Controlling the Internet: A Survey and Some New Results

Shao Liu, Tamer Başar, and R. Srikant
(shaoliu@uiuc.edu, tbasar@control.csl.uiuc.edu, rsrikant@uiuc.edu)

***Abstract*— In the last few years, significant progress has been made in the mathematical modelling of congestion control and congestion feedback mechanisms in the Internet. The resulting models have proved to be very useful in improving existing control and feedback mechanisms, and to make them scalable to networks that operate at very high speeds. Tools from convex optimization, control theory and stochastic processes have played a major role in the development of this Internet congestion control theory. In this paper, we focus on the control-theoretic aspects of the theory, and review some recent developments in the design of stable, scalable congestion control mechanisms. We also present a new scheme that can improve the performance of the Internet with minimal changes to the current architecture.**

***Keywords:* Congestion Control, Primal Algorithm, Dual Algorithm, Primal-Dual Algorithm, TCP, AQM, RED, AVQ, Generalized Nyquist Criterion.**

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 [1], 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* [1]. Congestion control schemes can be viewed as algorithms to compute the optimum or some suboptimum solution to this maximization problem.

Congestion control schemes can be divided into two classes: primal algorithms and 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 [1]. 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 [1], [2], [3].

A modified primal algorithm, called the AVQ (Active Virtual Queue) algorithm, was introduced in [4]. Here the link prices in the original primal algorithm [1] are slowly adjusted so that asymptotically in time, the link prices are equal to the Lagrange multipliers in Kelly's system problem [4]. Most importantly, in the presence of feedback delays, the parameters of this algorithm can be chosen such that the network is locally stable [5], [6]. 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 [7] to modify the dual algorithm to form a new algorithm where a slow adaptation at the sources was shown to yield the same benefits as the AVQ algorithm in [5], [6].

In this paper, we first briefly review the above developments and identify their role in allocating resource fairly in a network of competing users. We then introduce a new class of algorithms which are of the **primal-dual** type ([8], [9]), that is they feature dynamic adaptations at both user ends and link ends. In this paper, we provide design guidelines to choose the parameters of such algorithms for general topology networks, with heterogeneous feedback delays. In this class of algorithms, the source dynamics are similar to those in the primal algorithm in [1] and [10] while the link dynamics are similar to those in the dual algorithm [2]. Specifically, following the approach introduced in [10], we obtain a local stability that shows that stability does not depend on the source adaptation speed, but depends only on the link adaptation speed. This result has the dual algorithm as its limiting case when the source adaptation speed approaches infinity. We also show that the primal-dual algorithm can be implemented by making small modifications to TCP-Reno and RED (Random Early Detection), the two most widely used protocols today. Our modification to RED sets the packet marking probability as 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 E-RED (Exponential-RED). We present some *ns-2* simulation results to show that E-RED outperforms RED in the sense that it can stabilize TCP-Reno and achieve high bandwidth utilization and low queueing delay at the same time.

II. Overview of Prior Work

Suppose we have a set of users/routes, R, and a set of links/resources, L. For each user $r \in R$, its route involves 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 the relationship $y = Ax$.

Research supported by the NSF ITR Grant CCR 00-85917 and DARPA Grant F30602-00-2-0542

Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois, 1308 West Main Street, Urbana, IL 61801-2307, USA.

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 over the range $x_r \geq 0$, and further assume that utilities are additive, so that the aggregate utility for the system is $U(x) = \sum_{r\in R} U_r(x_r)$. The so-called system problem, first introduced in this context by Kelly [11], is the concave programming problem:

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

where $c = (c_l, l \in L)$, a column vector. The Lagrangian for the system problem 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 [12] 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. Suppressing the *hat*, a pair (x,p) is in saddle-point equilibrium if and only if [11]

$$U_r'(x_r) = q_r \quad \forall r \in R\,, \tag{4}$$

$$p_l(c_l - y_l) = 0 \;\text{ and }\; p_l \geq 0, \;\text{ and }\; c_l \geq y_l \quad \forall l \in L\,. \tag{5}$$

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 *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 among the user adaptation laws, and AQM algorithms are among the link adaptation laws.

For the logarithmic utility function, $U_r(x_r) = w_r \log x_r$, introduced above, Kelly introduced the Primal Algorithm [1]: 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 computes its price as

$$p_l(t) = f_l(y_l(t)), \;\text{ where } f_l > 0, f_l' > 0\,. \tag{7}$$

Kelly [1] has shown that this primal algorithm globally converges to a point which uniquely maximizes the function

$$\widetilde{U}(x) = \sum_{r\in R} w_r \log x_r - \sum_{l\in L}\int_0^{y_l} f_l(y)\,dy\,. \tag{8}$$

The above development has not incorporated delay into the algorithm. If we consider propagation delay and neglect queueing delay, thus treating delay as a constant, then we have the relationships:

$$y_l(t) = \sum_{r:l\in r} x_r(t - \tau_{lr}^f)\,, \tag{9}$$

$$q_r(t) = \sum_{l\in r} p_l(t - \tau_{lr}^b)\,, \tag{10}$$

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

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

Then we have the relationships [10]

$$Y(s) = A(s)X(s)\,, \tag{11}$$

$$Q(s) = \text{diag}(e^{-sT_r})A^T(-s)P(s)\,, \tag{12}$$

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

User end:

$$\dot{x_r}(t) = k_r(w_r - x_r(t - T_r)q_r(t))\,. \tag{13}$$

Link end:

$$p_l(t) = f_l(y_l(t))\,. \tag{14}$$

In the presence of delays, it is very hard to prove global stability for this general framework. Conditions for local stability were first conjectured in [13], and then proved and extended in [14], [10], [15].

Note that the link price computation in the above primal algorithm above is static. An algorithm called the AVQ algorithm was introduced in [4], 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 shown in [16] without delay, and local stability has been shown in [5], [6] in the presence of feedback delays.

B. Dual Algorithm

The dual algorithm introduced in [2] employs the following algorithms at the link and user ends:

Links:

$$\dot{p}_l = \gamma_l (y_l - c_l)\,, \tag{15}$$

where $\gamma_l > 0$ is an adaptation parameter. In [2], γ_l is chosen to be $1/c_l$.

Users:

$$x_r = x_r(q_r) = U_r'^{-1}(q_r)\,. \tag{16}$$

The above dual algorithm is a special case of the dual algorithm in [1], [17]. It was also derived in [3] for the special case of logarithmic utility functions.

