Fairness and stability of end-to-end congestion control *

Frank Kelly †

Abstract

In recent years the Internet has attracted the attention of many theoreticians, eager to understand the remarkable success of this diverse and complex artefact. A central element of the design philosophy that shaped the Internet is the end-to-end argument, and a key illustration of the argument is provided by the congestion avoidance algorithm of the Transmission Control Protocol (TCP). Why does this algorithm work so well? How might, or should, it evolve in the future? In this paper we outline some of the mathematical models that have been developed to help address these questions.

We review the equilibrium and dynamic properties of primal and dual algorithms, concentrating upon fairness, delay instability and stochastic instability. Primal algorithms broadly correspond with congestion control mechanisms where noisy feedback from the network is averaged at endpoints, using increase and decrease rules generalizing those of TCP. Vinnicombe has shown that delay instability is characterized in terms of the increase rule; Ott has shown that stochastic instability is primarily influenced by the decrease rule. The need to control both forms of instability places constraints on possible variants of TCP, and on attempts to remove TCP's round-trip time bias.

Dual algorithms broadly correspond with congestion control mechanisms where averaging at resources precedes the feedback of more explicit information to endpoints, and may be especially appropriate where round-trip times are short, as in ad-hoc networks. Previous work has concentrated on delay-based dual algorithms, which find fairness and stability difficult to reconcile. We describe a family of fair dual algorithms, with attractive stability properties.

*Paper prepared for a plenary address to the European Control Conference, Cambridge, September 2003, and available from www.statslab.cam.ac.uk . This version is a corrected version of a paper that appeared in the European Journal of Control 2003; 9: 159-176.

†Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge CB3 0WB UK. Email: f.p.kelly@statslab.cam.ac.uk

Keywords: Internet, TCP, proportional fairness, Ornstein-Uhlenbeck process

1 Introduction

A central element of the design philosophy that shaped the Internet [8, 46] is the end-to-end argument [40], summarized as follows [4]: "*An end-to-end protocol design should not rely on the maintenance of state (i.e., information about the state of the end-to-end communication) inside the network. Such state should be maintained only in the endpoints, in such a way that the state can only be destroyed when the endpoint itself breaks.*" Intelligence and control is end-to-end rather than hidden in the network. The resulting interconnection of communication links is sometimes termed a dumb network, to emphasise a contrast with the earlier telephony infrastructure where a smart network connects endpoints (telephones) that have little responsibility for control. A dumb network allows new applications to be designed, prototyped and deployed without requiring changes to the underlying network, and has resulted in an extraordinary flowering of innovation. It also places a considerable responsibility for cooperative behaviour on endpoints.

A key illustration of the end-to-end argument is TCP, the transmission control protocol of the Internet, and its congestion avoidance algorithm, due to Jacobson [15]. The rate at which packets enter the network is controlled by TCP, implemented as software on the computers (the endpoints) that are the source and destination of the data. The general approach is as follows. When a link within the network becomes overloaded, one or more packets are lost; loss of a packet is taken as an indication of congestion, the destination informs the source, and the source slows down. The TCP then gradually increases its sending rate until it again receives an indication of congestion. This cycle of increase and decrease serves to discover and utilize available bandwidth, and to share it between flows.

Jacobson's algorithm has been outstandingly successful, as the Internet has evolved from a small-scale research network to today's interconnection of tens of millions of endpoints and links. This in itself is a striking observation. Each of a large but indeterminate number of flows is controlled by a feedback loop which can know only of its own experience of congestion and of its own feedback delay. A flow does not know how many other flows are sharing a link on its path, or even how many links are on its path. The links vary in capacity by many orders of magnitude, as do the propagation delays that are a consequence of geographical diversity and the finite speed of light. That end-to-end congestion control can have achieved so much, in such a rapidly growing and heterogeneous network, is remarkable.

However there are several interrelated developments that pose a challenge. Buffering at queues within the Internet has been important since the early days of store-and-forward communication networks [22], to smooth statistical fluctuations and, it is commonly believed, to help stabilize the network. But the huge capacity of tomorrow's links, together with the desire to carry delay-sensitive traffic, may cause an evolution towards a network with much smaller queueing delays. Is end-to-end congestion control feasible in such a network, and, if so, how should TCP evolve?

This paper explores possible answers to this question, and outlines some of the important insights that have been obtained from simplified mathematical models of congestion control. The selection of material has been primarily influenced by recent experiments with new, incrementably deployable, TCPs [10, 21, 38], the related modelling advances [35, 37, 43], and by proposals for variants where more explicit information is returned to endpoints [3, 6, 17, 20, 26].

The organization of the paper is as follows. In the next section we describe a tractable mathematical model of a network, following the development in [19]. An optimization framework leads naturally to two classes of rate control algorithm: primal algorithms, which broadly correspond with end-to-end congestion control mechanisms where noisy feedback from the network is averaged at endpoints; and dual algorithms, which broadly correspond with averaging at resources prior to the feedback of more explicit information to endpoints. Both types of algorithm reach an equilibrium which is proportionally fair. Fairness is a central issue in networks where responsibility for cooperation is devolved to endpoints, and various definitions of fairness have been suggested as the basis for behavioural norms. Weighted proportional fairness is a criterion with appealing properties from either an optimization, a game-theoretic or an economic viewpoint.

In Section 3 we outline how TCP's congestion avoidance algorithm can be interpreted within the optimization framework, and discuss the round-trip time bias of TCP. Through a simple example we describe the potentially distorting effect of this bias on network structure. It has not proved easy to remove round-trip time bias: variations to TCP's algorithm designed to shift the equilibrium point can easily have the unintended side-effect of destabilizing the equilibrium.

The stability of the equilibrium point may be compromised by two interacting effects [16]: *delay instability*, due to the combination of high gains and time delays; and *stochastic instability*, due to inherent randomness. In Section 4, we describe the important recent work of Vinnicombe [43] on delay instability, and on a potentially stable and scalable variant of TCP. In Section 5 we discuss stochastic instability, including Ott's scale-invariance property [35]. Delay instability is characterized in terms of TCP's increase rule, while stochastic instability is primarily influenced by the decrease rule. We use this separation to interpret a proposed TCP variant [20] that aims to remove round-trip time bias while simultaneously controlling both forms of instability.

In Section 6 we discuss dual algorithms, starting with the delay-based scheme of Paganini *et al.* [37]. This section describes a variety of other dual algorithms, including a family of fair dual algorithms with attractive stability properties. Fair dual algorithms have promise in circumstances where more explicit feedback is available and either: a link can estimate the average round-trip time of packets passing through it, as in the proposal of Katabi *et al.* [17]; or propagation delays are not large, as in home networks [3] or ad hoc networks [6]. Section 7 concludes.

2 Fairness

How should available bandwidth be shared between competing users of a network? In this Section we describe a mathematical framework for rate control which allows us to reconcile potentially conflicting notions of fairness and efficiency.

Consider a network with a set J of *resources*. Let a *route* r be a non-empty subset of J, and write $j \in r$ to indicate that route r passes through resource j. Let R be the set of possible routes. Set $A_{jr} = 1$ if $j \in r$, so that resource j lies on route r, and set $A_{jr} = 0$ otherwise. This defines a $0-1$ incidence matrix $A = (A_{jr}, j \in J, r \in R)$.

Consider the system of differential equations

$$\frac{d}{dt} x_r(t) = \kappa_r \left(w_r - x_r(t) \sum_{j \in r} \mu_j(t) \right) \tag{1}$$

where

$$\mu_j(t) = p_j \left(\sum_{s : j \in s} x_s(t) \right). \tag{2}$$

(Here and throughout we assume that, unless otherwise specified, r ranges over the set R and j ranges over the set J.) Assume that $w_r, \kappa_r > 0$, and that the function $p_j(y)$, $y \geq 0$, is a non-negative, continuous, strictly increasing function of y.

We may motivate the relations (1-2) as follows. Suppose that resource j marks a proportion $p_j(y)$ of packets with a feedback signal when the total flow through resource j is y, and that each feedback signal is viewed as a congestion indication requiring some reduction in the flow x_r. Then equation (1) corresponds to a rate control algorithm for the flow on route r that comprises two components: a steady increase at rate proportional to w_r, and a steady decrease at rate proportional to the stream of congestion indication signals received. Following [19], we shall call the system (1-2) the *primal algorithm*.

Define the *dual algorithm*

$$\frac{d}{dt}\mu_j(t) = \kappa_j\left(\sum_{r:j\in r} x_r(t) - q_j(\mu_j(t))\right) \tag{3}$$

where

$$x_r(t) = \frac{w_r}{\sum_{k\in r}\mu_k(t)}, \tag{4}$$

$\kappa_j > 0$, and $q_j(\eta) = p_j^{-1}(\eta)$ is a strictly increasing function. Again the algorithm has a straightforward interpretation. If we view $p_j(y)$ as a load-dependent price at resource j, then $q_j(\eta)$ is the flow through resource j which generates a price at resource j of η. Thus an economist would describe the right hand side of equation (3) as the vector of excess demand at prices $\mu = (\mu_j(t), j \in J)$, and would recognise equations (3-4) as a tatonnement process [41] by which prices adjust according to supply and demand.

Both the primal algorithm (1-2) and the dual algorithm (3-4) have a unique stable point (x, μ), which is the same for both algorithms, and to which all trajectories of either algorithm converge. At the stable point

$$x_r = \frac{w_r}{\sum_{j\in r}\mu_j}. \tag{5}$$

This equation has a simple interpretation in terms of a charge per unit flow: the variable μ_j is the *shadow price* per unit of flow through resource j. The allocation $x = (x_r, r \in R)$ given by equation (5) has an interpretation in terms of a *weighted proportional fairness* criterion [7, 18], satisfying certain natural assumptions from cooperative game theory as to what constitutes fairness [31, 34]. The weight w_r is the aggregate shadow price of the flow x_r, and the vector of weights $(w_r, r \in R)$ is proportional to the share of scarce resources obtained by different flows.

