Exponential-RED: A Stabilizing AQM Scheme for Low- and High-Speed TCP Protocols

Shao Liu, *Student Member, IEEE*, Tamer Başar, *Fellow, IEEE*, and R. Srikant, *Senior Member, IEEE*

***Abstract*—This paper introduces and analyzes a decentralized network congestion control algorithm which has dynamic adaptations at both user ends and link ends, a so-called general primal-dual algorithm. We obtain sufficient conditions for local stability of this algorithm in a general topology network with heterogeneous round-trip delays. Then, as an implementation of this algorithm in the Internet, we introduce an AQM (Active Queue Management) scheme called Exponential-RED (E-RED), which outperforms RED and is inherently stable when combined with TCP-Reno or its variants for high-speed networks.**

***Index Terms*—Congestion control, dual algorithm, primal algorithm, primal-dual algorithm, RED, TCP.**

I. Introduction

RECENTLY, there has been a flurry of research activity on decentralized end-to-end network congestion control algorithms. A widely-used framework, introduced in [15], is to associate a utility function with each flow and maximize the aggregate system utility function subject to link capacity constraints—an optimization problem known as *Kelly's System Problem* [15]. Congestion control schemes can be viewed as decentralized source and router algorithms to drive the system operating point to the optimum or some suboptimum solution of this maximization problem.

Congestion control schemes can be divided into three classes: primal algorithms, dual algorithms and primal-dual algorithms. In primal algorithms, the users adapt the source rates dynamically based on the route prices, and the links select a static law to determine the link prices directly from the arrival rates at the links [15]. In dual algorithms, on the other hand, the links adapt the link prices dynamically based on the link rates, and the users select a static law to determine the source rates directly from the route prices and the source parameters [15], [25], [36]. Primal-dual algorithms combine these two schemes and dynamically compute both user rates and link prices [1], [35]. For a comprehensive survey of these algorithms, see [31].

A modified primal algorithm, called the Active Virtual Queue (AVQ) algorithm, was introduced in [20]. Here the link prices in the original primal algorithm [15] are slowly adjusted so that asymptotically in time, the link prices become equal to the Lagrange multipliers in Kelly's system problem [20]. More importantly, in the presence of feedback delays, the parameters of this algorithm can be chosen such that the network is locally stable [21], [22]. The main benefit of this algorithm is that it achieves arbitrary fairness among the users and leads to full link utilization. This idea was adopted in [30] to modify the dual algorithm to allow slow adaptation at the sources and achieve the same benefits as the AVQ algorithm.

Both the modified primal algorithm and the modified dual algorithm have dynamic adaptations at both sources and routers, and thus can be regarded primal-dual. However, all the algorithms in the primal family relate the network congestion measure directly with the link aggregate rate, which corresponds to averaging the feedback from the network at the sources; and all the algorithms in the dual family relate the source rate directly with the route congestion measure, which corresponds to averaging the source rates at the links before the feedback of more explicit congestion information to the sources [14].

In this paper, we first briefly review the above developments and identify their role in allocating resources fairly in a network of competing users. We then generalize the class of **primal-dual** algorithms, and provide design guidelines to stabilize these algorithms in general topology networks with heterogeneous feedback delays. In this class of algorithms, the source dynamics are similar to those in the primal algorithm in [15] and [33] while the link dynamics are similar to those in the dual algorithm [25]. Following the approach introduced in [33], we obtain a local stability result which shows that stability does not depend on the source adaptation speed, but depends only on the link adaptation speed. This result subsumes the dual algorithm as its limiting case when the source adaptation speed approaches infinity.

From the stability analysis of the general primal-dual algorithm, we also show that Random Early Detection (RED) could stabilize TCP-Reno if modified slightly. Our modification to RED sets the packet marking probability to be an exponential function of the length of a virtual queue whose capacity is slightly smaller than the link capacity. Due to the exponential marking profile, we call it Exponential-RED (E-RED). From our analysis, it can be shown that E-RED stabilizes TCP-Reno and all its packet loss/mark based variations. Compared with other queue-length-based AQM schemes, like RED [7], REM [2], PI [12] and BLUE [6], E-RED is the first such scheme that can be proved to stabilize TCP-Reno for a general topology network with heterogeneous delays. In parallel with our work, another recent paper [10] has also obtained scaling rules for RED and PI to stabilize TCP-Reno.

We finally perform some *ns*-2 and Matlab simulations to compare E-RED with RED and to discuss the dependence of

Manuscript received January 20, 2004; revised October 25, 2004; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Low. This work was supported by the National Science Foundation under Information Technology Research (ITR) Grant CCR 00-85917, the Defense Advanced Research Projects Agency under Grant F30602-00-2-0542, and the Air Force Office of Scientific Research under URI F49620-01-1-0365. An earlier version of this paper appeared in the Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, HI, December 2003.

The authors are with the Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801-2307 USA (e-mail: shaoliu@ifp.uiuc.edu; tbasar@control.csl.uiuc.edu; rsrikant@uiuc.edu).

Digital Object Identifier 10.1109/TNET.2005.857110

E-RED's performance on the network scenario and the E-RED parameter choices. The simulation results show that E-RED outperforms RED when combined with TCP-Reno in the sense that it achieves less queue length oscillation, higher bandwidth utilization, and lower queueing delay at the same time. The simulation results also show that E-RED works well with HighSpeed TCP [8] and Scalable TCP [18].

II. Overview of Prior Work

Consider a set of users/routes, R, and a set of links/resources, L. For each user $r \in R$, its route consists of a set of links, which is a subset of L, denoted L_r. Each link $l \in L$ may be used by several routes; accordingly, we write $l \in r$ if $l \in L_r$. Each user $r \in R$ has an associated flow rate x_r and a utility function $U_r(x_r)$. Furthermore, each link $l \in L$ has an associated *link aggregate rate* $y_l = \sum_{r \in R: l \in r} x_r$, and a fixed capacity c_l. Introduce the source rates (column) vector $x := (x_r, r \in R)$, the link rate (column) vector $y := (y_l, l \in L)$, and the routing matrix $A := (A_{lr}, l \in L, r \in R)$, where $A_{lr} = 1$ if $l \in r$ and 0 otherwise; then we have $y = Ax$.

We impose the standard conditions on each user's utility function $U_r(x_r)$, namely that it be increasing, strictly concave, and continuously differentiable in x_r for $x_r \geq 0$. Further assume that $U_r'(x_r) \to -\infty$ as $x_r \to 0$, and that the utilities are additive so that the aggregate utility for the system is $U(x) = \sum_{r \in R} U_r(x_r)$. The *system problem*, first introduced in this context by Kelly [16], is the concave programming problem:

$$\max_x U(x) \text{ subject to } y \leq c \quad \text{over} \quad x \geq 0 \tag{1}$$

where $c = (c_l, l \in L)$, a column vector. The Lagrangian is:

$$L(x,p) = \sum_{r \in R} U_r(x_r) + \sum_{l \in L} p_l(c_l - y_l) \tag{2}$$

where p_l is the Lagrange multiplier associated with link $l, l \in L$. This multiplier is the shadow price of the link, and in many algorithms it also summarizes the link congestion information. Associate with each user r an aggregate route price q_r defined as $q_r := \sum_{l \in r} p_l$. Introduce the link price (column) vector $p = (p_l, l \in L)$ and the route price (column) vector $q = (q_r, r \in R)$. Then we have $q = A^T p$, and noting that $p^T y = p^T Ax = x^T A^T p = x^T q$, the Lagrangian can be rewritten as:

$$L(x,p) = \sum_{r \in R} (U_r(x_r) - x_r q_r) + \sum_{l \in L} p_l c_l. \tag{3}$$

From the Kuhn-Tucker theorem [3] of nonlinear programming, $\hat{x}$ solves (1) if and only if there exists a $\hat{p}$ such that the pair $(\hat{x}, \hat{p})$ constitutes a saddle point for the function $L(x,p)$, where x is the maximizer and p is the minimizer, with $x \geq 0$ and $p \geq 0$. Making the natural assumption that the constraint set for x is not empty and is bounded (and thus compact), the strict concavity of the aggregate utility function leads to the existence of a unique optimal solution $\hat{x}$ to (1). This then guarantees the existence of $\hat{p}$, and thus the existence of $\hat{q}$. The uniqueness of $\hat{x}$ determines the uniqueness of $\hat{q}$. If A has full row rank, then p is uniquely determined from q, and thus $\hat{p}$ is also unique. A pair $(\hat{x}, \hat{p})$ is in saddle-point equilibrium if and only if [16]

$$U_r'(\hat{x}_r) = \hat{q}_r \quad \forall\, r \in R \tag{4}$$

$$\hat{p}_l(c_l - \hat{y}_l) = 0 \text{ and } \hat{p}_l \geq 0, \text{and } c_l \geq \hat{y}_l \quad \forall\, l \in L. \tag{5}$$

A. Primal Algorithm

An issue of importance, driven by both theoretical and practical considerations, is the distributed computation of $(\hat{x}, \hat{p})$ using only the realistic decentralized information available to individual users. The congestion control problem deals precisely with the tasks of finding such algorithms that are decentralized, and selecting among them the one with the *best performance*. The users adapt x_r with respect to q_r, and the links adapt p_l with respect to y_l. The TCP algorithms are user adaptation laws, and AQM algorithms are link adaptation laws.

For the logarithmic utility function, $U_r(x_r) = w_r \log x_r$, Kelly *et al.* introduced the Primal Algorithm [15], where each user r implements the following algorithm:

$$\dot{x}_r = k_r(w_r - x_r q_r) \tag{6}$$

where k_r is a scaling factor or the step size. Each Link l, on the other hand, computes its price as

$$p_l = f_l(y_l), \quad \text{where } f_l > 0, f_l' > 0. \tag{7}$$

Without incorporating delay into the algorithm, it was shown in [15] that this primal algorithm is globally stable. If we include propagation delay and neglect queueing delay, thus treating delay as a constant, then we have the relationships:

$$y_l(t) = \sum_{r: l \in r} x_r \left(t - \tau_{lr}^f\right) \tag{8}$$

$$q_r(t) = \sum_{l \in r} p_l \left(t - \tau_{lr}^b\right) \tag{9}$$

where τ_{lr}^f is the forward delay from source r to link l, and τ_{lr}^b is the backward delay from link l back to source r, and we have round trip delay $T_r := \tau_{lr}^f + \tau_{lr}^b$, for any $l \in r$.

Denote the Laplace transforms of $x(t)$, $y(t)$, $p(t)$, and $q(t)$, respectively, by $X(s)$, $Y(s)$, $P(s)$, and $Q(s)$, and define the routing matrix with delay in the frequency domain as $A(s) = (A_{lr}(s), l \in L, r \in R)$, where

$$\begin{aligned} A_{lr}(s) &= \exp\left(-s\tau_{lr}^f\right), \quad \text{if } l \in r \\ &= 0, \qquad\qquad\qquad\quad \text{otherwise.} \end{aligned}$$

Then we have the relationships [33]

$$Y(s) = A(s)X(s) \tag{10}$$

$$Q(s) = \text{diag}(e^{-sT_r}) A^T(-s) P(s) \tag{11}$$

and the primal algorithm with delay is given as follows [13]:

$$\text{User end}: \quad \dot{x_r}(t) = k_r\left(w_r - x_r(t - T_r) q_r(t)\right) \tag{12}$$

$$\text{Link end}: \quad p_l(t) = f_l\left(y_l(t)\right). \tag{13}$$

Conditions for local stability were first conjectured in [13], and then proved and extended in [32]–[34]. For a certain class of functions $f_l(\cdot)$, global stability has been established in [37].

Note that the link price computation in the primal algorithm above is static. An algorithm called AVQ was introduced in [20], where users choose the same dynamic adaptation, but at the links, $p_l(t)$ is a function of both $y_l(t)$ and a virtual queue capacity $\widetilde{c}_l$ which is a time-varying link parameter. The global stability of this algorithm has been established in [23], [35] without delay, and local stability has been established in [21], [22] in the presence of feedback delays.

