,s}\{\tau_{l's} + \tau_{sl} \mid s \in S(l) \cap S(l')\}$. We will see below that $\bar{d}$ is the maximum time for a price change in link l' to affect the price at link l through shared sources s. In particular if delays are symmetric, $\tau_{ls} = \tau_{sl}$ for all l, s, then $\bar{d}$ is the maximum round trip delay of the network.

The basic OFC algorithm A1 becomes:

$$p_l(t+1) = \left[p_l(t) + \gamma\Big(\sum_{s\in S(l)} x_s(t-\tau_{sl}) - c_l\Big)\right]^+ \quad (7)$$

$$x_s(t) = x_s\left(\sum_{l\in L(s)} p_l(t-\tau_{ls})\right) \quad (8)$$

Hence link l computes the next price using the delayed source rates $x_s(t-\tau_{sl})$ and source s computes a new rate using delayed prices $p_l(t-\tau_{ls})$. It is important to note that the algorithm described by (7–8) does not require the links and sources to know their delays τ_{ls}, τ_{sl}. This algorithm is a special case of the asynchronous model considered in [4] where links and sources update every period using the most recent data. According to [4, Theorem 2] the algorithm converges to yield the optimal source rates provided the stepsize γ is small enough.

Suppose there is a unique dual optimal price p^*.[2] This would be the case, e.g., if the utility functions are $U_s(x_s) = a_s \log x_s$ and every link l with $p_l^* > 0$ has a single–link connection $s(l)$, i.e., for all l, there exists $s(l)$ such that $L(s(l)) = \{l\}$. We may assume without loss of generality that $p_l^* > 0$ for all l, because links l with $p_l^* = 0$ are not saturated in equilibrium and hence can be omitted from consideration. Then provided that the stepsize is sufficiently small the sequence $\{p(t)\}$ generated by (7–8) converges to p^*. Moreover for all sufficiently large t, we have $p_l(t) > 0$ for all l, and we can omit the projection operation in price computation for sufficiently large t. Assume also that $U_s'^{-1}(M_s) = 0$, $U_s'^{-1}(m_s) = \infty$ so that $x_s(p) = U_s'^{-1}(p)$. Then the system is described by:

$$p_l(t+1) = p_l(t) + \gamma\Big(\sum_{s\in S(l)} x_s(t-\tau_{sl}) - c_l\Big) \quad (9)$$

$$x_s(t) = U_s'^{-1}\left(\sum_{l\in L(s)} p_l(t-\tau_{ls})\right) \quad (10)$$

We will regard (9–10) as a discrete time system to be stabilized, where the states are the link prices.

For the rest of the section we first consider the linearized system of (9–10) in the neighbourhood of the unique equilibrium p^*. Then we will design a deadbeat controller for the linearized system to speed up its convergence. This control law however requires a link to know the entire network

[2]The simulation in Section IV also shows what might happen when dual optimal prices are nonunique: link prices may oscillate between two dual optimal limit points while source rates converge to the unique primal optimal vector.

topology and is therefore impractical. Next we derive an approximate control law that can be implemented by each link using local information. Finally we apply the control law to the REM algorithm.

Even though the control laws are derived for a linear system around the equilibrium p^* their performance in the nonlinear setting away from the equilibrium is investigated in the next section through simulation.

A. Linearized system

Let $\overline{p}(t)$ be the $(\overline{d}+1)L$ dimensional expanded 'state':

$$\overline{p}(t) = \begin{bmatrix} p(t) \\ p(t-1) \\ \vdots \\ p(t-\overline{d}) \end{bmatrix}$$

where each $p(t-\tau), \tau = 0, \ldots, \overline{d}$, is a L-vector. The (l,τ)th element of $\overline{p}(t)$ is $p_l(t-\tau)$. Let $\overline{p}^*$ be the $(\overline{d}+1)L$ dimensional vector with each of $p(t-\tau), \tau = 0, \ldots, \overline{d}$, replaced by the unique equilibrium p^*.

