A new TCP/AQM for Stable Operation in Fast Networks

Fernando Paganini Zhikui Wang Steven H. Low John C. Doyle

Abstract— **This paper is aimed at designing a congestion control system that scales gracefully with network capacity, providing high utilization, low queueing delay, dynamic stability, and fairness among users. In earlier work we had developed fluid-level control laws that achieve the first three objectives for arbitrary networks and delays, but were forced to constrain the resource allocation policy. In this paper we extend the theory to include further dynamics at TCP sources, preserving the earlier features at fast time-scales, but permitting sources to match their steady-state preferences at a slower time-scale, provided a bound on round-trip-times is known.**

We develop a packet-level implementation of this protocol, where the congestion measure is communicated back to sources via marking of an ECN bit. We discuss parameter choices for the marking and estimation system, and demonstrate using ns-2 simulations the stability of the protocol and its equilibrium features in terms of utilization, queueing and fairness, in comparison with existing protocols.

I. Introduction

Since their inception in the late 1980s [1], the congestion control mechanisms in TCP have been extremely successful in keeping the Internet under control while it underwent a dramatic growth. Despite continued work in improving some details such as retransmission, and on the active queue management (AQM) side [2], the basic additive-increase-multiplicative-decrease (AIMD) structure in TCP congestion avoidance has remained unchanged. What is the incentive for research on replacing it?

One justification comes from the desire to improve the quality of service provided by the Internet, both in reducing queueing delays, and in allowing for more control over resource allocation, which is currently indirectly determined by the protocol. As argued in [13], [14], both objectives could be improved significantly by Explicit Congestion Notification (ECN) marks communicating *shadow prices* through the network, without the need for other higher complexity solutions that are being considered (e.g., differentiated services).

F. Paganini and Z. Wang are with University of California, Los Angeles; emails: {paganini,zkwang}@ee.ucla.edu. S. Low, and J. Doyle are with the California Institute of Technology; emails: {slow@its,doyle@cds}.caltech.edu.

Another motivation comes from deficiencies of AIMD as the network further scales up in capacity. In such fast networks, congestion windows will easily scale up into the thousands, which creates two problems. On one hand, AI is too slow, since it can only change the window by one every round-trip-time, so changes of thousands can easily take minutes. On the other hand, MD is too fast: recent studies [3], [4], [5], [6] have shown that in these high window regimes, TCP combined with RED [2] is unstable, leading to dramatic oscillations in network queues that not only cause delay jitter but can even impact utilization.

It is quite possible that incremental modifications to the current protocols are able to deal with these issues in a satisfactory way. However, in recent years large strides have been taken in the analytical front, with tools from convex optimization coming into play to analyze resource allocation [12], [15], and advances in control theory to analyze stability [19], [7], [8], [17], which for the first time can tackle the case of truly large-scale networks. Given this scenario, it appears worth exploring how far could one go with analytical methods if we were to "do it all again". This is the motivation for the present paper.

In earlier work [7], we developed, at the level of fluid-flow models, a TCP/AQM congestion control system that could achieve high link utilization, low delay and scaled itself to provide dynamic stability for arbitrary networks and delays. We remark here that although instabilities such as oscillations could perhaps be tolerated in the network context, the boundary of stability is worth characterizing since it reflects the limits of predictable behavior in the network; the control laws in [7] (reviewed in Section II) are aimed at operating precisely at that boundary. What this solution does not allow is freedom in the resource allocation between sources; instead, a response curve must be imposed on sources,

depending on their round-trip-time (RTT), that will determine their allocated throughput.

A first contribution of this paper is to extend the theory in [7] to allow for an arbitrary choice of source utility functions; this could be used for instance to impose fairness among users who see the same bottleneck, independently of their RTT. This property is obtained by a new source control that uses separation of time-scales, running the "fairness" loop slower than a commonly agreed bound on the RTT. Other than this restriction, the stability proof extends to an arbitrary network.

The second objective of this paper is to go beyond fluid-flow models and pursue this family of protocols to the level of a packet implementation, within the constraints of mechanisms currently available in the Internet. We employ an explicit congestion notification (ECN) bit and the technique of random exponential marking (REM, [11]), as a means for communicating the price signal from links back to sources[1]. In Section IV we examine some practical considerations as to the choice of the marking parameter, and the price estimation process at the sources, and we describe the discretization of the flow control employed at sources and links.

In Section V we perform some tests to demonstrate the performance of the protocol and its comparisons with other versions of TCP and AQM, in highly stressed congestion scenarios and high capacity links. In particular, we consider tests of persistent flows and also of file transfers drawn from a heavy-tailed distribution. The results show this protocol significantly enhances link utilization while keeping queues empty, and is also able to adjust fairness in the case of persistent flows.

Conclusions are given in Section VI

II. Problem formulation and earlier work

A. Fluid-flow model and control objectives

We are concerned with a system of L communication links shared by a set of S sources. The routing matrix R, of dimensions $L \times S$, is defined by

$$R_{li} = \begin{cases} 1 & \text{if source } i \text{ uses link } l \\ 0 & \text{otherwise} \end{cases},$$

and assumed fixed. The theory will be based on a fluid-flow abstraction of the TCP/AQM congestion control problem. Each source i has an associated transmission rate $x_i(t)$; the set of transmission rates determines the aggregate flow $y_l(t)$ at each link, by the equation

$$y_l(t) = \sum_i R_{li} x_i(t - \tau_{li}^f), \tag{1}$$

in which the forward transmission delays τ_{li}^f between sources and links are accounted for. Each link has a capacity c_l in packets per second.

Next, we model the feedback mechanism which communicates to sources the congestion information about the network. The key idea is to associate with each link l a congestion measure or *price* $p_l(t)$ [12], [15], and assume sources have access to the *aggregate* price of all links in their route,

$$q_i(t) = \sum_l R_{li} p_l(t - \tau_{li}^b). \tag{2}$$

Here again we allow for backward delays τ_{li}^b in the feedback path from links to sources. As discussed in [16], [6], this feedback model includes, to a good approximation, the mechanism present in existing protocols, with a different interpretation for price in different protocols (e.g. loss probability in TCP Reno, queueing delay in TCP Vegas). The total RTT for the source thus satisfies

$$\tau_i = \tau_{li}^f + \tau_{li}^b \tag{3}$$

for every link in the source's path. The vectors x, y, p, q collect the above quantities across sources and links.

In this framework, a congestion control system is specified by choosing (i) how the links fix their prices based on link utilization; (ii) how the sources fix their rates based on their aggregate price.