Suppose that route r is associated with a *user*, representing a higher level entity served by the flow on route r. Suppose if a rate $x_r > 0$ is allocated to the flow on route r then this has *utility* $U_r(x_r)$ to the user. Assume that the utility $U_r(x_r)$ is an increasing, strictly concave function of x_r over the range $x_r > 0$. To simplify the statement of results, we shall assume further that $U_r(x_r)$ is continuously differentiable, with $U_r'(x_r) \to \infty$ as $x_r \downarrow 0$ and $U_r'(x_r) \to 0$ as $x_r \uparrow \infty$. Let $C_j(y)$ be defined by

$$C_j(y) = \int_0^y p_j(z)\, dz.$$

From our assumptions on $p_j(y)$, the function $C_j(y)$ is strictly convex. We might view $C_j(y)$ as a form of cost incurred at resource j, that increases more rapidly as the resource becomes more heavily loaded.

Next suppose that user r is able to monitor its rate $x_r(t)$ continuously, and to vary smoothly the parameter $w_r(t)$ so as to satisfy

$$w_r(t) = x_r(t) U_r'(x_r(t)) : \tag{6}$$

this would correspond to a user who observes a charge per unit flow of $\lambda_r = w_r(t)/x_r(t)$, and chooses $w_r = w_r(t)$ to solve the optimization problem

$$\begin{array}{ll} \text{maximize} & U_r\left(\dfrac{w_r}{\lambda_r}\right) - w_r \\ \text{over} & w_r \geq 0. \end{array}$$

This in turn corresponds to price-taking behaviour on the part of user r, who does not anticipate the impact of its own choice of $w_r(t)$ on the system.

If w_r is replaced in (1) and (4) by the time-varying form (6) then both the resulting algorithms have a unique stable point, which is the same for both algorithms, and to which all trajectories of either algorithm converge [19]. The stable point is proportionally fair, i.e. it is of the form (5), with $w_r = w_r(\infty)$. The stable point maximizes the function

$$\mathcal{U}(x) = \sum_{r \in R} U_r(x_r) - \sum_{j \in J} C_j\left(\sum_{s : j \in s} x_s\right). \tag{7}$$

Thus if each user is able to choose its own weight, $w_r(t)$, and does this so as to optimize its own utility less payment, then either algorithm will converge to the rate allocation x maximizing the net utility (7). The parameter $w_r(t)$ may be viewed as the *willingness to pay* of user r. Alternatively, in a network of co-operative users, $w_r(t)$ may be viewed as a time-varying weight chosen by user r with resource, but no monetary, implications. The distinction will not be important in this paper.

Define the *demand function* $D_r(\lambda_r) = (U_r')^{-1}(\lambda_r)$, a continuous, strictly decreasing function. Then, at the stable point, $x_r = D_r(\sum_{j \in r} \mu_j)$. Mo and Walrand [33] have introduced a class of utility functions

$$U_r(x_r) = w_r \frac{x_r^{1-\alpha}}{1-\alpha}$$

($= w_r \log x_r$ if $\alpha = 1$) with derived demand functions

$$D_r(\lambda_r) = \left(\frac{w_r}{\lambda_r}\right)^{1/\alpha}. \tag{8}$$

Term the resulting allocations $(x_r = D_r(\sum_{j\in r}\mu_j), r \in R)$ *weighted α-fair*: if $w_r = 1, r \in R$, the cases $\alpha \to 0$, $\alpha = 1$ and $\alpha \to \infty$ correspond respectively to an allocation which achieves maximum throughput, is proportionally fair or is max-min fair, and we shall refer to a weighted version of the $\alpha = 2$ case in the next Section on TCP. The interpretation of the weights $(w_r, r \in R)$ as proportional to the share of scarce resources obtained by different flows is lost if $\alpha \neq 1$.

An attractive case of the dual algorithm, considered by Low and Lapsley [25], sets

$$q_j(\eta) = C_j I[\eta > 0],$$

where the scalar C_j is the capacity of link j. Although this choice of $q_j(.)$ violates our simplifying assumption that $q_j(.)$ is strictly increasing, there is again an identification of equilibrium points with maxima of the function (7), where now $C_j(y) = 0$ if $y \leq C_j$, and $C_j(y) = \infty$ otherwise. The identification fixes the vector x uniquely, but now the equilibrium vector μ may not be unique. A sufficient condition for uniqueness of μ is that the incidence matrix A have full row rank.

None of the above results depend upon the gains $\kappa_r, r \in R, \kappa_j, j \in J$, which could indeed have been fairly general positive functions of the state $(x(t), \mu(t))$. The choice of gains is, of course, constrained by stability conditions, the subject of Sections 4, 5 and 6. The primal algorithm (1-2) is a simple but crude caricature of an end-to-end congestion control mechanism. In the next Section we refine the caricature, and study the fairness of TCP.

3 Modelling TCP

Packets transferred by TCP across the Internet are acknowledged. If a packet is lost then the destination detects this; the detection of loss prompts the resending of the lost packet, and is interpreted as an indication of congestion. Using lost packets to signal congestion has obvious drawbacks. First, it is wasteful, since a dropped packet may have already consumed resources at earlier stages of its route and needs to be resent. Second, there are limits upon the quality that can be provided by a network if damage to packets (*e.g.* loss or delay) is an essential part of the network's control mechanism. These considerations have led naturally to proposals for the introduction of congestion marking, whereby a packet that encounters incipient congestion has a bit set in its header. The procedure is called *Explicit Congestion Notification*, or ECN [9]. Endpoints detecting ECN marks should respond by reducing their transmission rates. The result will be a system that can share resources without recourse to dropped packets, except in periods of exceptionally heavy use. ECN has now been made a "Proposed Standard" by the Internet Engineering Task Force (IETF), the body concerned with the evolution of the Internet architecture [39]. In this paper our models will make little distinction between whether a dropped or a marked packet is used to indicate congestion: in either case a packet crossing the Internet generates a single bit of information concerning congestion along its route.

Jacobson's algorithm [15] is *self-clocking*: the sender uses the acknowledgement from the receiver to prompt a step forward. The source maintains a window of sent but not yet acknowledged packets; the rate x and the window size `cwnd` satisfy the approximate relation $\texttt{cwnd} = xT$. We shall outline a general class of increase and decrease rules (first considered in [2, 32, 35]) for the window – Jacobson's algorithm will be a special case. We suppose the congestion window is incremented by $a\ \texttt{cwnd}^n$ for each positive acknowledgement, and decremented by $b\ \texttt{cwnd}^m$ for each congestion indication, where $n < m$.

The expected change in the congestion window `cwnd` per update step is approximately

$$a\ (xT)^n\ (1-p) - b\ (xT)^m\ p$$

where p is the probability of congestion indication at the update step. Since the time between update steps is about $T/\texttt{cwnd} = 1/x$, the expected change in the rate x per unit time is approximately

$$\frac{x}{T}\Big(a\ (xT)^n\ (1-p) - b\ (xT)^m\ p\Big).$$

Motivated by this calculation, we model the algorithm by the system of differential equations

$$\frac{d}{dt}\,x_r(t) = \frac{x_r(t)}{T_r}\ \cdot\Big(a_r\ (x_r(t)T_r)^n\ (1-\lambda_r(t)) - b_r\ (x_r(t)T_r)^m\ \lambda_r(t)\Big), \tag{9}$$

where

$$\lambda_r(t) = 1 - \prod_{j\in r}(1-\mu_j(t)), \tag{10}$$

$\mu_j(t)$ is again given by equation (2), and T_r is the round-trip time for the connection of user r. We again view $p_j(y)$ as the probability a packet collects a congestion indication signal at resource j when the total load through

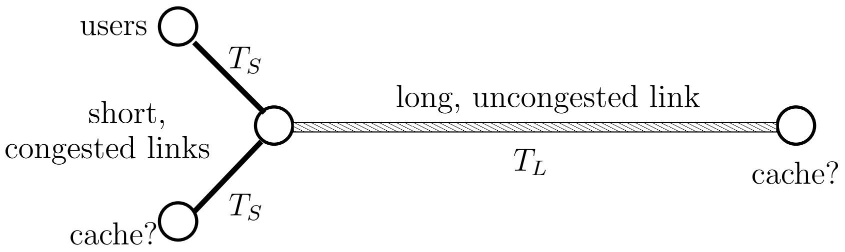

Figure 1: Round-trip time bias and cache location. Suppose the round-trip time over each of the short links is T_S, and the round-trip time over the long link is T_L. Better TCP throughput will be received from the nearer cache if $T_L > (2\sqrt{3} - 1)T_S$, even though this cache uses two congested links.

resource j is y. Equation (10) models the situation where congestion indication is provided by a dropped packet or a single bit, and corresponds to an approximation that packet drops or marks at different resources are independent.

The system (2), (9) and (10) has a unique equilibrium point, to which all trajectories converge (Appendix I). The equilibrium point has the form

$$x_r = \frac{1}{T_r}\left(\frac{a_r}{b_r} \cdot \frac{1-\lambda_r}{\lambda_r}\right)^{1/\alpha}, \tag{11}$$

where $\alpha = m - n$. The case $a = 1, b = \frac{1}{2}, m = 1, n = -1$ corresponds to Jacobson's TCP; the case $a_r = M_r, b_r = 1/(2M_r), m = 1, n = -1$ corresponds to Crowcroft and Oechslin's MulTCP [7], where M_r is a parameter influencing the share of resources achieved by flow r. In either case, $\alpha = 2$: thus equation (11) recovers the inverse square root dependence on the probability of packet loss familiar from the literature on TCP [30]. And if λ_r is small enough that $\lambda_r \approx \sum_{j \in r} \mu_j$, then the equilibrium is approximately weighted α-fair, with weight $w_r = a_r/(b_r T_r^2)$. The weights, and the form (11), exhibit *round-trip time bias* [11]: for a given packet loss probability, λ_r, the flow on route r is inversely proportional to the round-trip time T_r. Several variants of TCP have been proposed to correct this bias against flows with larger values of T_r, some of which we will discuss later in Section 5.

A less than obvious consequence of round-trip time bias is illustrated in Figure 1. Suppose that a cache is to be placed at one of two locations. One location is connected to a large body of potential users via two short congested links, on each of which the packet loss probability is $p < \frac{1}{2}$ and the round-trip time is T_S. The other location is connected via a short congested link and a long uncongested link, over which there is no packet loss and the round-trip time is T_L. The flow rate achieved to a user from the two locations will be proportional to

$$\frac{1}{2T_S}\left(\frac{(1-p)^2}{1-(1-p)^2}\right)^{1/2}, \quad \frac{1}{T_S+T_L}\left(\frac{1-p}{p}\right)^{1/2}$$

respectively, from relation (11). It follows that better TCP throughput will be received from the nearer cache if $T_L > (2\sqrt{3}-1)T_S$, even though this cache makes *twice* the use of scarce resources.[1] Decisions on cache location, capacity expansion, the topology of overlay networks, are often based on such local, decentralized optimizations. Thus we should expect the round-trip time bias to have consequences for the efficiency of the evolving network structure, as well as for the short-term fairness of the rates achieved by competing flows. In particular, round-trip time bias will encourage underuse and underdevelopment of long links, and overuse and overdevelopment of short links, relative to an efficient network structure.