B. Dual Algorithm

The dual algorithm introduced in [25] employs the following algorithms at the link and user ends:

$$\text{Links}: \quad \dot{p}_l = \gamma_l(y_l - c_l) \tag{14}$$

where $\gamma_l > 0$ is an adaptation parameter. In [30], γ_l was chosen to be $1/c_l$.

$$\text{Users}: \quad x_r = x_r(q_r) = U_r'^{-1}(q_r). \tag{15}$$

The above dual algorithm is a special case of the dual algorithm in [14], [15]. It was also derived in [36] for the special case of logarithmic utility functions.

For this dual algorithm, global stability was proved in [25], while local stability was proved for a particular choice of utility functions in [29]. The restriction on the choice of utility functions means that arbitrary relative fairness among the users cannot be achieved using the algorithm in [29]. Following the time-scale decomposition idea in [21]–[23], to track an arbitrary utility function, the dual algorithm was modified in [30] by introducing slow time-scale dynamics at the user end. In other words, $x_r = x_r(\xi_r, q_r)$, where ξ_r is a slowly adjusted source parameter, just like the virtual queue capacity $\widetilde{c}_l$ in the AVQ algorithms. This ensures that the dual algorithm can also allocate the network resources fairly.

C. Primal-Dual Algorithm

Both the primal algorithms and the dual algorithms have dynamics at one end (user end for the primal algorithm and link end for the dual algorithm) and static adaptation at the other. However, in the current Internet, TCP is a dynamic source algorithm, and most AQM algorithms, like RED, Vegas, etc, relate the price to the queue length or queueing delay, and thus are dynamic link algorithms. This motivates the work on primal-dual algorithms, to directly relate $\dot{x}_r$ to q_r and $\dot{p}_l$ to y_l.

As already mentioned, $\hat{x}$ solves the system problem if and only if there exists a $\hat{p}$ such that $(\hat{x}, \hat{p})$ is the saddle point of $L(x,p)$. So we can approach this optimization problem from a game theoretical point of view: $L(x,p)$ is the gain for player 1 ($P1$) who controls x, and loss for player 2 ($P2$) who controls p. So $P1$ is the maximizer and $P2$ is the minimizer, and we have a 2-player zero-sum game, with $(\hat{x}, \hat{p})$ the unique saddle point:

$$L(x,\hat{p}) \le L(\hat{x},\hat{p}) \le L(\hat{x},p) \quad \forall\, x \ge 0,\ p \ge 0.$$

To reach this unique saddle point equilibrium, we can choose iterative algorithms with dynamic adaptations from both players. The simplest such algorithm is the gradient algorithm:

$$\dot{x}_r = k_r \frac{\partial L(x,p)}{\partial x_r} = k_r\,(U_r'(x_r) - q_r) \tag{16}$$

$$\dot{p}_l = -\gamma_l \frac{\partial L(x,p)}{\partial p_l} = \gamma_l(y_l - c_l). \tag{17}$$

Such algorithms were considered in [1] and [35], which can be regarded as primal-dual algorithms in the sense that $\dot{x}_r$ depends on q_r and $\dot{p}_l$ depends on y_l. At equilibrium, we have

$$x_r = U_r'^{-1}(q_r) \tag{18}$$

and

$$y_l \begin{cases} = c_l, & \text{if } p_l > 0 \\ \le c_l, & \text{if } p_l = 0 \end{cases} \tag{19}$$

for all l,r. So the equilibrium point solves the system problem (1) exactly. In [35], a passivity approach was used to show global stability of this equilibrium without considering delay. In the next section, we will embed this algorithm within a more general class of primal-dual algorithms, and for this general class we will establish local stability in the presence of delay.

D. Remarks on the Link Model

As we have already stated, most of the AQM algorithms choose the marking probability as a function of the queue length. So it is intuitive to choose p_l to be a function of the queue length b_l, and not as a function of the aggregate link arrival rate y_l. This is the model chosen by the dual and primal-dual approaches. However, in the primal approach, p_l is chosen to be a function of y_l. Does this then mean that one has to measure the link arrival rate, and not the queue length, to compute the marking probability? In general, the answer seems to be *no* when there are stochastic disturbances in the network. In [17], Kelly argues that if the queue length hits zero several times within a round-trip time, even if the marking is queue-length-based, the source acts as though the marking probability is a function of the stochastically averaged version of the queue length. Thus, in the deterministic model of congestion control, the source only sees the price p_l as a static function of y_l. However, the conditions under which the queue length hits zero frequently have not been provided in [17]. Recently, in [4], it has been shown that the parameters of popular AQM schemes determine whether the queue length hits zero frequently or not. Specifically, if both the queue length and the capacity are scaled in proportion to the number of users in the network, and the marking probability is based on the queue length divided by the number of users in the network, then we have a queue-based model; else, the queue length hits zero frequently and we have the rate-based model. Thus, the primal-dual algorithm considered here requires that the AQM parameters are chosen such that the queue length does not hit zero frequently. Since we also would like to have negligible queueing delay, this suggests that the AQM algorithm based on the primal-dual algorithm analysis must adjust its marking probability according to the occupancy of a virtual queue, whose capacity is slightly smaller than that of the real link [9]. Thus, we can maintain a large virtual queue, which always stays nonzero, as well as a small real queue, yielding negligible queueing delay. We will discuss this further in later sections.

III. A General Primal-Dual Algorithm and Its Local Analysis

The primal-dual algorithm (16), (17) is too specific and still far from being interpreted as a TCP-AQM combination. We introduce here a general primal-dual algorithm, whose source law includes current versions of TCP as special cases and whose link law includes most current AQM algorithms as special cases. Define $[x]^+ := \max(0,x)$ and

$$[h(x)]_x^+ := \begin{cases} h(x), & \text{if } x > 0 \\ [h(x)]^+, & \text{if } x = 0. \end{cases}$$

The general primal-dual algorithm is given by

$$\dot{x}_r = f_r(x_r)\,[U_r'(x_r) - q_r]_{x_r}^+ \tag{20}$$

$$\dot{p}_l = g_l(p_l)[y_l - c_l]_{p_l}^+ \tag{21}$$

where $f_r(x_r)$ and $g_l(p_l)$ are continuous functions, lower bounded by constants $\underline{f_r} > 0$ and $\underline{g_l} > 0$, respectively.

A. Global Stability Without Delay

It immediately follows that the equilibrium point of the general primal-dual algorithm (20), (21) solves the system problem exactly. Using an approach similar to the one in [35], one can show that the general primal-dual algorithm without delay is globally asymptotically stable (GAS). Define[1]

$$H_r(x_r) := \int_{\hat{x}_r}^{x_r} \frac{u - \hat{x}_r}{f_r(u)} du$$

$$J_l(p_l) := \int_{\hat{p}_l}^{p_L} \frac{v - \hat{p}_l}{g_l(v)} dv$$

and introduce

$$V(x,p) = \sum_{r \in R} H_r(x_r) + \sum_{l \in L} J_l(p_l).$$

Using these, we can establish the following theorem on global stability of the system; see [31] for a proof.

Theorem III.1: $V(x,p)$ is a Lyapunov function for the system described by (20), (21), and thus the system is globally asymptotically stable.

B. Local Stability Analysis With Delay

With feedback delay, we can modify (20)–(21) to

$$\dot{x}_r(t) = f_r\left(x_r(t), x_r(t-T_r)\right)\left[U_r'\left(x_r(t)\right) - q_r(t)\right]_{x_r}^{+} \quad (22)$$

$$\dot{p}_l(t) = g_l\left(p_l(t)\right)\left[y_l(t) - c_l\right]_{p_l}^{+} \quad (23)$$

where, by slightly abusing notation, we allow f_r to be a continuous function of both $x_r(t)$ and $x_r(t-T_r)$, which is again lower bounded by the constant $\underline{f_r} > 0$.

Before we proceed with the local analysis by linearizing the algorithm around the equilibrium values, we first note the following on the boundary conditions in (22) and (23) and the constant delay assumption in the system modeling. For the users, $\hat{x}_r > 0, \forall r \in R$, since $U_r(x_r) \to \infty$ as $x \to 0$. For the links, we restrict the link set L to the active link set $L_a := \{l \in L : \hat{p}_l > 0\}$, i.e., we consider only the bottleneck links. Since the set $L \backslash L_a$ does not affect the local analysis, for the rest of this paper we will assume that $L = L_a$. Then, the boundary conditions in (22) and (23) can be ignored for local analysis. Also, we assume that the queueing delay can be neglected when compared with propagation delay, and thus the round trip time can be taken to be a constant. This is a reasonable assumption if one assumes that each router uses a virtual queue to compute link prices [9].

Let us now linearize (22). At the equilibrium point, we have $\hat{q}_r = U_r'(\hat{x}_r)$. Linearization around it gives

$$\dot{x}_r(t) = \hat{f}_r\left(U_r''(\hat{x}_r)x_r(t) - q_r(t)\right)$$

where $\hat{f}_r = f_r(\hat{x}_r, \hat{x}_r)$. Note that in the above equation and in the rest of the paper, by slightly abusing notation, we use x_r, q_r, y_l, and p_l also to represent the perturbations from their equilibrium values $\hat{x}_r$, $\hat{q}_r$, $\hat{y}_l$, and $\hat{p}_l$, respectively.

In the Laplace domain, we have

$$sX_r(s) = -\hat{f}_r Q_r(s) + \hat{f}_r U_r''(\hat{x}_r) X_r(s).$$

Solving for $X_r(s)$ yields

$$\begin{aligned} X_r(s) &= -\hat{f}_r \frac{1}{s - \hat{f}_r U_r''(\hat{x}_r)} \\ Q_r(s) &= -\frac{1}{u_r''(\hat{x}_r)} \frac{\theta_r}{sT_r + \theta_r} Q_r(s) \end{aligned} \quad (24)$$

where

$$\theta_r = \hat{f}_r T_r \left(-U_r''(\hat{x}_r)\right). \quad (25)$$

Note that $\theta_r > 0$, since $U_r(x_r)$ is strictly concave and thus $U_r''(x_r) < 0$.

For the link dynamics, linearization yields

$$\dot{p}_l = g_l(\hat{p}_l) y_l.$$

In the Laplace domain, we have

$$P_l(s) = \frac{g_l(\hat{p}_l)}{s} Y_l(s). \quad (26)$$

The return ratio of the closed-loop feedback system (10), (11), (24) and (26) is given by

$$L(s) = \text{diag}\left(\frac{1}{-U_r''(\hat{x}_r)} \frac{\theta_r e^{-sT_r}}{sT_r + \theta_r}\right) A^T(-s) \text{diag}\left(\frac{g_l(\hat{p}_l)}{s}\right) A(s). \quad (27)$$

Notice that we can rewrite $L(s)$ as:

$$L(s) = \text{diag}\left(\frac{T_r}{-U_r''(\hat{x}_r)} \frac{\theta_r e^{-sT_r}}{sT_r(sT_r + \theta_r)}\right) \times A^T(-s) \text{diag}\left(g_l(\hat{p}_l)\right) A(s). \quad (28)$$

This now leads to the following theorem on stability, whose proof can be found in the Appendix.

Theorem III.2: The closed-loop system described by (22), (23), (8), and (9) is locally asymptotically stable around the equilibrium point if the following two conditions hold:

$$-U_r''(\hat{x}) \geq \frac{\hat{q}_r}{a_r \hat{x}_r} \quad (29)$$

$$\frac{g_l(\hat{p}_l)}{\hat{p}_l} \leq \frac{1}{2c_l} \frac{1}{a_l T_m} \quad (30)$$

where a_r is a positive constant for each r, and $a_l = \max_{r \in l} a_r$, $T_m = \max_{r \in R} T_r$.

Remark 1: Condition (29) is satisfied for the general class of utility functions introduced in [28]. For example, if $U_r(x_r)$ is of the form $w_r \log x_r$ (logarithmic utility function), we have $a_r = 1$. If $U_r(x_r)$ is of the form $-w_r/x_r^{n_r}$ (power utility function of order n_r), we have $a_r = 1/(1+n_r)$.