We remark that we are directly modeling only persistent sources, i.e. those long enough to be controlled. From the point of view of these "elephants", what matters mainly is that the system reaches an equilibrium point x_0, y_0, p_0, q_0 with high network utilization and adequate resource allocation among them. The network is, however, also shared by short "mice", which don't last long enough to be controlled, and for which no "equilibrium" exists, but who are affected by the dynamic properties of the control (e.g. how fast it responds to freely available bandwidth), and by the queuing delay they experience. We will not model them explicitly here (they

[1] This does not mean using the AQM method in [11], only the price "coding" technique.

could be treated as noise in link rates), but will bear these objectives in mind for our design.

Specifically, we lay out the following design objectives:

1. Network utilization. Link equilibrium rates y_{0l} should of course not exceed the capacity c_l, but also should attempt to track it.

2. Equilibrium queues should be empty to avoid queueing delays.

3. Resource allocation. We will assume sources have a demand curve

$$x_{0i} = f_i(q_{0i}) \tag{4}$$

that specifies their desired equilibrium rate as a decreasing function of price. This is equivalent to assigning them a concave *utility function* $U_i(x_i)$, in the language of [12]; in this case $f_i = (U_i')^{-1}$. We would like the control system to reach an equilibrium that accommodates these demands. This does not in itself ensure fairness, or address differnetiated services, but provides a tuning knob in which to tackle these kind of issues.

4. Stability. The equilibrium should be (at least locally) stable.

What makes this a challenging problem is that we require the above to hold for *arbitrary* networks, and that one must work with very tight information constraints: sources and links have only access to their respective variables, and nobody knows what the overall network is.

B. Control laws with scalable stability

We describe here the control laws of [7], which achieve three of the above objectives.

At the links, the price dynamics is defined as

$$\dot{p}_l = \begin{cases} \frac{y_l - c_{0l}}{c_{0l}}, & \text{if } p_l > 0 \text{ or } y_l > c_{0l}; \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where c_{0l} is a target "virtual" capacity. At equilibrium, bottlenecks with nonzero price will have $y_{l0} = c_{0l}$, and non-bottlenecks with $y_{l0} < c_l$ will have zero price. Note that with this choice the price is the "virtual" queueing delay one would get if the capacity was c_{0l}. Choosing c_{0l} slightly below the actual capacity c_l ensures high utilization and at the same time that the real queues are empty, as intended by the first two design objetives.

At the sources, a static rate function of the aggregate price is proposed:

$$x_i = x_{\max,i}\, e^{-\frac{\alpha_i q_i}{M_i \tau_i}}. \tag{6}$$

Here τ_i is the RTT, α_i a constant, and M_i a bound on the number of bottlenecks in source i's path. $x_{\max,i}$ is a maximum rate parameter, which can depend on M_i, τ_i (but not on q_i).

The above control laws are a special case of those in [15], and therefore define a unique equilibrium point. The main result of [7] is that the equilibrium is locally stable for arbitrary networks, parameters, and delays. This is shown by considering a perturbation $x = x_0 + \delta x$, $y = y_0 + \delta y$, $p = p_0 + \delta p$, $q = q_0 + \delta q$ around equilibrium, and writing the linearized equations in the Laplace domain:

$$\delta\bar{y}(s) = \bar{R}_f(s)\delta x(s), \tag{7}$$

$$\delta q(s) = \bar{R}_b(s)^T \delta\bar{p}(s), \tag{8}$$

$$\delta\bar{p} = \mathcal{C}\frac{I}{s}\delta\bar{y}, \tag{9}$$

$$\delta x = -\mathcal{K}\delta q. \tag{10}$$

Here we use the notation $\delta\bar{p}, \delta\bar{y}$ to indicate the reduced vectors obtained by eliminating non-bottleneck links, which do not contribute to the linear dynamics. Thus (1-2) linearize to (7-8), where the matrices $\bar{R}_f(s)$ and $\bar{R}_b(s)$ are obtained by eliminating non-bottleneck rows from R, and also replacing the "1" elements respectively by the delay terms $e^{-\tau^f_{i,l}s}$, $e^{-\tau^b_{i,l}s}$. The diagonal matrices

$$\mathcal{C} = \mathrm{diag}(\frac{1}{c_{0l}}), \quad \mathcal{K} = \mathrm{diag}(\kappa_i)$$

are derived from the linearization of (5-6); in particular the number of integrators in (9) equals the number of bottlenecks, and the diagonal gains of (10) are

$$\kappa_i = \frac{\alpha_i x_{0i}}{M_i \tau_i}. \tag{11}$$

The above equations lead to a formula for the overall multivariable loop transfer function

$$L(s) = \bar{R}_f(s)\mathcal{K}\bar{R}_b^T(s)\mathcal{C}\frac{I}{s}. \tag{12}$$

The stability of such loops with integral control is studied in [7] via the following proposition.

Proposition 1: Consider a standard unity feedback loop, with $L(s) = F(s)\frac{I}{s}$. Suppose:
(i) $F(s)$ is analytic in $Re(s) > 0$ and bounded in $Re(s) \geq 0$.
(ii) $F(0)$ has strictly positive eigenvalues.
(iii) For all $\gamma \in (0, 1]$, -1 is not an eigenvalue of $\gamma L(j\omega)$, $\omega \neq 0$.

Then the closed loop is stable.

We have the following general result on scalable stability for arbitrary networks under a mild rank restriction.

Theorem 2 ([7]) Suppose the matrix $\bar{R} := \bar{R}_f(0) = \bar{R}_b(0)$ is of full row rank, and that $\alpha_i < \frac{\pi}{2}$. Then for arbitrary delays and link capacities, the system under control laws (5-6) has a unique equilibrium point which is locally stable.

Taking $F(s) = \bar{R}_f(s)\mathcal{K}\bar{R}_b^T(s)\mathcal{C}$, it is easy to establish here that (i) holds, and (ii) follows from the rank assumption on $\bar{R}$. The more delicate step is (iii), as we will see below when discussing a generalization.

The laws of [7] satisfy the equilibrium objectives on the link side, and stability. However the exponential laws in (6) specify a fixed demand curve for the sources, or equivalently a fixed utility function. Some degrees of freedom are left in the choice of $x_{\max,i}$, and one could further generalize these laws as indicated in [7]. Nevertheless we do not have complete freedom in the choice of the demand curve, as we had aimed for in Section II-A. In particular, we would like to be able to eliminate the dependence of the equilibrium structure on the RTT, which is also present in current protocols. We remark that parallel work in [9] has derived solutions with scalable stability and arbitrary utility functions, but where the link utilization requirement is relaxed. Indeed, it appears that one must choose between the equilibrium conditions on either the source or the link side, if one desires a scalable stability theorem. In the next section we show how this difficulty is overcome if we slightly relax our scalability requirement.