Next suppose that $\alpha = 1$, and that λ_r is small enough that $\lambda_r \approx \sum_{j\in r} \mu_j$. Then equation (11) becomes approximately the proportionally fair allocation (5), with weight $w_r = a_r/(b_r T_r)$. Might it be possible to design a variant of TCP with these features, so that the parameters a_r, b_r could compensate for the round-trip time bias, or more generally provide a straightforward control over the share of resources achieved by a flow? We return to this question, after looking at the influence of the increase and decrease rules upon stability.

4 Delay stability

In the last Section we have seen how the self-clocking feature of the algorithm can cause the round-trip time of a flow to have an important effect on the share of resources allocated to the flow. Next we consider how the differential equations of the last Section should be amended to model delayed feedback, following Vinnicombe [43].

4.1 Propagation delays

For each j, r such that $j \in r$ let T_{rj} be the propagation delay from the source of flow on route r to the resource j, and let T_{jr} be the return delay from

[1]In this simple example, a halving of distance can compensate for as much as a quadrupling of packet loss probability. The discussion assumes that TCP's performance is primarily determined by its congestion avoidance phase, rather than its initial slow start phase. There are clearly advantages in using nearby caches for the transfer of short files.

resource j to the source. In the current Internet

$$T_{rj} + T_{jr} = T_r \quad j \in r, r \in R, \tag{12}$$

where T_r is the round-trip propagation delay on route r: the identity (12) is a direct consequence of the end-to-end nature of the signalling mechanism, whereby congestion on a route is perceived by the destination, which then informs the source. In the current Internet the total round-trip time is composed of not just propagation delays, but also queueing delays at resources and delays at endpoints. But these additional delays may not be fundamental to the end-to-end model: higher capacities reduce queueing delays, and ECN marking and faster processors will further reduce delays at endpoints. An aim of this paper is to explore the end-to-end model when T_r is reduced to the inescapable minimum, the propagation delay. We shall return to discuss queueing delays later, but until then we assume that delays other than propagation delays are negligible.

Consider, then, how the system (2), (9) and (10) should be amended to include the effect of delayed feedback. The argument leading to equation (9) used the approximation[2] that the time between update steps is about $T/\texttt{cwnd} = 1/x(t)$. But the time between update steps at the source for route r is determined by the flow that left the source a time T_r previously, and hence the time between update steps is about $1/x(t - T_r)$, giving instead

$$\frac{d}{dt} x_r(t) = \frac{x_r(t - T_r)}{T_r} \cdot \Big(a_r \, (x_r(t) T_r)^n \, (1 - \lambda_r(t)) - b_r \, (x_r(t) T_r)^m \, \lambda_r(t) \Big). \tag{13}$$

The feedback seen at the source for route r at time t is carried on a flow that passed through resource j a time T_{jr} previously: hence

$$\lambda_r(t) = 1 - \prod_{j \in r} (1 - \mu_j(t - T_{jr})). \tag{14}$$

Similarly the flow on route s that is seen at resource j at time t left the source for route s a time T_{sj} previously: hence

$$\mu_j(t) = p_j \left(\sum_{s: j \in s} x_s(t - T_{sj}) \right). \tag{15}$$

Given the round-trip times $T_r, r \in R$, the delays T_{rj}, T_{jr} do not affect the equilibrium point $(x(t), \lambda(t)) = (x, \lambda)$, at which equation (11) remains

[2] A more refined model of a window algorithm could be developed; the model discussed here corresponds more closely to a rate-paced version of such an algorithm.

satisfied. Next we analyze the local stability of the equilibrium point. Let $x_r(t) = x_r + u_r(t)$, $\lambda_r(t) = \lambda_r + (1-\lambda_r)\nu_r(t)$, and write $y_j = \sum_{s:j\in s} x_s$, $p_j = p_j(y_j), p'_j = p'_j(y_j)$. Then, linearizing the system (13-15) about (x, λ), and using the relation (11), we obtain the equations

$$T_r \frac{d}{dt} u_r(t) = -a_r\ (x_r T_r)^n (1-\lambda_r) \left(\alpha u_r(t) + \frac{x_r}{\lambda_r} \nu_r(t) \right) \tag{16}$$

where

$$\nu_r(t) = \sum_{j\in r} \frac{p'_j}{1-p_j} \sum_{s:j\in s} u_s(t - T_{sj} - T_{jr}). \tag{17}$$

Vinnicombe has shown ([42, 43], Appendix II) that the system (16-17) is stable if for $r \in R$

$$a_r\ (x_r T_r)^n \frac{1-\lambda_r}{\lambda_r} \sum_{j\in r} \frac{y_j p'_j}{1-p_j}\ <\ \frac{\pi}{2}. \tag{18}$$

Suppose next that

$$y p'_j(y) \leq \beta p_j(y), \quad j \in J, \tag{19}$$

a relation that will hold with equality if $p_j(y) = (y/C_j)^\beta$. Then

$$\begin{aligned} \frac{1-\lambda_r}{\lambda_r} \sum_{j\in r} \frac{y_j p'_j}{1-p_j}\ &\leq\ \frac{\beta}{\lambda_r} \sum_{j\in r} p_j \frac{1-\lambda_r}{1-p_j} \\ &= \beta \frac{\text{Prob(packet on route } r \text{ marked exactly once)}}{\text{Prob(packet on route } r \text{ marked at least once)}}\ \leq\ \beta, \end{aligned}$$

and hence a sufficient condition for stability is

$$a_r\ (x_r T_r)^n < \frac{\pi}{2\beta} \tag{20}$$

for $r \in R$, a simple decentralized condition expressed in terms of the window increase rule. Further, if $n = 0$, so that the increment upon a positive acknowledgement is just a constant, the condition becomes simpler still, and independent of the window size $x_r T_r$.

The condition (20) becomes tight when α is small, there is a single congested resource, and the routes using this resource all share the same round-trip time T – in this case $u_r(t) = x_r \sin(\pi t/2T)$ solves equations (16-17) (with $\alpha = 0$, $\beta = y_j p'_j/p_j$, and $a_r(x_r T_r)^n = \pi/2\beta$), an oscillatory solution with period $4T$.

Recall that Jacobson's algorithm corresponds to the choice $a_r = 1, n = -1$. Condition (18) or (20) then becomes a *lower* bound on the size of the congestion window $x_r T_r$. This may seem counter-intuitive, but note that small congestion windows may indicate a large number of flows through a congested resource, a more intuitively plausible cause of instability.

A current major concern for TCP is that it is slow to adapt when the window size is large, for example on long distance routes with large capacities [21, 38]. And indeed, for $n = -1$, condition (18) suggests that adaptation will be unnecessarily slow when the window $x_r T_r$ is large. Following Vinnicombe's work the natural suggestion is to let $n = 0$: but in an evolving network this would raise concerns over fairness between different forms of TCP. In [10, 21] there is proposed a variant of TCP where the increment upon a positive acknowledgement becomes a constant, rather than 1/`cwnd`, but only when the window size on a route exceeds a threshold: experiments reported in [21] suggest a substantial improvement for transfers over long distance routes with large capacities, with negligible impact on other traffic.

4.2 Queueing delays

Next we briefly consider how the analysis would be affected if queueing delays at resources could be substantial. Suppose that the round-trip time is

$$T_r = T_r^{\text{prop}} + \sum_{j \in r} Q_j(\mu_j) \quad r \in R \tag{21}$$

where T_r^{prop} is the round-trip propagation delay, and $Q_j(\mu_j)$ is the queueing delay at resource j when the loss probability there is μ_j. Suppose that $Q_j(\mu_j)$ is an increasing continuous function of $\mu_j \in [0,1]$. Then substitution from equations (10), (21) into equation (11), followed by substitution from this equation into equation (2), defines a continuous mapping from the compact convex set $\{\mu \in [0,1]^J\}$ into itself: hence, by the Brouwer fixed point theorem, there exists a solution μ, λ, x to the equations (2), (10-11), (21). It may not be unique: for the following discussion we fix on one solution and assume that it may be perturbed continuously, the generic case.

The first point to note is that T_r as well as λ_r acts to limit the equilibrium flow x_r, through equation (11). Thus $Q_j(\mu_j)$ as well as μ_j acts to limit the load on resource j, allowing generally lower packet loss probabilities at equilibrium than if queueing delays were negligible.

The second point concerns dynamic properties when $n = -1$, corresponding to current TCP. Suppose that capacities and buffers are such that queueing delays can be substantial: further, as a crude model of the current Internet, suppose that a resource either is not fully utilized, in which case the packet loss and queueing delay are near zero, or is fully utilized, in which case the packet loss is variable but the queueing delay is near constant, at the time taken for a packet to pass through a near full buffer. Then the round-trip time on a route will depend upon which resources along the route are fully utilized, but not sensitively upon the packet loss probabilities at these resources. The analysis leading to equation (18) will apply approximately: hence the presence of a full buffer at a resource will, by increasing the congestion windows of all flows through that resource, help stabilize the equilibrium point. [3]

In summary, within the current Internet, we should expect queueing delays at resources not only to allow lower loss probabilities at an equilibrium, but also to help stabilize an equilibrium. Conversely, if queueing delays within the Internet are to be reduced, then algorithms will be needed, such as the $n = 0$ variants of the last subsection, that maintain stability without the help of queueing delays.

---

[3] The deduction presented here is incorrect. It implicitly assumes the functions $p(.)$ satisfy equation (19) with approximate equality. But for the resource model considered in this paragraph, where $p(.)$ is the proportion of packets overflowing a large buffer, a more reasonable approximation is the form $p(y) = [y-C]^+/y$, introduced by S. Kunniyur and R. Srikant, End-to-end congestion control: utility functions, random losses and ECN marks, IEEE/ACM Transactions on Networking. I am very grateful to R. Srikant for pointing out the problem with this paragraph.

