A Control Theoretical Look at Internet Congestion Control

Fernando Paganini[1], John Doyle[2], and Steven H. Low[2]

[1] UCLA Electrical Engineering, Los Angeles, CA 90095, USA
[2] California Institute of Technology, Pasadena, CA 91125, USA

Abstract. Congestion control mechanisms in today's Internet represent perhaps the largest scale artificial feedback system ever deployed, and yet one that has evolved mostly outside the scope of control theory. This can be explained by the tight constraints of decentralization and simplicity of implementation in this problem, which would appear to rule out most mathematically-based designs. Nevertheless, a recently developed framework based on fluid flow models has allowed for a belated injection of control theory into the area, with some pleasant surprises. As described in this chapter, there is enough special structure to allow us to "guess" designs with mathematically provable properties that hold in arbitrary networks, and which involve a modest complexity in implementation.

1 Introduction

At the heart of today's Internet lies a feedback system, in charge of managing the allocation of bandwidth resources between competing traffic streams. In contrast to the telephony network where resources are allocated by the network core at call admission time, the Internet's resources are allocated in real-time, mainly by the end systems themselves. This solution is motivated by the desire to accommodate widely heterogeneous demands, from "mice" made of a few packets, to long "elephants" greedy for whatever bandwidth is available, and to avoid the complexity of a centralized allocation mechanism. The fact that end-systems must control their throughput with little information about the overall network necessitates the use of feedback; such mechanisms have been incorporated since the late 1980s [5] into the transport (TCP) layer of the Internet protocol stack. For a survey of these algorithms, see [12].

While the feedback component has significant performance implications, it was historically designed by computer scientists working largely outside the orbit of feedback control theory. This can be explained in part by the cultural distance between mathematical theory and the desire for simplicity of Internet engineers. There is, however, a more fundamental reason that stems from the Internet design principle [3] of keeping the network simple, and moving complexity to the end systems: in the congestion control problem, this creates a radically decentralized, yet highly coupled feedback system, for which control theory has little to offer. Consequently, most contributions to

congestion control from the control community (e.g. [1,16,14]) have focused on problems under centralized information which are relevant to other network scenarios (e.g., ATM), but have limited bearing on the Internet case.

Given the apparent success of the Internet in satisfying its demands, one might wonder about the relevance of mathematical theory to this endeavor: maybe this "hacked" system has managed to solve the problem. There are, however, deficiencies of the current solutions that have serious impact in the further scalability of the network, and which have proven difficult to address without the aid of mathematical tools. A first issue concerns understanding, and potentially improving, the resource allocation equilibrium that results from current TCP, and avoiding some of its undesirable side-effects, such as induced queueing delays. There are also dynamic limitations: algorithms tuned to react quickly to changing conditions have often been found to produce dramatic oscillations.

In the last few years, significant progress has been made in the theoretical understanding of both these issues, following seminal work by Kelly and coworkers [7,8] (for more references see [12]). Key to these advances is to work at the correct level of aggregation (namely, fluid flow models), and to explicitly model the *congestion measure* fed back to sources from congested links. In practice this measure can correspond to packet loss probability, or queueing delay, depending on the protocol variant. Interpreting such signals as *prices* has allowed for economic interpretations [8,10] that make explicit the equilibrium resource allocation policy specified by the control algorithms. Congestion measures allow also for *dynamic* models of TCP, that have been successful in matching empirical observations on oscillatory behavior [15,11]. In particular, these models predict that oscillatory instabilities will become more prevalent as network capacity scales up, if protocols are left unchanged.

The availability of mathematical models now stimulate the following question: how much could control engineers improve on these systems if we were to "do it all again"? Given the decentralized information structure and other tight implementation constraints, the prospect does not look easy: nevertheless, it turns out there is enough structure in this problem to allow for mathematical "hacks" with provable properties of stability and scalability. This chapter describes one of these solutions.

2 Problem Formulation

2.1 Fluid flow models

The starting point of our analysis will be a flow-level abstraction of the TCP congestion control problem. Here, each of the traffic-sources i which share the network has an associated rate x_i, and these rates get aggregated in accordance to their particular routing into flows y_l at each network link l, which in turn has a capacity c_l. All these real-valued quantities are in data packets/second.

To put this abstraction into context it is worth looking at the actual network in closer detail. TCP sources send individual packets across the network to their destinations, and receive from them an acknowledgement (ACK) packet, which serves as confirmation of correct reception and is also used for timing the following transmission. Sources maintain a *congestion window* variable w that determines how many packets can be sent before receiving an ACK; in this way, the transmission rate of the source is roughly

$$x \approx \frac{w}{\tau}, \tag{1}$$

where τ is the round-trip-time (RTT) of the communication. Clearly, the above approximation can only have meaning at longer time-scales than the RTT, and ignores all the complexity of individual packet arrival times.