III. A NEW FLOW CONTROL WITH ENHANCED FAIRNESS

The reason we are getting restrictions on source utility is that for static laws, the elasticity of the demand curve (the control gain at DC) coincides with the high frequency gain, and is thus constrained by stability. One way of decoupling the two gains is to replace the linearized source control by a dynamic, lead-lag compensation of the form

$$\delta x_i = -\frac{\kappa_i(s+z)}{s+\frac{z\kappa_i}{\nu_i}}\delta q_i. \qquad (13)$$

Here the high frequency gain κ_i is the same as in (11), "socially acceptable" from a dynamic perspective. The DC gain $\nu_i = -f_i'(q_{i0})$ is the elasticity of source demand based on its own "selfish" demand curve $x_{i0} = f_i(q_{i0})$, that need no longer be of the form (6). The *zero* z is assumed fixed across sources.

- If $\nu_i \leq \kappa_i$, a static source controller based on its utility would be within the limits of the earlier stability theorem, without any need for compensation. In this case, the above controller provides a *higher* gain at cross-over frequency, so that the network utilization loop reacts as fast as possible compatible with stability. It also gives phase *lead*, which reinforces the idea that stability is not compromised.
- If $\nu_i > \kappa_i$, the compensation forces the aggressive source to reduce its gain at cross-over frequency to maintain stability. Note that here the source pole is lower than z; the more aggressive the source tries to be, the slower this response becomes, to keep the high frequency behavior roughly intact. We also have a phase *lag* in this case, which means that care must be taken in the stability analysis.

A. Local Stability Results

With the new local source control, we will proceed to study the linearized stability of the closed loop, generalizing the method of Theorem 2. We first write down the overall loop transfer function

$$L(s) = R_f(s)\mathcal{K}(s)R_b^T(s)\mathcal{C}\frac{I}{s}, \qquad (14)$$

which is analogous to (12) except that now

$$\mathcal{K}(s) = \mathrm{diag}\big(\kappa_i V_i(s)\big), \text{ with } V_i(s) = \frac{s+z}{s+\frac{z\kappa_i}{\nu_i}},$$

κ_i as in (11). The stability argument is based again on Proposition 1, the key step being once more the study of the eigenvalues of $\gamma L(j\omega)$.

As in [7], the key structure that is employed is the relationship

$$\bar{R}_b(s) = \bar{R}_f(-s)\mathrm{diag}(e^{-\tau_i s}),$$

which follows from (3), and allows us to write

$$L(j\omega) = R_f(j\omega)X_0\mathcal{M}\Lambda(j\omega)R_f(j\omega)^*\mathcal{C}, \qquad (15)$$

$$\begin{aligned} L(j\omega) &= R_f(j\omega)X_0\mathcal{M}\Lambda(j\omega)R_f(j\omega)^*\mathcal{C}, \\ \text{with } X_0 &= \mathrm{diag}(x_{0i}), \quad \mathcal{M} = \mathrm{diag}(\frac{1}{M_i}), \\ \Lambda(j\omega) &= \mathrm{diag}(\lambda_i(j\omega)). \end{aligned}$$

The only change with respect to [7] is that we have added the lead-lag term $V_i(s)$ to the diagonal elements of $\Lambda(s)$,

$$\lambda_i(s) = \frac{\alpha_i e^{-\tau_i s}}{\tau_i s} V_i(s). \quad (16)$$

Proceeding with the method of [7], we write $eig(\gamma L(j\omega)) = \mathrm{eig}\Big(\gamma P(j\omega)\Lambda(j\omega)\Big)$, where

$$P(j\omega) := \mathcal{M}^{\frac{1}{2}} X_0^{\frac{1}{2}} R_f(j\omega)^* \, \mathcal{C} \, R_f(j\omega) X_0^{\frac{1}{2}} \mathcal{M}^{\frac{1}{2}} \geq 0;$$

it follows in a similar way that $\rho(\gamma P) \leq \rho(P) \leq 1$. Then using Vinnicombe's lemma [8], the eigenvalues of $L(j\omega)$ are convex combinations of the $\lambda_i(j\omega)$, and the origin.

It remains to give conditions so that the convex combinations of the $\lambda_i(j\omega)$, which now include an extra lead-lag term, do not reach the critical point -1. Figure 1 contains various Nyquist plots of $\lambda_i(j\omega)$, for τ_i ranging between 1ms and 1sec, and ratios ν_i/κ_i ranging between 0.1 and 1000. The value of z is fixed at 0.2, and $\alpha = 1$.

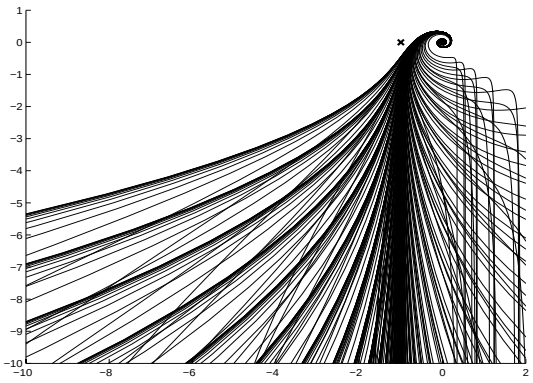

Fig. 1. Nyquist plots of $\lambda_i(j\omega)$, $z = 0.2$, $\alpha = 1$, various τ_i and ν_i/κ_i.

A first comment is that here the plots do not coincide, as they did in the "scale-invariant" case of [7], when we used the source control κ_i (only the high frequency portion of the above plots coincide).

Secondly, we note that there is not an obvious separation between the convex hull of these points and the critical point -1. One could think of obtaining convex separation through a slanted line; this however, would imply a lower limit $-\pi + \theta$, $\theta > 0$ on the phase of $\lambda_i(j\omega)$ at low frequencies, which in turn implies, based on (16), a limit on the lag-lead gain ratio ν_i/κ_i. This may be acceptable, but would not allow us to accommodate *arbitrary* utilities.

The alternative is to treat the low-frequency portion of the above curve separately, ensuring for instance that it doesn't reach phase $-\pi$. This, however, implies a common notion of what "low-frequency" means, so that we are not operating in different portions of the curve for sources with different RTTs. This can be obtained through a fixed bound $\bar{\tau}$ on the RTT, as follows.

Proposition 3: Assume that for every source i, $\tau_i \leq \bar{\tau}$. In the source controllers (13), choose $\alpha_i = \alpha < \frac{\pi}{2}$ and $z = \frac{\eta}{\bar{\tau}}$. Then for a small enough $\eta \in (0, 1)$ depending only on α, $-1 \notin eig(L(j\omega))$.