If

$$p_j(y) = [y - C_j]^+/y$$

then, using equation (11), we can rewrite the sufficient condition (18) as

$$b_r\ (x_r T_r)^m\ M_r < \frac{\pi}{2}$$

where $M_r = \sum_{j \in J} I[p_j > 0] A_{jr}$, the number of saturated resources on route r. For TCP, $b_r = 1/2, m = 1$, and so larger congestion windows will make it *harder* to satisfy the sufficient condition. (Throughout the paper we ignore the fact that a dropped packet is not seen at later resources on its route - see, for example, equation (2). This is likely to matter for the model of heavily saturated resources considered in the paragraph in question.)

In the current Internet, it seems plausible that queueing delays may help stabilize an equilibrium not by improving the delay stability of the differential equation models, but rather by keeping the congestion windows large enough and the packet loss probabilities low enough to avoid time-outs. If ECN were used to keep the load on a resource less than its capacity, then there are a wide range of possible functions $p(.)$ that could be realized, including functions $p(.)$ that satisfy equation (19) with approximate equality [20].

5 Stochastic stability

Variability about the equilibrium will be caused by two interacting effects: oscillations caused by the combination of high gains and time delays, and perturbations caused by the random nature of packet loss or packet marking. In Section 4.1 we analysed the first effect in isolation: in this section we analyse the second effect in isolation.

If packets on route r are marked independently, each with probability λ_r, then the number of marks received on route r in unit time will be approximately binomially distributed with mean $x_r\lambda_r$ and variance $x_r\lambda_r(1-\lambda_r)$ (when x_r is measured in packets per unit time). The corresponding Brownian perturbation of equation (9) is

$$dx_r(t) = \frac{x_r(t)}{T_r} a_r\ (x_r(t)T_r)^n\ dt - \frac{1}{T_r}\Big(a_r\ (x_r(t)T_r)^n\ + b_r\ (x_r(t)T_r)^m\Big) \\ \cdot\ \Big(x_r(t)\lambda_r(t)\ dt - (x_r(t)\lambda_r(t)(1-\lambda_r(t))^{\frac{1}{2}}\ dB_r(t)\Big), \quad (22)$$

where $(B_r(t), r \in R)$ are independent standard Brownian motions: we have replaced a deterministic term $x_r(t)\lambda_r(t)$ in equation (9), giving the rate at which marks are received on route r, by a Brownian perturbation with the same mean and the required variance.

The linearization of this stochastic differential equation has, as its solution, a multidimensional Ornstein-Uhlenbeck process, centred on the equilibrium point of the differential equations (9-10). The stationary distribution for $(x_r(t), r \in R)$ under the linearization is, in consequence, a multivariate normal distribution, $N(x, \Sigma)$, whose covariance matrix Σ is determined explicitly in terms of the parameters of the network (Appendix III).

5.1 Delay invariance

If

$$a_r = \bar{a}_r T_r^{1-n}, \quad b_r = \bar{b}_r T_r^{1-m} \quad r \in R \quad (23)$$

then neither x nor Σ depend upon $(T_r, r \in R)$: we can deduce this from Appendix III, or directly from the observation that equations (22), under the substitution (23), lose their dependence upon $(T_r, r \in R)$. The equilibrium point is given by

$$x_r = \left(\frac{\bar{a}_r}{\bar{b}_r} \cdot \frac{1-\lambda_r}{\lambda_r}\right)^{1/\alpha};$$

the parameters $\bar{a}_r, \bar{b}_r$ control the share of resources achieved by route r.

We next describe three examples where condition (23) is met.

The case $n = -1, m = 1, a_r = \bar{a}T_r^2, b_r = \bar{b}, r \in R$, was explored in [11, 13], as a mechanism to remove round-trip time bias in the allocation x. As noted in [11], the parameters n, a_r cause a source, in the absence of congestion feedback, to increase its throughput by a constant $\bar{a}$ packets per second in one second, regardless of the round-trip time (expressed in seconds). Since the parameters also satisfy (23), the covariance matrix Σ, as well as the mean vector x, is independent of $(T_r, r \in R)$. However, in view of the condition (18), we should expect possible delay instability on routes r where the ratio T_r/x_r is large, or convergence that is slower than necessary on routes where the ratio is small.

The case $n = -1, m = 0, a_r = \bar{a}_r T_r^2, b_r = \bar{b}T_r, r \in R$, was considered in [12]. The parameters m, b_r ensure that the effect of a single congestion indication bit is predictable: each marked packet will reduce the flow through a resource by $\bar{b}$, regardless of the round-trip time of the packet carrying the mark. The parameters satisfy (23), and so again both x and Σ are independent of $(T_r, r \in R)$. And again, in view of condition (18), we should expect instability or unnecessary sluggishness on a route r where the ratio $\bar{a}_r T_r/x_r$ is, respectively, especially large or small.

A third example satisfying (23) is provided by the choice

$$n = 0, a_r = \bar{a}T_r, \quad m = 1, b_r = \bar{b}_r, \quad r \in R. \tag{24}$$

In this case the covariance matrix Σ has the relatively simple form

$$\Sigma = \frac{\bar{a}}{2} X(\alpha\Lambda X + XA^T P' AX)^{-1} X, \tag{25}$$

where $X = \mathrm{diag}(x_r, r \in R)$, $\Lambda = \mathrm{diag}(\lambda_r, r \in R)$ and $P' = \mathrm{diag}(p'_j/(1 - p_j), j \in J)$. In view of condition (18), we should expect instability or unnecessary sluggishness on a route r where T_r is, respectively, especially large or small.

5.2 Scale invariance

Next we explore some properties of the covariance matrix Σ, for general n, m, a_r, b_r, for the very special case where the network comprises a single resource.

Consider the case of N flows through a single resource, where all flows share the same values of a_r, b_r, T_r, as well as m and n, and let $\beta = y_j p'_j/p_j$. Then the variance of a flow is

$$\mathrm{Var}(x_r(t)) = \frac{b_r x_r^{m+1} T_r^{m-1}}{2N} \left(\frac{1}{\alpha(1-\lambda_r) + \beta} + \frac{N-1}{\alpha(1-\lambda_r)} \right). \tag{26}$$

If N is large or β is small, then there will be little interaction between two flows. In either of the limits $N \to \infty$ or $\beta \to 0$ the expression (26) becomes

$$\mathrm{Var}(x_r(t)) = \frac{b_r x_r^{m+1} T_r^{m-1}}{2\alpha(1-\lambda_r)}. \tag{27}$$

This is also the stationary distribution of the system (22) in the case where $\lambda_r(t)$ is replaced by a constant λ_r, which corresponds with the model analysed in some detail by Ott [35] (Ott assumes the marking probability is fixed and small, and provides a precise analysis of the stationary distribution of the congestion window).

A key feature of the choice $m = 1$ is that it causes the expressions (26-27) to scale with x_r^2. Hence the coefficient of variation (i.e. the ratio: standard deviation/mean) of $x_r(t)$ does not depend upon x_r – an important scale invariance property first identified by Ott [35].

The scale invariance property of the choice $m = 1$ extends to more general networks, although care must be taken with its formulation since the equilibrium value of a single flow will affect the variance of other flows with which it shares resources, a coupling captured in the general covariance matrix Σ calculated in Appendix III. We illustrate the coupling, and the impact of b_r on the coefficient of variation, with a simple example where x_r varies with r.

Consider the case of N flows through a single resource, where $m = 1, n = 0$, $a_r = \bar{a}T_r, r \in R$. Allow b_r, and hence x_r, to vary with r, and let $\beta = y_j p'_j/p_j$. (Observe this is a special case of the third example of Section 5.1.) Then

$$\mathrm{Var}(x_r(t)) = \frac{1}{2} b_r x_r^2 \left(\frac{x_r}{\sum_s x_s} . \frac{1}{1-\lambda_r+\beta} + \left(1 - \frac{x_r}{\sum_s x_s}\right) . \frac{1}{1-\lambda_r} \right). \tag{28}$$

There is now heterogeneity amongst x_r, but note that the final term of expression (28) will be approximately the same for all r, unless a single flow occupies a large proportion of the resource. If we ignore the dependence of this final term on r, then the coefficient of variation of $x_r(t)$ is proportional to $b_r^{1/2}$. This illustrates a phenomenon, again identified by Ott [35], that occurs more generally when $n = 0, m = 1$: if x_r is a small proportion of the flow through each of the resources on route r, and if λ_r is not large, then the coefficient of variation of $x_r(t)$ is approximately equal to $(b_r/2)^{1/2}$.

5.3 Discussion

We have seen in Section 4 that delay stability is characterized in terms of the increase parameters n and a_r, and in this Section we have explored the impact of the decrease parameters m and b_r on stochastic stability. We now compare and contrast these insights.

The choice $n = 0$ is suggested by delay stability considerations: this choice leaves the condition (20) independent of window size. The choice $m = 1$ is suggested by stochastic stability considerations: this choice has Ott's scale-invariance property. The combined choice $n = 0$ and $m = 1$ gives $\alpha = 1$, and hence weighted proportional fairness.

But when we look more closely, at the choices of the parameters a_r, b_r, we see that there is a tension between delay and stochastic stability. If a_r is given by the form (23) we should, in view of condition (20), expect delay instability or unnecessary sluggishness on a route r where $\bar{a}_r x_r^n T_r$ is, respectively, especially large or small, as we have seen in the examples of Section 5.1.

In a network with heterogeneous delays, some of which may be substantial, we have seen that the choices $n = 0, m = 1, a_r = \bar{a} < \pi/2\beta, b_r = \bar{b}/T_r$ seem very desirable from the point of view of earlier sections: these choices remove the round-trip bias from the equilibrium point of the system (13-15), and stabilize the equilibrium point within the deterministic model. But these choices may lead to overly high variances for routes r with low values of T_r.

In contrast, in a network where random effects predominate, the choices $n = 0, m = 1, a_r = \bar{a}T_r, b_r = \bar{b}$ have the effect of removing the round-trip bias from the equilibrium point, and of making variances independent of round-trip times.