Contrast this with the viewpoint of queueing theory: here the packet is the essential unit, and stochastic models are used to characterize inter-arrival times, which are then used to find probability distributions of relevant quantities such as network queues. This viewpoint is in fact so ingrained that the word "randomness" would commonly be used in place of "complexity" at the end of the previous paragraph, and the fluid approximation would be presented as a first-moment analysis of the probability distributions. Note, however, that when the traffic sources remain fixed, their packet transmission times are deterministically "clocked" by the ACK process, whose complexity depends only on issues like initial ordering in queues. This is very different from the traditional abstraction of individual customers arriving at a queue following e.g. a renewal process (which could apply naturally to the arrival of new TCP *sessions*) so it is unclear that a stochastic model can give an accurate characterization at a finer scale than the rate abstraction.

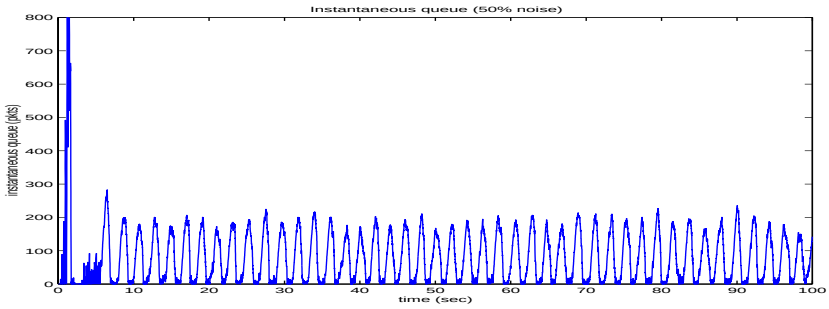

Fig. 1. Simulation example: queue oscillations.

Fortunately, recent research has shown that fluid flow models have substantial predictive power, particularly in regard to large-scale questions such as the achieved equilibrium rates and the stability of the dynamics. For instance, Figure 1 from [11] shows the result of a packet-level simulation of the

standard TCP protocol combined with the RED queue management scheme [4]. The figure shows an essentially periodic oscillation of the queue of backlogged packets over time, which can in fact be explained [15,11] as a limit cycle oscillation in fluid flow models of the type we consider here. Note that there is little "randomness" observed despite the fact that 50 % of the traffic is generated by uncontrolled "noise" sources.

Still, there is one issue that is not trivially resolved when ignoring packet-level effects: what is the adequate fluid-flow model of a queue? Thinking in terms of actual fluids and buffers as "tanks", a natural choice is to write

$$\dot{b}_l = \begin{cases} y_l - c_l, & \text{if } b_l > 0 \text{ or } y_l > c_l; \\ 0 & \text{otherwise;} \end{cases} \tag{2}$$

where b_l is the queue backlog. Namely, b_l integrates the excess rate over capacity, and is saturated to be non-negative. This model is successful in predicting slow, deterministic phenomena like the oscillations of Figure 1.

An alternative viewpoint is to say that nonzero queues build up due to packet randomness even before y_l reaches c_l, and use queueing theory formulas of the form $b_l = f(y_l, c_l)$ relating expected queues to e.g. Poisson rates. From a dynamic point of view, a static function is very different from the integrator in (2), so both models could lead to very different predictions. Note, however, that these static formulas apply only to steady-state; an improvement based on approximate transient analysis of M/M/1 queues was recently done in [18], yielding an interpolation between the two types of models. It must, however, rely on the above traffic model which is hard to justify in the context of controlled TCP sources.

A pragmatic solution to this modeling difficulty is to avoid giving network queues a key role in congestion feedback. This is also consistent with the objective, described later, of decoupling feedback from queueing. Below, we will base our congestion signals on a *virtual* queue which by construction can be made to operate fully in the integrator regime.

2.2 The congestion control loop

We return now to specifying the model in more detail. The link rates are modeled by

$$y_l(t) = \sum_i R_{li} x_i(t - \tau_{li}^f), \tag{3}$$

in which the forward transmission delays τ_{li}^f between sources at links are accounted for, and the *routing matrix* R is defined by

$$R_{li} = \begin{cases} 1 & \text{if link } l \text{ belongs to source } i\text{'s route} \\ 0 & \text{otherwise} \end{cases}.$$

The next step is to model the feedback mechanism which communicates to sources the congestion information about the network. The key idea associate with each link l a *congestion measure* $p_l(t)$, which is a positive real-valued quantity. Due to its economic interpretations we will call this variable a "price" associated with using link l. The fundamental assumption we make is that sources have access to the *aggregate* price of all links in their route,

$$q_i(t) = \sum_l R_{li} p_l(t - \tau^b_{li}). \tag{4}$$

Here again we allow for *backward* delays τ^b_{li} in the feedback path. As discussed in [12], such model can be used to approximate, at a fluid level, the feedback mechanism in existing protocols. The total RTT by source is given by

$$\tau_i = \tau^b_{i,l} + \tau^f_{i,l}; \tag{5}$$

this quantity is available to sources in real time.