For any scalar p define $\beta_s(p)$ by

$$\beta_s(p) = \frac{1}{-U_s''(x_s(p))}$$

where $x(p)$ is given by (6).

Then after some manipulations we have

$$\frac{\partial x_s}{\partial \overline{p}_{l\tau}}(t) = -\beta_s(\overline{p}^s(t))\, 1(l \in L(s), \tau_{ls} = \tau)$$

where $\overline{p}^s(t) = \left(\sum_{l \in L(s)} p_l(t-\tau_{ls})\right)$. Hence the first order term in the Taylor expansion of $x_s(t)$ is

$$\begin{aligned} &\nabla_{\overline{p}} x_s(\overline{p}^*)(\overline{p}(t) - \overline{p}^*) \\ = \; &-\beta(p^{*s}) \sum_{l' \in L(s)} (p_{l'}(t-\tau_{l's}) - p^*_{l'}) \end{aligned}$$

Linearizing (9–10) around the equilibrium p^* and letting $\tilde{p}(t) = p(t) - p^*$ we thus have, after rearranging,

$$\tilde{p}_l(t+1) = \tilde{p}_l(t) - \gamma \sum_{l'} b_{ll'} \tilde{p}_{l'}(t - \tau_{l's} - \tau_{sl}) \quad (11)$$

where $b_{ll'} = \sum_{s \in S(l) \cap S(l')} \beta_s(p^*)$. The last equality uses the fact that $x^l(p^*) = c_l$ by complementary slackness, since by assumption $p^*_l > 0$. (11) makes the interdependence of link prices apparent. It says that the new price $p_l(t+1)$ at link l depends not only on the current price at link l, but also on past prices $p_{l'}(t - \tau_{l's} - \tau_{sl})$ at other links l'. This is because in response to a price change at link l', sources $s \in S(l) \cap S(l')$ that traverse both links l' and l adjust their rates, which then affect the price at link l. Hence there is a delay of $\tau_{l's} + \tau_{sl}$ for a price update at link l' to affect the price at link l through shared sources s. This coupling of link prices is described by $b_{ll'}$.

To express (11) in matrix form, define the $L \times L$ matrices $A(\tau), \tau = 0, \ldots, \overline{d}$, by:

$$[A(\tau)]_{ll'} = \sum_{s \in S(l) \cap S(l')} \beta_s(p^*)\, 1(\tau_{l's} + \tau_{sl} = \tau)$$

Then we have, defining $\tilde{q}(t) = [\tilde{p}(t)\;\; \tilde{p}(t-1)\;\; \cdots\;\; \tilde{p}(t-\overline{d})]^T$,

$$\tilde{q}(t+1) = \begin{bmatrix} I - \gamma A(0) & \cdots & -\gamma A(\overline{d}-1) & -\gamma A(\overline{d}) \\ I & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \tilde{q}(t)$$

This describes the local behavior of the gradient projection algorithm around the (unique) equilibrium p^* when the feedback delays are nonzero.

Note the difference between our system and that of [18]. Their system has two important features that simplify significantly the controller design. First their control (the explicit rates) are calculated at the network link where the current as well as past buffer levels are available. Second, all sources receive the same explicit rate (single congestion node case), except possibly with different feedback delay, and these past rates are also available at the link for calculation. These allow a simple proportional–plus–derivative controller and lead to the simple close loop equations (38) and (39) in their paper. In our case, current source rates are not available at a link; moreover a link may not know the value of τ_{sl} and hence may not be able to use past source rates in price adjustment (except the most recent one). On the other hand a link always has the most current price, so we consider a controller that uses only past values of local prices.