For the Internet, where delays are highly heterogeneous and random effects are ever present, the above discussion helps us understand the compromise advocated in [20]. If $n = 0, m = 1$, and

$$\begin{aligned} a_r &= w_r T_r \bar{b}_r, \quad & b_r &= \bar{b}_r \qquad & &\text{if } T_r \le \frac{\bar{a}}{w_r \bar{b}_r} \\ a_r &= \bar{a}, & b_r &= \frac{\bar{a}}{w_r T_r} & &\text{otherwise} \end{aligned}$$

then the round-trip time bias is removed from the equilibrium point, and the speed of adaptation is delay limited on long routes and variance limited on short routes. The parameter w_r controls the share of resources allocated to flow r. Provided λ_r is not too large, flow r receives approximately w_r marks per unit time, resulting in a weighted proportionally fair allocation, with weights $(w_r, r \in R)$. If $w_r = \bar{w}, r \in R$, then we obtain approximate proportional fairness. The parameter $\bar{b}_r$ expresses flow r's trade-off between speed of convergence and variance. If flow r occupies a small proportion of each resource on its route, then the coefficient of variation of $x_r(t)$ is approximately $(b_r/2)^{1/2}$, where b_r depends upon T_r but is bounded above by

$\bar{b_r}$.

6 Dual algorithms

We have seen that there are several families of primal algorithms, with varying fairness and stability properties. Similarly, there are many variants of dual algorithm, and we shall discuss two families in this Section. Dual algorithms were initially motivated by the possibility of using queueing delay, rather than packet loss, as the feedback signal from resources to endpoints [27], and the first family we consider will be the delay-based dual algorithms analysed in detail by Paganini *et al.* [37]. More generally, dual algorithms correspond with averaging at resources prior to the feedback of more explicit information to endpoints, and we shall see that there are advantages in using feedback with a different scaling from delay.

6.1 Delay-based dual algorithms

Following [37], let

$$\frac{d}{dt}\mu_j(t) = \kappa_j\left(\sum_{s:j\in s} x_s(t-T_{sj}) - C_j I[\mu_j(t)>0]\right) \tag{29}$$

where

$$x_r(t) = D_r\left(\lambda_r(t)\right),\ \lambda_r(t) = \sum_{j\in r}\mu_j(t-T_{jr}) \tag{30}$$

and $D_r(\eta), \eta \geq 0$, is a non-negative continuous, strictly decreasing function. Assume that the matrix A has full row rank. These conditions are sufficient to allow the construction of a strictly concave Lyapunov function, and hence to deduce (*cf* [19, 36]) that the system (29-30) has a unique equilibrium point $(x(t), \mu(t)) = (x, \mu)$. Assume that link j is saturated, that is $\mu_j > 0$, for each $j \in J$: thus

$$\sum_j A_{js}x_s = C_j, \quad j \in J. \tag{31}$$

In what follows we could, equivalently, assume that the set J is reduced to include only the saturated links, and that there are no almost saturated links, at which both $\mu_j = 0$ and condition (31) holds.

Let $x_r(t) = x_r + u_r(t)$. Then, linearizing the system (29-30) about x, we obtain

$$\frac{d}{dt}u_r(t) = D_r'(\lambda_r)\sum_{j\in r}\kappa_j \sum_{s:j\in s} u_s(t-T_{sj}-T_{jr}). \tag{32}$$

A sufficient condition for the system (32) to be stable is (Appendix II) that

$$-\frac{T_r}{x_r} D'_r(\lambda_r) \sum_{j \in J} A_{jr} \kappa_j \sum_{s \in R} A_{js} x_s < \frac{\pi}{2}. \tag{33}$$

The algorithm (29-30), with $\kappa_j = C_j^{-1}$, has a natural interpretation in terms of queueing delays. Suppose that link j is modelled as a buffer with a fluid inflow at rate $\sum_{s:j \in s} x_s(t - T_{sj})$, a queue size of $C_j \mu_j(t)$, and an outflow rate of C_j whenever the queue size is positive. For example, in TCP Vegas [27], a variant of TCP, the endpoints for route r estimate the sum of the queueing delays along the route, $\sum_{j \in r} \mu_j(t - T_{jr})$, as the difference between measurements of round-trip times, including queueing delay, and longer term estimates of propagation delay. The model (29-30) ignores the impact of queueing delays on round-trip times: this may be reasonable if queueing delays are small compared with propagation delays. Alternatively, Paganini *et al.* [37] use the variable $\mu_j(t)$ to represent *virtual* queueing delay, obtained by setting C_j to be slightly lower than the outflow rate of the link, so that real queueing delays are zero at the equilibrium point of the deterministic model (29-30).

In TCP Vegas the demand function of user r is [27, 33] $D_r(\lambda_r) = 1/\lambda_r$, corresponding to a proportionally fair equilibrium. With this demand function, and with $\kappa_j = C_j^{-1}$, the stability condition (33) becomes the bound

$$\lambda_r \geq \frac{2}{\pi} T_r M_r,$$

where

$$M_r = \sum_{j \in J} A_{jr},$$

the number of saturated resources on route r.

Might it be possible to choose a demand function that ensures stability for all values of λ_r? Observe that condition (33) will be satisfied if

$$-\frac{D'_r(\lambda_r)}{D_r(\lambda_r)} \leq \frac{\pi}{2T_r M_r}, r \in R, \quad \kappa_j < \frac{1}{C_j}, j \in J. \tag{34}$$

The first condition will be satisfied with equality by

$$D_r(\lambda_r) = D_r^{\max} \exp\left(-\frac{\pi \lambda_r}{2T_r M_r}\right), \tag{35}$$

the form identified by Paganini *et al.* [37]. This demand function has an undesirable dependence on T_r as well as M_r, with fairness consequences we return to discuss in Section 6.4.

With more information available at resources, there are other ways to ensure stability of the system (32). For example, a different sufficient condition for stability is that

$$-\frac{1}{M_r}\sum_{j\in J} A_{jr}\kappa_j \sum_{s\in R} A_{js}D'_s(\lambda_s)M_sT_s < \frac{\pi}{2} \tag{36}$$

(Appendix II). Let

$$N_j = \sum_{s\in R} A_{js},$$

the number of flows passing through resource j, and let

$$D_r(\lambda_r) = \left[D_r^{\max} - \frac{\lambda_r}{T_rM_r}\right]^+ .$$

Then condition (36) will be satisfied if

$$\kappa_j N_j < \frac{\pi}{2}, \quad j \in J,$$

a bound on a link's gain in terms of the number of flows through it. Or if

$$D_r(\lambda_r) = \left[D_r^{\max} - \frac{\lambda_r}{M_r}\right]^+ . \tag{37}$$

then condition (36) will be satisfied if

$$\kappa_j \sum_{s\in R} A_{js}T_s < \frac{\pi}{2}, \quad j \in J.$$

In this example, the demand function (37) has the attractive feature that it has no dependence on T_r, and a link's gain is bounded in terms of the sum of the round-trip times of the flows through it.

6.2 Fair dual algorithms

The algorithm (29-30) allowed a natural interpretation of $\mu_j(t)$ as either a real or virtual queueing delay. It was, however, difficult to reconcile fairness with stability. We now show that it is possible to design dual algorithms that can achieve weighted α-fairness, and have straightforward delay and stochastic stability properties.

Consider

$$\frac{d}{dt}\mu_j(t) = \kappa_j\mu_j(t)\left(\sum_{s:j\in s} x_s(t-T_{sj}) - C_j\right) \tag{38}$$

where again $x_r(t)$ is defined by (30), $D_r(\eta), \eta \geq 0$, is a non-negative continuous, strictly decreasing function. Let $(x(t), \mu(t)) = (x, \mu)$ be an equilibrium point of the system (30), (38), and assume $\mu_j > 0$ for $j \in J$.

Let $x_r(t) = x_r + u_r(t)$. Then, linearizing the system (30), (38) about x, we obtain

$$\frac{d}{dt} u_r(t) = D'_r(\lambda_r) \sum_{j \in r} \kappa_j \mu_j \sum_{s:j \in s} u_s(t - T_{sj} - T_{jr}). \tag{39}$$

A sufficient condition for the system (39) to be stable is (Appendix II) that

$$-\frac{T_r}{x_r} D'_r(\lambda_r) \sum_{j \in J} A_{jr} \kappa_j \mu_j \sum_{s \in R} A_{js} x_s < \frac{\pi}{2}. \tag{40}$$

Condition (40) will be satisfied if

$$D_r(\lambda_r) = D_r(1) \lambda_r^{-1/T_r}, r \in R, \quad \kappa_j C_j < \frac{\pi}{2}, j \in J,$$

an example that parallels the earlier case (34-35).

But an alternative and preferable sufficient condition for the system (39) to be stable is (Appendix II) that

$$-\frac{D'_r(\lambda_r)}{x_r} \sum_{j \in J} A_{jr} \kappa_j \mu_j \sum_{s \in R} A_{js} x_s T_s < \frac{\pi}{2}. \tag{41}$$

Now suppose that $D_r(\lambda_r)$ is defined by equation (8), corresponding to weighted α-fairness, so that $D_r(\lambda_r) = -\alpha \lambda_r D'_r(\lambda_r)$. Then condition (41) will be satisfied if

$$\kappa_j C_j \bar{T}_j < \frac{\pi}{2} \alpha, \quad j \in J \tag{42}$$

where

$$\bar{T}_j = \frac{\sum_{s \in R} A_{js} x_s T_s}{\sum_{s \in R} A_{js} x_s},$$

the average round trip time of packets through resource j.

We term the system (8), (30), (38) the *fair dual algorithm*: it is able to achieve weighted α-fairness with a natural delay stability condition (42) on resource gains. We shall see in the next subsection that it also possesses scale-invariant stochastic stability properties.

The proposal of Katabi *et al.* [17] for more explicit feedback from resource to endpoints requires that a packet should contain the sending endpoint's estimate of its round-trip time: we note this as a possible mechanism that would allow each resource to estimate its own value of $\bar{T}_j$.

6.3 Stochastic stability of dual algorithms

For the primal algorithms we considered, each packet crossing the network provided a single bit of information concerning congestion along its route: this noisy feedback was averaged at endpoints, a process whose variance was analysed in Section 5. We idealize dual algorithms as providing enough bits of information per packet that there is essentially no averaging necessary at endpoints. This will leave the estimation process at a resource as a possible source of randomness, and in this subsection we study its variance.