Using vector notation c, y, p, x, q to collect the above variables across links or sources, we reach the following network model in the Laplace domain:

$$y(s) = R_f(s)x(s), \tag{6}$$

$$q(s) = R_b(s)^T p(s). \tag{7}$$

Here ^{T} denotes transpose, and R_f and R_b are the delayed forward and backward routing matrices, obtained by replacing the "1" elements of the matrix R respectively by the pure delay terms $e^{-\tau^f_{i,l}s}$, $e^{-\tau^b_{i,l}s}$.

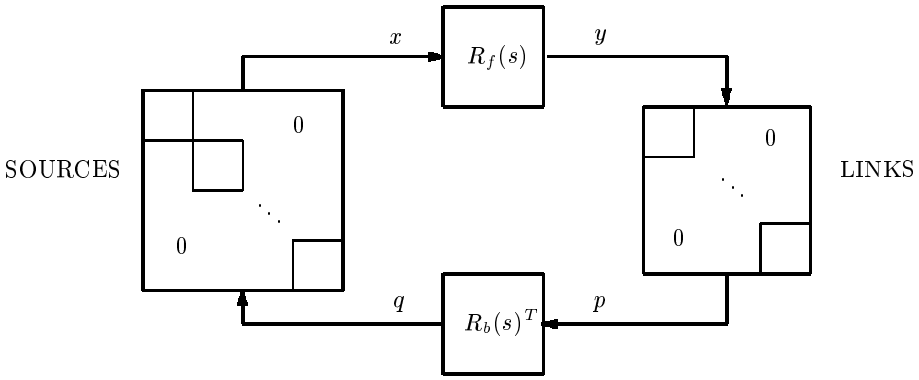

Fig. 2. General congestion control structure.

Figure 2 represents the resulting congestion control feedback loop. Tacitly assumed in the development is that both the routing and the sources participating in the feedback, remain fixed. In practice, routing usually varies at

a slower time-scale, and we are focusing on the control of "elephant" TCP flows that last long enough to be controlled; the only way to model short "mice" would be as additive noise.

What remains to be specified is: (i) How the links fix their prices based on link utilization; (ii) how the sources fix their rates based on their aggregate price. These operations are up to the designer, but have a main restriction: both must be *decentralized*, as indicated in the figure by the block-diagonal structure. For instance the source rate x_i can only depend on the corresponding aggregate price q_i.

2.3 Control objectives

The objective of this feedback is for source rates to converge, as quickly as possible to an equilibrium point x_0, y_0, p_0, q_0 that satisfies some desired static properties. More specifically, we lay out the following design objectives:

1. Network utilization. Link equilibrium rates y_{0l} should of course not exceed the capacity c_l, but also should attempt to track it. Clearly, there may be some bottleneck links that prevent others from being at capacity, but at least one bottleneck for each source should be at almost full capacity.
2. Empty equilibrium queues. In this way we avoid queueing delays, which are particularly relevant for uncontrolled "mice" that share the network with our controlled sources.
3. Resource allocation. We will assume sources have a demand curve

$$x_{0i} = f_i(q_{0i}) \tag{8}$$

that specifies their desired equilibrium rate as a decreasing function of price. This is equivalent to assigning them a *utility function* $U_i(x_i)$, in the language of [8]; in this case $f_i = (U_i')^{-1}$. We would like the control system to reach an equilibrium that accommodates these demands. This does not in itself ensure "fairness", but provides a tuning knob in which to address these issues; for more discussion see [8].
4. Dynamic asymptotic stability.

We aim at achieving these objectives for an *arbitrary* choice of network: topology, routing, and parameters such as link capacities and round trip times. Here lies the biggest challenge for design.

Based on historical experience, it appears that network engineers rank the above objectives roughly in decreasing order. High utilization is a feature of protocols since TCP-Reno [5], and efforts at reducing queueing delay have come later [4]; as of today, TCP has no mechanism for influencing the resource allocation policy. As for stability, window-based protocols have built-in boundedness due to conservation of packets, but oscillatory behavior as in Figure 1 does not create the alarm it would cause in other control engineering domains.

Perhaps due to these priorities, initial analytical work in [8,10] developed control laws guided mainly by equilibrium considerations, and only considered dynamic aspects after the fact. It is, however, very difficult to satisfy stability restrictions in this way, and one ends up having to make parameter choices which are very conservative in terms of dynamic response. Due to this difficulty (and not because of a change in priorities), we will address the stability question from early on in the design, attempting to negotiate the equilibrium objectives under this restriction.