For this dual algorithm, global stability was proved in [2] while local stability was proved for a particular choice of utility functions in [18]. The restriction on the choice of utility functions means that arbitrary relative fairness among the users cannot be achieved using the algorithm in [18]. Following the time-scale decomposition idea in [16], [5], [6], to track an arbitrary utility function, the dual algorithm was modified by introducing slow time-scale dynamics at the user end [7]. 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 and 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 the 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:

$$\dot{x}_r = k_r \frac{\partial L(x,p)}{\partial x_r} = k_r \left(U_r'(x_r) - q_r \right)\,, \tag{17}$$

$$\dot{p}_l = -\gamma_l \frac{\partial L(x,p)}{\partial p_l} = \gamma_l (y_l - c_l)\,. \tag{18}$$

Such algorithms were considered in [9] and [8], 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.

For this algorithm, at equilibrium,

$$x_r = U_r'^{-1}(q_r)\,, \tag{19}$$

and

$$y_l \begin{cases} = c_l & \text{if} \quad p_l > 0\,, \\ \le c_l & \text{if} \quad p_l = 0\,, \end{cases} \tag{20}$$

for all l, r. So the equilibrium point solves the system problem (1) exactly. In [8], Wen and Arcak show by a passivity approach the 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 establish local stability in the presence of delay.

D. Remarks on the Link Model

As we have stated, most of the AQM algorithms choose the marking probability as a function of the queue length. So it is intuitive to choose the 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 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 [19], 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 often are not provided in [19]. Recently, in [20], 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 get a queue-based model; else, the queue length hits zero frequently and we get the rate-based model. Thus, the primal-dual algorithm considered here assumes that the AQM parameters are chosen such that the queue length does not hit zero frequently. Since we wish to have a low queueing delay, it suggests that the AQM algorithm based on the primal-dual algorithm analysis must base its marking probability on the occupancy of a virtual queue, whose capacity is slightly smaller than the real link capacity. Thus, we can maintain a large virtual queue length as well as a small real queue length. We will discuss this further in later sections.

III. General Primal-Dual Algorithm and Its Local Analysis

The primal-dual algorithm (17)-(18) is too specific and still far from 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

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

The general primal-dual algorithm is given by

$$\dot{x}_r = \left[f_r(x_r)(U_r'(x_r) - q_r)\right]_{x_r}^+\,, \tag{21}$$

$$\dot{p}_l = \left[g_l(p_l)(y_l - c_l)\right]_{p_l}^+\,, \tag{22}$$

where $f_r(x_r)$ and $g_l(p_l)$ are positive and continuous functions.

A. Global Stability without Delay

It immediately follows that the equilibrium point of the general primal-dual algorithm (21)-(22) solves the system problem exactly. Using the same passivity approach in [8], one can show that the general primal-dual algorithm without delay is globally asymptotically stable (GAS). Define

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

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

and introduce

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

Then, it can be verified that $V(x, p)$[1] is a Lyapunov function for the system determined by (21)-(22), which leads to GAS.

B. Local Stability Analysis with Delay

In the presence of feedback delay, we can generalize (21-(22) to

$$\dot{x}_r(t) = \left[f_r(x_r(t), x_r(t-T_r))(U_r'(x_r(t)) - q_r(t))\right]_{x_r}^+\,, \tag{23}$$

$$\dot{p}_l(t) = \left[g_l(p_l(t))(y_l(t) - c_l)\right]_{p_l}^+\,, \tag{24}$$

where, by slightly abusing notation, we allow f_r to be a function of both $x_r(t)$ and $x_r(t - T_r)$.

Before we proceed with the local analysis by linearizing the algorithm around the equilibrium values, we first consider the boundary conditions in (21) and (22) and the constant delay assumption in the system modelling. For the users, we restrict the user set R to the active user set $R_a := \{r \in R : \hat{x}_r > 0\}$. If for one user r, $U_r'(0)$ is larger than any possible value of q_r, then, r is willing to join the system and thus $\hat{x}_r > 0$. 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 sets $R \setminus R_a$ and $L \setminus L_a$ do not affect the local analysis, for the rest of this paper we will assume that $R = R_a$ and $L = L_a$. Then, the boundary conditions in (21) and (22) can be ignored. Also, we assume that the queueing delay can be neglected compared with propagation delay, 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 [21].

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

Let us first linearize (23). At the equilibrium point, we have $\hat{q}_r = U_r'(\hat{x}_r)$. Linearization around it gives:

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

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 $X(s)$, we have:

$$\begin{aligned} X_r(s) &= -\hat{f}_r \frac{1}{s - \hat{f}_r U_r''(\hat{x}_r)} Q_r(s) \\ &= -\frac{1}{u_r''(\hat{x}_r)} \frac{\theta_r}{sT_r + \theta_r} Q_r(s)\,, \end{aligned} \tag{25}$$

where

$$\theta_r = \hat{f}_r T_r(-U_r''(\hat{x}_r))\,. \tag{26}$$

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

For the link dynamics, the linearization yields:

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

In the Laplace domain, we have:

$$P_l(s) = \frac{g_l(\hat{p}_l)}{s} Y_l(s)\,. \tag{27}$$

The return ratio of the closed-loop feedback system (25) and (27) is given by

$$\begin{aligned} L(s) \quad &= \quad \text{diag}\Big(\frac{1}{-U_r''(\hat{x}_r)} \frac{\theta_r e^{-sT_r}}{sT_r + \theta_r}\Big) A^T(-s) \\ &\quad \times \text{diag}\Big(\frac{g_l(\hat{p}_l)}{s}\Big) A(s)\,. \end{aligned}$$

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

$$\begin{aligned} L(s) \quad &= \quad \text{diag}\Big(\frac{T_r}{-U_r''(\hat{x}_r)} \frac{\theta_r e^{-sT_r}}{sT_r(sT_r + \theta_r)}\Big) A^T(-s) \\ &\quad \times \text{diag}(g_l(\hat{p}_l)) A(s)\,. \end{aligned} \tag{28}$$

This now leads to the following theorem on stability, whose proof is given in the Appendix.

Theorem III.1. *The closed-loop system described by (23), (24), (9), and (10) is locally asymptotically stable around the equilibrium point if*

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

and

$$\frac{g_l(\hat{p}_l)}{\hat{p}_l} \leq \frac{1}{c_l} \min_{r:l\in r} \frac{1}{a_r T_r}\,, \tag{30}$$

where a_r is a positive constant.