Suppose that packets flowing along route r form a random stream, with the number in unit time having mean $x_r(t)$ and variance $\epsilon_r x_r(t)$. For example, if packets form a Poisson stream then $\epsilon_r = 1$. Consider the system

$$d\mu_j(t) = \kappa_j \mu_j(t)^m \left(\sum_{s:j\in s} (x_s(t)\, dt + (\epsilon_s x_s(t))^{\frac{1}{2}}\ dB_s(t)) - C_j I[\mu_j(t) > 0]\, dt \right), \tag{43}$$

where $x_r(t) = D_r(\sum_{j\in r} \mu_j)$, and $(B_s(t), s \in R)$ are independent standard Brownian motions. The cases $m = 0$ and $m = 1$ describe Brownian perturbations of delay-based and fair dual algorithms respectively. In each case we ignore the time delays T_{sj}, T_{jr}. The linearization of this stochastic differential equation has, as its stationary distribution for $(\mu_j(t), j \in J)$, a multivariate normal distribution, $N(\mu, \Sigma)$, whose covariance matrix Σ is determined explicitly, in terms of the parameters of the network, in Appendix III.

As a very special case, consider a single resource, $J = \{j\}$, where $A_{jr} = 1, \epsilon_r = \epsilon, r \in R$, and $D_r(\lambda_r), r \in R$, takes the form (8). Allow w_r, and hence x_r, to vary with r. Then (Appendix III)

$$\mathrm{Var}\,(\mu_j(t)) = \frac{\epsilon\alpha\kappa_j}{2}\mu_j^{m+1}, \quad \mathrm{Var}\,(x_r(t)) = \frac{\epsilon\kappa_j}{2\alpha}\mu_j^{m-1}x_r^2. \tag{44}$$

If $m = 1$, corresponding to the fair dual algorithm, then the coefficient of variation of neither $\mu_j(t)$ nor $x_r(t)$ depends on μ_j, an attractive scale-invariance property.

As for primal algorithms, there is a tension between delay and stochastic stability. If κ_j is chosen to satisfy the delay stability condition (42) with equality then the variance of $x_r(t)$ will not depend upon α, and the variance of both $\mu_j(t)$ and $x_r(t)$ will be inversely proportional to both the capacity of resource j and the average round-trip time of packets through resource j. A resource may prefer a smaller value of κ_j in order to control its coefficient of variation, especially if its capacity or its average round-trip time is small. In home networks [3] or ad hoc networks [6] where propagation delays are small, stochastic instability may dominate delay stability. The choice $\alpha =$

$1, \kappa_j = \bar{\kappa}/C_j$ was used in [6], a choice that makes comparable the rate of convergence across different resources and causes variances to be inversely related to capacities. From equation (42), a sufficient condition for delay stability is then that

$$\bar{\kappa}\bar{T}_j < \frac{\pi}{2}, \quad j \in J,$$

an upper bound, uniform over resources, on the average round-trip time of packets through a resource.

6.4 Discussion

The constraints imposed by delay stability take different forms for primal and dual algorithms: for primal algorithms there are restrictions, such as (19), on resource behaviour; while for dual algorithms there are restrictions, such as (34), on demand functions.

Delay-based dual algorithms are effective at fully utilizing resources, but are less effective at fairly sharing resources when delays are heterogeneous. In contrast primal algorithms can achieve fairness, but are less effective at utilizing resources fully. Kunniyur and Srikant [23, 24] have shown that by slowly adapting the marking function $p_j(y)$ at resources, primal algorithms can also control resource utilization; and Paganini*et al.* [38] have shown that by slowly adapting the demand function $D_r(\lambda)$ at sources, delay-based dual algorithms can also control fairness. For a discussion of the resulting primal-dual schemes, which aim to achieve both fairness and high utilization, see Low and Srikant [28]. Without propagation delays, global stability can be obtained for primal-dual schemes [1, 45]. With heterogeneous delays and averaging at both sources and resources, Vinnicombe [42] has established an important robust stability result, briefly mentioned in Appendix II.

For the primal algorithm Johari and Tan [16] observe that the form $p_j(y) = (y/C_j)^\beta$, for β integral, is the probability that a packet arriving at an M/M/1 queue will find β or more packets already present; more generally, the restriction (19) may be plausible in connection with queueing phenomena. For the fair dual algorithm, the restriction upon demand functions, that $D_r(\lambda_r) = -\alpha\lambda_r D_r'(\lambda_r)$, corresponds precisely with the definition of weighted α-fairness, and hence this algorithm's ability to achieve fairness and full utilization.

In a network comprising a single resource that knows the number of flows through it, much more can be done: see Hollot *et al.* [14] for an analysis of several schemes, including some which correspond to classical proportional and proportional-integral control.

7 Conclusion

In this paper we have reviewed recent work on the fairness and stability of end-to-end congestion control. We have described models that provide some insight into the success of the congestion avoidance algorithm of TCP, and into how it might, or should, evolve in the future. In particular, we have seen that it is in principle possible for an arbitrary collection of overlapping flows to share resources in a fair, stable and scalable manner, using end-to-end mechanisms where each flow knows only of its own experience of congestion and of its own feedback delay.

The heterogeneity of the Internet makes it important to understand what can be achieved with minimal, incrementably deployable, changes. We have seen that in networks with long propagation delays, a single bit of congestion information per packet may be ample: delay stability requires relatively slow adaptation at endpoints, slow enough to allow averaging of congestion information over many packets. For home or ad-hoc networks different issues arise. Homogeneity of equipment may allow substantial changes in protocols, and when round-trip times are short more explicit feedback can substantially reduce variances.

We have not discussed work on the initial phase of TCP, *i.e.* Jacobson's slow-start algorithm [15], or work on the dynamics of flow arrivals and departures, both areas which give complementary insights into network behaviour. And it is salutary to note that in the Linux source, less than 1% of the TCP code concerns congestion window updates. These are important lines, governing the way the network shares resources in a fair and stable manner; but they are not all there is to TCP.

Acknowledgement

I am grateful to Tom Kelly for many illuminating discussions on TCP and for pointing out the importance of Ott's [35] coefficient of variation calculation, and to Jon Crowcroft, Ramesh Johari, Sven Östring, Gaurav Raina and Damon Wischik for their helpful comments on earlier drafts of this paper. I benefitted from taking part in the Spring 2002 Program on Large Scale Communication Networks at the Institute for Pure and Applied Mathematics, UCLA; I am grateful both to IPAM, and to John Doyle and Walter Willinger for their work in support of this Program.

References

[1] Arrow KJ, Hurwicz L. Gradient method for concave programming III: further global results and applications to resource allocation. In: Arrow KJ, Hurwicz L, Uzawa H (eds). Studies in linear and non-linear programming. Stanford University Press. 1958, pp 133-145.

[2] Bansal D, Balakrishnan H. Binomial congestion control algorithms. Proc. IEEE INFOCOM 2001.

[3] Barham P. Explicit congestion avoidance: cooperative mechanisms for reducing latency and proportionally sharing bandwidth. Microsoft Research Technical Report MSR-TR-2001-100, 2001.

[4] Carpenter B (ed). Architectural principles of the Internet. Network Working Group RFC-1958, 1996.

[5] Clark DD, Blumenthal MS. Rethinking the design of the Internet: the end to end arguments vs the brave new world. 2000.

[6] Crowcroft J, Gibbens R, Kelly F, Östring S. Modelling incentives for collaboration in mobile ad hoc networks. Proc. WiOpt'03, Sophia Antipolis, France. 2003.

[7] Crowcroft J, Oechslin P. Differentiated end-to-end Internet services using a weighted proportionally fair sharing TCP. ACM Computer Communications Review 1998; 28: 53-67.

[8] Clark DD. The design philosophy of the DARPA Internet protocols. ACM Computer Communication Review 1998; 18: 106-114.

[9] Floyd S. TCP and Explicit Congestion Notification, ACM Computer Communication Review 1994; 24: 10-23.

[10] Floyd S. HighSpeed TCP for large congestion windows. Internet Draft <draft-floyd-tcp-highspeed-02.txt> , February 2003. Work in progress.

[11] Floyd S, Jacobson V. (1992) On traffic phase effects in packet-switched gateways. Internetworking: Research and Experience 1992; 3: 115-156.

[12] Gibbens RJ, Kelly FP. Resource pricing and the evolution of congestion control. Automatica 1999; 35: 1969-1985.

[13] Henderson TR, Katz RH. TCP performance over satellite channels. UCB Computer Science Technical Report 99-1083, December 1999.

[14] Hollot CV, Misra V, Towsley D, Gong WB. Analysis and design of controllers for AQM routers supporting TCP flows. IEEE Transactions Automatic Control 2002; 47: 945-959.

[15] Jacobson V. Congestion avoidance and control. Proc ACM Sigcomm 1988: 314-329.

[16] Johari R, Tan DKH. End-to-end congestion control for the Internet: delays and stability. IEEE/ACM Transactions on Networking 2001; 9: 818-832.

[17] Katabi D, Handley M, Rohrs C. Internet congestion control for future high bandwidth-delay product environments. Proc ACM Sigcomm 2002.

[18] Kelly FP. Mathematical modelling of the Internet. In: Engquist B, Schmid W (eds). Mathematics Unlimited – 2001 and Beyond. Springer-Verlag, Berlin. 2001, pp 685-702.

[19] Kelly FP, Maulloo AK, Tan DKH Rate control in communication networks: shadow prices, proportional fairness and stability. Journal of the Operational Research Society 1998; 49: 237-252.

[20] Kelly T. On engineering a stable and scalable TCP variant. Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR.435, June 2002.

[21] Kelly T. Scalable TCP: improving performance in highspeed wide area networks. First International Workshop on Protocols for Fast Long-Distance Networks, 2003.

[22] Kleinrock L. Communication Nets: Stochastic Message Flow and Delay. McGraw-Hill, New York. 1964.

[23] Kunniyur S, Srikant R. A time-scale decomposition approach to adaptive ECN marking. IEEE Transactions on Automatic Control 2002; 47: 882-894.

[24] Kunniyur S, Srikant R. Stable, scalable, fair congestion control and AQM schemes that achieve high utilization in the Internet. 2002.

[25] Low SH, Lapsley DE. Optimization flow control, I: basic algorithm and convergence. IEEE/ACM Transactions on Networking 1999; 7:861-875.

[26] Low SH, Paganini F, Doyle JC. Internet congestion control. IEEE Control Systems Magazine 2002; 22: 28-43.

[27] Low SH, Peterson LL, Wang L. Understanding Vegas: a duality model. Journal of ACM 2002; 49: 207-235.