B. Deadbeat controller

Suppose in computing a new price each link l averages its past link prices and diagonally scale the gradient:

$$\begin{aligned} p_l(t+1) = &\sum_{\tau=0}^{\overline{d}} \mu_l(\tau) p_l(t-\tau) \\ &+\gamma_l\Big(\sum_{s \in S(l)} x_s(t-\tau_{sl}) - c_l\Big) \end{aligned} \quad (12)$$

Note that link l uses only the most recent source rates and this does *not* require it to know the value of τ_{sl}. The goal is to compute the averaging parameters $\mu_l(\tau)$ and the scaling parameter γ_l in order to place the poles of the discrete time system (12) at the origin, i.e., deadbeat control law. In equilibrium both sides of (12) must be p^* and hence $\mu_l(\tau)$ must satisfy (since $\sum_{s\in S(l)} x_s^* = c_l$ in equilibrium)

$$\sum_{\tau=0}^{\bar{d}} \mu_l(\tau) = 1, \quad \text{for all } l \tag{13}$$

Following similar derivation as in the last subsection, the linearized system around p^* is

$$\tilde{q}(t+1) = \begin{bmatrix} M(0) - GA(0) & \cdots & M(\bar{d}) - GA(\bar{d}) \\ I & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \tilde{q}(t) \tag{14}$$

where $M(\tau) = diag(\mu_l(\tau))$, $\tau = 0, \ldots, \bar{d}$, and $G = diag(\gamma_l)$ are $L \times L$ diagonal matrices. The following result explains how to choose the averaging matrices $M(\tau)$ and the scaling matrix G. For simplicity of exposition we assume a large network where all delays are nonzero, $\tau_{ls} \geq 1$, $\tau_{sl} \geq 1$ (otherwise, condition (15) below is replaced by $\det(\lambda I - M(0) + GA(0)) = \lambda^L$).

Theorem 1: Suppose all delays are nonzero. Then all poles of (14) are at the origin if and only if $M(\tau)$ and G are chosen to satisfy

$$M(0) = 0 \tag{15}$$

$$\det(M(\tau) - GA(\tau)) = 0, \quad \tau = 0, \ldots, \bar{d} \tag{16}$$

$$\sum_{\tau=0}^{\bar{d}} M(\tau) = I \tag{17}$$

Moreover the above conditions uniquely determine the diagonal matrices $M(\tau)$ and G. ■

Interestingly the theorem says that if all delays are nonzero then current prices $p_l(t)$ should not be used in the weighted average of past prices.

Note that the above theorem ensures rapid convergence only around the unique equilibrium point p^*. In order to choose the averaging parameters $\mu_l(\tau)$ to place all poles at the origin, we have given up the choice of stepsize γ (which now becomes a matrix). This loss of freedom may upset global stability. This is indeed observed in our simulation, presented in the next section.

C. Approximate deadbeat controller

Theorem 1 describes how to choose the averaging matrices $M(\tau)$ and the gain matrix G to enforce rapid convergence towards the equilibrium p^* when the system is in a neighborhood of p^*. However the equations (15–16) in the theorem can only be solved centrally, as the off–diagonal elements of the matrices $A(\tau)$ imply that a link needs to know information on sources at other links, and hence is impractical. In this subsection we derive an approximation to the control law of the last section that can be implemented by individual links locally. The idea is to approximate the matrices $A(\tau)$ by their diagonal terms.

Let $B(\tau) = diag(\beta^l(\tau), l \in L)$ be the $L \times L$ diagonal matrices whose diagonal elements are

$$\beta^l(\tau) = \sum_{s\in S(l)} \beta_s(p^*)\, 1(\tau_{ls} + \tau_{sl} = \tau)$$

In words the lth diagonal elements $\beta^l(\tau)$ of $B(\tau)$ are the sums of $\beta_s(p^*)$, sum over all sources s that traverse link l and have a round trip delay of τ. With this approximation there is enough degree of freedom, provided $\bar{d} \geq 2$, that we can reduce the gain matrix G to an *arbitrary* scalar $\gamma > 0$. This means that we can choose γ to be sufficiently small to ensure global convergence; contrast this with the exact deadbeat control law of the last subsection. We will see in the next section that the performance of this controller can be better than that of the exact deadbeat controller whose stepsize γ is fixed by the condition (13).