3 Control design with linear scalable stability

We will design nonlinear control laws at sources and links that are meant to operate universally across networks; in each case, they will result in an equilibrium point x_0, y_0, p_0, q_0, and determines the dynamics around it. The objective of obtaining a stable equilibrium in every case means that the system must "schedule its gains" automatically; this severely narrows the family of suitable laws, a fact we will exploit in our search.

Consider first the objective of link utilization: we can use the principle of integral control to impose that the equilibrium rates y_{0l} track a target capacity c_{0l}; namely, writing the price dynamics

$$\dot{p}_l = \begin{cases} \mu_l(y_l - c_{0l}), & \text{if } p_l > 0 \text{ or } y_l > c_{0l}; \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where μ_l is a constant. This law is of the type considered in [10]. Comparing to (2), we see that if $c_{0l} = c_l$, prices would be proportional to queue backlogs; given our second objective of eliminating the latter in equilibrium, we will choose c_{0l} to be slightly smaller than capacity (a "virtual" capacity, see [9]). If this system reaches equilibrium, bottlenecks with nonzero price will have $y_{0l} = c_{0l}$, and non-bottlenecks with $y_{0l} < c_l$ will have zero price. This ensures every source will see a bottleneck, unless its own maximum demand is insufficient to fill it.

To guide our search for the source control law, we will impose the requirement that the closed loop must be locally stable for arbitrary networks and delays. To begin, consider a single link, running (9), and a single source, with the linearized static control law (between incremental quantities)

$$\delta x = -\kappa \delta q,$$

combined through the delay $e^{-\tau s}$. It is easily seen that this loop would be unstable for large τ, unless κ compensates for it. Fortunately, sources can measure their RTT so we can set[1] $\kappa = \frac{\alpha}{\tau}$, which gives a loop transfer function

$$L(s) = \alpha\mu \frac{e^{-\tau s}}{\tau s}. \tag{10}$$

[1] In fact, this compensation is implicit in any window protocol due to (1).

We call the above expression, with the frequency variable scaled by τ *scale-invariant*: this means that Nyquist plots for all values of τ would fall on a single curve Γ, depicted below for $\alpha\mu = 1$. In the time domain, closed loop responses for different τ's would be the same except for time-scale.

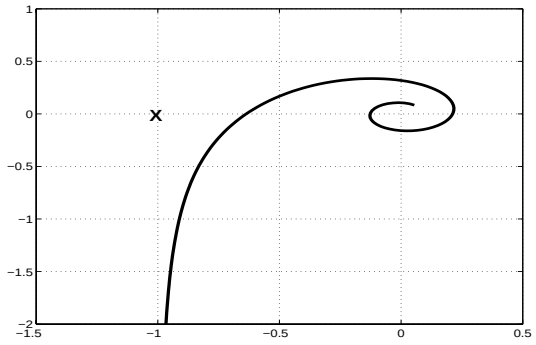

Fig. 3. Nyquist plot Γ of $e^{j\theta}/j\theta$.

Since Γ touches the negative real axis at the point $-2/\pi$, we see that our loop achieves scalable stability for all τ provided that the gain $\alpha\mu < \pi/2$.

For a single link/source, the above gain condition could be imposed a priori. Suppose, however, that we have N identical sources sharing a bottleneck link. It is not difficult to see that the effective loop gain is scaled up by N; this must be compensated for if we want stability, but in these networks neither sources nor links know what N is: how can they do the right "gain-scheduling"?

The key idea in our solution is to exploit the conservation law $c_{0l} = \sum_i x_{0i}$ implicit in the network equilibrium point, by choosing $\mu_l = \dfrac{1}{c_{0l}}$ at each link, and a gain x_{0i} at each source, in addition to the $1/\tau_i$ factor.

In the case of a single link, but now many sources with heterogeneous delays, this gives a loop transfer function of

$$L(j\omega) = \sum_i \frac{x_{0i}}{c_l} \frac{e^{-j\tau_i\omega}}{\tau_i\omega},$$

which is a *convex combination* of points in Γ. It follows that this convex combination will remain stable by a Nyquist argument.

Will this strategy work if there are multiple bottleneck links contributing to the feedback? Intuitively, there could be an analogous increase in gain that must be compensated for. Therefore we introduce a gain $\frac{1}{M_i}$ at each source, M_i being a bound on the number of bottleneck links in the source's path, which we assume is available (see Section 5). This leads to a local source

controller

$$\delta x_i = -\kappa_i \delta q_i = -\frac{\alpha_i x_{0i}}{M_i \tau_i} \delta q_i, \tag{11}$$

where $\alpha_i < \pi/2$ is a parameter. For this basic source controller, we will prove linear stability for an arbitrary network.