Remark 2: From (28) and the proof of Theorem III.2 in the Appendix, we see that the sufficient condition for local stability (l.s.) depends only on the link price adaptation speed $g_l(p_l)$ and the user utility function $U_r(x_r)$, and not on the source rate adaptation speed $f_r(x_r(t), x_r(t-T_r))$. If we adapt the source rate infinitely fast (i.e., choosing $f_r(\cdot) \equiv \kappa_r$ and letting $\kappa_r \to \infty$), we arrive at the dual algorithm. The corresponding sufficient condition for stability of the dual algorithm also depends only on

[1]The definitions of $H_r(x_r)$ and $J_l(p_l)$ were suggested by Murat Arcak in his correspondence with John Wen and the third author of this paper in a different context.

the link adaptation speed and the user utility function. If $g_l(p_l)$ is fixed to be $1/c_l$ as in the dual algorithm [25], then the utility function must be specified as in [29]. To allow arbitrary utility functions, we could either add a slow adaptation at the source [30] or choose a different $g_l(p_l)$. Thus, the interesting observation from our analysis is that slow adaptation at the source (as in [30]) is not necessary for the dual algorithm to be stable, and achieve full utilization and arbitrary fairness. The dual algorithm can also be stabilized with slow adaptation at the links since the l.s. result does not depend on $f_r(x_r)$.

IV. Implementation in the Internet and E-RED AQM Scheme

In the current Internet, the source law is TCP and the link law is an AQM scheme, such as DropTail or RED. We wish to make as little changes to the current protocols as possible, but still achieve guaranteed stability and better performance. In this section we utilize the local stability analysis for the primal-dual algorithm to modify TCP+RED such that the combination is locally stable.

A. General Form of TCP

Consider the general form of TCP introduced in [33], which contains all possible TCP variants. Let W_r represent the window size of source r. Suppose the window size increases by $m_r W_r^{i_r}$ for each unmarked acknowledgment and decreases by $n_r W_r^{d_r}$ for each marked acknowledgment, where m_r, n_r, i_r and d_r are constants, with m_r and n_r being positive and $i_r < d_r$ being both integers. We note that the choice of $i_r = -1$ and $d_r = 1$ corresponds to TCP-Reno ($m_r = 1$, $n_r = 2/3$ for the standard version), and the choice of $i_r = 0$ and $d_r = 1$ corresponds to the Scalable TCP in [18].

This window based adaptation can be written in differential equation form as

$$\dot{W}_r = x_r(t - T_r)\left(m_r W_r^{i_r}(t) - n_r W_r^{d_r}(t) q_r(t)\right). \quad (31)$$

We note here the implicit assumption that q_r is sufficiently small so that we do not distinguish between 1 and $1 - q_r(t)$. This is widely accepted in the modeling of TCP [19], [27].

The source rate is $x_r = W_r/T_r$, so we have

$$\begin{aligned}\dot{x}_r &= \frac{x_r(t-T_r)}{T_r}\left(m_r T_r^{i_r} x_r^{i_r}(t) - n_r T_r^{d_r} x_r^{d_r}(t) q_r(t)\right)\\ &= n_r T_r^{d_r - 1} x_r(t - T_r) x_r^{d_r}(t)\\ &\quad \times \left(\frac{m_r}{n_r} x_r^{i_r - d_r}(t) T_r^{i_r - d_r} - q_r(t)\right). \end{aligned} \quad (32)$$

Comparing this with the source law (22) of the primal-dual algorithm, and considering the fact that $f_r(x_r)$ is lower bounded by $\underline{f_r} > 0$ and that (32) represents the dynamics of TCP only when x_r is not close to zero, we have

$$f_r\left(x_r(t), x_r(t - T_r)\right) = \max\left(\underline{f_r}, n_r x_r(t) x_r^{d_r}(t - T_r) T_r^{d_r - 1}\right)$$

and

$$U_r'\left(x_r(t)\right) = \frac{m_r}{n_r} x_r^{i_r - d_r}(t) T_r^{i_r - d_r}.$$

Thus,

$$\hat{f}_r = n_r \hat{x}_r^{d_r + 1} T_r^{d_r - 1} \text{ and } U_r''\left(x_r(t)\right) = U_r'\left(x_r(t)\right) \frac{i_r - d_r}{x_r(t)}$$

So, we have

$$U_r''(\hat{x}_r) = -\frac{(d_r - i_r)\hat{q}_r}{\hat{x}_r} \text{ and } a_r = \frac{1}{d_r - i_r}.$$

In particular, TCP-Reno and HighSpeed TCP correspond to a power utility function of order 1 and $a_r = 1/2$, while Scalable-TCP corresponds to a logarithmic utility function and $a_r = 1$.

B. Exponential RED (E-RED)

From Theorem III.2, we know that to stabilize the window adaptation source law in the TCP family, the link dynamic adaptation should satisfy

$$\frac{g_l(\hat{p}_l)}{\hat{p}_l} \le \frac{1}{2 a_l T_m c_l}. \quad (33)$$

Consider the link dynamics

$$\dot{p}_l = \max\left(\underline{g_l}, \beta_l \frac{p_l}{c_l}\right)\left[y_l - c_l\right]_{p_l}^{+} \quad (34)$$

with $\beta_l < 1/(2 a_l T_m)$. It satisfies the stability condition (33) if $\hat{p}_l$ is not close to zero (more specifically, $\hat{p}_l > \underline{g_l} c_l/\beta_l$). Throughout this paper, we assume that the above condition on $\hat{p}_l$ is satisfied. Equation (34) can be implemented at the packet level by setting the packet marking probability (the link price) p_l as an exponential function of the virtual queue length b_l:

$$p_l = \begin{cases} 0, & \text{if } 0 \le b_l < th_{\min,l} \\ p_{\min,l} e^{\frac{\beta_l}{c_l}(b_l - th_{\min,l})}, & \text{if } th_{\min,l} < b_l < th_{\max,l} \\ 1, & \text{if } b_l \ge th_{\max,l} \end{cases} \quad (35)$$

where $th_{\min,l} < th_{\max,l}$ are the two queue length thresholds between which the exponential marking is selected and $p_{\min,l}$ is the marking threshold when $b_l = th_{\min,l}$. Let $p_{\max,l}$ denote the marking probability when $b_l = th_{\max,l}$; then, $th_{\min,l}$, $th_{\max,l}$, and $p_{\min,l}$ must be such that $p_{\max,l} < 1$. If $p_{\min,l}$ and $p_{\max,l}$ are selected properly such that $p_{\min,l} < \hat{p}_l < p_{\max,l}$, then between the two probability thresholds, the derivative of p_l in (35) is just (34) and the linearization satisfies (33). Then, this marking scheme locally stabilizes TCP given that the fluid model of the system dynamics is valid. We call the above marking scheme Exponential-RED or E-RED in short, since the marking profile is an exponential function of the queue length. We note that the form of (35) is similar to the fair dual algorithm in [14]. However, the source control laws are static in the fair dual algorithm, and hence the stability conditions in [14] are different than the ones in this paper.

In the TCP-AQM scheme, the prices p_l and q_r are expressed in terms of the marking probabilities, which is why we use p_l, q_r to represent both the prices and the marking probabilities. However, the price feedback from p_l to q_r is linear in (9). while the probabilistic feedback is actually not linear:

$$q_r = 1 - \prod (1 - p_l). \quad (36)$$

We notice, however, that if p_l and q_r are sufficiently small for all $l \in L, r \in R$, we can approximate (36) by (9). In this and the next subsection, we assume that the small probability condition holds; subsequently in Section IV-D, we consider the stability issue for the general case.

Given the linear feedback from p_l to q_r, from Theorem III.2, the condition $\beta_l < 1/(2a_l T_m)$ guarantees the local stability of E-RED combined with any window adaptation source law in the TCP family. Note that E-RED simply uses a marking profile different than RED, and employs a virtual queue.

Remark 3: The slow link adaptation makes the queue length very large. To maintain a small queue length, so that the queueing delay is negligible and thus the RTT can be modeled as a constant, we need to implement the E-RED algorithm in a virtual queue whose capacity is $\tilde{c}_l = \gamma_l c_l$, where $0 < \gamma_l < 1$ is the link utilization parameter. Then, b_l is the queue length in that virtual queue, and we can have a large b_l while maintaining a small real queue.

C. E-RED With the Average Queue Length

In RED, the marking probability is a linear function of the average queue length. In E-RED, we can also take the marking probability to be an exponential function of the average value of the virtual queue length. We call this variant E-RED-aq and the original E-RED without averaging E-RED-iq (iq stands for instantaneous queue length).

For this variant, we replace b_l with r_l in the marking (35), where r_l is the exponentially weighted moving average of b_l based on samples taken every δ_l seconds. Thus,

$$r_l\left((k+1)\delta_l\right) = r(k\delta_l)(1-\alpha_l) + \alpha_l b_l \tag{37}$$

where $0 < \alpha_l < 1$ is the weight. As in [27], we use the following continuous approximation to (37):

$$\dot{r}_l = -\kappa_l(r_l - b_l) \tag{38}$$

where $\kappa_l = -\log(1-\alpha_l)/\delta_l$. In the Laplace domain, we have:

$$r(s) = \frac{\kappa_l}{s(s+\kappa_l)} y_l(s).$$

The dynamics of E-RED-aq yields

$$\dot{p}_l = p_l \frac{\beta_l}{c_l}\dot{r}_l.$$

Since at equilibrium $\dot{r}_l = 0$, linearization yields

$$\dot{p}_l = \frac{\beta_l \hat{p}_l}{c_l}\dot{r}_l \tag{39}$$

and in the Laplace domain, we have

$$P_l(s) = \frac{\beta_l \hat{p}_l}{c_l}\frac{\kappa_l}{s(s+\kappa_l)} Y_l(s). \tag{40}$$

Combining the source algorithm (20) with logarithmic or power utility function and E-RED-aq, we arrive at the following loop function:

$$\begin{aligned} L(s) &= \text{diag}\left(\frac{T_r}{-U''(\hat{x}_r)}\frac{\theta_r e^{-sT_r}}{T_r(sT_r+\theta_r)}\right) A^T(-s) \\ &\quad \times \text{diag}\left(\frac{\beta_l \hat{p}_l}{c_l}\frac{\kappa_l}{s(s+\kappa_l)}\right) A(s) \\ &= \text{diag}\left(\frac{T_r}{-U''(\hat{x}_r)}\frac{\theta_r e^{-sT_r}}{T_r(sT_r+\theta_r)}\right) A^T(-s) \\ &\quad \times \text{diag}\left(\frac{\beta_l \hat{p}_l}{c_l}\frac{\kappa_l}{s+\kappa_l}\right) A(s) \end{aligned} \tag{41}$$

This now leads to the following theorem on the stability condition for the source algorithm (20) and E-RED-aq:

Theorem IV.1: The closed-loop system composed of the source algorithm (20) and E-RED-aq is locally asymptotically stable around the equilibrium if the two conditions in (29) and (30) are satisfied.

Proof: See the Appendix.

This theorem suggests that E-RED-aq requires the same link adaptation as E-RED-iq, so the averaging of the queue length as in RED is not detrimental to stability.

D. Probabilistic Feedback

In this subsection, we will show that even under exact probabilistic feedback, the local stability results still hold. We introduce two pseudo price variables v_r and z_l, defined as

$$v_r = -\log(1-q_r), \quad z_l = -\log(1-p_l).$$

Then (36) leads to the linearized relationship

$$v_r(t) = \sum_{l \in r} z_l\left(t - \tau_{lr}^b\right). \tag{42}$$

Let v and z be the column vectors of all v_r and z_l, respectively, and let $V(s)$ and $Z(s)$ be Laplace transforms of $v(t)$ and $z(t)$. Then, we have a relationship similar to (9):

$$V(s) = \text{diag}(e^{-sT_r})A^T(-s)Z(s). \tag{43}$$

Comparing (28) with (41) and noticing that for E-RED, $g_l(p_l) = \beta_l p_l/c_l$, we see that the loop function of either E-RED-iq or E-RED-aq combined with (20) can be written as

$$L(s) = \text{diag}\left(\frac{\hat{x}_r}{\hat{q}_r}\tilde{f}_r(s)e^{-sT_r}\right) A^T(-s)\text{diag}\left(\frac{\hat{p}_l}{c_l}\tilde{g}_l(s)\right) A(s) \tag{44}$$

where $\tilde{f}_r(s)$ and $\tilde{g}_l(s)$ are two functions of s. The stability results build on the fact that the spectral radius of the matrix $\text{diag}(\sqrt{\hat{p}_l/\hat{y}_l})R(s)\text{diag}(\sqrt{\hat{x}_r/\hat{q}_r})$ is bounded by 1, and thus we can select the parameters such that the eigenloci of L cross the real line to the right of -1. Then, the stability condition is obtained from the generalized Nyquist Criterion.