Proof:

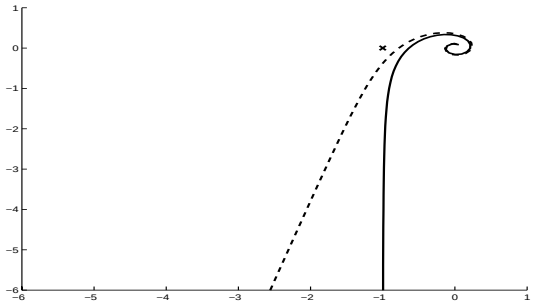

Fig. 2. Plots of ∇ (solid) and ∇_η (dashed)

For fixed $\alpha < \frac{\pi}{2}$, let $\nabla = \{\frac{\alpha e^{-j\theta}}{j\theta} : \theta > 0\}$. This curve, depicted by the solid curve in Figure 2, would be the Nyquist plot of $\lambda_i(j\omega)$ for $\nu_i = \kappa_i$. ∇ is strictly away from the critical point.

We now quantify the extra gain and phase introduced by the lag-lead $V_i(j\omega)$ for frequencies $\omega \geq \frac{1}{\bar{\tau}}$:

$$\left|\frac{j\omega + z}{j\omega + p}\right| \leq \sqrt{1 + \frac{z^2}{\omega^2}} \leq \sqrt{1 + \eta^2}, \quad (17)$$

$$\text{phase}\left(\frac{j\omega + z}{j\omega + p}\right) \geq -\arctan(\frac{z}{\omega}) \geq -\arctan(\eta) \quad (18)$$

This means for this frequency range, $\lambda_i(j\omega)$ will lie always below the perturbed curve

$$\nabla_\eta := \sqrt{1 + \eta^2} e^{-j \arctan(\eta)} \nabla.$$

(a slight clockwise rotation and expansion of ∇), depicted by dashed lines in Figure 2. By appropriately small choice of η, depending only on α, we can make sure that this curve stays below the critical point. It follows that convex combinations of $\lambda_i(j\omega)$ cannot reach the critical point for $\omega \geq \frac{1}{F\bar{\tau}}$.

It remains to consider the frequencies $\omega \in (0, \frac{1}{\bar{\tau}})$. We will argue that in this frequency range, $\lambda_i(j\omega)$ is

always in the lower half-plane (negative imaginary part), and hence again one cannot obtain the critical point by convex combinations.

To see this, compute

$$\begin{aligned}\text{phase}(\lambda_i(j\omega)) &= -\frac{\pi}{2} - \tau_i\omega + \text{phase}(V_i(j\omega)) \\ &> -\pi - \tau_i\omega + \arctan(\frac{\omega}{z}) \\ &\geq -\pi - \bar{\tau}\omega + \arctan(\frac{\bar{\tau}\omega}{\eta})\end{aligned}$$

Thus it suffices to show that for $\omega \in (0, \frac{1}{\bar{\tau}})$,

$$\arctan(\frac{\bar{\tau}\omega}{\eta}) > \bar{\tau}\omega,$$

or equivalently

$$\eta < \frac{\bar{\tau}\omega}{\tan(\bar{\tau}\omega)}$$

The right hand-side is decreasing in $\bar{\tau}\omega < 1$, so it suffices to choose $\eta < \frac{1}{\tan(1)} \approx 0.64$. ■

B. Nonlinear implementation

We now discuss how to embed our new linearized source control law in global nonlinear laws. The requirements are:

- The equilibrium matches the desired utility function, $U_i'(x_{0i}) = q_{0i}$, or equivalently the demand curve (4) for $f_i = (U_i)^{-1}$.
- The linearization is (13), with the zero z being fixed, independently of the operating point and the RTT.

We now present a nonlinear implementation that satisfies these conditions, of a similar nature to laws obtained in the "primal" approach [12], [19], [10].

$$\tau_i\dot{\xi}_i = \beta_i(U_i'(x_i) - q_i), \tag{19}$$

$$x_i = x_{m,i}e^{(\xi_i - \frac{\alpha_i q_i}{M_i\tau_i})}. \tag{20}$$

Note that (20) corresponds exactly to the rate control law in (6), with the change that the parameter $x_{\max}$ is now varied exponentially as

$$x_{\max,i} = x_{m,i}e^{\xi_i},$$

with ξ_i as in (19). If β_i is small, the intuition is that the sources use (6) at fast time-scales, but slowly adapt their $x_{\max_i}$ to achieve an equilibrium rate that matches their utility function, as follows clearly from equation (19).

We now find the linearization around equilibrium; the source subscript i is omitted for brevity. For increments $\xi = \xi_0 + \delta\xi$, $x = x_0 + \delta x$, $q = q_0 + \delta q$, we obtain the linearized equations:

$$\begin{aligned}\tau\delta\dot{\xi} &= \beta\big(U''(x_0)\delta x - \delta q\big) = \beta\big(-\frac{\delta x}{\nu} - \delta q\big), \\ \delta x &= x_0(\delta\xi - \frac{\alpha}{M\tau}\delta q) = x_0\delta\xi - \kappa\delta q.\end{aligned}$$

Here we have used the fact that $U''(x_0) = \frac{1}{f'(q_0)} = -\frac{1}{\nu}$, and the expression (10) for κ . Some algebra in the Laplace domain leads to the transfer function

$$\delta x = -\kappa\left(\frac{s + \frac{\beta x_0}{\kappa\tau}}{s + \frac{\beta x_0}{\nu\tau}}\right)\delta q,$$

that is exactly of the form in (13) if we take

$$z = \frac{\beta x_0}{\kappa\tau} = \frac{\beta M}{\alpha}.$$

By choosing β, the zero of our lead-lag can be made independent of the operating point, or the delay, as desired.

We recapitulate the main result as follows.

Theorem 4: Consider the source control (19-20) where $U_i(x_i)$ is the source utility function, and the link control (5). At equilibrium, this system will satisfy the desired demand curve $x_{i0} = f_i(q_{i0})$, and the bottleneck links will satisfy $y_{0l} = c_{0l}$, with empty queues. Furthermore, under the rank assumption in Theorem 2, $\alpha_i < \frac{\pi}{2}$, and $z = \frac{\beta_i M_i}{\alpha_i}$ chosen as in Proposition 3, the equilibrium point will be locally stable.

We have thus satisfied all the objectives set forth in Section II-A, except for the fact that an overall bound on the RTT had to be imposed.

IV. Packet-level Implementation

In this section, we describe a packet-level implementation in ns-2 of the new algorithms in section III, including the mechanism of price estimation and transmission, and the pseudo code for the source and link algorithms.