The conditions corresponding to (15–17) in Theorem 1 reduce to the following choice of weights $\mu_l(\tau)$:

$$\mu_l(0) = \gamma\beta^l(0), \quad l \in L \tag{18}$$

$$\mu_1(\tau) = \gamma\beta^1(\tau), \quad \tau = 1, \ldots, \bar{d}-1 \tag{19}$$

$$\mu_1(\bar{d}) = 1 - \gamma \sum_{\tau=0}^{\bar{d}-1} \beta^1(\tau) \tag{20}$$

$$\mu_L(\bar{d}) = \gamma\beta^L(\bar{d}) \tag{21}$$

and for all other l and τ, $\mu_l(\tau)$ are chosen to satisfy

$$\sum_{\tau=0}^{\bar{d}} \mu_l(\tau) = 1 \tag{22}$$

This control law is much simpler than that expressed in Theorem 1: a link l can choose its own weights $\mu_l(\tau)$ based on the information about sources that traverse link l; we will come back to this point in Section V.

With this approximate deadbeat controller the corre-

sponding optimization algorithm becomes:

$$p_l(t+1) = \left[\sum_{\tau=0}^{\bar{d}} \mu_l(\tau) p_l(t-\tau) + \gamma\Big(\sum_{s\in S(l)} x_s(t-\tau_{sl}) - c_l\Big)\right]^+ \quad (23)$$

$$x_s(t) = U_s'^{-1}\left(\sum_{l\in L(s)} p_l(t-\tau_{ls})\right) \quad (24)$$

where the weights $\mu_l(\tau)$ are given by (18–22). The linearized system around the unique equilibrium p^* is then given by, after some algebra,

$$\tilde{q}(t+1) = \begin{bmatrix} M(0)-\gamma A(0) & \cdots & M(\bar{d})-\gamma A(\bar{d}) \\ I & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \tilde{q}(t) \quad (25)$$

Even though the linearized system (25) may not have all poles at the origin, we can ensure that the actual optimization algorithm (23–24) is globally stable and converges exponentially to the equilibrium p^* locally around p^*.

Theorem 2: With the control law (18–22), provided $\gamma > 0$ is sufficiently small, the optimization algorithm (23–24) converges globally to p^*. Moreover the linearize system (25) around the unique equilibrium p^* has all poles within the unit disc. ■

D. Enhanced REM

In this subsection we apply the control law (18–22) to the REM algorithm. The only modification is in Step 1 of A2: instead of setting price $p_l(t) = \gamma b_l(t)$ to be a fraction of its buffer occupancy, link l updates its price according to (23) with weights $\mu_l(\tau)$ given by (18–22).

IV. Simulation results

In this section we illustrate the effectiveness of the control laws through preliminary simulations. We will present results that compare the performance of gradient projection algorithm A1 with the deadbeat control law defined by Theorem 1, and with its approximation (18–22). We will then compare the performance of the original REM algorithm A2 with the enhanced version described in Section III-D. We emphasize that though the control laws are derived from linearized systems, the (nonlinear) optimization algorithm (23–24) is simulated, where the weights $\mu_l(\tau)$ are defined by the control laws.

The simulation study is carried out for the network in Figure 2 shared by three connections, with sources Si and destinations Di, $i = 1, 2, 3$. Connection S1–D1 spanned all links 1, 2; connection S2–D2 spanned link 1; connection S3–D3 spanned link 2. All links have capacity 200 packets/sec. Source S1 transmitted data from time 7ms to time 300ms. The start times of the other sources are staggered with S2 starting at time 80ms, S3 at time 160ms. Once turned on, sources S2 and S3 remained active until time 240ms. 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 $-\frac{a_s}{2}(M_s - x_s)^2$, with a_s and M_s equal to 1 and 300 respectively for all the sources S1,S2 and S3. The feedforward and feedback delays between sources and links are given in table V.