Now, for the probabilistic feedback, with the relationships (10) and (43), we have

$$\begin{aligned} X_r(s) &= \frac{\hat{x}_r}{\hat{q}_r}\tilde{f}_r(s)Q_r(s)\frac{\hat{x}_r}{\hat{q}_r}\tilde{f}_r(s)\frac{\partial q_r}{\partial v_r}V_r(s) \\ &= \frac{\hat{x}_r}{\hat{q}_r}\tilde{f}_r(s)(1-\hat{q}_r)V_r(s) \\ Z_l(s) &= \frac{\partial z_l}{\partial p_l}P_l(s) = \frac{1}{(1-\hat{p}_l)}P_l(s) \\ &= \frac{1}{(1-\hat{p}_l)}\frac{\hat{p}_l}{c_l}\tilde{g}_l(s)Y_l(s). \end{aligned}$$

So the new loop function is in the following form:

$$\begin{aligned} L(s) = \text{diag}&\left(\frac{\hat{x}_r(1-\hat{q}_r)}{\hat{q}_r}\tilde{f}_r(s)e^{-sT_r}\right) A^T(-s) \\ &\times\text{diag}\left(\frac{\hat{p}_l}{c_l(1-\hat{p}_l)}\tilde{g}_l(s)\right) A(s). \end{aligned} \tag{45}$$

Thus, for the probabilistic feedback law (42), the same stability conditions as those in the price feedback law (9) are obtained if we could prove that the spectral radius of the matrix $\text{diag}(\sqrt{\hat{p}_l/c_l(1-\hat{p}_l)})A(s)\text{diag}(\sqrt{\hat{x}_r(1-\hat{q}_r)/\hat{q}_r})$ is also

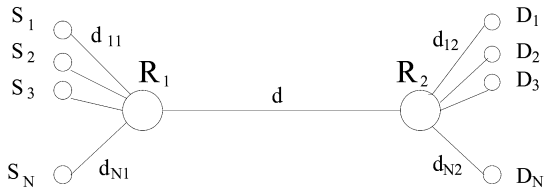

Fig. 1. The network topology: a single bottleneck link and multiple flows with heterogeneous delays.

bounded by 1. Since $\sum_{r:r\in l}\hat{x}_r \le c_l\ \forall l$, we just need to show that

$$\sum_{l\in r}\frac{p_l}{1-p_l}\le\frac{q_r}{1-q_r}. \tag{46}$$

Theorem IV.2: The inequality (46) holds, and thus the stability results for the linear price feedback case also hold under probabilistic feedback.

Proof: See the Appendix.

V. Simulations

We implemented the E-RED algorithm in the *ns*-2 package and performed simulations on a network whose sources use TCP-Reno, and whose routers choose E-RED-iq, E-RED-aq or RED. Let $\xi_l := \beta_l T_m/2$; then, $\xi_l \le 1/2$ is the local stability condition for E-RED-iq with TCP-Reno. We make all sources ECN-capable, and set the maximum window size to be 1000 instead of 64, so the maximum window size will not be a threshold for the flow rate in all scenarios. We also let all sources send packets of identical size, 1040 bytes, including the IP and TCP header. For E-RED, we choose the parameters ξ_l, $p_{\min,l}$, $p_{\max,l}$, $th_{\min,l}$, γ_l, and α_l ($\alpha_l = 1$ corresponds to E-RED-iq and $\alpha_l < 1$ corresponds to E-RED-aq), and compute $th_{\max,l}$ from $p_{\min,l}$, $p_{\max,l}$, $th_{\min,l}$.

For most simulations in this section, we consider a network with a single bottleneck link and multiple flows, as shown in Fig. 1. Since there is a large population of users in the Internet and the fluid model makes sense only when the number of users is large, we choose a sufficiently large number of users for most of the simulations. We vary the values of N, c, d, d_{11} to d_{N2} to obtain different network configurations. We set the buffer limit (of the real queue) always to be $\max(0.3c, 25)$ packets, where c is measured in Mb/s, which guarantees that the bottleneck queueing delay for sufficiently large capacity is bounded by 2.5 ms (actually, we will see that if E-RED stabilizes the system, then the real queue length is much smaller than its buffer limit and thus the queueing delay is much smaller than this bound). As for the parameter settings, for RED, we set $thresh$ to be 1/5 of the buffer limit and $maxthresh$ to be $3 * thresh$, keep all the other parameters at their default settings, and choose ECN marking instead of dropping; for E-RED, we set $th_{\min,l}$ to be 1/5 of the buffer limit (of the real queue). The other parameters for E-RED will be varied for different scenarios.

A. Stability and Performance Comparison of RED and E-RED

We first compare the stability and performances of E-RED and RED. From the analysis in [11] and [26], we know that RED becomes unstable when the capacity or the round trip delay becomes large. In the first simulation scenario, we choose a large link capacity network with $c = 1$ Gb/s, $N = 2000$, $d = 10$ ms, and the delays of all other links being uniformly distributed between 1 and 20 ms. So the maximum round trip time is $T_m = 100$ ms. For the E-RED parameters, we set $p_{\min,l} = 0.0005$, $p_{\max,l} = 0.1$, $\gamma = 0.95$, $\xi = 1$. Even though our sufficient stability condition suggests $\xi \le 1/2$, we have chosen $\xi = 1$ in order to understand how conservative our stability results are. For RED, we monitor the instantaneous queue length, and measure the throughput with a time granularity of 0.1 s; they are shown in Fig. 2. From the graphs, we see that RED is unstable in the sense that both the queue length and the throughput show significant oscillations, and the queue length hits zero and the full buffer size frequently. For E-RED, we monitor the real queue length and the virtual queue length, and measure the throughput with the same time granularity; they are shown in Fig. 3. From the graphs, we see that E-RED is stable in the sense that the virtual queue length oscillates around its equilibrium very slightly, the throughput is less oscillatory than that of RED, and the real queue size is always small after the transient phase. So E-RED stabilizes TCP-Reno in large capacity scenarios while RED does not. To compare the performances numerically, we measure the average queue length (Avg(qlen)), standard deviation of queue length (Std(qlen)), and average throughput (Avg(thr)) for both algorithms after the transient phase (over the interval between 10 s and 39 s for this scenario); these values are listed in Table I. In the second scenario, we choose a large round trip delay network with $c = 300$ Mb/s, $N = 1000$, $d = 10$ ms, and the delays of all other links being uniformly distributed between 45 and 95 ms. So the maximum round trip time is $T_m = 400$ ms. We set $p_{\min,l} = 0.000\,67$, $p_{\max,l} = 0.1$, $\gamma = 0.95$, $\xi = 1$. The queue length and throughput for RED are shown in Fig. 4, and we see that RED is unstable. The real queue length, virtual queue length and throughput for E-RED are shown in Fig. 5; we see that E-RED is stable and achieves a higher and less oscillatory throughput and lower and less oscillatory queueing delay than RED. For the numerical comparison, the average throughput,

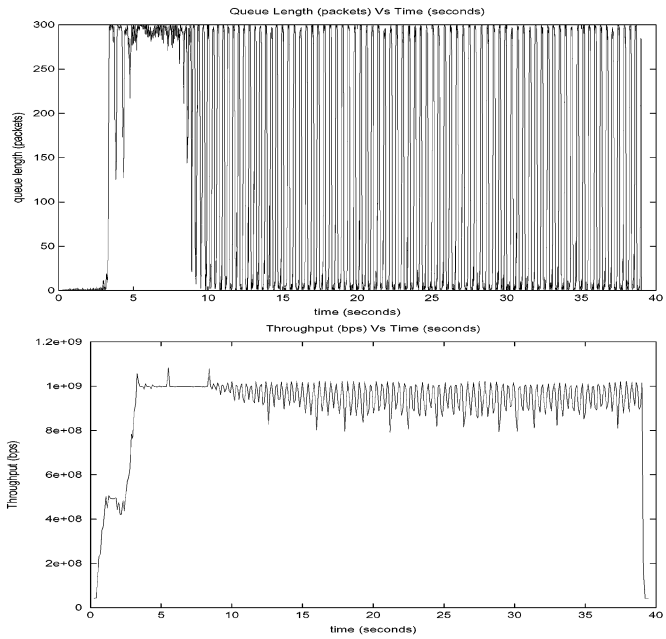

Fig. 2. RED in a large capacity network: instantaneous queue length (top graph), and throughput (bottom graph).

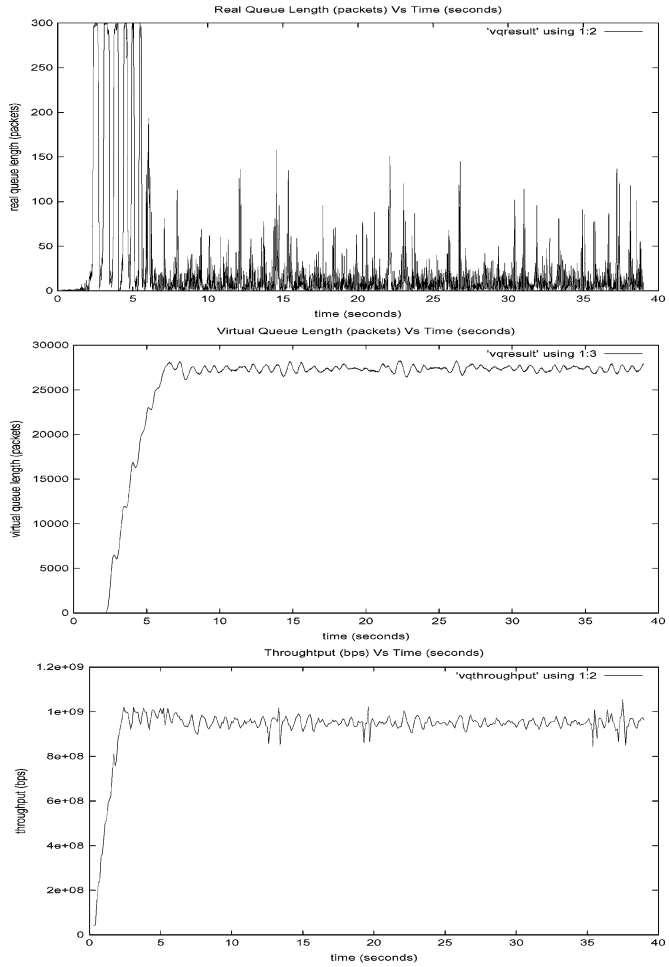

Fig. 3. E-RED in a large capacity network: real queue length (top graph), virtual queue length (middle graph), and throughput (bottom graph).