A. Marking and Estimation

The key requirement for the implementation of the above protocols is the communication of price signals from links back to sources, which then use them to adapt their rates. We explore in this section the

use of an ECN bit in the packet header as a means of communicating prices in an additive way over congested links.

We employ the technique of Random Exponential Marking (REM, [11]), in which an ECN bit would be marked at each link l with probability

$$1 - \phi^{-p_l}$$

where $\phi > 1$ is a global constant, known by everyone in the network. Assuming independence between links, the overall probability that a packet from source i gets marked is (see [11])

$$\mathcal{P}_i = 1 - \phi^{-q_i}, \tag{21}$$

and therefore q_i can be estimated from marking statistics. For example, a shift-register of the last N received marks can be maintained, the fraction of positive marks providing an estimate $\hat{\mathcal{P}}_i$ of the marking probability, from which an estimate $\hat{q}_i$ could be derived.

While simple in principle, two related issues are important to make this scheme practical:
1. Is there a global parameter ϕ such that the new protocol could work in the practical network ?
2. The estimation window introduces additional *delay*, which if excessive can compromise stability. This limits the size N of the estimation window to be used.

For the first issue, we can only estimate prices accurately enough if the marking probability is not too close to 0 or 1. Now for a fixed ϕ, restricting the marking probability to, say, the range 5% to 95%, means restricting the price to a certain absolute interval; the table below shows this interval for different choices of ϕ.

ϕ	$q_{\min}$ (sec)	$q_{\max}$ (sec)
10	0.022	1.3
100	0.011	0.65
1000	0.007	0.43

Is there any reason to expect prices to be largely confined to one of these absolute ranges, regardless of the network scenario? One reason for optimism is that our prices are *virtual queueing delays*, i.e. the queueing delay that would be experienced if the link capacities were slightly reduced. One can argue that for this reason they should be of the order of RTTs currently observed in the network, which has significant queues. Invoking [18], it appears that a range of 0.02 to 1 second seems to cover most occurrences. So perhaps a uniform ϕ in the order given above would be successful in the current Internet.

Another comment is that one could allow the marking probability to drop below 5% for low prices, provided one retains accuracy in higher prices, which are more critical to congestion. In Section V we will show simulations of satisfactory behavior with marking probabilities as low as 0.3%.

With respect to the issue of estimation, we note that a window of size N implies a delay in the receipt of the price signal which is of the order of the time it takes to receive $\frac{N}{2}$ packets. This means an additional delay

$$\tau_{est} \approx \frac{N}{2w}\tau, \tag{22}$$

where w is the current congestion window. To see this, assume that packets are arriving with uniform spacing $h = \frac{\tau}{w}$; then the estimator behaves like a moving average filter of frequency response

$$H(j\omega) = \frac{1}{N}\sum_{k=0}^{N-1} e^{-j\omega kh} = e^{-j\omega h\frac{N-1}{2}} H_0(j\omega),$$

where the frequency function $H_0(j\omega)$ is real valued, and has gain bounded by 1. Then this filter would contribute a linear phase of

$$\omega h \frac{N-1}{2}$$

approximately equivalent to a pure delay τ_{est} given in (22). From the point of view of linear stability, Theorem 2 requires that the feedback delay be compensated by a decrease in gain. $H_0(j\omega)$ may or may not provide attenuation, depending on the critical frequency, so stability could be compromised. To avoid, this, we make the following recommendations:

- Avoid too high estimation windows. The estimation variance will decay as $1/N$, so it appears that going beyond $N = 100$ would be unnecessary: some random fluctuations in the estimated price (and thus in the window control) can be tolerated, since they will average out in the longer term.
- Use a value of the gain parameter α that is not too close to the stability limit. For instance, the value $\alpha = 0.37$ gives a critically damped response in the case of many identical sources, when we ignore estimation delays. It can tolerate some additional delay without becoming unstable.

• If the congestion window becomes small, reduce the estimation window to keep the additional delay in (22) under control.

To recapitulate our discussion in this section: going from an ideal feedback of the price to an implementation based on ECN bits requires some practical engineering considerations. In particular, the system would not work under a careless choice of the design parameters ϕ and N. Fortunately, it appears that there is enough maneuvering room to yield a satisfactory implementation in scenarios relevant to the current Internet. This will be explored in more detail in Section V.

B. Source and link algorithms

From Theorem 4, the utility function could be selected arbitrarily according to the requirements of the sources. In the following implementation, we use the utility function [12] $U_i(x_i) = K_i \log(x_i)$, which induces the so-called "proportional fairness". If we further set K_i the same constant for all the sources of the network, then sources seeing the same bottlenecks (and thus the same price) would receive an equal allocation of bandwidth.

We then proceed to discretize in time the differential equations (19-20), using a sampling interval T_s. Noting that the congestion window w can be approximated as $x\tau$, we write

$$\xi_i(k) = \xi_i(k-1) + \beta_i \left(\frac{K_i}{w_i} - \frac{q_i(k)}{\tau_i} \right) T_s, \quad (23)$$

$$w_i(k) = w_{m,i} e^{\left(\xi_i(k) - \frac{\alpha_i q_i(k)}{M_i \tau_i} \right)}. \quad (24)$$

Therefore, one direct implementation is to maintain a timer to act as an integrator and update the window upon timeout. The source operations are described by the pseudo-code in Figure 3.

At the sources, the price from the links is estimated from the ECN bits in the latest N packets on every ACK arrival. The difference equations (23-24) are used to calculate the the expected congestion window. We set the $baseRTT$ as the minimal RTT over time. Also we impose the capping on the change of the congestion window per ACK to mitigate the noise from price estimation. Furthermore, the output packet flows are paced uniformly over each RTT.

We remark that this window protocol is the result of our initial experimentation to validate the theory; more experience is required before we settle into a definitive solution. Obviously, improvements could be made to get higher efficiency or better performance. For instance, to avoid the complex calculation of the exponential function, we could derive another window management scheme. From equations (19-20), we have

Every $intInterval$ seconds:

$$\xi \leftarrow \xi + \beta * \left(\frac{K}{expWnd} - \frac{estPrice}{baseRTT} \right) * intInterval;$$

$$expWnd \leftarrow W_m * exp\left(\xi - \frac{\alpha * estPrice}{M * baseRTT} \right);$$

On each ACK arrival:

Estimate $estProb$ using last N ACKs;
$tmp \leftarrow CWnd - expWnd;$
$if(tmp > maxDecrement)$
  $CWnd \leftarrow CWnd - maxDecrement;$
$elseif(tmp < -maxIncrement)$
$CWnd \leftarrow CWnd + maxIncrement;$
$else$
  $CWnd \leftarrow expWnd;$

Variables:
ξ: state variable;
$estPrice$: estimated price from the marking probability;
$expWnd$: expected congestion window.