[28] Low SH, Srikant R. A mathematical framework for designing a low-loss, low-delay Internet. Networks and Spatial Economics 2003.

[29] Massoulié L. Stability of distributed congestion control with heterogeneous feedback delays. IEEE Transactions on Automatic Control 2002; 47: 895-902.

[30] Mathis M, Semke J, Mahdavi J, Ott T. The macroscopic behaviour of the TCP congestion avoidance algorithm. Computer Communication Review 1997; 27: 67-82.

[31] Mazumdar R, Mason LG, Douligeris C. Fairness in network optimal flow control: optimality of product forms. IEEE Transactions on Communications 1991; 39: 775–782.

[32] Misra A, Baras J, Ott T. Generalized TCP congestion avoidance and its effect on bandwidth sharing and variability. Proc. GLOBECOM 2000

[33] Mo J, Walrand J. Fair end-to-end window-based congestion control. IEEE/ACM Transactions on Networking 2000; 8: 556-567.

[34] Nash JF. The bargaining problem. Econometrica 1950: 28; 155-162.

[35] Ott TJ. ECN protocols and the TCP paradigm. 1999.

[36] Paganini F. A global stability result in network flow control. Systems and Control Letters 2002; 46: 165-172.

[37] Paganini F, Doyle JC, Low SH. Scalable laws for stable network congestion control. IEEE CDC, Orlando, FL, December 2001.

[38] Paganini F, Wang Z, Low SH, Doyle JC A new TCP/AQM for stable operation in Fast networks. 2002.

[39] Ramakrishnan KK, Floyd S, Black D. The addition of Explicit Congestion Notification (ECN) to IP. Internet Engineering Task Force, June 2001.

[40] Saltzer JH, Reed DP, Clark DD. End-to-end arguments in system design. ACM Transactions on Computer Systems 1984; 2: 277-288.

[41] Varian HR (1992). *Microeconomic Analysis*. Third edition. Norton, New York.

[42] Vinnicombe G. On the stability of end-to-end congestion control for the Internet. Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR.398. 2000.

[43] Vinnicombe G. On the stability of networks operating TCP-like congestion control. IFAC 2002.

[44] Vinnicombe G. Robust congestion control for the Internet. 2002.

[45] Wen T, Arcak M. A unifying passivity framework for network flow control. 2003.

[46] Willinger W, Doyle J. Robustness and the Internet: design and evolution. In: Jen E (ed). Robust design: a repertoire from biology, ecology, and engineering. Oxford University Press. 2003.

8 Appendix I: global stability

Consider the system of differential equations

$$\frac{d}{dt}\,x_r(t) = \frac{x_r(t)}{T_r}\,\Big(a_r(x_r(t)T_r)\;(1-\lambda_r(t)) - b_r(x_r(t)T_r)\;\lambda_r(t)\Big), \tag{45}$$

for $r \in R$, where $\lambda_r(t)$ is defined by equations (2) and (10). Assume that, for $r \in R$, $a_r(w)$ and $b_r(w)$ are continuous functions on $(0,\infty)$, and that $a_r(w)/b_r(w)$ is a strictly decreasing function, with $a_r(w)/b_r(w) \to \infty$ as $w \downarrow 0$ and $a_r(w)/b_r(w) \to 0$ as $w \uparrow \infty$. Assume that, for $j \in J$, the function $p_j(y), y \geq 0$, is a continuous, non-decreasing function, taking values in the interval $[0,1]$ and not identically either zero or one.

Theorem 8.1 *The form (7) is a Lyapunov function for the system of differential equations (2), (10), (45) under the choices*

$$U_r(x_r) = \frac{1}{T_r}\int_0^{x_rT_r} \log\left(1+\frac{a_r(w)}{b_r(w)}\right)\,dw,$$

$$C_j(y) = -\int_0^y \log\left(1-p_j(z)\right)dz.$$

The unique value x maximizing the form (7) is a stable point of the system, to which all trajectories converge. At the stable point

$$a_r(x_r T_r)(1-\lambda_r) = b_r(x_r T_r)\lambda_r. \tag{46}$$

Proof. The assumptions on $a_r(w), b_r(w), r \in R$, and on $p_j, j \in J$, ensure that $\mathcal{U}(x)$, given by expression (7), is strictly concave on the positive orthant with an interior maximum; the maximizing value of x is thus unique. Observe that

$$\frac{\partial}{\partial x_r}\mathcal{U}(x) = \sum_{j \in r} \log\left(1 - p_j\Big(\sum_{s: j \in s} x_s\Big)\right) - \log\frac{b_r(x_r T_r)}{a_r(x_r T_r) + b_r(x_r T_r)}\,; \tag{47}$$

setting these derivatives to zero identifies the maximum. Further, the expression (47) has the same sign as

$$\prod_{j \in r}\left(1 - p_j\Big(\sum_{s: j \in s} x_s\Big)\right) - \frac{b_r(x_r T_r)}{a_r(x_r T_r) + b_r(x_r T_r)}\,. \tag{48}$$

Now, from equation (45),

$$\begin{aligned}\frac{d}{dt}x_r(t) = \frac{x_r(t)}{T_r}\;&\Big(a_r(x_r(t)T_r) + b_r(x_r(t)T_r)\Big)\\ &\cdot\left(\,(1-\lambda_r(t)) - \frac{b_r(x_r(t)T_r)}{a_r(x_r(t)T_r) + b_r(x_r(t)T_r)}\right).\end{aligned} \tag{49}$$

We can deduce that the partial derivative (47) has the same sign as the derivative (49), and they are zero together. Now

$$\frac{d}{dt}\mathcal{U}\big(x(t)\big) = \sum_{r \in R}\frac{\partial \mathcal{U}}{\partial x_r}\cdot\frac{d}{dt}x_r(t) \tag{50}$$

and so $\mathcal{U}(x(t))$ is strictly increasing with t unless $x(t) = x$, the unique x maximizing $\mathcal{U}(x)$. The strict concavity of $\mathcal{U}(x)$ and the continuity of the derivative (45) implies that the derivative (50) is bounded away from zero outside any open neighbourhood of x, and the result follows.

□

The choices

$$a_r(w) = a_r\, w^n, \quad b_r(w) = b_r\, w^m \quad r \in R$$

with $\alpha = m - n > 0$ correspond to equations (9). Given a strictly concave continuously differentiable function $U_r(x_r)$ with $U_r'(x_r) \to \infty$ as $x_r \downarrow 0$ and $U_r'(x_r) \to 0$ as $x_r \uparrow \infty$, choices satisfying the conditions of the Theorem are

$$a_r(w) = a, \quad b_r(w) = a \left(\exp(U_r'(w_r/T_r)) - 1 \right)^{-1},$$

or

$$a_r(w) = b \left(\exp(U_r'(w_r/T_r)) - 1 \right), \quad b_r(w) = b.$$

The interpretation of congestion control as a distributed algorithm solving a global optimization problem is reviewed in [18, 26, 28]. The above theorem generalizes Theorem 4 of [18].

9 Appendix II: local stability under time delays

Consider the linear system

$$T_r \frac{d}{dt} u_r(t) = -\gamma_r u_r(t) - \kappa_r x_r \nu_r(t) \tag{51}$$

for $r \in R$, where

$$\nu_r(t) = \sum_{j \in J} A_{jr} \kappa_j \sum_{s \in R} A_{js} u_s(t - T_{sj} - T_{jr}). \tag{52}$$

Theorem 9.1 *Suppose that $\gamma_r, \kappa_r, x_r \geq 0$ for $r \in R$; $\kappa_j \geq 0$ for $j \in J$; $A_{jr}, T_{jr}, T_{rj} \geq 0$ for $j \in J, r \in R$, and the identity (12) is satisfied. Then the system (51-52) is stable if*

$$\kappa_r \sum_{j \in J} A_{jr} \kappa_j \sum_{s \in R} A_{js} x_s < \frac{\pi}{2} \quad r \in R. \tag{53}$$

There are two striking aspects of this result. Firstly, the condition (53) is local, in the sense that it involves κ_j only for resources j for which $A_{jr} > 0$ and, for these resources, it involves x_s only for routes s for which $A_{js} > 0$. Secondly, the delays T_{jr}, T_{sj} are not part of the condition: the term T_r in equation (51) is sufficient to scale the gain on route r.

That a result of the above form might be possible was first conjectured by Johari and Tan [16]. They showed that the identity (12) leads to an elegant decomposition of the transfer function into a product of a diagonal and an Hermitian matrix, and used this to establish their conjecture in the case where all round-trip times are the same. The clear formulation in [16] of the essential problem stimulated considerable interest, and three independent papers reported exciting further results [29, 37, 42]. Massoulié [29] established the above result in the case where $\gamma = 0$ and the right-hand side of condition (53) has $\pi/2$ replaced by 1; an alternative form of the same result was proved by Paganini, Doyle and Low [37], who established condition (34), with $\pi/2$ replaced by 1, as a sufficient condition for local stability of the dual algorithm (29-30). Vinnicombe's lemma [42], a key bound on the eigenvalues of the product of a matrix and a diagonal matrix, allows the result as stated to be proved [42, 43].

Theorem 9.1 is really a *family* of sufficient conditions for stability. Given the vector $(\kappa_r x_r, r \in R)$, different choices for the vectors $(\kappa_r, r \in R), (x_r, r \in R)$ will leave the system (51-52) unaltered, but will give different conditions (53). Thus the system (32) is an example of equations (51-52) with the choice $\gamma_r = 0, \kappa_r x_r = -D'_r(\lambda_r)T_r$. The stability condition (33) corresponds to the choice $\kappa_r = -D'_r(\lambda_r)T_r/x_r$, while the stability condition (36) has $\kappa_r = 1/M_r, x_s = -D'_s(\lambda_s)M_sT_s$. Similarly the system (39) is an example of equations (51-52) but with κ_j replaced by $\kappa_j\mu_j$. The stability condition (40) corresponds to the choice $\kappa_r = -D'_r(\lambda_r)T_r/x_r$, while the stability condition (41) has $\kappa_r = -D'_r(\lambda_r)/x_r$.