A. Basic OFC

In this section we present in Figure 3 three sets of results on the system dynamics under the basic OFC algorithm A1, under the deadbeat controller described in equations (12)–(17), and under the approximation to the deadbeat controller described by equations (18)–(22). Each set of results consists of a graph of the source transmission rate x_s, link price p_l and buffer occupancy q_l.

We can see from Figure 3 that all of the algorithms converge to the theoretically optimum operating point (given in table I). The major differences are in the degree of oscillation and speed of convergence. We can see from the figures that the application of the deadbeat controller (both in exact and approximate forms) can speed up the convergence significantly. Moreover the buffer requirement under the approximate deadbeat controller is 25% less than the other two schemes. However, as noted after Theorem 1 the inability of the exact deadbeat control law to choose the stepsize (and indeed the direction) of the optimization algorithm may upset global stability. This is illustrated in Figure 3(b) where the source rate of the first source oscillates around the equilibrium from time 7ms to 80ms. Furthermore, as noted in Section III, when dual optimal prices are nonunique, link prices may oscillate between two dual optimal limit points without ever converging while source rates converge to the unique primal optimal vector. This is illustrated in figure 3(c) where the link prices oscillate but the source rates have converged to the unique primal optimal.

B. Random Early Marking

Random Early Marking simplifies greatly communication between links and sources. In this section we present

Time	x_1	x_2	x_3	p_1	p_2
7–79	200	-	-	50*	50*
80–160	100	100	-	200	0
160–240	33.3	166.6	166.6	133.3	133.3
240–300	200	-	-	50*	50*

TABLE I
EQUILIBRIUM VALUES

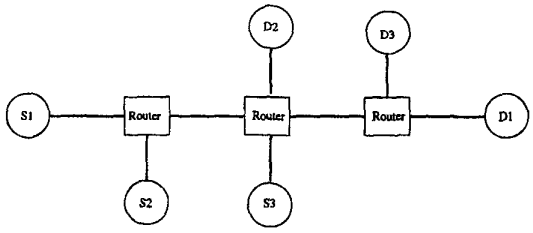


Fig. 2. Network Topology

the results of our simulations using Random Early Marking. The simulation scenario is identical to that used for the results in the previous section. Figure 4 shows the results for REM: first using the algorithm A2, then with the exact deadbeat controller, and finally using the approximate deadbeat controller. The results are similar to those of the last subsection. While the use of binary feedback introduces extra oscillation around the equilibrium values, the oscillations are relatively small in magnitude. Again the use of the deadbeat controller (both in the exact and approximate form) speeds up system convergence and reduces the buffer requirement.

V. CONCLUSION

We have derived an enhanced REM algorithm for Internet flow control and illustrated the performance improvement through simulation. The basic idea is to stabilize the transient behavior of the optimization algorithm of which REM is an implementation, by averaging over past prices. The averaging allows us to place the poles of the linearized system around the origin, ensuring exponential convergence locally. Simulation shows significant improvement in convergence and performance (buffering requirement) over the original REM. More importantly, perhaps, this procedure demonstrates the advantage of the optimization framework which allows a systematic refinement of practical flow control schemes.

We now comment on limitations of this preliminary work. The control law of Section III-C, though much simpler than the exact deadbeat controller, still involves significant overhead. It requires a link l to know $\beta_s(p^*)$ of sources s going through link l and their round trip delays. This may be impractical for a large network. $\beta_s(p^*)$ may be easy to determine, e.g., for the quadratic utility functions used in Section IV, $\beta_s(p^*)$ is (approximately) a constant which can be communicated to the links during connection setup. We have also tried simple rules such as choosing μ_l to be the reciprocal of the number of sources at link l and it seems to work fine. We are currently investigating ways to systematically derive simpler control laws with provably good performance.