TABLE I
NUMERICAL COMPARISON OF RED AND E-RED IN LARGE CAPACITY (LC) AND LARGE DELAY (LD) NETWORK SCENARIOS

AVQ/Scenario	Avg(qlen) (pkts)	Std(qlen) (pkts)	Avg(thr) (Mbps)
RED/LC	117	131	939
E-RED/LC	9.74	11.8	$951(\approx \gamma c)$
RED/LD	27.7	34.3	278
E-RED/LD	11.8	16.2	$285(\approx \gamma c)$

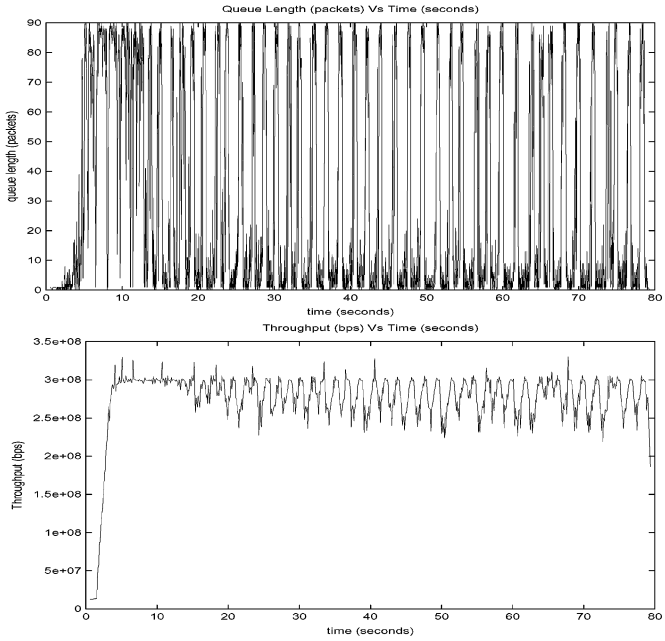

Fig. 4. RED in a large delay network: instantaneous queue length (top graph) and throughput (bottom graph).

average queue length, and standard deviation of queue length (measured after the transient phase) for both algorithms are listed in Table I.

So for both large capacity and large delay networks, given that the number of users is sufficiently large, E-RED stabilizes TCP-Reno while RED does not. Furthermore, E-RED achieves higher bandwidth utilization and smaller queueing delay.

One may argue that the advantage of E-RED over RED is mainly due to the use of a virtual queue. We have performed simulations with RED in a virtual queue (REDvq) and found that REDvq is unstable if the virtual queue limit and the REDvq parameters are taken to be the same as those of RED in the real queue. The reason for the instability is that the link adaptation speed is much larger than what Theorem III.2 suggests. From Fig. 3, we see that at equilibrium, $b_l \approx 27\,000$ packets. Since we have chosen $\xi = 1$, we get

$$\hat{p}_l = 0.0005 \exp\left(\frac{2(27000-60)\times 8320}{0.1\times 10^9}\right) = 0.0442$$

and the following stability condition on link adaptation speed:

$$g_l(\hat{p}_l) \le \frac{2\times 0.0442\times 8320}{0.1\times 10^9} = 7.36\times 10^{-6}$$

For RED, the default setting for $p_{\max}$ is 0.1 and this requires

$$maxthresh - thresh \ge \frac{0.1}{7.36\times 10^{-6}} = 13\,584$$

and hence a much larger virtual queue length. If we increase the virtual queue limit 120 fold so that the difference between the two thresholds is now 14 400 (which is larger than 13 584), then the simulations (not shown here due to space limitations) indicate that the system is stable under these new parameter settings. So both linear and exponential marking profiles can stabilize the system if the adaptation speed is chosen according to the condition in Theorem III.2. However, for linear profile to satisfy the stability condition, we need the knowledge of $\hat{p}_l$, while E-RED does not require that. This auto-configuration is one of the reasons for choosing the exponential marking profile.

Remark 4: From the above simulation results, we see that once the virtual queue becomes stable (oscillates around its equilibrium point slightly), the system will become stable, and the real queue size will be small and the throughput will oscillate around $\gamma_l c_l$ slightly. In the following simulations, to save space, we provide the graphs of only the virtual queues for the E-RED algorithm.

B. E-RED-iq and E-RED-aq

We then test the stability and performance of E-RED-aq. We still choose the topologies as in the previous subsection, and choose $\alpha_l = 0.001$ and $\xi = 0.9$. The virtual queue size of the two network scenarios are shown in Fig. 6. From the graphs, we see that E-RED-aq is also stable with TCP-Reno.

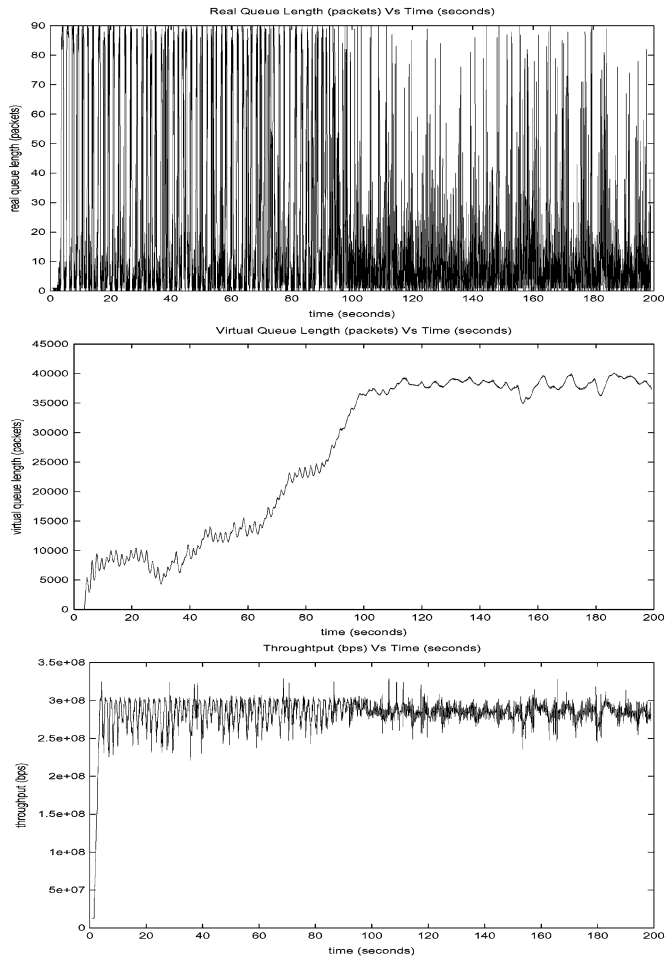

Fig. 5. E-RED in a large delay network: real queue length (top graph), virtual queue length (middle graph), and throughput (bottom graph).

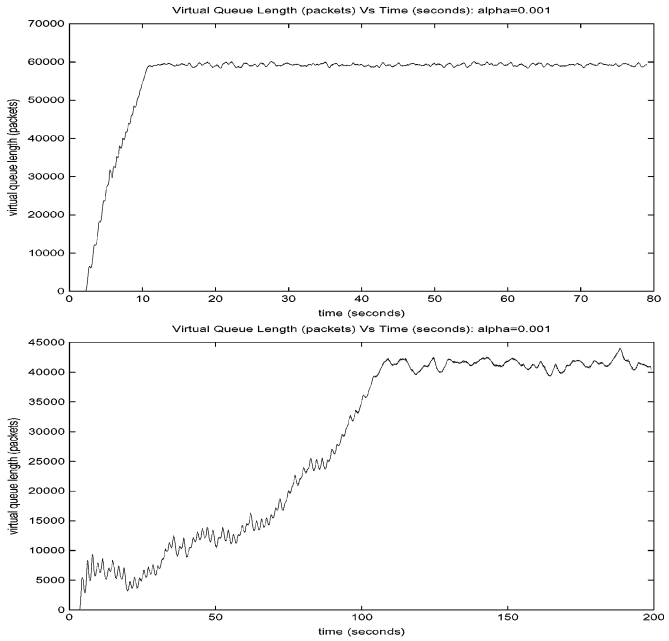

Fig. 6. The virtual queue of E-RED-aq in large capacity (top graph) and large delay networks (bottom graph): $\alpha_l = 0.001$.

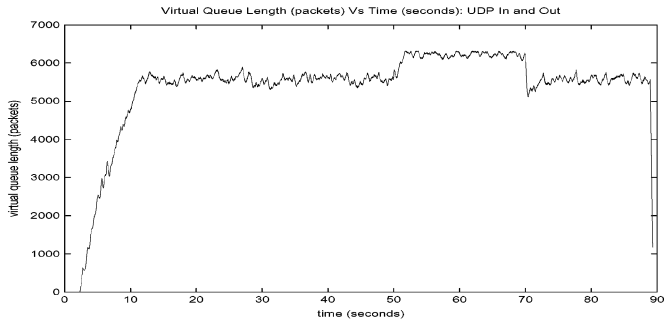

Fig. 7. UDP traffic in and out, virtual queue.

C. Sudden Traffic Changes

To study the transient response of the proposed algorithm, we consider a scenario where a large traffic source suddenly joins the network and leaves after a while. We choose $c = 100$ Mb/s and $N = 200$, and the round trip delays to be from 20 ms to 100 ms. One UDP flow, with fixed rate of $c/3$ joins at 50 and leaves at 70. The virtual queue is shown in Fig. 7. We see that the transient response is fast.

D. Dependence of Stability on the Parameters: ξ, γ and $p_{\min,l}$

We then study the influence of the parameters on stability and performance. We have shown that $\xi = 1$ also stabilizes the system while $\xi \leq 1/2$ is suggested in Theorem III.2. To further test how conservative the condition is, we have chosen ξ to be 2, 4 and 8. The network parameters are chosen as $c =$ 200 Mb/s, $N = 200$, and RTT between 20 and 100 ms. The E-RED parameters are chosen as $p_{\min,l} = 0.0005$, $p_{\max,l} = 0.1$, $\gamma = 0.9$. We have plotted the virtual queue lengths for the three ξ values in Fig. 8. We see that as ξ increases, the virtual queue shows higher oscillation, and when $\xi = 8$, the system is unstable. So this simulation shows that the slow link adaptation condition is not very conservative.

In simulations not shown here, we have also studied the influence of γ on the system performance. We studied two scenarios, one with $c = 150$ Mb/s and $N = 100$, and another one with the same c but $N = 250$. For the first scenario, we have seen that the system is unstable for $\gamma > 0.93$, and is stable for $\gamma \leq 0.93$. For the second scenario, we have seen that the system is unstable for $\gamma > 0.97$, and is stable for $\gamma \leq 0.97$. For both scenarios, once γ is small enough for the system to be stable, the oscillation level does not change much as γ decreases. Thus we know that for a fixed network scenario, there is a threshold value of γ, under which the system is stable. This threshold value can be larger if the number of sources is increased while keeping the per flow throughput constant. This is to be expected since what we have observed is essentially a statistical multiplexing effect.

In simulations not shown here, we have also seen that if $p_{\min,l}$ is smaller than the equilibrium value $\hat{p}$, the value of $p_{\min,l}$ only changes the initial setup time, and does not influence the stability and transient response. However, if $p_{\min,l} > \hat{p}$, we will never reach the operating point and the system cannot be stable.

E. TCP-Reno, Highspeed-TCP, and Scalable-TCP

We studied the combination of E-RED with two popular TCP variants for high-speed networks: HighSpeed-TCP [8] and Scalable-TCP [18]. We performed *ns*-2 simulation for HighSpeed-TCP, and performed Matlab simulation for Scalable-TCP since

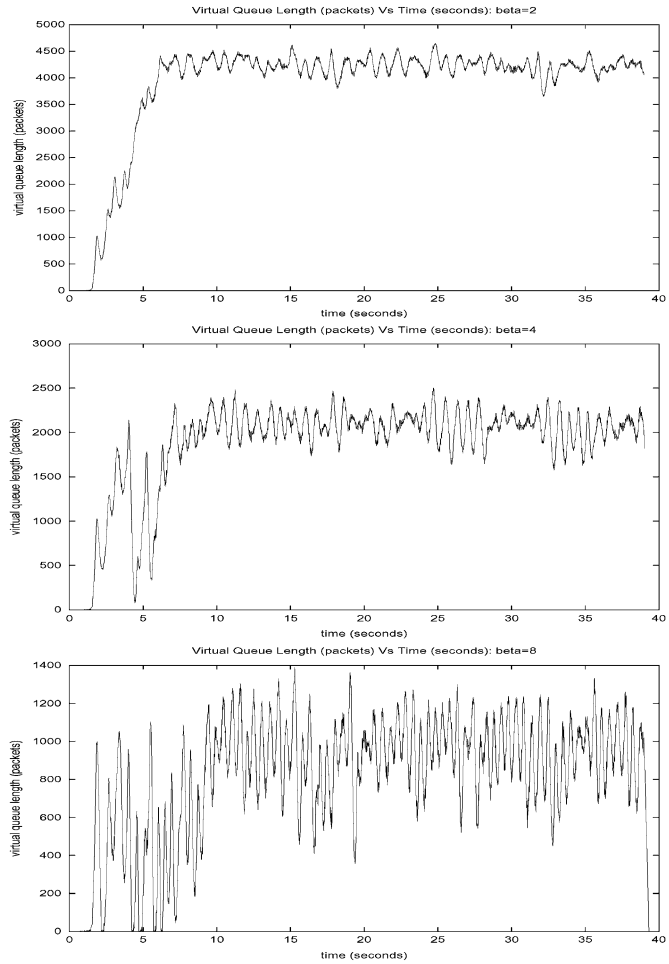

Fig. 8. The virtual queue length of E-RED, $\xi = 2, 4, 8$ (in top, middle, and bottom graphs, respectively).