Remark 1. *The condition (29) is generally satisfied. 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. *The general primal-dual algorithm requires the control gain at the router to be inversely proportional with the maximum round-trip delay of any source using that router. This is a relaxation of the condition for local stability of the AVQ algorithm [5], [6] which requires the gain at the router to be inversely proportional with the maximum round-trip delay in the network.*

Remark 3. *From Theorem III.1, we see that local stability depends only on the link price adaptation speed, and not on the source rate adaptation speed. In other words, the stability condition does not depend on $\{\hat{f}_r\}$. 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. Thus, the interesting observation from our analysis is that slow adaptation at the source (as in [7]) 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 local stability result does not depend on κ_r.*

IV. Implementation in the Internet and Relationship to TCP+RED

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 and achieve 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. Always Additive Increment (AAI)-TCP

Consider the following implementation of TCP. Let W_r represent the window size of source r. Suppose the window size increases by m/W_r for each acknowledgement and decreases by nW_r for each marked acknowledgement, where m and n are some positive constants. This adaptation can be written in differential equation form as

$$\dot{W}_r = x_r(t-T_r)(m\frac{1}{W_r(t)} - nW_r(t)q_r(t))\,. \quad (31)$$

For the current version of TCP, $m = 1, n = 2/3$. Further, note that the window size is increased by m/W_r, irrespective of whether the acknowledgement contains a mark or not. For this reason, we call (31) AAI-TCP.

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

$$T_r\dot{x}_r = x_r(t-T_r)(m\frac{1}{T_r x_r(t)} - nT_r x_r(t)q_r(t))\,, \quad (32)$$

$$\begin{aligned} \dot{x}_r &= \frac{x_r(t-T_r)}{T_r}\left(m\frac{1}{T_r x_r(t)} - nT_r x_r(t)q_r(t)\right) \\ &= \frac{m x_r(t-T_r)}{x_r(t)T_r^2}(1 - \frac{1}{m}nT_r^2 x_r^2(t)q_r(t))\,. \end{aligned} \quad (33)$$

Comparing this with the source law of the primal-dual algorithm (23), we have:

$$f_r(x_r(t), x_r(t-T_r)) = nx_r(t)x_r(t-T_r)\,, \quad \hat{f}_r = n\hat{x}_r^2\,,$$

and

$$U_r'(x_r(t)) = \frac{m}{nT_r^2 x_r^2(t)}\,.$$

Thus,

$$U_r(x) = -\frac{m}{nxT_r^2}\,, \quad U_r''(x) = -\frac{2m}{nx^3T_r^2}\,,$$
$$U_r''(\hat{x}_r) = -\frac{2\hat{q}_r}{\hat{x}_r}\,, \text{ and } a_r = \frac{1}{2}\,.$$

B. E-RED: Exponential RED

From Theorem III.1, we know that to stabilize the AAI-TCP source law, the link dynamic adaptation should satisfy

$$\frac{g_l(\hat{p}_l)}{\hat{p}_l} \le \frac{2}{T_{ml}c_l}\,, \quad (34)$$

where $T_{ml} := \max_{r:l\in r} T_r$.

Consider the link dynamics

$$\dot{p}_l = \beta_l\frac{p_l}{c_l}(y_l - c_l) \quad \text{if} \quad p_l > 0\,, \quad (35)$$

with $\beta_l < 2/T_{ml}$. It satisfies the stability condition (34) and can be implemented easily by setting the packet marking probability (the link price) p_l as an exponential function of the queue length b_l:

$$p_l = \begin{cases} 0 & \text{if} \quad 0 \le \tilde{b}_l < th_{min,l}\,, \\ p_{min}e^{\frac{\beta_l}{c_l}(\tilde{b}_l - th_{min,l})} & \text{if} \quad th_{min,l} < \tilde{b}_l < th_{max,l}\,, \\ 1 & \text{if} \quad \tilde{b}_l \ge th_{max,l}\,, \end{cases} \quad (36)$$

where, $\tilde{b}_l = b_l$, and $th_{min,l} < th_{max,l}$ are the two queue length thresholds between which the exponential marking is selected and p_{min} is the marking threshold when $b_l = th_{min,l}$. Let p_{max} denote the marking probability when $b_l = th_{max,l}$; then, $th_{min,l}$, $th_{max,l}$, and p_{min} must be such that $p_{max} < 1$. We call the above marking scheme E-RED. We note that the form of (36) is similar to the fair dual algorithm in [17]. However, the source control laws are static in the fair dual algorithm, and hence the stability conditions in [17] are different than the ones presented in this paper.

From Theorem III.1, the condition $\beta_l < 2/T_{ml}$ guarantees the local stability of AAI-TCP and E-RED. Note that E-RED simply uses a different marking profile than RED, and chooses a virtual 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 enhancement E-RED-aq and the original E-RED without averaging E-RED-iq (iq stands for instantaneous queue length).

For this enhancement, we still use the marking equation (36), but with $\tilde{b}_l = r_l$, where r_l is the exponentially weighted moving average of b_l based on samples taken every δ_l seconds. Thus,

$$r_l((k+1)\delta_l) = r(k\delta_l)(1-\alpha_l) + \alpha_l b_l\,, \quad (37)$$

where $0 < \alpha_l < 1$ is the weight. As in [22], 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, $\dot{r}_l = 0$, the linearization of the above equation 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 (21) with logarithmic or power utility function and E-RED-aq, we have the following loop function:

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

We now present the following theorem on the stability condition for the source algorithm (21) and E-RED-aq; its proof can be found in the Appendix.

Theorem IV.1. *The closed-loop system composed of the source algorithm (21) and E-RED-aq is locally asymptotically stable around the equilibrium if*

$$-U_r''(\hat{x}) \geq \frac{\hat{q}_r}{a_r \hat{x}_r}\,, \tag{42}$$

and

$$\beta_l \leq \frac{1}{\eta_{aq}} \min_{r:l\in r} \frac{1}{a_r T_r}\,, \tag{43}$$

where a_r is a positive constant, and

$$\eta_{aq} = \max(1 + \frac{1}{M}, \frac{2}{\pi} + \frac{4}{\pi M})\,, \tag{44}$$

where M is defined as

$$M := \min_{r\in R, l\in L} \kappa_l T_r\,, \tag{45}$$

Remark 4. *This theorem suggests that E-RED-aq requires a slower adaptation speed than E-RED-iq. Also, as α_l decreases, and thus κ_l decreases, we need to decrease the adaptation speed β_l. From the definition of κ_l, we know that $\kappa_l \leq \alpha_l/\delta_l$. If the average queue length is updated once every L packet arrivals, at equilibrium, $\delta_l \approx L/c_l$, where c_l is the capacity in packets per second. So if we define M' to be*

$$M' := \min_{r\in R, l\in L} \frac{\alpha_l c_l}{L} T_r\,, \tag{46}$$

and replace M in (44) with M', then we get a larger η_{aq} and a weaker condition for β_l and Theorem IV.1 still holds. The new definition of η_{aq} shows clearly how the stabilizing link adaptation speed β_l is influenced by the averaging parameter α_l.