REFERENCES

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

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

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

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

[5] Steven H. Low, "Optimization flow control with on-line measurement," in *Proceedings of the ITC*, June 1999, vol. 16.

[6] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.

[7] Sanjeewa Athuraliya and Steven Low, "Newton-like algorithm for optimization flow control," in *Proceedings of Globecom'99*, November 1999.

[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.*, 1980, pp. 779–784.

[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] 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.

[11] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 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*, vol. 49, no. 3, pp. 237–252, March 1998.

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

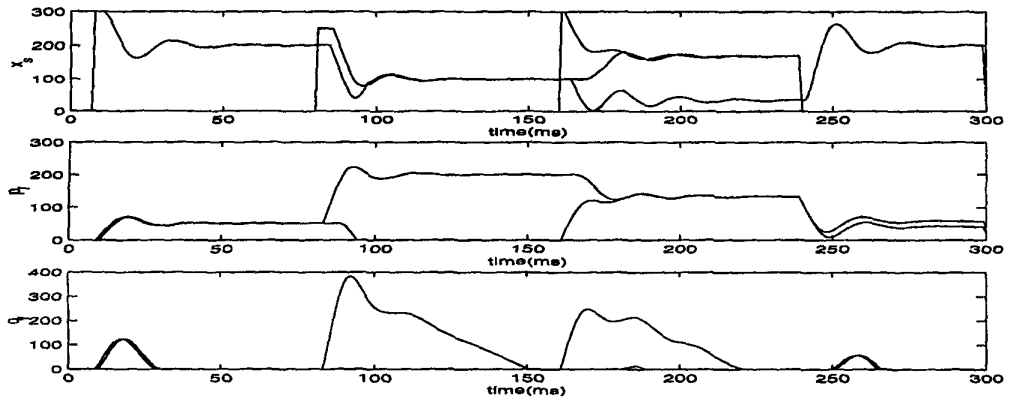
[14] Dimitri P. Bertsekas and John N. Tsitsiklis, *Parallel and distributed computation*, Prentice-Hall, 1989.

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

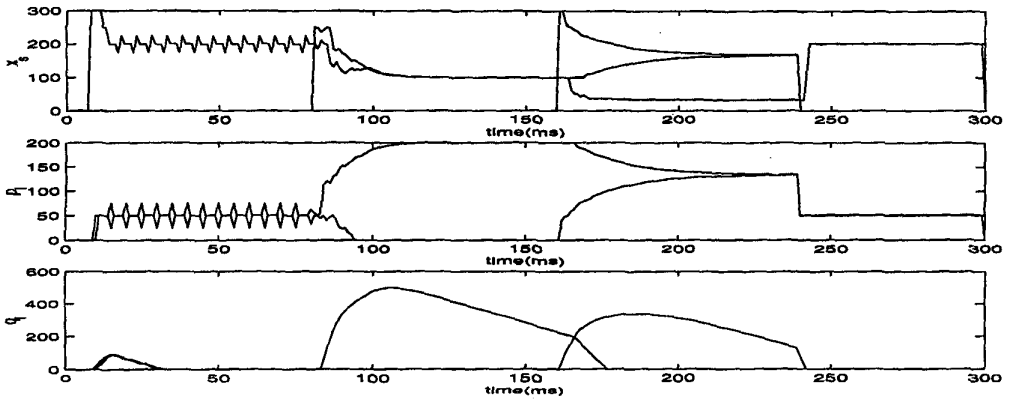
[16] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communication Review*, vol. 24, no. 5, October 1994.

[17] 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.