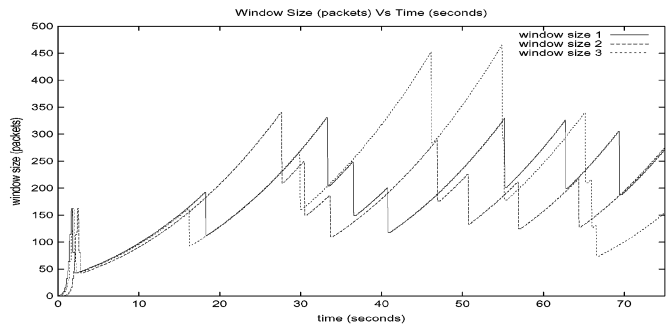

Fig. 9. The window sizes of the three HS-TCP flows with E-RED.

it is not included in the current *ns*-2 package. For both simulations, the scenario is the following: $N = 3$, the capacity per user is 10 Mb/s, and the round trip delay for each user is 200 ms. The window behavior of the three users under HighSpeed TCP is shown in Fig. 9. The average throughput per user measured over 75 seconds is $0.8c$. For a small number of sources, the use of fluid models is questionable since there are not enough sources for the law of large numbers to hold [31]. Despite this, the combination of E-RED with HighSpeed TCP gives a reasonably high throughput. The fact that the fluid model does not apply was noticed in the simulations from the behavior of the

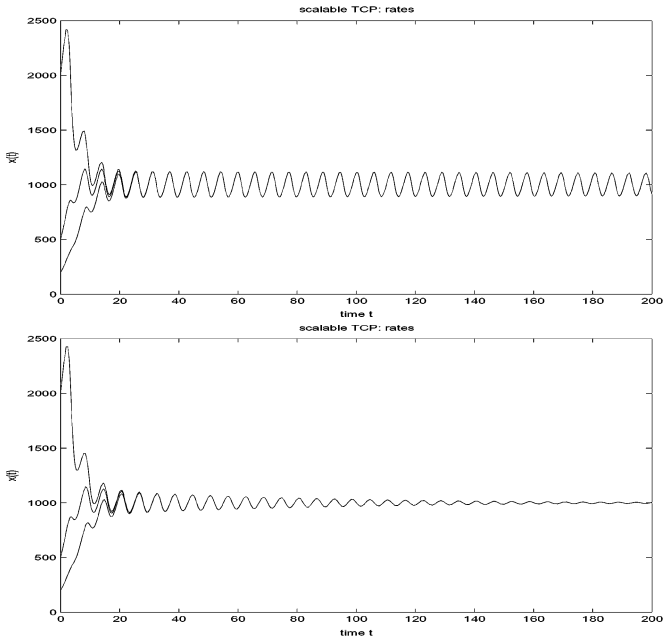

Fig. 10. Source rates for Scalable TCP+E-RED: $\xi = 1$ (top graph) and $\xi = 0.9$ (bottom graph).

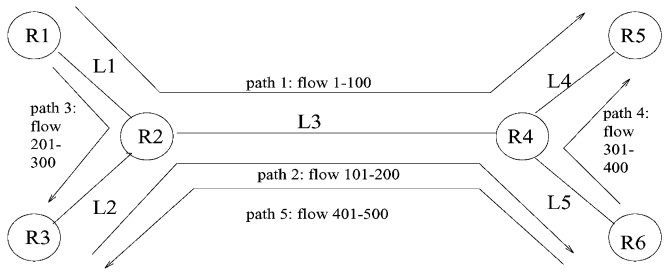

Fig. 11. The multilink network topology.

queue length. The queue length does not converge to its equilibrium value. However, simulations (not shown here) have indicated that if N is sufficiently large, the system converges to its equilibrium under E-RED.

The Matlab simulation for Scalable-TCP shows that if $\xi = 1$, both the source rates (shown in Fig. 10) and the link marking probability (not shown here) oscillate, without a trend of either divergence or convergence. If $\xi = 0.9$, all rates and probability converge. These results suggest that $\xi = 1$ is the critical point between stability and instability. This Matlab simulation also supports our earlier claim that the slow adaptation condition in Theorem III.2 is not only sufficient but also very close to being necessary.

F. Multi-Link Network Simulations

All the simulations above were for a single-link network. Our theoretical analysis applies to a general topology network and in this subsection we will examine the performance of E-RED in a multi-link network. The network is composed of 6 routers and 500 source-destination pairs (users), as shown in Fig. 11. The 500 flows from the 500 users are all FTP traffic and they go along 5 different paths: $R1 - R2 - R4 - R5$ $(L1)$, $R3 - R2 - R4 - R6$ $(L2)$, $R1 - R2 - R3$ $(L3)$, $R6 - R4 - R5$ $(L4)$, and $R6 - R4 - R2 - R3$ $(L5)$, with each path containing 100 flows. All the inter-router links have a capacity of 100 Mb/s and

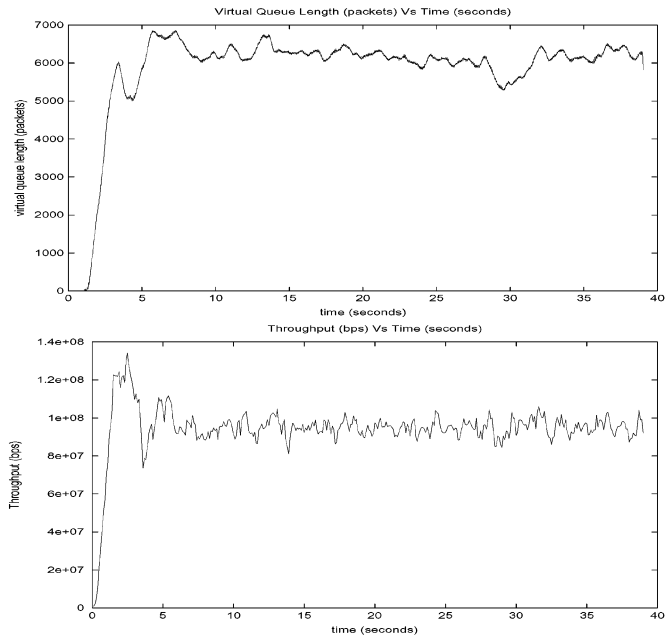

Fig. 12. The virtual queue length (top graph) and throughput (bottom graph) of the link between $R2$ and $R4$ in a multilink network.

a delay of 10 ms; and all the user-router links have a capacity of 10 Mb/s (large enough for these links not to be bottlenecked) and a delay uniformly distributed between 0 ms and 10 ms. We plot the virtual queue length and throughput of the link between $R2$ and $R4$ in Fig. 12; those of the other inter-router links show similar behavior. From the graphs, we see that E-RED stabilizes TCP-Reno for this multi-link multi-path multi-flow network and achieves satisfactory performance.

G. Summary of the Simulations

We summarize here our results from the simulations:

- For the general network with heterogeneous delays, E-RED stabilizes TCP-Reno, even when the capacity or the round trip delays are very large, given the condition that the number of flows is sufficiently large.
- E-RED-aq also stabilizes TCP-Reno, by reducing the link adaptation speed (Theorem IV.1).
- The transient response is fast when there are sudden changes in the available bandwidth.
- For a fixed network scenario, there is a threshold value of γ to guarantee stability, and for most realistic scenarios, this value of γ could be between 0.9 and 0.95. As N increases, this value of γ can be made very close to 1.0.
- The choice of $p_{\min,l}$ does not influence the stability and transient response, given that $p_{\min,l} < \hat{p}$,
- If $\xi > 1/2$, the system might be unstable. If $\xi \leq 1/2$ and the system is unstable because N is not large enough, the system cannot be stabilized by making ξ even smaller. This suggests that the large oscillations for small N are due to the TCP's AIMD behavior and cannot be eliminated through slow adaptation speed.

VI. Conclusion

In this paper, we have introduced a general class of primal-dual algorithms and the E-RED algorithm. This class of primal-dual algorithms has been shown to be locally stable in the presence of delay. The E-RED algorithm has been shown to stabilize TCP-Reno and its high speed variants, both analytically as well as through simulations. The simulations were performed under different scenarios and different parameter choices. These simulations clearly show that E-RED outperforms RED with larger throughput and lower queueing delay. They also suggest guidelines for tuning the E-RED parameters.

Appendix

In this appendix we provide proofs for Theorems III.2, IV.1, and IV.2. First, we present the following lemmas as preliminary to the proof of Theorem III.2 and Theorem IV.1:

Lemma 1: If $\theta_1 > 0$, $\theta_2 > 0$, $T_1 > 0$, $T_2 > 0$, $T_m = \max(T_1, T_2)$, $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, $\lambda_1 + \lambda_2 \leq 1$, then

$$f(\omega) = \lambda_1 \frac{T_1}{T_m} \frac{\theta_1 e^{-j\omega T_1}}{j\omega T_1(j\omega T_1 + \theta_1)} + \lambda_2 \frac{T_2}{T_m} \frac{\theta_2 e^{-j\omega T_2}}{j\omega T_2(j\omega T_2 + \theta_2)}$$

crosses the real line at the right of -2 as ω varies from $-\infty$ to ∞.

Proof: It is easy to see that $f(-\omega) = f^*(\omega)$, where $f^*(\omega)$ is the convex conjugate of $f(\omega)$, so we only need to consider the case of $\omega > 0$. Without loss of generality, we assume $T_1 \leq T_2$, then $T_m = T_2$. Let $x_1 = \omega T_1$, $x_2 = \omega T_2$, then $x_2 \geq x_1 > 0$, and $T_1/T_m = x_1/x_2$, $T_2/T_m = 1$. Define

$$C_i(x_i) := \frac{\theta_i e^{-jx_i}}{jx_i(jx_i + \theta_i)} = \frac{1}{x_i} \cos\phi_i e^{-jx_i - \phi_i - \frac{\pi}{2}}$$

where $\phi_i = \arctan(x_i/\theta_i) \in (0, \pi/2)$. Then, $f(\omega) = \lambda_1(x_1/x_2)C_1(x_1) + \lambda_2 C_2(x_2)$. Let $R := Re(f(\omega))$ and $I := Im(f(\omega))$. Then,

$$R = -\frac{\lambda_1 \cos\phi_1 \sin(\phi_1 + x_1) + \lambda_2 \cos\phi_2 \sin(\phi_2 + x_2)}{x_2}$$

and

$$I = -\frac{\lambda_1 \cos\phi_1 \cos(\phi_1 + x_1) + \lambda_2 \cos\phi_2 \cos(\phi_2 + x_2)}{x_2}.$$

To show that the curve crosses the real line to the right of -2, it is sufficient to show that $R > -2$ if $I = 0$. Since $R > -2$ if $x_2 > 1/2$, we only need to consider the case of $x_1 \leq x_2 \leq 1/2$. From $I = 0$, we get

$$\lambda_1 \cos\phi_1 \cos(\phi_1 + x_1) = \lambda_2 \cos\phi_2 \left(-\cos(\phi_2 + x_2)\right)$$

We consider the following two cases:

Case 1: $\cos(\phi_1 + x_1) > 0$ and $\cos(\phi_2 + x_2) < 0$, which lead to $\pi > \phi_2 + x_2 > \pi/2 > \phi_1 + x_1 > 0$. Then, we have $\cos\phi_2 = \sin(\pi/2 - \phi_2) < \sin x_2 < x_2$, and $0 < -\cos(\phi_2 + x_2) = \sin(\phi_2 + x_2 - \pi/2) < \sin x_2 < x_2$.