3.1 Linear stability result

Consider a small perturbation around equilibrium in the equations (6-7): $x = x_0 + \delta x$, $y = y_0 + \delta y$, $p = p_0 + \delta p$, $q = q_0 + \delta q$. Assuming the set of bottlenecks is unchanged by this perturbation, δp_l is only non-zero for bottleneck links. Therefore for the local analysis to follow, we write the reduced model

$$\delta \bar{y}(s) = \bar{R}_f(s) \delta x(s), \tag{12}$$

$$\delta q(s) = \bar{R}_b(s)^T \delta \bar{p}(s), \tag{13}$$

where the matrices $\bar{R}_f$, $\bar{R}_b$, and the vectors $\delta \bar{p}$, $\delta \bar{y}$ are obtained by eliminating the rows corresponding to non-bottleneck links. We will assume that after this row elimination, the resulting static matrix $\bar{R} := \bar{R}_f(0) = \bar{R}_b(0)$ is of full row rank, which appears to be a generic assumption.

With the source and link controllers described above, we have an open loop return ratio of the overall system given by

$$L(s) = \bar{R}_f(s) \mathcal{K} \bar{R}_b^T(s) \mathcal{C} \frac{I}{s}, \tag{14}$$

where the rightmost matrix of integrators has the dimension of the number of links, and

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

Note that there are no unstable pole/zero cancellations within $L(s)$; the proposition below provides stability conditions for such multivariable loops with integral control. It can be established with elementary tools.

Proposition 1. *Consider a standard unity feedback loop, with* $L(s) = \gamma F(s) \frac{I}{s}$. *Suppose:*

(i) $F(s)$ *is analytic in* $Re(s) > 0$ *and bounded in* $Re(s) \geq 0$.
(ii) $F(0)$ *has strictly positive eigenvalues.*
(iii) For all $\gamma \in (0, 1]$, -1 *is not an eigenvalue of* $L(j\omega)$, $\omega \neq 0$.

Then the closed loop is stable for all $\gamma \in (0, 1]$.

In essence, the above conditions are a "nominal" stability requirement for small γ, that says that we have strictly negative feedback of enough rank to stabilize all the integrators, and a "robustness" argument that says we can perform a homotopy to $\gamma = 1$ without bifurcating into instability.

Applying this to the $L(s)$ in (14), we take $F(s) = \bar{R}_f(s)\mathcal{K}\bar{R}_b^T(s)\mathcal{C}$; we will later add the scaling γ. Note that (i) is automatically satisfied. Since

$$eig(F(0)) = eig(\mathcal{C}^{\frac{1}{2}}\bar{R}\mathcal{K}\bar{R}^T\mathcal{C}^{\frac{1}{2}}),$$

condition (ii) holds provided $\bar{R}$ has full row rank. Here we see the importance of putting the integrators at the links (the lower dimensional portion). If, instead, we tried to integrate at the sources, the resulting feedback matrix at DC would not have enough rank to stabilize the larger number of integrators.

What remains is to establish (iii). The key structure we will exploit in this problem is the equation

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

which follows from (5), and allows us to write $\bar{R}_b^T(s) = \mathrm{diag}(e^{-\tau_i s})\bar{R}_f^{\sim}(s)$, where $\bar{R}_f^{\sim}(s) = \bar{R}_f^T(-s)$ is the adjoint system. Bringing in the notation

$$X_0 = \mathrm{diag}(x_{0i}), \quad \mathcal{M} = \mathrm{diag}(\frac{1}{M_i}), \quad \Lambda(s) = \mathrm{diag}(\lambda_i(s)), \quad \lambda_i(s) = \frac{\alpha_i e^{-\tau_i s}}{\tau_i s},$$

we can now rewrite $L(s)$, for $s \neq 0$, as

$$L(s) = \bar{R}_f(s)X_0\mathcal{M}\Lambda(s)\bar{R}_f^{\sim}(s)\mathcal{C}. \tag{15}$$

We now tackle the robustness argument.

Theorem 1. *Consider an equilibrium point where rates match target capacity, i.e.* $c_0 = \bar{R}_f(0)x_0$. *Let* $\alpha_i < \frac{\pi}{2}$ *and the delays be arbitrary. Then with* $L(s)$ *as in (15),* $-1 \notin eig(L(j\omega))$, $\omega \neq 0$.

Proof: Since nonzero eigenvalues are invariant under commutation, and also many of the factors in (15) are diagonal, we observe that

$$-1 \in eig\Big(L(j\omega)\Big) \Longleftrightarrow -1 \in eig\Big(P(j\omega)\Lambda(j\omega)\Big),$$

$$P(j\omega) := \mathcal{M}^{\frac{1}{2}}X_0^{\frac{1}{2}}\bar{R}_f(j\omega)^*\ \mathcal{C}\ \bar{R}_f(j\omega)X_0^{\frac{1}{2}}\mathcal{M}^{\frac{1}{2}} \geq 0.$$