Remark 5. *In other work on modelling TCP [23], [22], equation (31) has been widely accepted as the model for current TCP-Reno. In other words, the fact that the window is only increased when there is no mark in the acknowledgement has been ignored in [23], [22]. In our* ns-2 *simulations, we do not modify TCP-Reno and the simulation results suggest that TCP-Reno with E-RED is stable.*

Remark 6. *The slow link adaptation makes the queue length very large. To maintain a small queue length, 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.*

Remark 7. *In the TCP-AQM scheme, the prices p_l and q_r are expressed in terms of the marking probabilities. In the optimization model, the feedback from p_l to q_r is linear in (10). However, the probabilitistic feedback is actually not linear:*

$$q_r = 1 - \prod (1 - p_l)\,. \tag{47}$$

Hence for the above local analysis to hold, p_l has to be sufficiently small for all $l \in L$, so that (47) can be approximated to be the linear relationship (10). In the case when the marking probability is large or the number of bottleneck links is large, this approximation will not hold. We have shown however, that even under the exact probabilistic feedback, the local stability results still hold. Details can be found in [24].

V. Simulations

We have implemented the E-RED algorithm in the *ns*-2 package and performed simulations on the network whose sources choose TCP-Reno and whose routers choose E-RED or RED. Let $\xi_l := \beta_l T_{lm}/2$; then, E-RED chooses the parameters ξ_l, p_{min}, p_{max}, $th_{min,l}$, γ, and computes $th_{max,l}$ from p_{min}, p_{max}, $th_{min,l}$. $\xi_l = 1$ guarantees stability, but a larger ξ_l leads to faster adaption of the link price, with a possibility of instability. We have chosen different ξ_l values, to see the influence of ξ_l (the link adaptation speed) on stability. We consider a network with a single bottleneck link and multiple flows, as shown in Fig. 1. We assume that each source chooses TCP-Reno and sends packets of the same size, 1040 bytes, including the IP and TCP header.

We first compare the stability and performances of E-RED and RED. We choose a large link capacity network. The scenario that we consider is one where $c = 1$ Gbps, $N = 2000$, $d = 10$ms, and the delays of all other links are uniformly

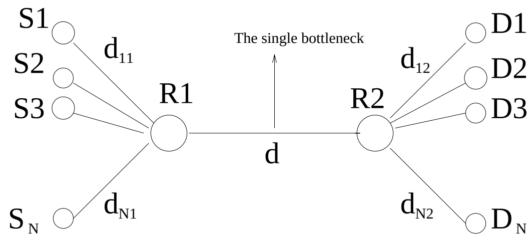

Fig. 1. The network topology: single bottleneck link and multiple flows with heterogeneous delays

distributed between 1 and 20 ms. So the maximum round trip time is $T_m = 50$ms. The buffer limit of the bottleneck link is 300 packets. For RED, we keep all parameters at their default settings and choose ECN marking instead of dropping. For E-RED, we set th_{min} to be $1/5$ of the buffer limit, $p_{min} = 0.0005$, $p_{max} = 0.1$, $\gamma = 0.95$, $\xi = 1$. For RED, we monitor the instantaneous and average queue length, and they are shown in Fig. 2. From the graphs, we see that RED is unstable in the sense that the queue length shows significant oscillation, and hits zero and the full buffer size frequently. For E-RED, we monitor the real queue length and the virtual queue length, and they are shown in Fig. 3 and Fig. 4. From the graphs, we see that E-RED is stable in the sense that the virtual queue length oscillates around its equilibrium value only slightly and the real queue size is always small after the transient phase. So RED does not stabilize TCP-Reno when the link capacity is very large, which agrees with the analysis in [25]. However, E-RED does. As for the performance, we show the throughputs of RED and E-RED in Fig. 5 and Fig. 6. It shows clearly that the throughput achieved by E-RED is less oscillatory than that of RED. We then measure the average throughput and queue length of both algorithms after the transient phase. We average over the interval between 10s and 39s; the average throughput and queue length for RED are 0.939621 Gbps and 117.255 packets, while those for E-RED are 0.950129 Gbps (almost the same as γc) and 12.2429 packets. So E-RED achieves a higher bandwidth utilization as well as a much lower queueing delay.

In the second simulation, we have chosen different ξ values to see the influence of the adaptation speed on stability and transient response. We now choose $c = 200$ Mbps, $N = 200$, and the delays are the same as in the first scenario. The bottleneck link buffer limit is 60 packets. The E-RED parameters are chosen as $th_{min} = 12$, $p_{min} = 0.0005$, $p_{max} = 0.1$, $\gamma = 0.9$. We have chosen ξ values to be $1, 2, 3, 4, 5$ and monitor the virtual queue lengths for the five ξ values and they are shown in Figs. 7-11. From the graphs, we see that as ξ increases, the virtual queue shows higher oscillations; even the recovery time is best for $\xi = 1$. So this simulation suggests that our sufficient condition on local stability is also pretty close to a necessary condition.

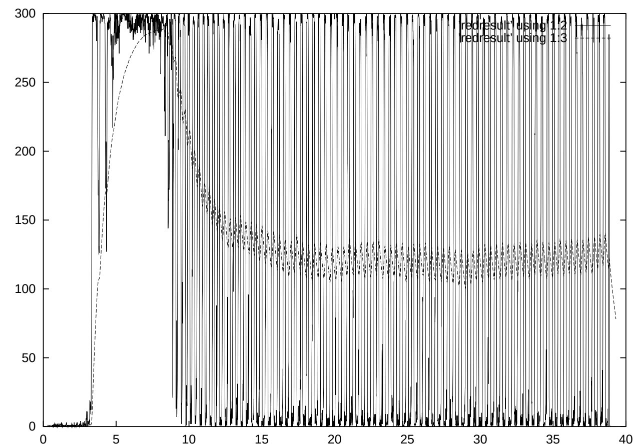

Fig. 2. RED: Instantaneous Queue and Average Queue.