If $\phi_1 + x_1 \leq \phi_2$, then $\cos(\phi_1 + x_1) \geq \cos\phi_2$. So we have

$$\begin{aligned} |R| &= \frac{\lambda_2}{x_2}\left[\frac{-\cos\phi_2 \cos(\phi_2 + x_2)\sin(\phi_1 + x_1)}{\cos(\phi_1 + x_1)} + \cos\phi_2 \sin(\phi_2 + x_2)\right] \\ &\leq \frac{\lambda_2}{x_2}[-\cos(\phi_2 + x_2)\sin(\phi_1 + x_1) + \cos\phi_2 \sin(\phi_2 + x_2)] \\ &\leq \frac{1}{x_2}[(-\cos(\phi_2 + x_2)) + \cos\phi_2] < \frac{x_2 + x_2}{x_2} = 2 \end{aligned}$$

If $\phi_1 + x_1 > \phi_2$, then $\phi_1 > \phi_2 - x_1 \geq \phi_2 - x_2 > 0$, since $\phi_2 > \pi/2 - 1/2 > 1/2 \geq x_2$. So $\cos\phi_1 < \cos(\phi_2 - x_2) = \cos\phi_2 \cos x_2 + \sin\phi_2 \sin x_2$ and thus we have

$$\begin{aligned}|R| &= \frac{1}{x_2}\left[\lambda_1 \cos\phi_1 \sin(\phi_1 + x_1) + \lambda_2 \cos\phi_2 \sin(\phi_2 + x_2)\right] \\ &\leq \frac{1}{x_2}\left[\lambda_1(\cos\phi_2 \cos x_2 + \sin\phi_2 \sin x_2)\sin(\phi_1 + x_1)\right. \\ &\qquad \left. + \lambda_2 \cos\phi_2 \sin(\phi_2 + x_2)\right] \\ &\leq \frac{1}{x_2}\left[\lambda_1(\cos\phi_2 + \sin x_2) + \lambda_2 \cos\phi_2\right] \\ &< \frac{1}{x_2}\left[\lambda_1(x_2 + x_2) + \lambda_2 x_2\right] \leq 2.\end{aligned}$$

So under case 1, we have $R > -2$.

Case 2: $\cos(\phi_1 + x_1) \leq 0$ and $\cos(\phi_2 + x_2) \geq 0$, which lead to $\pi > \phi_1 + x_1 \geq \pi/2 \geq \phi_2 + x_2 > 0$. Then, we have $\cos\phi_1 = \sin(\pi/2 - \phi_1) \leq \sin x_1 < x_1 < x_2$, and $-\cos(\phi_1 + x_1) = \sin(\phi_1 + x_1 - \pi/2) < \sin x_1 < x_1 \leq x_2$.If $\phi_2 + x_2 \leq \phi_1$, then $\cos(\phi_2 + x_2) \geq \cos\phi_1$. So we have

$$\begin{aligned}|R| &= \frac{\lambda_1}{x_2}\left[\cos\phi_1 \sin(\phi_1 + x_1)\right. \\ &\qquad \left. + \frac{\cos\phi_1\,(-\cos(\phi_1 + x_1))\sin(\phi_2 + x_2)}{\cos(\phi_2 + x_2)}\right] \\ &\leq \frac{\lambda_1}{x_2}\left[\cos\phi_1 \sin(\phi_1 + x_1)\right. \\ &\qquad \left. + (-\cos(\phi_1 + x_1))\sin(\phi_2 + x_2)\right] \\ &\leq \frac{1}{x_2}\left[\cos\phi_1 + (-\cos(\phi_1 + x_1))\right] < \frac{x_2 + x_2}{x_2} = 2\end{aligned}$$

If $\phi_2 + x_2 > \phi_1$, then $\phi_2 > \phi_1 - x_2 > 0$, since $\phi_1 \geq \pi/2 - 1/2 > 1/2 \geq x_2$. So $\cos\phi_2 < \cos(\phi_1 - x_2) = \cos\phi_1 \cos x_2 + \sin\phi_1 \sin x_2$ and thus we have

$$\begin{aligned}|R| &= \frac{1}{x_2}\left[\lambda_1 \cos\phi_1 \sin(\phi_1 + x_1) + \lambda_2 \cos\phi_2 \sin(\phi_2 + x_2)\right] \\ &\leq \frac{1}{x_2}\left[\lambda_1 \cos\phi_1 \sin(\phi_1 + x_1)\right. \\ &\qquad \left. + \lambda_2(\cos\phi_1 \cos x_2 + \sin\phi_1 \sin x_2)\sin(\phi_2 + x_2)\right] \\ &\leq \frac{1}{x_2}\left[\lambda_1 \cos\phi_1 + \lambda_2(\cos\phi_1 + \sin x_2)\right] \\ &< \frac{1}{x_2}\left[\lambda_1 x_2 + \lambda_2(x_2 + x_2)\right] \leq 2.\end{aligned}$$

So under Case 2, we still have $R > -2$. Then, we have proved that $R > -2$ if $I = 0$ and thus the curve crosses the real line to the right of -2. □

Lemma 2: If $\lambda_i \geq 0$, $\theta_i > 0$, $T_i > 0$, $\forall i = 1, 2, 3$ and $T_m = \max(T_1, T_2, T_3)$, $\lambda_1 + \lambda_2 + \lambda_3 = 1$, then

$$f(\omega) = \sum_{i=1}^{3} \lambda_i g_i(\omega) = \sum_{i=1}^{3} \lambda_i \frac{T_i}{T_m} \frac{\theta_i e^{-j\omega T_i}}{j\omega T_i(j\omega T_i + \theta_i)}$$

crosses the real line to the right of -2 as ω varies from $-\infty$ to ∞.

Proof: Suppose $f(\omega)$ crosses the real line to the left of -2; then there exists an ω^* such that $Im(f(\omega^*)) = 0$ and $Re(f(\omega^*)) \leq -2$. If $s := \sum_{i=1}^{3} \lambda_i < 1$, then we can pick $\lambda_i' = \lambda_i/s$, and the new combination $f'(\omega^*) = \sum_{i=1}^{3} \lambda_i' g_i(\omega^*)$ also lies on the real line to the left of -2. So we only need to consider the case when $\sum_{i=1}^{3} \lambda_i = 1$, under which $f(\omega^*)$ lies in the triangle formed by A_1, A_2 and A_3, where A_i, $i = 1, 2, 3$ are the points in the complex plan corresponding to $g_i(\omega^*)$, $i = 1, 2, 3$. From the triangle property, there exists some point on one of the three edges of the triangle that is on the real line to the left of -2. From Lemma 1, we know that this cannot be true. □

Lemma 3: For any integer $N > 0$, if $\lambda_i \geq 0$, $\theta_i > 0$, $T_i > 0$, $\forall i = 1, 2, \ldots, N$ and $T_m = \max(T_i : i = 1, 2, \ldots, N)$, $\sum_{i=1}^{N} \lambda_i \leq 1$, then

$$f(\omega) = \sum_{i=1}^{N} \lambda_i g_i(\omega) = \sum_{i=1}^{N} \lambda_i \frac{T_i}{T_m} \frac{\theta_i e^{-j\omega T_i}}{j\omega T_i(j\omega T_i + \theta_i)}$$

crosses the real line at a point to the right of -2 as ω varies from $-\infty$ to ∞.

Proof: From Caratheodory's Theorem [3, Prop. B.6, p. 679], let X be a subset of R^n; then every element of conv(X) can be represented as a convex combination of no more than $n+1$ elements of X. So the convex combination of $g_i(\omega)$'s can be represented as a convex combination of three $g_i(\omega)$'s. From Lemma 2, we know that it is true for any integer N. □

With Lemma 3 at hand, we can now prove Theorem III.2 and Theorem IV.1:

Proof of Theorem III.2: Let $\tilde{a}_r := -\hat{q}_r/(\hat{x}_r U_r''(\hat{x}_r))$. Then, from (29), $\tilde{a}_r \leq a_r$.

Let $\beta_l := (g_l(\hat{p}_l)c_l/\hat{p}_l)$. Then from (28) and (29), we have

$$L(s) = \text{diag}\left(\frac{T_r \tilde{a}_r \hat{x}_r}{\hat{q}_r} \frac{\theta_r e^{-sT_r}}{sT_r(sT_r + \theta_r)}\right) A^T(-s) \text{diag}\left(\frac{\beta_l \hat{p}_l}{c_l}\right) A(s).$$

Since the open-loop system is stable, by the generalized Nyquist Criterion [5] the closed-loop system is asymptotically stable if the eigenloci of $L(j\omega)$ do not encircle -1. Now the eigenvalues of $L(j\omega)$ are identical with those of

$$\hat{L}(j\omega) = \text{diag}\left(\frac{T_r}{T_m} \frac{\theta_r e^{-j\omega T_r}}{2j\omega T_r(j\omega T_r + \theta_r)}\right) \hat{A}^T(-j\omega)\hat{A}(j\omega)$$

where

$$\hat{A}(j\omega) = \text{diag}\left(\sqrt{\frac{\beta_l \hat{p}_l}{c_l}}\right) A(j\omega) \text{diag}\left(\sqrt{\frac{2T_m \tilde{a}_r \hat{x}_r}{\hat{q}_r}}\right).$$

Let $\sigma(Z)$ denote the largest singular value of a square matrix Z and $\rho(Z)$ its spectral radius. Given the condition in (30), we have:

$$\begin{aligned}\sigma^2\left(\hat{A}(j\omega)\right) &= \rho\left(\hat{A}^T(-j\omega)\hat{A}(j\omega)\right) \\ &= \rho\left(\text{diag}\left(\frac{2\tilde{a}_r T_m \hat{x}_r}{\hat{q}_r}\right) A^T(-j\omega)\right. \\ &\qquad \left. \times \text{diag}\left(\frac{\beta_l \hat{p}_l}{c_l}\right) A(j\omega)\right) \\ &\leq \left\|\text{diag}\left(\frac{2a_r T_m}{\hat{q}_r}\right) A^T(-j\omega)\text{diag}(\beta_l \hat{p}_l)\right\| \\ &\qquad \times \left\|\text{diag}\left(\frac{1}{\hat{y}_l}\right) A(j\omega)\text{diag}(\hat{x}_r)\right\| \\ &\leq 1 \times 1 = 1.\end{aligned}$$

The last inequality above uses the facts that

$$\sum_{r:l\in r} \hat{x}_r = \hat{y}_l = c_l \;\forall\, l \in L,\; \sum_{l:l\in r} \hat{p}_l = \hat{q}_r \;\forall\, r \in R \qquad (47)$$

and

$$\beta_l < \frac{1}{2a_r T_m} \quad \forall\, r : l \in r$$

and thus the absolute row sums of these matrices are all less than or equal to 1.

Now, if λ is an eigenvalue of $L(j\omega)$, then

$$\lambda \in \mathrm{Co}\left(0 \cup \left\{\frac{T_r}{2T_m}\frac{\theta_r e^{-j\omega T_r}}{j\omega T_r(j\omega T_r+\theta_r)}\right\}\right).$$

From Lemma 3, the eigenloci of $L(j\omega)$ do not encircle -1; thus by the generalized Nyquist Criterion [5], the closed-loop system is stable. □

Remark 5: We can also show that $(\theta_r e^{-j\omega T_r}/j\omega T_r(j\omega T_r+\theta_r))$ always crosses the real axis to the left of the point $-2/\pi$, which suggests that our local stability condition will not be too conservative regardless of the value θ takes in $[0,\infty)$.