Claim:

$$0 \leq P \leq I. \tag{16}$$

This amounts to bounding the spectral radius

$$\rho(P) = \rho\Big(\mathcal{M}\bar{R}_f(j\omega)^*\ \mathcal{C}\ \bar{R}_f(j\omega)X_0\Big) \leq \|\mathcal{M}\bar{R}_f(j\omega)^*\| \cdot \|\mathcal{C}\ \bar{R}_f(j\omega)X_0\|.$$

Any induced norm will do, but if we use the l_∞-induced (max-row-sum) norm, we find that

$$\|\mathcal{C}\bar{R}_f(j\omega)X_0\|_{\infty-ind} = \max_l \frac{1}{c_{0l}} \sum_{i \text{ uses } l} |e^{-\tau_{i,l}^f j\omega} x_{0i}| = \max_l \frac{1}{c_{0l}} \sum_{i \text{ uses } l} x_{0i} = 1;$$

note we are dealing with bottlenecks. Also $\|\mathcal{M}\bar{R}_f^*\| = 1$, because each row contains exactly M_i elements of magnitude $1/M_i$. So $\rho(P) \leq 1$ as claimed. Indeed, $\rho(P) = 1$ at $\omega = 0$, the eigenvector being the vector of all ones.

Now suppose $-1 \in eig(P(j\omega)\Lambda(j\omega))$ for some ω. We thus have a vector u, $|u| = 1$ such that $y = \Lambda u, \quad u = -Py$. Now

$$u^*y = u^*\Lambda u = \sum_i \lambda_i |u_i|^2$$

is a convex combination of the $\{\lambda_i\}$, which are points in the curve Γ of Figure 3, scaled by $\alpha_i < \frac{\pi}{2}$. It is clear that such convex combinations and scaling cannot reach any point in the half-line $(-\infty, -1]$. However, we also have

$$1 + u^*y = u^*u + u^*y = y^*P(P-I)y \leq 0,$$

using (16). So $u^*y \in (-\infty, -1]$, a contradiction.

■

Remark 1. Some elements of the proof, in particular the use of l_∞ induced norms to prove a spectral radius bound, are inspired by the work of [6] for the control laws in [8]. More recently [17] has extended the stability argument for the laws in [8] in a parallel fashion to our work.

Theorem 1 establishes (iii) in Proposition 1; note that scaling down by γ is equivalent to making the α_i smaller. To summarize, we have:

Theorem 2. *Let $\bar{R}_f(s)$, $\bar{R}_b(s)$ denote the routing matrices of sources in relation to the bottleneck links. Suppose $\bar{R}_f(0) = \bar{R}_b(0)$ has full row rank, and that $\alpha_i < \frac{\pi}{2}$. Then the system with link control (9) and linearized source control (11) is locally stable for arbitrary delays and link capacities.*

Our stability theorem covers the simplest possible control laws consistent with our utilization requirement, namely integrators at links and static gains at sources. Could the argument be generalized to include additional dynamics? We give the following observations:

- Clearly one could include a fixed stable, inversely stable filter at all links, and its inverse at all sources, but this would have to be universally chosen.
- There can be no more pure integrators. Otherwise the Nyquist plot in Figure 3 would branch towards $-\infty$, and convex combinations of such points could reach the critical point. In particular, strategy in [2] of adding another integrator to clear link queues would not qualify.
- The source controller could include additional scalable dynamics, function of $\tau_i s$; this would result in a modified Nyquist curve Γ, which is acceptable as long as its convex hull does not touch the critical point.

4 Nonlinear laws and the equilibrium structure

4.1 Static source laws with scalable stability

We have provided the global law (9) with $\mu_l = \frac{1}{c_{0l}}$ for price generation at the links, but so far we have only characterized sources by their linearization (11). For static source control laws, however, specifying its linearization at every equilibrium point essentially determines its nonlinear structure.

Consider a static source control of the form $x_i = f_i(q_i, \tau_i, M_i)$. The linearization requirement (11) imposes that

$$\frac{\partial f_i}{\partial q_i} = -\frac{\alpha_i f_i}{M_i \tau_i},$$

for some $0 < \alpha_i < \pi/2$. Let us assume initially that α_i is constant. Then the above differential equation can be solved analytically, and gives the static source control law

$$x_i = f_i(q_i) := x_{\max,i}\, e^{-\frac{\alpha_i q_i}{M_i \tau_i}}. \tag{17}$$

Here $x_{\max,i}$ is a maximum rate parameter, which can vary for each source, and can also depend on M_i, τ_i (but not on q_i). This *exponential backoff* of source rates as a function of aggregate price can provide the desired control law, together with the link control in (9).

We can achieve more freedom in the control law by letting the parameter α_i be a function of the operating point: in general, we would allow any mapping $x_i = f_i(q_i)$ that satisfies the differential inequality

$$0 \geq \frac{\partial f_i}{\partial q_i} \geq -\frac{\pi}{2}\frac{f_i}{M_i \tau_i}. \tag{18}$$

The essential requirement is that the slope of the source rate function (the "elasticity" in source demand) decreases with delay τ_i, and with the number of bottlenecks M_i.