Parameters:
α: stability parameter;
β: constant from the zero point;
Φ: constant for price communication;
N: the length of the estimation window;
K: from the utility function $Klog(x)$;
$intInterval$: integral period.

Fig. 3. pseudo code of the source algorithm

$$\dot{x}_i = \beta_i K - x_i \left(\frac{K q_i}{\tau_i} + \frac{\alpha_i}{M_i \tau_i} \dot{q}_i \right), \quad (25)$$

yielding the discrete window update equations as

$$tmp = 1 + \left(\frac{K T_s}{\tau_i} + \frac{\alpha_i}{M_i \tau_i} \right) q(k) - \frac{\alpha_i}{M_i \tau_i} q(k-1), \quad (26)$$

$$w(k) = \frac{1}{tmp} w(k-1) + \beta_i K T s. \quad (27)$$

On the link side, we discretize similarly the equation (5), with interval $\tilde{T}_s$, as following:

$$p(k) = \left[p(k-1) + \frac{y_l(k) - \gamma c_l}{\gamma c_l} \tilde{T}_s \right]^+. \quad (28)$$

Here $y_l(k)\tilde{T}_s$ means number of arrivals at the queue during the interval. The pseudo code is given in Figure 4.

Every packet enque:

$pktCounter++$.

Every $updInterval$ seconds:

$aveInput \leftarrow \dfrac{pktCounter}{updInterval}$;

$price \leftarrow price + \left(\dfrac{aveInput}{virtCap} - 1\right) * updInterval$;

$prob \leftarrow 1 - \Phi^{-price}$;

$pktCounter \leftarrow 0$.

Every packet dequeue:

$temp \leftarrow$ uniform();

if $temp \geq prob$ marking the packet .

Variables:
$price$, $prob$, $pktCounter$.

Parameters:
Φ: constant, the same as that of the sources;
$virtCap$: Virtual Capacity, $\gamma * Capacity$;
$updInterval$: update period.

Fig. 4. pseudo code of the link algorithm

V. Simulations and Results

In this section, we simulate the algorithms with ns-2 to validate the desired performance of the new protocol, including the scalable stability, empty queue, fairness and high utilization. We also compare it with TCP NewReno sources combined with other active queue management schemes such as Reno/RED, Reno/AVQ and Reno/PI.

A. Performance with two-way long-lived traffic

All our simulations use two-way long-lived traffic on a single bottleneck link with one way capacity of 2Gbps (250pkts/ms with mean packet size 1000bytes). It is shared by 512 ftp flows in each direction. The number of flows in *each* direction is doubled every 20 seconds, from 32, to 64, 128, 256, and finally to 512 flows. These groups of flows have round trip propagation delays of 40ms, 80ms, 120ms, 160ms and 200ms respectively. This scenario is designed to stress a high-capacity link with heterogeneous flows.

For the new protocol, we set the target link utilization to be $c_{0l}/c_l = 0.95$, stability parameter $\alpha = 0.37$, the utility function $50\log(x)$, and other parameters as following: $M_i = 1, N = 31, \phi = 100, \beta = 1.5, T_s = 5ms, \tilde{T}_s = 5ms$. All AQM schemes have ECN marking. We set RED parameters as $thresh = 100, maxthresh = 2500$, $q\_weight = 0.002$, and the PI parameter $q\_ref = 100$. In the case of two-way traffic, all the $qib\_$ parameters are set to be $true$. To validate the performance around equilibrium, we use large buffer to avoid overflow.

The simulation results of the different schemes are shown in Figure 5, 6, 7, 8 and 9. The queues shown in the plots are in packets, and the rates in packets/sec. The utilization of the new protocol is averaged over 20ms, and those of the other schems 2sec. For NewReno (with all AQM schemes), the interval between source activation is doubled to 40 sec in order to clearly see the equilibrium behavior.

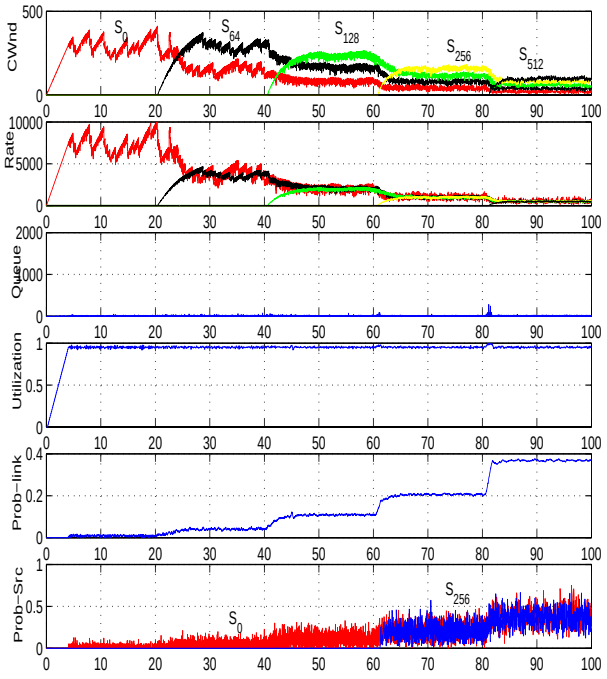

Fig. 5. new protocol with long-lived traffic

Compared with the other schemes, the new protocol shows the desired performance under these conditions.

First, the source rates and the link prices (marking probability) track the expected equilibria when new sources activate. The window is much smoother than those of the AIMD schemes. The new protocol works well with both large and small equilibrium windows. Although the estimated probability