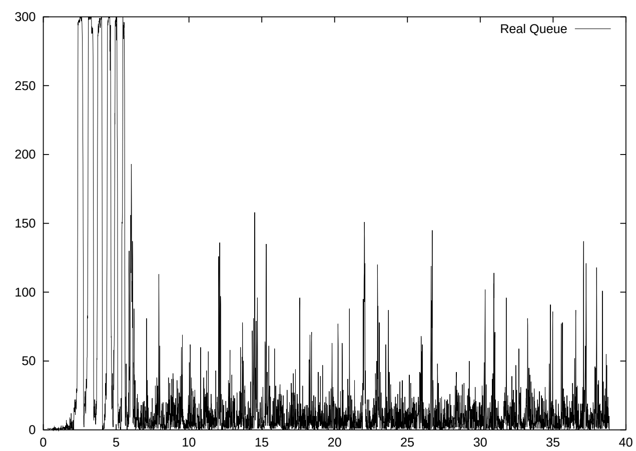

Fig. 3. E-RED: Real Queue Length.

REFERENCES

[1] F. P. Kelly, A. Maulloo, and D. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[2] S. H. Low and D. E. Lapsley, “Optimization flow control, I: Basic algorithm and convergence,” *IEEE/ACM Transactions on Networking*, pp. 861–875, December 1999.

[3] H. Yaiche, R. R. Mazumdar and C. Rosenberg, ”A game-theoretic framework for bandwidth allocation and pricing in broadband network,” *IEEE/ACM Transactions on Networking*, pp. 667–678, October 2000.

[4] S. Kunniyur and R. Srikant, “Analysis and design of an adaptive virtual queue algorithm for active queue management,” in *Proc. ACM Sigcomm*, San Diego, CA, August 2001, pp. 123–134.

[5] S. Kunniyur and R. Srikant, “Note on the stability of the AVQ scheme,” in *Proc. Conference on Information Sciences and Systems*, Princeton, NJ, March 2002.

[6] S. Kunniyur and R. Srikant, “Designing AVQ parameters for a general topology network,” in *Proc. Asian Control Conference*, Singapore, September 2002.

[7] F. Paganini, Z. Wang, J. Doyle, and S. Low, “A new TCP/AQM for stable operation in fast networks,” in *Proc. IEEE Infocom*, San Francisco, CA, April 2003.

[8] J. Wen and M. Arcak, “A unifying passivity framework for network flow control,” in *Proc. IEEE Infocom*, San Francisco, CA, April 2003.

[9] T. Alpcan and T. Başar, “A game-theoretic framework for congestion control in general topology networks,” in *Proc. 41th IEEE Conference on Decision and Control*, Las Vegas, Nevada, December 2002.

[10] G. Vinnicombe, “On the stability of networks operating TCP-like congestion control,” in *Proc. IFAC World Congress*, Barcelona, Spain, 2002, University of Cambridge Technical Report CUED/F-INFENG/TR.398. Available at http://www.eng.cam.ac.uk/˜gv.

[11] F. Kelly, “Charging and rate control for elastic traffic,” *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[12] D. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

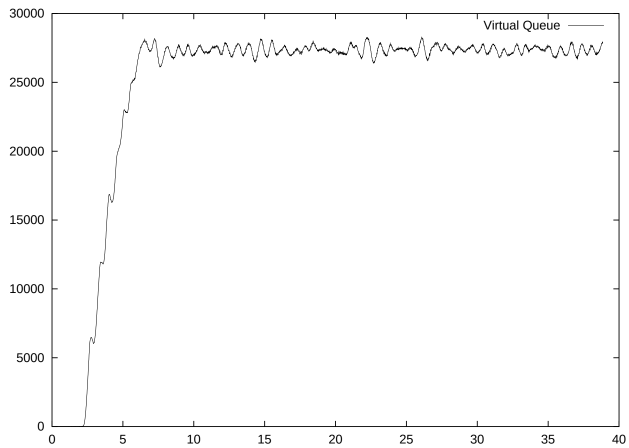

Fig. 4. E-RED: Virtual Queue Length.

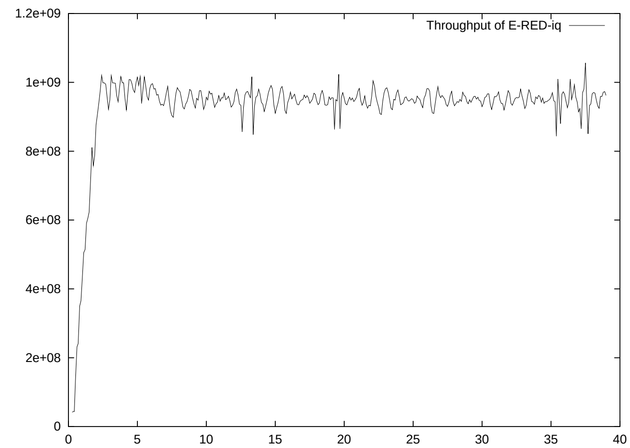

Fig. 6. E-RED: Throughput.

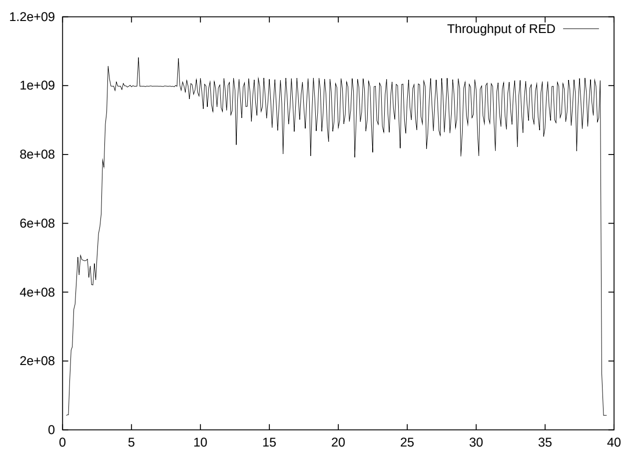

Fig. 5. RED: Throughput.