So we find that in order to obtain this very general scalable stability theorem, some restrictions apply to the sources' demand curves (or their utility functions). This is undesirable from the point of view of our objective 3 in Section 2.3; we would prefer to leave the utility functions completely up to the sources; in particular, to have the ability to allocate equilibrium rates independently of the RTT. We remark that parallel work in [18] has derived solutions with scalable stability and arbitrary utility functions, but where the link utilization requirement is relaxed. Indeed, it appears that one must choose between the equilibrium conditions on either the source or on the link side, if one desires a scalable stability theorem. Below we show how this difficulty is overcome if we slightly relax our scalability requirement.

4.2 A lead-lag alternative for source control

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

$$\delta x_i = -\varphi_i(s)\delta q_i = -\frac{\kappa_i(s+z)}{s+\frac{z\kappa_i}{\nu_i}}\delta q_i. \tag{19}$$

Here the high frequency gain κ_i is the same as in (11), "socially acceptable" from a dynamic perspective. The DC gain $\nu_i = -f_i'(q_{i0})$ is the elasticity of source demand based on its own "selfish" demand curve $x_{i0} = f_i(q_{i0})$ (here f_i need no longer be of the form (17). The *zero* z is assumed fixed across sources.

Can a stability theorem be obtained under these new laws? The main requirement would be that at cross-over frequency all sources respond according to their high-frequency gain, so that the previous analysis applies. The difficulty is that this implies a common agreement on the frequency scale, which means forgoing complete scalability with respect to time delay. While less elegant, this is not too serious in practice, where one can assume a known bound on the network's RTT. We have the following result.

Proposition 2. *Assume that for every source i, $\tau_i \le \bar{\tau}$. In the assumptions of Theorem 2 replace the source control by (19), with $\alpha_i = \alpha < \frac{\pi}{2}$ and a $z = \frac{\eta}{\bar{\tau}}$. Then for a small enough $\eta \in (0,1)$ depending only on α, the closed loop is linearly stable.*

We omit the proof, but remark that it is based again on a Nyquist argument via the eigenvalues of the loop transfer function $L(j\omega)$; a perturbed version of the argument in Theorem 1 is used at frequencies above $\frac{1}{\bar{\tau}}$, and a different argument at low frequencies; the fact that the source zero z is fixed across sources is essential to this decomposition.

4.3 Nonlinear implementation of dynamic source laws

We wish to find a source control law whose equilibrium matches the desired utility function, $U_i'(x_{0i}) = q_{0i}$, (equivalently, the demand curve $x_{i0} = f_i(q_{i0})$), and with linearization (19). This is not as easy as before, in particular the ability to fix the zero z independently of the operating point and the RTT.

Below is a candidate solution, of a similar nature to the "primal" laws proposed in [8].

$$\tau_i\dot{\xi}_i = \beta_i(U_i'(x_i) - q_i), \tag{20}$$

$$x_i = x_{m,i}e^{\left(\xi_i - \frac{\alpha_i q_i}{M_i\tau_i}\right)}. \tag{21}$$

Note that (21) corresponds exactly to the rate control law in (17), with the change that the parameter $x_{\max}$ is now varied exponentially as

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

with ξ_i as in (20). If β_i is small, the intuition is that the sources use (17) at fast time-scales, but slowly adapt their $x_{\max_i}$ to achieve an equilibrium rate that matches their utility function, as follows clearly from equation (20).

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

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

Here we have used the fact that $U''(x_0) = \frac{1}{f'(q_0)} = -\frac{1}{\nu}$. This leads after some algebra in Laplace to the transfer function

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

This is exactly of the form in (19) if we take

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

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

We recapitulate the main result as follows.

Theorem 3. *Consider the source control (20-21) where $U_i(x_i)$ is the source utility function, and the link control (9). At equilibrium, this system will satisfy the desired demand curve $x_{i0} = f_i(q_{i0})$, and the bottleneck links will satisfy $y_{0l} = c_{0l}$, with empty queues. Furthermore, under the rank assumption in Theorem 2, $\alpha_i < \frac{\pi}{2}$, and $z = \frac{\beta_i M_i}{\alpha_i}$ chosen as in Proposition 2, the equilibrium point will be locally stable.*

We have thus satisfied all the objectives set forth in Section 2.3, except for the fact that an overall bound on the RTT had to be imposed.

5 Signaling requirements

We briefly discuss here the information needed at sources and links to implement our dynamic laws, and the resulting communication requirements.

Links generate prices by integrating the excess flow $y_l - c_{0l}$ with respect to the virtual capacity; this is easily implemented by maintaining a "virtual queue" variable, incremented upon packet arrival, and decremented at the virtual capacity rate. Note that true bottlenecks will operate away from saturation, so the integrator model (2) for this queue is justified.