Proof of Theorem IV.1: Define $\tilde{a}_r$ as in the proof of Theorem III.2. Then, $\tilde{a}_r \le a_r \forall r$, and (41) becomes

$$L(s) = \mathrm{diag}\left(\frac{T_r\tilde{a}_r\hat{x}_r}{\hat{q}_r}\frac{\theta_r e^{-sT_r}}{sT_r(sT_r+\theta_r)}\right)A^T(-s) \times \mathrm{diag}\left(\frac{\beta_l\hat{p}_l}{c_l}\frac{\kappa_l}{s+\kappa_l}\right)A(s).$$

Similar to the proof of Theorem III.2, we need to show that the eigenloci of $L(j\omega)$ do not encircle -1. Now, the eigenvalues of $L(j\omega)$ are identical with those of

$$\hat{L}(j\omega) = \mathrm{diag}\left(\frac{T_r}{T_m}\frac{\theta_r e^{-j\omega T_r}}{2j\omega T_r(j\omega T_r+\theta_r)}\right)\hat{A}^T(-j\omega)\hat{A}(j\omega)$$

where

$$\hat{A}(j\omega) = \mathrm{diag}\left(\sqrt{\frac{\beta_l\hat{p}_l}{c_l}\frac{\kappa_l}{j\omega+\kappa_l}}\right)A(j\omega)\mathrm{diag}\left(\sqrt{\frac{2T_m\tilde{a}_r\hat{x}_r}{\hat{q}_r}}\right).$$

Given the condition in (30), we have:

$$\begin{aligned}\sigma^2\left(\hat{A}(j\omega)\right) &= \rho\left(\hat{A}^T(-j\omega)\hat{A}(j\omega)\right)\\ &= \rho\left(\mathrm{diag}\left(\frac{2\tilde{a}_r T_m\hat{x}_r}{\hat{q}_r}\right)A^T(-j\omega)\right.\\ &\qquad \left.\times\mathrm{diag}\left(\frac{\beta_l\hat{p}_l}{c_l}\frac{\kappa_l}{|j\omega+\kappa_l|}\right)A(j\omega)\right)\\ &\le \left\|\mathrm{diag}\left(\frac{2a_r T_m}{\hat{q}_r}\right)A^T(-j\omega)\right.\\ &\qquad \left.\times\mathrm{diag}\left(\beta_l\hat{p}_l\left|\frac{\kappa_l}{j\omega+\kappa_l}\right|\right)\right\|\\ &\qquad \times\left\|\mathrm{diag}\left(\frac{1}{\hat{y}_l}\right)A(j\omega)\mathrm{diag}(\hat{x}_r)\right\| \le 1\times 1 = 1.\end{aligned}$$

The last inequality above uses (47) and

$$\beta_l\left|\frac{\kappa_l}{j\omega+\kappa_l}\right| \le \beta_l \le \frac{1}{2a_r T_m} \quad \forall r, l : l \in r,$$

and thus the absolute row sums of these matrices are all less than or equal to 1.

Now, if λ is an eigenvalue of $L(j\omega)$, then

$$\lambda \in \mathrm{Co}\left(0 \cup \left\{\frac{\theta_r e^{-j\omega T_r}}{2j\omega T_m(j\omega T_r+\theta_r)}\right\}\right).$$

From Lemma 3, the eigenloci of $L(j\omega)$ do not encircle -1; thus by the generalized Nyquist Criterion [5], the closed-loop system is stable. □

We finally prove Theorem IV.2.

Proof of Theorem IV.2: Note that:

$$\begin{aligned}&\sum_{l\in r} p_l(1-p_l) \le q_r(1-q_r)\\ \iff &\sum_{l\in r}(e^{z_l}-1) \le e^{v_r}-1\\ \iff &\sum_{l\in r}(\tilde{a}_l-1) \le \prod_{l\in r}\tilde{a}_l - 1\\ \iff &\sum_{i=1}^{N} a_i \le \prod_{i=1}^{N} a_i + N - 1\end{aligned}$$

where $\tilde{a}_i := e^{z_l} \ge 1$ for all l, and $(a_i, 1 \le i \le N)$ is a permutation of $(\tilde{a}_l, l \in r)$. Let us now prove the last inequality by induction: it is obviously true for $N=1$ and $N=2$. Let us suppose that it is true for N. Then, for $N+1$, we have

$$\begin{aligned}\sum_{i=1}^{N+1} a_i &\le \prod_{i=1}^{N} a_i + N - 1 + a_{N+1}\\ &= \prod_{i=1}^{N+1} a_i + \prod_{i=1}^{N} a_i(1-a_{N+1}) + N + (a_{N+1}-1)\\ &= \prod_{i=1}^{N+1} a_i + N - \left(\prod_{i=1}^{N} a_i - 1\right)(a_{N+1}-1)\\ &\le \prod_{i=1}^{N+1} a_i + ((N+1)-1)\end{aligned}$$

Thus the inequality also holds for $N+1$, and therefore for all N. □

REFERENCES

[1] T. Alpcan and T. Başar, "A game-theoretic framework for congestion control in general topology networks," presented at the 41th IEEE Conf. Decision and Control, Las Vegas, NV, Dec. 2002.

[2] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, May/Jun. 2001.

[3] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[4] S. Deb and R. Srikant, "Rate-based versus queue-based models of congestion control," presented at the ACM Sigmetrics, New York, Jun. 2004.

[5] C. A. Desoer and Y. T. Wang, "On the generalized Nyquist stability criterion," *IEEE Trans. Autom. Contr.*, vol. 25, no. 2, pp. 187–196, Apr. 1980.

[6] W.-C. Feng, K. G. Shin, D. D. Kandalur, and D. Saha, "The BLUE active queue management algorithms," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 513–528, Aug. 2002.

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

[8] S. Floyd. (2003, Aug.) HighSpeed TCP for large congestion windows, Internet draft. [Online]. Available: http://www.icir.org/floyd/hstcp.html

[9] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, pp. 1969–1985, 1999.

[10] H. Han, C. V. Hollot, Y. Chait, and V. Misra, "TCP network stabilized by buffer-based AQMs," presented at the IEEE INFOCOM, Hong Kong, Mar. 2004.

[11] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "A control theoretic analysis of RED," presented at the IEEE INFOCOM, Anchorage, AK, Apr. 2001.

[12] ——, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. Autom. Contr.*, vol. 47, no. 6, pp. 945–959, Jun. 2002.
[13] R. Johari and D. Tan, "End-to-end congestion control for the Internet: delays and stability," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 818–832, Dec. 2001.
[14] F. Kelly, "Fairness and stability of end-to-end congestion control," *Eur. J. Control*, pp. 149–165, 2003.
[15] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.
[16] F. P. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, pp. 33–37, 1997.
[17] ——, "Models for a self-managed Internet," in *Phil. Trans. Roy. Soc.*, 2000, vol. A358, pp. 2335–2348.
[18] T. Kelly. (2002, Dec.) Scalable TCP: Improving performance in high speed wide area networks. [Online]. Available: http://www-lce.eng.cam.ac.uk/ctk21/papers/
[19] S. Kunniyur and R. Srikant, "End-to-end congestion control: utility functions, random losses and ECN marks," presented at the IEEE INFOCOM, Tel Aviv, Israel, Mar. 2000.
[20] ——, "Analysis and design of an adaptive virtual queue algorithm for active queue management," in *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001, pp. 123–134.
[21] ——, "Designing AVQ parameters for a general topology network," presented at the Asian Control Conf., Singapore, Sep. 2002.
[22] ——, "Note on the stability of the AVQ scheme," presented at the Conf. Information Sciences and Systems, Princeton, NJ, Mar. 2002.
[23] ——, "A time-scale decomposition approach to adaptive ECN marking," *IEEE Trans. Autom. Contr.*, vol. 47, no. 6, pp. 882–894, Jun. 2002.
[24] S. Liu, T. Başar, and R. Srikant, "Controlling the Internet: a survey and some new results," presented at the IEEE Conf. Decision and Control, Maui, HI, Dec. 2003.
[25] S. H. Low and D. E. Lapsley, "Optimization flow control, I: basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 6, no. 6, pp. 861–875, Dec. 1999.
[26] S. Low, F. Paganini, J. Wang, S. Adlakha, and J. Doyle, "Dynamics of TCP/RED and a scalable control," presented at the IEEE INFOCOM, New York, Jun. 2002.
[27] V. Misra, W. Gong, and D. Towsley, "A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," presented at the ACM SIGCOMM, Stockholm, Sweden, Sep. 2000.
[28] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, Oct. 2000.
[29] F. Paganini, J. Doyle, and S. Low, "Scalable laws for stable network congestion control," presented at the IEEE Conf. Decision and Control, Orlando, FL, Dec. 2001.
[30] F. Paganini, Z. Wang, J. Doyle, and S. Low, "A new TCP/AQM for stable operation in fast networks," presented at the IEEE INFOCOM, San Francisco, CA, Apr. 2003.
[31] R. Srikant, *The Mathematics of Internet Congestion Control*. Boston, MA: Birkhäuser, 2004.
[32] G. Vinnicombe. (2001) On the stability of end-to-end congestion control for the Internet. University of Cambridge Tech. Report CUED/F-INFENG/TR.398. [Online]. Available: http://www.eng.cam.ac.uk/~gv
[33] ——, "On the stability of networks operating TCP-like congestion control," presented at the IFAC World Congress, Barcelona, Catalonia, Spain, 2002, [Online.] Available: http://www.eng.cam.ac.uk/~gv.
[34] ——, (2002) Robust Congestion Control for the Internet. University of Cambridge Tech. Report. [Online]. Available: http://www.eng.cam.ac.uk/~gv
[35] J. T. Wen and M. Arcak, "A unifying passivity framework for network flow control," *IEEE Trans. Autom. Contr.*, vol. 49, no. 2, pp. 162–174, 2004.
[36] H. Yaiche, R. R. Mazumdar, and C. Rosenberg, "A game-theoretic framework for bandwidth allocation and pricing in broadband networks," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 578–667, Oct. 2000.
[37] L. Ying, G. Dullerud, and R. Srikant, "Global stability of Internet congestion controllers with heterogeneous delays," presented at the American Control Conf., Boston, MA, Jun. 2004.

Shao Liu (S'05) received the B.S. degree from Peking University, Beijing, China, and the M.S. degree in electrical engineering from the University of Illinois at Urbana-Champaign, where he is currently working toward the Ph.D. degree in electrical engineering and performing research on congestion control for communication networks, and contention and congestion control for wireless ad hoc networks.

Tamer Başar (S'71–M'73–SM'79–F'83) received the B.S.E.E. degree from Robert College, Istanbul, Turkey, and the M.S., M.Phil., and Ph.D. degrees in engineering and applied science from Yale University, New Haven, CT.

After stints at Harvard University and Marmara Research Institute, Gebze, Turkey, he joined the University of Illinois at Urbana-Champaign in 1981, where he is currently the Fredric G. and Elizabeth H. Nearing Professor of Electrical and Computer Engineering. He has published extensively in systems, control, communications, and dynamic games, and has current interests in modeling and control of communication networks, control over heterogeneous networks, resource management and pricing in networks, and robust identification and control.

Dr. Başar is the Editor-in-Chief of *Automatica*, Editor of the Birkhäuser Series on Systems and Control, Managing Editor of the *Annals of the International Society of Dynamic Games* (ISDG), and member of editorial and advisory boards of several international journals. He has received several awards and recognitions over the years, among which are the Medal of Science of Turkey (1993), Distinguished Member Award (1993), Axelby Outstanding Paper Award (1995) and Bode Lecture Prize (2004) of the IEEE Control Systems Society (CSS), and the Quazza Medal (2005) of IFAC. He is a member of the National Academy of Engineering, a member of the European Academy of Sciences, a Fellow of IEEE, a past president of CSS, and a past (founding) president of ISDG.

R. Srikant (S'90–M'91–SM'01) received the B.Tech. degree from the Indian Institute of Technology, Madras, in 1985, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign in 1988 and 1991, respectively, all in electrical engineering.

He was a Member of Technical Staff at AT&T Bell Laboratories from 1991 to 1995. He is currently with the University of Illinois, where he is a Professor in the Department of Electrical and Computer Engineering and a Research Professor in the Coordinated Science Laboratory. His research interests include communication networks, stochastic processes, queueing theory, information theory, and game theory.

Dr. Srikant was an associate editor of *Automatica*, and is currently on the editorial boards of the IEEE/ACM TRANSACTIONS ON NETWORKING and IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He was the chair of the 2002 IEEE Computer Communications Workshop and will be a program co-chair of IEEE INFOCOM in 2007.