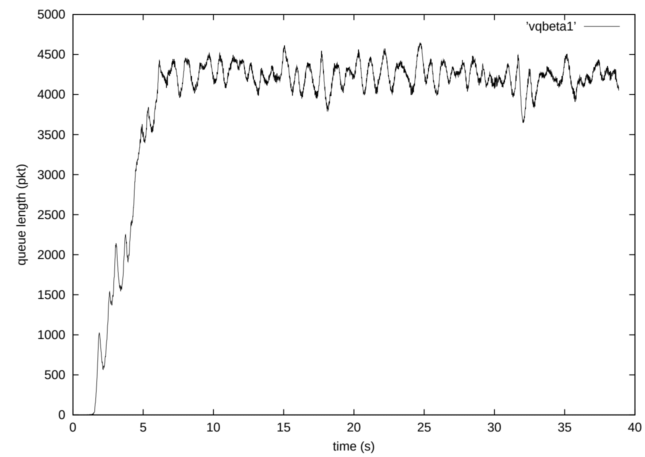

Fig. 7. E-RED: Virtual Queue, $\xi = 1$

[13] R. Johari and D. Tan, “End-to-end congestion control for the Internet: Delays and stability,” *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 818–832, December 2001.

[14] G. Vinnicombe, “On the stability of end-to-end congestion control for the Internet,” 2001, University of Cambridge Technical Report CUED/F-INFENG/TR.398. Available at http://www.eng.cam.ac.uk/˜gv.

[15] G. Vinnicombe, “Robust congestion control for the Internet,” 2002, University of Cambridge Technical Report. Available at http://www.eng.cam.ac.uk/˜gv.

[16] S. Kunniyur and R. Srikant, “A time-scale decomposition approach to adaptive ECN marking,” *IEEE Transactions on Automatic Control*, vol. 47, no. 6, pp. 882–894, June 2002.

[17] F. Kelly, ”Fairness and stability of end-to-end congestion control,” *European Journal of Control*, vol. 9, pp. 149–165, 2003

[18] F. Paganini, J. Doyle, and S. Low, “Scalable laws for stable network congestion control,” in *Proc. IEEE Conference on Decision and Control*, December 2001.

[19] F. Kelly, “Models for a self-managed Internet,” *Philosophical Transactions of the Royal Society*, vol. A358, pp. 2335–2348, 2000.

[20] S. Deb and R. Srikant, ”Queue and rate based models of congestion control,” Technical Report, 2003.

[21] R. Gibbens and F. Kelly, “Resource pricing and the evolution of congestion control,” *Automatica*, vol. 35, pp. 1969–1985, 1999.

[22] 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,” in *Proc. ACM Sigcomm*, Stockholm, Sweden, September 2000.

[23] S. Kunniyur and R. Srikant, “End-to-end congestion control: utility functions, random losses and ECN marks,” in *Proc. IEEE Infocom*, Tel Aviv, Israel, March 2000.

[24] S. Liu, T. Başar and R. Srikant, ”The general primal-dual algorithms and the E-RED algorithm,” Technical Report, 2003.

[25] S. Low, F. Paganini, J. Wang, S. Adlakha and J. Doyle, ”Dynamics of TCP/RED and a Scalable Control,” in *Proc. IEEE Infocom*, New York, NY, June 2002

[26] C. A. Desoer and Y. T. Wang, “On the generalized Nyquist stability criterion,” *IEEE Transactions on Automatic Control*, vol. 25, pp. 187–196, 1980.

VI. Appendix

In this appendix, we provide proofs for Theorem III.1 and Theorem IV.1. First we prove Theorem III.1, and toward this end, we state the following lemma:

Lemma VI.1. *As x is varied from $-\infty$ to ∞, $f(x) := \dfrac{\theta e^{-jx}}{jx(jx+\theta)}$ crosses the real line at a point to the right of -1 for all $\theta > 0$.*

Proof. Suppose $f(x)$ crosses the real line at $-b$ when $x = x_0$. Then we just need to prove that $b < 1$. We have:

$$\tan x_0 = \frac{\theta}{x_0}\,.$$

So

$$x_0 \tan x_0 = \theta\,,$$

$$b = \frac{\theta}{x_0\sqrt{x_0^2+\theta^2}} = \frac{\cos x_0}{x_0^2} x_0 \tan x_0 = \frac{\sin x_0}{x_0} < 1\,.$$

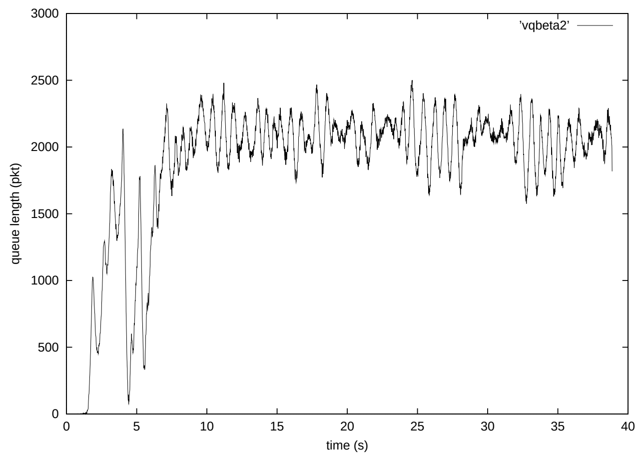

Fig. 8. E-RED: Virtual Queue, $\xi = 2$

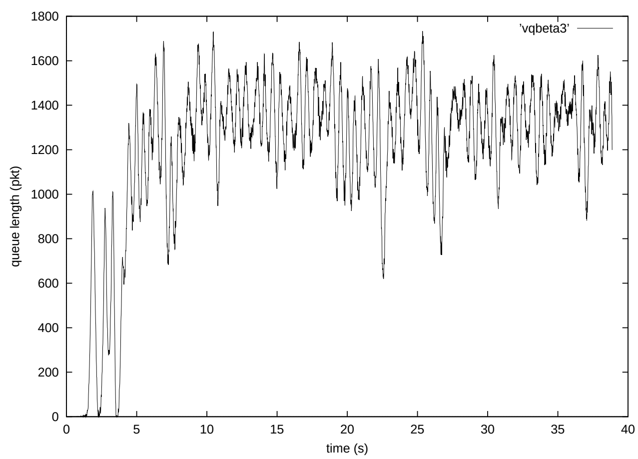

Fig. 9. E-RED: Virtual Queue, $\xi = 3$

$\square$

With Lemma VI.1 at hand, we can now prove Theorem III.1:

Proof. of Theorem III.1: 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

$$\begin{aligned} L(s) &= \text{diag}\Big(\frac{T_r\tilde{a}_r\hat{x}_r}{\hat{q}_r}\frac{\theta_r e^{-sT_r}}{sT_r(sT_r+\theta_r)}\Big)A^T(-s) \\ &\quad \times\text{diag}\Big(\frac{\beta_l\hat{p}_l}{c_l}\Big)A(s)\,. \end{aligned} \tag{48}$$

Since the open-loop system is stable, by the generalized Nyquist Criterion [26], 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

$$\hat{L}(j\omega) = \text{diag}\Big(\frac{\theta_r e^{-j\omega T_r}}{j\omega T_r(j\omega T_r+\theta_r)}\Big)\hat{A}^T(-j\omega)\hat{A}(j\omega)\,,$$

where

$$\hat{A}(j\omega) = \text{diag}\Big(\sqrt{\frac{\beta_l\hat{p}_l}{c_l}}\Big)A(j\omega)\text{diag}\Big(\sqrt{\frac{T_r\tilde{a}_r\hat{x}_r}{\hat{q}_r}}\Big)\,.$$

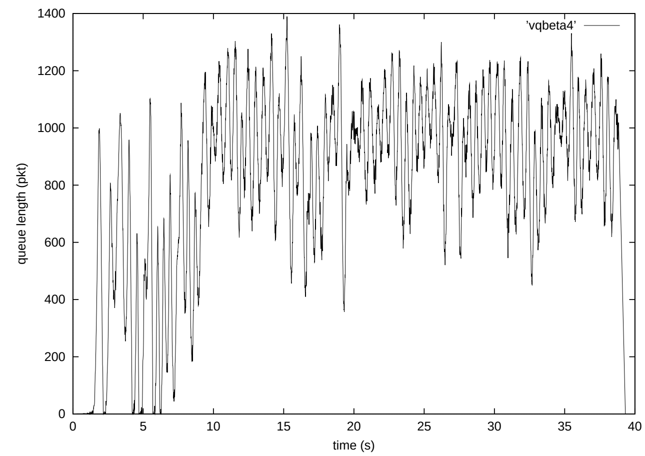

Fig. 10. E-RED: Virtual Queue, $\xi = 4$

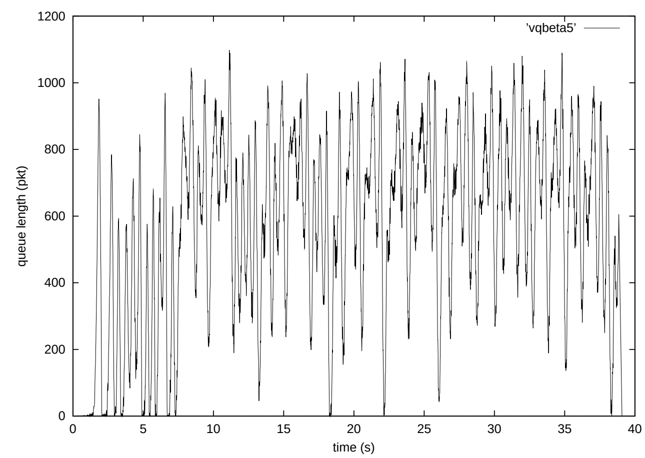

Fig. 11. E-RED: Virtual Queue, $\xi = 5$

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

$$\begin{aligned} \sigma^2(\hat{A}(j\omega)) &= \rho(\hat{A}^T(-j\omega)\hat{A}(j\omega)) \\ &= \rho(\text{diag}(\tfrac{\tilde{a}_rT_r\hat{x}_r}{\hat{q}_r})A^T(-j\omega)\text{diag}(\tfrac{\beta_l\hat{p}_l}{c_l})A(j\omega)) \\ &\leq \|\text{diag}(\tfrac{a_rT_r}{\hat{q}_r})A^T(-j\omega)\text{diag}(\beta_l\hat{p}_l)\| \\ &\quad \times\|\text{diag}(\tfrac{1}{\hat{y}_l})A(j\omega)\text{diag}(\hat{x}_r)\| \\ &\leq 1\times 1 = 1\,. \end{aligned}$$

The last inequality above uses the facts that

$$\sum_{r:l\in r}\hat{x}_r = \hat{y}_l = c_l, \forall l\in L\,,$$

and

$$\beta_l < \frac{1}{a_rT_r}, \forall r: l\in r \Longrightarrow \sum_{l\in r}\hat{p}_l\beta_l \leq \frac{\hat{q}_r}{a_rT_r}, \forall r: l\in r\,,$$

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

$$\lambda\in\text{Co}(0\cup\{\frac{\theta_r e^{-j\omega T_r}}{j\omega T_r(j\omega T_r+\theta_r)}\})\,.$$

(Here the reasoning is similar to that in [10].) From Lemma VI.1, $\left(\frac{\theta_r e^{-j\omega T_r}}{j\omega T_r(j\omega T_r+\theta_r)}\right)$ always crosses the real axis to the right of the point -1. So the eigenloci of $L(j\omega)$ do not encircle -1; thus by generalized Nyquist Criterion [26], the closed-loop system is stable. $\square$

We now move on to the proof of Theorem IV.1. First we have the following lemma.

Lemma VI.2. *As x is varied from $-\infty$ to ∞,*

$$f(x) := \frac{1}{\eta}\frac{mne^{-jx}}{jx(jx+m)(jx+n)}$$

crosses real line at a point to the right of -1 for all $m,n>0$, where

$$\eta = \max(1+\frac{1}{n}, \frac{2}{\pi}+\frac{4}{\pi n})$$

Proof. Suppose $f(x)$ crosses the left half of the real line at $-b/\eta$ when $x=x_0$. Then we just need to show that $b<\eta$. We have:

$$-\frac{\pi}{2} - x_0 - \arctan\frac{x_0}{m} - \arctan\frac{x_0}{n} = -\pi - 2k\pi\,.$$

or equivalently

$$x_0 + \alpha + \beta = \frac{\pi}{2} + 2k\pi\,,$$

where $\alpha = \arctan\frac{x_0}{m} \in (-\pi/2,\pi/2)$, $\beta = \arctan\frac{x_0}{n} \in (-\pi/2,\pi/2)$, and k is an integer.