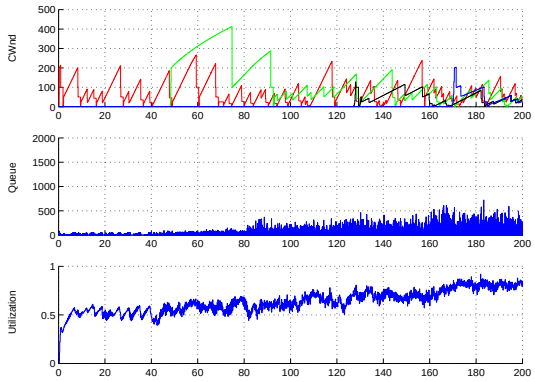

Fig. 6. NewReno/RED with long-lived traffic

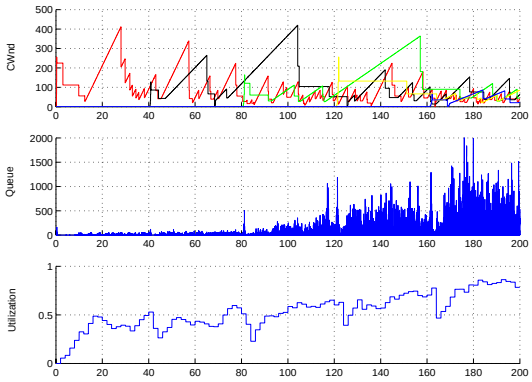

Fig. 8. NewReno/VQ with long-lived traffic

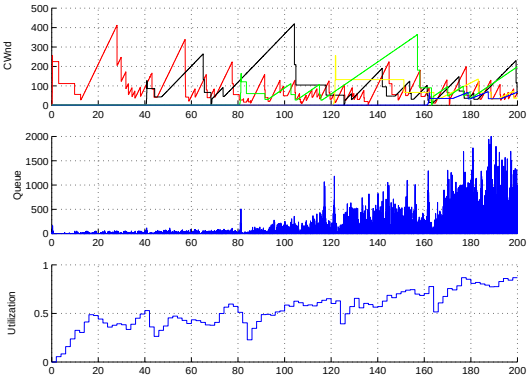

Fig. 7. NewReno/AdaptiveRED with long-lived traffic

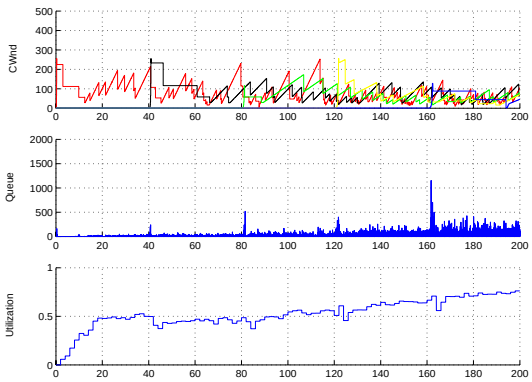

Fig. 9. NewReno/PI with long-lived traffic

of the source is extremely noisy, the new protocol still works well.

Second, proportional fairness is achieved at equilibrium. The bandwidth shares of the heterogeneous sources at equilibrium are independent of their RTTs under the new protocol.

Third, the queue is small (around 20 packets) and smooth almost all the time, both in transient and in equilibrium. The queue overshoot in Figure 5 is caused by the activation of 256 new flows in each direction in a short time. In the other AQM schemes with NewReno sources, the queueing delay increases when more flows are activated.

Finally, the utilization of the link with the new protocol is always around the 95% target. The utilization with AIMD is much lower at low load; as load increases, utilitization also improves but only with rising queuing delay. The various AQM schemes seem to have little difference in handling this tradeoff in our simulations.

Another observation is the approximately linear increase experienced by sources at the beginning, i.e. when the price is small. This is a consequence of the dynamics that can be seen most easily from (26-27), while q remains small. The slope of this increase is mainly determined by the product $\beta_i K$, and therefore could be affected by changing the utility function. The relatively slow increase helps avoid temporary queue overshoots as new sources start. Alternatively, one could tolerate larger overshoots to obtain a faster response.

When new flows are activated, the new equilibrium price increases. Since the price is equal to the virtual queueing delay, the new equilibrium virtual queue should increase to a higher value. It takes time for the virtual queue to built up to the new equilibrium, which causes the smooth increase of the price, but also contribute partly to the delay of the price feedback and overshoot of the real queue. By appropriate configuration of the parameters, these overshoots can be absorbed by excessive capacity.

We remark that, with regard to the parameters of RED, a queue with $2500 packets$ means $10ms$ of queue delay with capacity of $250 packets/ms$, which is reasonable. If we take smaller thresholds, e.g, 5 and 20 respectively, then we could have a small

and smooth queue, but with a utilization no more than 50%.

One issue is whether the new protocol could work with extremely small marking probability. By choosing a small value of ϕ, we choose a maximum marking (in the highly congested case) of 0.05; still Figure 10 shows a satisfactory, though somewhat noisier performance.

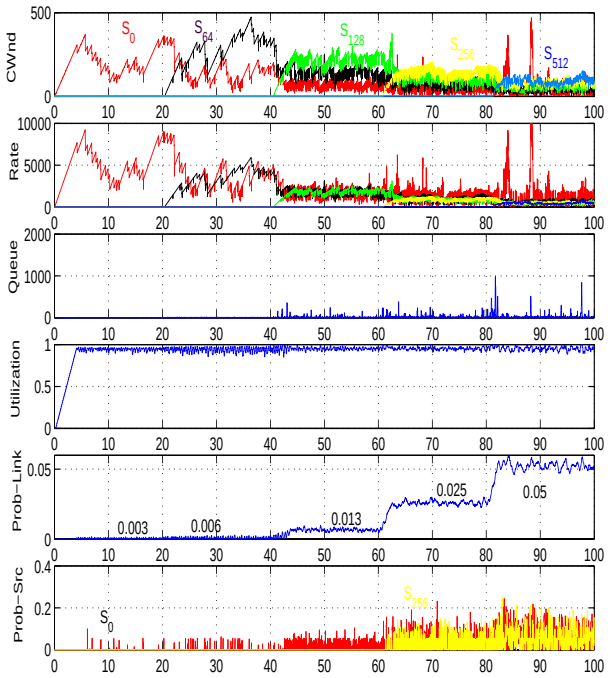

Fig. 10. New Protocol with small marking probability

B. *Performance with two-way heavy-tailed traffic*

The above simulations involve only long-lived traffic, where all sources stay long enough to be controlled. In a real network, flows have been found to follow a heavy-tailed distribution, going from the extremes of short "mice" that cannot be controlled, to long "elephants" that will respond as above, with flows with intermediate sizes. How does the new protocol behave in this environment?

We use the same setup as in the previous section except that all sources start at the same time and the link capacity is 1Gbps. To simulate heavy-tailed traffic, flow sizes are randomly generated according to a Pareto distrition. All the 1024 sources generate such flows with inter-arrival times exponentially distributed. After each flow transmission is finished, the tcp agent is reset to the initial states.

Figure 11 shows one sample of the Pareto series with shape 1.0 and scale 100, producing heavy-tailed flows. For instance, among the 34078 flows that generate a total of 3.33e+7 packets, more than 90% of the flows have sizes less than 1000 packets, but they only contribute around 25% of the overall packets. Elephants dominate, despite being small in number.