To obtain condition (18) consider the system

$$\frac{d}{dt}x_r(t) = \frac{x_r(t-T_r)}{T_r} \cdot \Big(a_r(x_r(t)T_r)\ (1-\lambda_r(t)) - b_r(x_r(t)T_r)\ \lambda_r(t)\Big), \quad (54)$$

together with equations (14), (15). Let $x_r(t) = x_r + u_r(t)$, $\lambda_r(t) = \lambda_r + (1-\lambda_r)\nu_r(t)$, and write $y_j = \sum_{s:j\in s} x_s$, $p_j = p_j(y_j), p'_j = p'_j(y_j)$. Let

$$n_r = \frac{w a'_r(w)}{a_r(w)} \quad m_r = \frac{w b'_r(w)}{b_r(w)},$$

both evaluated at $w = x_rT_r$, at which argument we assume $a_r(.), b_r(.)$ are differentiable. Then linearizing the system (14), (15), (54) about its unique equilibrium point (x, λ), and using relation (46), we obtain the equations

$$T_r\frac{d}{dt}u_r(t) = -a_r(x_rT_r)(1-\lambda_r)\left((m_r - n_r)u_r(t) + \frac{x_r}{\lambda_r}\nu_r(t)\right)$$

together with equations (17). Then a sufficient condition for stability is that $m_r > n_r$ and

$$a_r(x_rT_r)\,\frac{1-\lambda_r}{\lambda_r}\sum_{j\in r}\frac{y_jp'_j}{1-p_j} < \frac{\pi}{2},$$

as can be seen by setting $\kappa_r = a_r(x_r T_r)(1-\lambda_r)/\lambda_r$ and $\kappa_j = p'_j/(1-p_j)$ in Theorem 9.1.

In the model (14), (15), (54) the equation (15) represents the marking probability at a resource as a function of the instantaneous flow through the resource. Vinnicombe has also considered a variant where the marking probability at a resource is a function of an exponentially weighted average of the flow through the resource. Consider the system (14), (54) where, instead of equation (15),

$$\mu_j(t) = p_j(z_j(t)), \quad \delta_j \frac{d}{dt} z_j(t) = \sum_{s:j\in s} x_s(t - T_{sj}) - z_j(t).$$

In Vinnicombe [43, 44] it is shown that the linearization of this system about its equilibrium point is locally stable if

$$j \in r \Rightarrow a_r(x_r T_r) \cdot \frac{y_j p'_j}{p_j} < 1; \tag{55}$$

if, further,

$$j \in r \Rightarrow \delta_j < 2T_r \tag{56}$$

then the stability is robust to perturbations of the link and source dynamics. Now if $j \in r$ then the propagation delay through link j is a lower bound on T_r: condition (56) will thus be satisfied if the flow averaging at link j has a time constant δ_j less than twice the propagation delay through link j. Vinnicombe's results [43, 44] provide a family of sufficient conditions: for a given guarantee of robust stability, the coefficient of T_r appearing in the right hand side of the inequality in (56) may be increased, at the cost of a reduction of the right hand side of the inequality in (55).

10 Appendix III: variance calculations

In the previous Appendix the linearization faithfully represented feedback delays, and ignored random perturbations. In this appendix the linearization will model stochastic effects, but will ignore feedback delays.

Again let $x_r(t) = x_r + u_r(t)$, write $y_j = \sum_{s:j\in s} x_s$, and let $p_j = p_j(y_j), p'_j = p'_j(y_j)$. Then, linearizing the system (2),(9-10) about the unique equilibrium point (x, λ), we obtain the equations

$$T_r \frac{d}{dt} u_r(t) = -b_r(x_r T_r)^m \left(\alpha\lambda_r u_r(t) + x_r \sum_{j\in r} \frac{p'_j}{1-p_j} \sum_{s:j\in s} u_s(t) \right) \tag{57}$$

Next let $v_r(t) = u_r(t)/(b_r x_r^{m+1} T_r^{m-1})^{\frac{1}{2}}$, so that $x_r(t) = x_r + (b_r x_r^{m+1} T_r^{m-1})^{\frac{1}{2}} v_r(t)$. Let $v(t) = (v_r(t), r \in R)^T$: then we may rewrite equation (57) in matrix form as

$$\frac{d}{dt} v(t) = -B^{\frac{1}{2}} X^{\frac{m-1}{2}} T^{\frac{m-1}{2}} [\alpha \Lambda X + X A^T P' A X] T^{\frac{m-1}{2}} X^{\frac{m-1}{2}} B^{\frac{1}{2}} v(t)$$

where $B = \mathrm{diag}(b_r, r \in R)$, $X = \mathrm{diag}(x_r, r \in R)$, $\Lambda = \mathrm{diag}(\lambda_r, r \in R)$, $P' = \mathrm{diag}(p'_j/(1-p_j), j \in J)$, and (leaving the context to make clear this is not the transpose operator) $T = \mathrm{diag}(T_r, r \in R)$. Let

$$\Gamma^T \Phi \Gamma = B^{\frac{1}{2}} X^{\frac{m-1}{2}} T^{\frac{m-1}{2}} [\alpha \Lambda X + X A^T P' A X] T^{\frac{m-1}{2}} X^{\frac{m-1}{2}} B^{\frac{1}{2}} \tag{58}$$

where Γ is an orthogonal matrix, $\Gamma^T \Gamma = I$, and $\Phi = \mathrm{diag}(\phi_r, r \in R)$ is the matrix of eigenvalues, necessarily positive, of the real, symmetric, positive definite matrix (58). Then

$$\frac{d}{dt} v(t) = -\Gamma^T \Phi \Gamma v(t).$$

The corresponding linearization of equation (22) is

$$dv(t) = -\Big(\Gamma^T \Phi \Gamma v(t) dt + F dB(t)\Big) \tag{59}$$

where $F = \mathrm{diag}(f_r, r \in R)$ and

$$f_r^2 = \frac{a_r}{T_r} (x_r T_r)^n = \frac{b_r}{T_r} \frac{\lambda_r}{1 - \lambda_r} (x_r T_r)^m. \tag{60}$$

Under the stationary solution to the system (59), $v(t)$ has a multivariate normal distribution, whose covariance matrix is calculated in [19]. From this covariance matrix we can deduce that the linearization of the system (22) has, as its stationary solution, $x(t) \sim N(x, \Sigma)$ where

$$\Sigma = B^{\frac{1}{2}} X^{\frac{m+1}{2}} T^{\frac{m-1}{2}} \Gamma^T [\Gamma F; \Phi] \Gamma T^{\frac{m-1}{2}} X^{\frac{m+1}{2}} B^{\frac{1}{2}}$$

and

$$[\Gamma F; \Phi]_{rs} = \frac{[\Gamma F F^T \Gamma^T]_{rs}}{\phi_r + \phi_s}\,. \tag{61}$$

First observe that under condition (23) the matrix F, given by (60), does not depend upon the various round-trip times T: it follows that expression (61), and hence the covariance matrix Σ, does not depend upon T. (If $b_r = \bar{b}_r T_r^{1-m}, r \in R$ then from (60) and (61) we can write Σ in terms of X, with no explicit dependence on T, but of course X will itself depend upon T.)

Under the choice (24), $F = \bar{a}I$, and so $[\Gamma F; \Phi] = (\bar{a}/2)\Phi^{-1}$. Now $\Gamma^T \Phi^{-1} \Gamma = (\Gamma^T \Phi \Gamma)^{-1}$, and hence we deduce the form (25). More generally, if $m = 1$ then

$$\Sigma = B^{\frac{1}{2}} X \Gamma^T [\Gamma F; \Phi] \Gamma X B^{\frac{1}{2}}, \tag{62}$$

and so Σ does not depend upon T other than through X. The matrix $\Gamma^T[\Gamma F; \Phi]\Gamma$ captures coupling between routes. Its pre- and post-multiplication by X in expression (62) is a generalization to a network of Ott's scale-invariance property. The expression (28) gives the diagonal entries of the matrix (62) in a simple example that illustrates the scaling impact of X and B.

Next we calculate the covariance matrix Σ for some of the dual algorithms of Section 6, following [19]. Consider the system (43), and let $\mu_j(t) = \mu_j + (\kappa_j \mu_j^m)^{1/2} \eta(t)$. Assume A has full row rank, and $\mu_j > 0, j \in J$. Linearizing about the equilibrium point (x, μ), we obtain

$$d\,\eta(t) = -\Theta^T \Psi \Theta \eta(t) + G dB(t)$$

where

$$\Theta^T \Psi \Theta = -\kappa^{1/2} \mu^{m/2} A D' A^T \mu^{m/2} \kappa^{1/2}, \tag{63}$$

$\kappa = \mathrm{diag}(\kappa_j, j \in J), \mu = \mathrm{diag}(\mu_j, j \in J)$, $D' = \mathrm{diag}(D'_r(\lambda_r), r \in R)$, Θ is an orthogonal matrix, $\Theta^T \Theta = I$, $\Psi = \mathrm{diag}(\psi_j, j \in J)$ is the matrix of eigenvalues, necessarily positive (since A has full row rank), of the real, symmetric matrix (63), and

$$G_{jr} = (\kappa_j \mu_j^m)^{1/2} A_{jr} (\epsilon_r x_r)^{1/2}.$$

Define the symmetric matrix $[\Theta G; \Psi]$ by

$$[\Theta G; \Psi]_{jk} = \frac{[\Theta G G^T \Theta^T]_{jk}}{\psi_j + \psi_k}.$$

Then

$$\Sigma = \kappa^{1/2} \mu^{m/2} \Theta^T [\Theta G; \Psi] \Theta \mu^{m/2} \kappa^{1/2}.$$

For example, if $J = \{j\}, A_{jr} = 1, \epsilon_r = \epsilon, r \in R$, and $D_r(\lambda_r), r \in R$, takes the form (8), then Σ evaluates to the scalar $\frac{1}{2}\epsilon \alpha \kappa_j \mu_j^{m+1}$, giving the first part of equation (44); the second part then follows from the form of the assumed demand function (8). More generally, the matrix $\Theta^T[\Theta G; \Psi]\Theta$ captures coupling between resources.

We remark that one could attempt to combine the analyses of this and the previous appendix, and consider a linearized model that incorporates *both* feedback delays *and* Gaussian noise. The stationary solution would again be Gaussian, and the results of this and the previous appendix would presumably emerge as boundary cases. Similarly, we have assumed that the primary source of randomness in primal algorithms is the averaging process at endpoints, while the primary source of randomness in dual algorithms is the averaging process at resources. It would be interesting to study the interaction of both averaging processes, and to explore whether our variance analyses emerge as boundary cases.