If $k\neq 0$, then $|x_0|>\pi/2$, and

$$b = |\eta f(x_0)| \le \frac{1}{|x_0|} < \frac{2}{\pi} < \eta$$

If $k=0$, then x_0, α, β are all positive, and we have

$$\begin{aligned} b = |\eta f(x_0)| &= \frac{mn}{x_0\sqrt{x_0^2+m^2}\sqrt{x_0^2+n^2}} = \frac{\cos\alpha\cos\beta}{x_0} \\ &= \frac{\cos(\alpha+\beta)+\cos(\alpha-\beta)}{2x_0} = \frac{\sin x_0 + \sin(x_0+2\beta)}{2x_0} \end{aligned}$$

We know that $\beta \le \tan\beta$. If $x_0 + 2\tan\beta \le \pi/2$, we have

$$\begin{aligned} b &\le \frac{\sin x_0 + \sin(x_0+2\tan\beta)}{2x_0} = \frac{\sin x_0 + \sin(x_0+\frac{2x_0}{n})}{2x_0} \\ &< \frac{x_0 + x_0 + \frac{2x_0}{n}}{2x_0} < 1+\frac{1}{n} \le \eta\,. \end{aligned}$$

Else, we have $x_0 + 2\tan\beta = x_0(1+2/n) > \pi/2$, which leads to

$$x_0 > \frac{\pi}{2}\frac{1}{1+\frac{2}{n}}\,.$$

Then, we have

$$b \le \frac{1+1}{2x_0} < \frac{2}{\pi}(1+\frac{2}{n}) = \frac{2}{\pi} + \frac{4}{\pi n} \le \eta\,.$$

So $b<\eta$ for any positive m,n. $\square$

Equipped with Lemma VI.2, we can now prove Theorem IV.1.

Proof. of Theorem IV.1: Define $\tilde{a}_r$ as in the proof of Theorem III.1. Then, from (41), we have

$$\begin{aligned} L(s) &= \text{diag}(\frac{T_r\tilde{a}_r\hat{x}_r}{\hat{q}_r}\frac{\theta_r e^{-sT_r}}{sT_r(sT_r+\theta_r)})A^T(-s) \\ &\quad \times\text{diag}(\frac{\beta_l\hat{p}_l}{c_l}\frac{\kappa_l}{s+\kappa_l})A(s)\,. \end{aligned} \tag{49}$$

Since the open-loop system is stable, by the generalized Nyquist Criterion [26], 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 to those of

$$\begin{aligned} \hat{L}(j\omega) &= \text{diag}(\frac{1}{\eta_{aq}}\frac{\theta_r e^{-j\omega T_r}}{j\omega T_r(j\omega T_r+\theta_r)}) \\ &\quad \times\hat{A}^T(-j\omega)\text{diag}(\frac{\kappa_l}{s+\kappa_l})\hat{A}(j\omega)\,, \end{aligned}$$

where η_{aq} is defined in (44) and

$$\hat{A}(j\omega) = \text{diag}(\sqrt{\frac{\beta_l\hat{p}_l}{c_l}})A(j\omega)\text{diag}(\sqrt{\frac{T_r\hat{a}_r\hat{x}_r\eta_{aq}}{\hat{q}_r}}).$$

Given the condition in (43), we know, as in the proof of Theorem III.1, that we still have

$$\begin{aligned} \sigma^2(\hat{A}(j\omega)) &\le \|\text{diag}(\tfrac{1}{\hat{q}_r})A^T(-j\omega)\text{diag}(\hat{p}_l)\| \\ &\quad \times\|\text{diag}(\tfrac{1}{c_l})A(j\omega)\text{diag}(x_r)\| \le 1 \end{aligned}$$

since 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 there exists an eigenvector v, $\|v\|=1$, such that

$$\begin{aligned} \lambda v &= \text{diag}(\frac{\kappa_l}{j\omega+\kappa_l})\hat{A}(j\omega) \\ &\quad \times\text{diag}(\frac{1}{\eta_{aq}}\frac{\theta_r e^{-j\omega T_i}}{j\omega T_i(j\omega T_i+\theta_r)})\hat{A}^T(-j\omega)v\,. \end{aligned}$$

So we have

$$\lambda = \frac{v^*\hat{A}(j\omega)\text{diag}(\frac{1}{\eta_{aq}}\frac{\theta_r e^{-j\omega T_i}}{j\omega T_i(j\omega T_i+\theta_r)})\hat{A}^T(-j\omega)v}{v^*\text{diag}(j\frac{\omega}{\kappa_l}+1)v}\,.$$

The denominator is a complex scalar with the real part being 1, so we can write the denominator as $1+j\omega/\tilde{\kappa}$, where $\tilde{\kappa}$ is a real positive scalar, which depends on v and is no smaller than $\min_{l\in L}\kappa_l$. Dividing each element of the diagonal matrix in the numerator by this complex scalar, we have:

$$\begin{aligned} \lambda &= v^*\hat{A}(j\omega)\text{diag}(\tfrac{1}{\eta_{aq}}\tfrac{\theta_r\tilde{\kappa}T_r e^{-j\omega T_i}}{j\omega T_i(j\omega T_i+\theta_r)(j\omega T_i+\tilde{\kappa}T_r)}) \\ &\quad \times\hat{A}^T(-j\omega)v\,, \end{aligned}$$

Define

$$\tilde{\eta}_r = \max(1+\frac{1}{\tilde{\kappa}T_r}, \frac{2}{\pi}+\frac{4}{\pi\tilde{\kappa}T_r})\,.$$

Then, $\tilde{\eta}_r \le \eta_{aq}$, since $\tilde{\kappa}T_r \ge M$. And also since $\|\hat{A}^T(-j\omega)v\|^2 \le 1$, we have

$$\lambda \in \text{Co}(0 \cup \{\frac{\theta_r\tilde{\kappa}T_r e^{-j\omega T_r}}{\tilde{\eta}_r j\omega T_r(j\omega T_r+\theta_r)(j\omega T_r+\tilde{\kappa}T_r)}\})\,.$$

From Lemma VI.2, $\left(\frac{\theta_r\tilde{\kappa}T_r e^{-j\omega T_r}}{\tilde{\eta}_r j\omega T_r(j\omega T_r+\theta_r)(j\omega T_r+\tilde{\kappa}T_r)}\right)$ always crosses the real axis to the right of the point -1. So the eigenloci of $L(j\omega)$ do not encircle -1; thus by generalized Nyquist Criterion [26], the closed-loop system is stable. $\square$