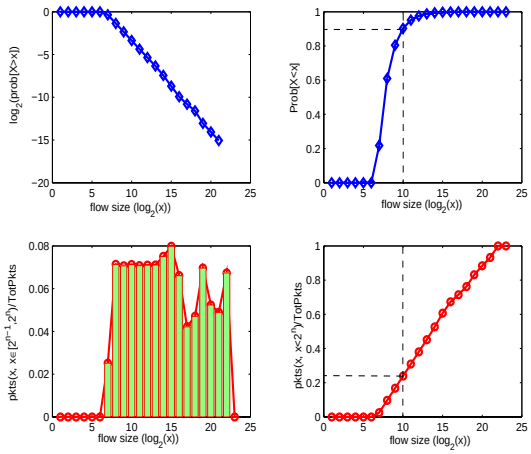

Fig. 11. sample of random series with pareto distribution

Compared with the other four schemes, the new protocol still keeps high utilization and small queue as in the long-connection cases, see Figure 12. There exists only a few overshoots. Also, the elephants share the bandwidth fairly.

It has been widely verified that NewReno/RED works quite well over the web traffic. However, to get a high utilization, we have to tolerate high queueing delay, see Figure 13. Again, if we set the thresholds small, say, 5 and 20, the utilization will fall to less than 50%. Under this scenario, the other AQM schemes, including Adaptive RED, AVQ and PI work similarly, with noisier and larger queues than the new protocol and lower utilization of less than 80%. We omit the plots here.

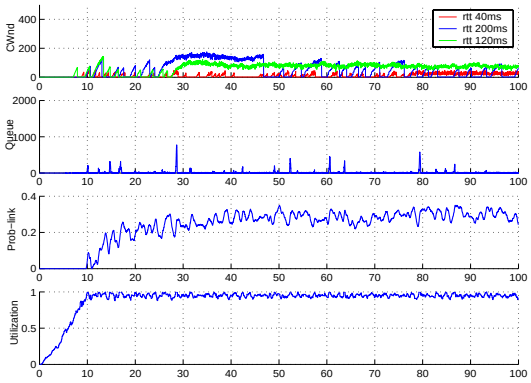

Fig. 12. new protocol with heavy-tailed traffic

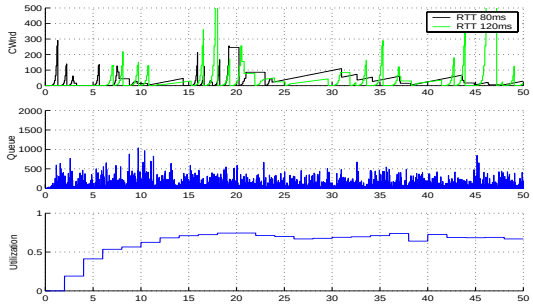

Fig. 13. NewReno/RED with heavy-tailed traffic

VI. Conclusion

An ideal TCP congestion avoidance method that can achieve high utilization, small queueing delay, freedom from oscillations and fairness in bandwidth allocation has been a major objective of networking research in recent years. Our results show first that at the level of fluid-flow quantities, these objectives are indeed achievable if one assumes a price signal that can be fed back to sources from links. We have further demonstrated a practical version of the protocol, based on ECN marking, that appears to successfully approximate these objectives in certain high capacity scenarios where current protocols exhibit limitations. More research remains to be done to verify the generality of these conclusions, as well as on finding universal parameter choices for the protocol.

The main obstacle to the implementation of this protocol is that it requires substantial changes to current practice at both the links and the sources. This brings the question of the ability of this protocol to be incrementally deployed with the current TCP. We have deliberately postponed this issue until a viable version of the protocol itself had been tested, but it is the next question in line for future research.

Acknowledgments

The authors would like to acknowledge Sanjeewa Athuraliya, Jiantao Wang and Polly Huang for the early work on simulation.

References

[1] 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.

[2] 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, ftp://ftp.ee.lbl.gov/papers/early.ps.gz

[3] Victor Firoiu and Marty Borden, A study of active queue management for congestion control, in *Proceedings of IEEE Infocom*, March 2000.

[4] Chris Hollot, Vishal Misra, Don Towsley, and Wei-Bo Gong, A control theoretic analysis of RED, in *Proceedings of IEEE Infocom*, April 2001, http://www-net.cs.umass.edu/papers/papers.html.

[5] S. H. Low, F. Paganini, J. Wang, S. A. Adlakha, and J. C. Doyle, Dynamics of TCP/RED and a scalable control, to appear in Proceedings of IEEE Infocom, July 2002.

[6] Steven H. Low, Fernando Paganini, John C. Doyle, Internet congestion control: an analytical perspective, *IEEE Control Systems Magazine*, February 2002.

[7] Fernando Paganini, John C. Doyle,and Steven H. Low, Scalable laws for stable network congestion control, in *Proceedings of 2001 Conference on Decision & Control*, Decemeber 2001.

[8] Glenn Vinnicombe, On the stability of end-to-end congestion control for the Internet, Tech. Rep., Cambridge University, CUED/F-INFENG/TR.398, December 2000.

[9] Glenn Vinnicombe Robust congeston control for the Internet, submitted for publication, Feberary 2002, http://www-control.eng.cam.ac.uk/gv/internet/index.html.

[10] Glenn Vinnicombe, On the stability of networks operating TCP-like congestion control, to appear on IFAC'02. http://www-control.eng.cam.ac.uk/gv/internet/index.html.

[11] Sanjeewa Athuraliya, Victor H. Li, Steven H. Low, and Qinghe Yin, REM: active queue management, , *IEEE Network*, vol. 15, no. 3, pp.48-53, May/June 2001.

[12] Frank P. Kelly, Aman Maulloo, and David Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, *Journal of Operations Resarch Soceity*, vol. 49, no.3, pp 237-252, March 1998.

[13] F.P. Kelly, Models for a self-managed Internet, *Philosophical Transactions of the Royal Soceity*, A358(2000) 2335-2348. http://www.statslab.cam.ac.uk/ frank/smi.html.

[14] R.J. Gibbens and F.P.Kelly, Resource pricing and the evolution of congestion control, *Automatica* 35 (1999), 1969-1985. http://www.statslab.cam.ac.uk/ frank/evol.html.

[15] Steven H. Low and David E. Lapsley, Optimization flow control, I: basic algorithm and convergence, *IEEE/ACM Transactions on Networking*, vol.7, no.6,pp861-874, December 1999. http://netlab.caltech.edu.

[16] Steven 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.

[17] Srisankar Kunniyur and R. Srikant, A time-scale decomposition approach to adaptive ECN marking, in *Proceedings of IEEE infocom*, April 2001, http://comm.csl.uiuc.edu:80/srikant/pub.html.

[18] M. Allman, A web server's view of the transport layer, *ACM Computer Communication Review*, vol.30, no. 5, October 2000.

[19] R. Johari and D. Tan, End-to-end congestion control for the Internet: delays and stability, *IEEE/ACM Transactions on Networking* 9(2001) 818-832.