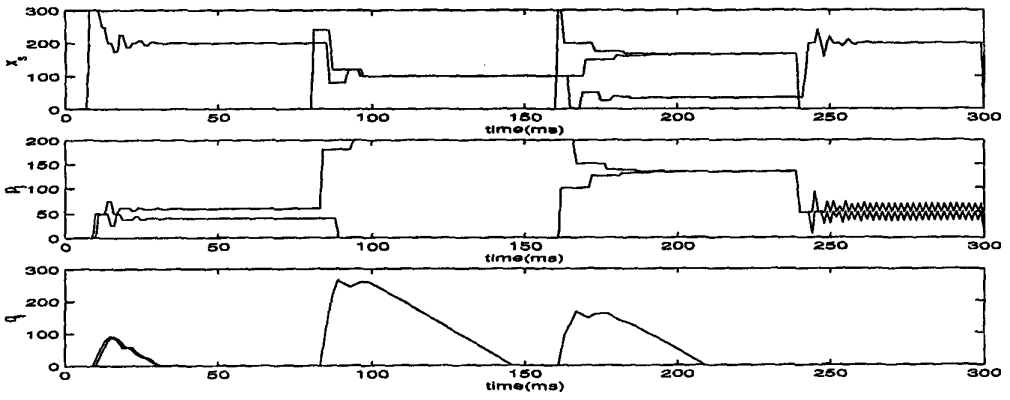
[18] L. Benmohamed and S. M. Meerkov, "Feedback control of congestion in store-and-forward networks: the case of a single congested node," *IEEE/ACM Transactions on Networking*, vol. 1, no. 6, pp. 693–707, December 1993.