The resulting price must be communicated to sources in additive way across links. For this purpose we can employ the Explicit Congestion Notification bit available in the packet header, and the technique of random exponential marking [2]: here the bit would be marked at link l with probability $1 - \phi^{-p_l}$, where $\phi > 1$ is a global constant. Assuming independence, the overall probability that a packet from source i gets marked is (see [2])

$$1 - \phi^{-q_i},$$

and therefore q_i can be estimated from marking statistics. Note that the estimation process will add noise, and additional delay in the feedback loop. The latter can be accounted for in the source compensation.

Sources must have access to the round-trip time τ_i, which can be obtained by timing packets and their acknowledgments. They also need the bound M_i on the number of bottlenecks, which is not so easy to obtain, although it can be argued that in practice this number is typically not large (e.g. 2 bottlenecks per source). Alternatively, one could think of using another ECN bit to communicate this information. Once a rate is computed by the source, (1) can be used to set the congestion window.

An initial implementation of such protocol has been programmed in the standard simulator ns-2 [13]; while validation work is in process, early results are encouraging.

6 Conclusion

The abstraction of fluid-flow models has allowed us to cast the congestion control problem in the familiar language of linear multivariable control. Although, due to decentralization, feedback design can only be handled in an ad-hoc way, we have found that the special structure of the problem allows us to go after a very ambitious objective: scalable local stability for arbitrary networks and delays, together with tracking of link utilization. When in addition we want to give sources the freedom of choosing their rate demand curves, we found a solution based on separation of time-scales assuming a known bound on the round-trip times.

Going from flow models and theorems to actual protocols based on packet level mechanisms, requires of course a layer of "hacks" and experimentation. Whether this transition will eventually yield viable new protocols will depend on engineering aspects which are mostly outside the scope of the theory; for instance, the restriction of one ECN bit per packet, or the issue of incremental deployment of these protocols in the current network. Regardless of this

outcome, it is reassuring to discover that control theory is still relevant in the world of complex networks.

References

1. E. Altman, T. Basar and R. Srikant. "Congestion control as a stochastic control problem with action delays", *Automatica*, December 1999.
2. Sanjeewa Athuraliya, Victor H. Li, Steven H. Low, and Qinghe Yin, "REM: active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, May/June 2001.
3. D.D. Clark, "The design philosophy of the DARPA Internet protocols", *Proc. ACM SIGCOMM '88, in: ACM Computer Communication Reviews*, Vol. 18, No 4., pp. 106-114, 1988.
4. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", *IEEE/ACM Trans. on Networking*, 1(4):397-413, Aug. 1993.
5. V. Jacobson, "Congestion avoidance and control", *Proc. ACM SIGCOMM '88.*
6. R. Johari and D. Tan, "End-to-End Congestion Control for the Internet: Delays and Stability", *Cambridge Univ. Statistical Lab. Research Report 2000-2.*
7. F. P. Kelly, "Mathematical modeling of the Internet", Fourth International Congress on Industrial and Applied Mathematics, Edinburgh, Scotland, July 1999.
8. F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness, and stability". *Jour. Oper. Res. Soc.*, 49(3), pp. 237-252, 1998.
9. S. Kunniyur and R. Srikant, "Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management", Sigcomm 2001, San Deigo, Aug 2001
10. S. H. Low and D. E. Lapsley, "Optimization flow control – I: basic algorithm and convergence" *IEEE/ACM Trans. on Networking*, Vol 7(6) Dec 1999.
11. S. H. Low, F. Paganini, J. Wang, S. A. Adlakha, and J. C. Doyle. "Dynamics of TCP/RED and a scalable control", *Proc. IEEE Infocom 2001,*, New York.
12. Steven H. Low, Fernando Paganini, and John C. Doyle, "Internet congestion control" *IEEE Control Systems Magazine*, February 2002.
13. F. Paganini, Z. Wang, S. H. Low, J. Wang and J. C. Doyle. "A new TCP/AQM for Stable Operation in Fast Networks" in preparation.
14. S. Mascolo, "Congestion control in high-speed communication networks using the Smith principle", *Automatica*, 1999.
15. V. Misra, W.-B Gong, and D. Towsley. "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED." In *Proceedings of ACM SIGCOMM*, 2000.
16. H. Ozbay, S. Kalyanaraman, A. Iftar, "On rate-based congestion control in high-speed networks: design of an H_∞ based flow controller for single bottleneck", *Proc. American Control Conference*, 1998.
17. G. Vinnicombe, "On the stability of networks operating TCP-like congestion control", to appear in *2002 IFAC World Congress*, Barcelona, Spain.
18. G. Vinnicombe, "Robust congestion control for the Internet", preprint, Feb 2002.