(a) Basic Algorithm

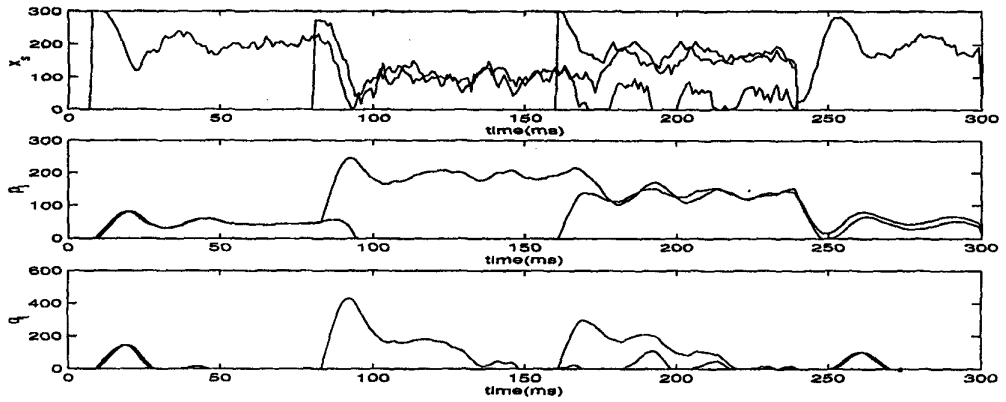


(b) Exact Deadbeat Controller

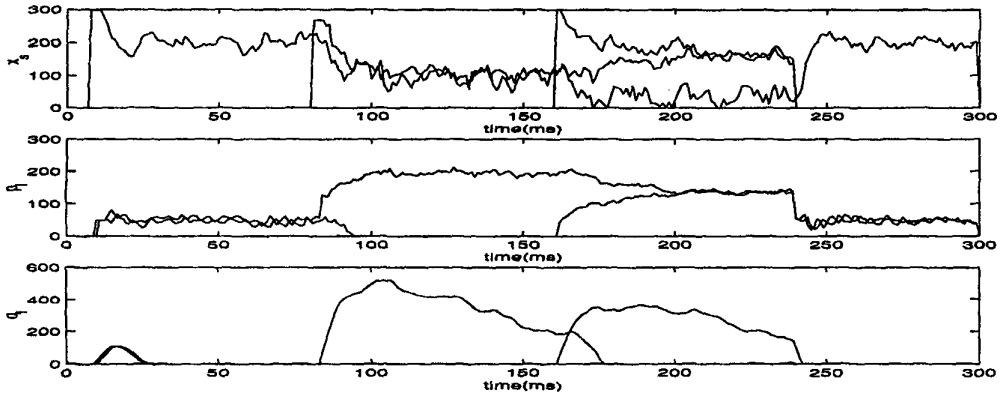


(c) Approximate Deadbeat Controller

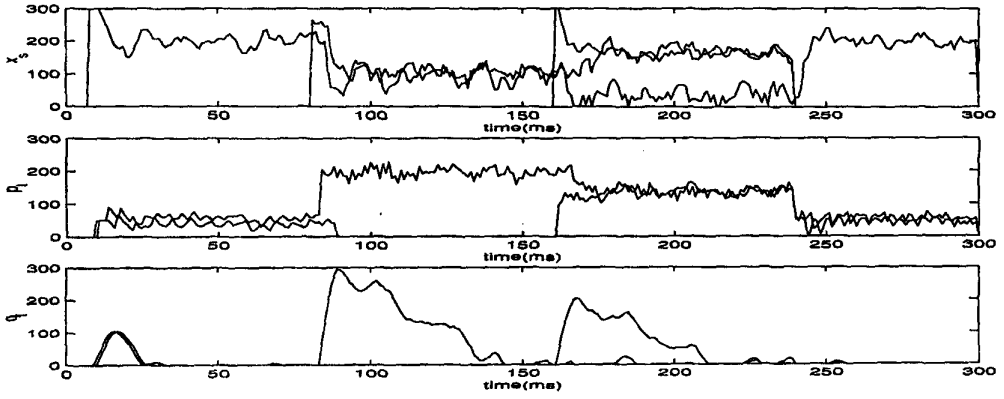
Fig. 3. Basic OFC



(a) Basic Algorithm



(b) Exact Deadbeat Controller



(c) Approximate Deadbeat Controller

Fig. 4. Random Early Marking

Source	*starting time (ms)*	*finish time (ms)*
1	7	300
2	80	240
3	160	240

TABLE II

SOURCE STARTING AND FINISHING TIMES

	link 1				*link 2*			
time (ms)	7-79	80-159	160-239	240-300	7-79	80-159	160-239	240-300
$\mu_l(0)$	0	0	0	0	0	0	0	0
$\mu_l(1)$	0	0	0.2	0	0	0	0.2	0
$\mu_l(2)$	0	0	0	0	0	0	0.2	0
$\mu_l(3)$	0	0.25	0.2	0	0	0.33	0.2	0
$\mu_l(4)$	0.5	0.25	0.2	0.5	0	0	0	0
$\mu_l(5)$	0.5	0.25	0.2	0.5	0.5	0.33	0.2	0.5
$\mu_l(6)$	0	0.25	0.2	0	0.5	0.33	0.2	0.5
γ	0.5	0.25	0.2	0.5	0.5	0.33	0.2	0.5

TABLE III

WEIGHTS USED IN THE EXACT ALGORITHM

	link 1				*link 2*			
time (ms)	7-79	80-159	160-239	240-300	7-79	80-159	160-239	240-300
$\mu_l(0)$	0	0	0	0	0	0	0	0
$\mu_l(1)$	0	0	0	0	0	0	0	0
$\mu_l(2)$	0	0	0	0	0	0	0.5	0
$\mu_l(3)$	0	0	0	0	0	0	0	0
$\mu_l(4)$	1	0.5	0.5	1	0	0	0	0
$\mu_l(5)$	0	0	0	0	0	0	0	0
$\mu_l(6)$	0	0.5	0.5	0	1	1	0.5	1
γ	0.5	0.5	0.5	0.5	0.5	1	0.5	0.5

TABLE IV

WEIGHTS USED IN THE APPROXIMATE ALGORITHM

Source	*link1*		*link2*	
	forward delay	feedback delay	forward delay	feedback delay
1	2	2	3	3
2	3	3	N/A	N/A
3	N/A	N/A	1	1

TABLE V

FEEDFORWARD AND FEEDBACK DELAYS BETWEEN LINKS AND SOURCES