Analysis and design of congestion control in synchronised communication networks

R.N.Shorten*, D.J.Leith*, J.Foy, R.Kilduff
Hamilton Institute, NUI Maynooth

June 20, 2003

Abstract

We present a simplified model of a network of TCP-like sources that compete for a shared bandwidth. We show that: (i) networks of communicating devices operating AIMD congestion control algorithms may be modelled as a positive linear system; (ii) that such networks possess a unique stationary point; and (iii) that this stationary point is globally exponentially stable. Using these results we establish conditions for the fair co-existence of traffic in networks employing heterogeneous AIMD algorithms. A new protocol for operation over high-speed links is proposed and its dynamic properties discussed as a positive linear system.

1 Introduction

A basic problem in the design of networks is the development of congestion control algorithms. Conventional congestion control algorithms were deployed for two principal reasons: (a) to ensure avoidance of network congestion collapse [1, 2] ; and (b) to ensure a degree of network fairness. Roughly speaking, network fairness refers to the situation whereby a data source receives a fair share of available bandwidth, whereas congestion collapse refers to the situation whereby an increase in network load results in a decrease of useful work done by the network (usually due to retransmission of data). Attempts to deal with network congestion have resulted in the widely applied Transmission Control Protocol (TCP) [3]. While the current TCP congestion control algorithm has proved remarkably durable, it is likely to be less effective on next generation networks featuring gigabit-speed connectivity and heterogeneous traffic and sources. These considerations have led to widespread acceptance that new congestion control algorithms must be developed to accompany the realization of next generation systems (and perhaps also to better exploit the resources of existing networks) [3, 4, 5].

The task of developing such algorithms is non-trivial. In addition to the requirement of avoiding congestion collapse, fundamental requirements of congestion control algorithms include: efficient use of bandwidth; fair allocation of bandwidth among sources; and responsiveness (the network should rapidly reallocate bandwidth as required). These requirements must be met while respecting key constraints including: decentralised design (TCP sources have restricted information available to them); scalability (the qualitative properties of networks employing congestion control algorithms should be independent of the size of the network and of a wide variety of network conditions); and suitable backward compatibility

*Joint first author

with conventional TCP sources. We argue that tools from the design of feedback systems, systems theory and hybrid systems, based upon appropriate mathematical models, may provide a framework, and an enabling technology, for the principled design of such protocols.

In this paper we consider a framework for the design and analysis of networks of devices that compete for available bandwidth (or more generally a shared resource). Under certain simplifying assumptions, we show that: (i) networks of communicating devices operating additive-increase multiplicative-decrease (AIMD) congestion control algorithms may be modelled as a positive linear system; (ii) such networks possess a unique stationary point; and (iii) this stationary point is globally exponentially stable. Using these results we also establish conditions for fair co-existence of traffic in networks employing a mix of AIMD algorithms.

2 Definitions and mathematical preliminaries

Throughout, the following notation is adopted: $\mathbb{R}$ denotes the real numbers; Z denotes the integers; $\mathbb{R}^n$ denotes the n-dimensional real Euclidean space; $\mathbb{R}^{n\times n}$ denotes the space of $n \times n$ matrices with real entries; x_i denotes the i^{th} component of the vector x in $\mathbb{R}^n$; a_{ij} denotes the entry in the (i, j) position of the matrix A in $\mathbb{R}^{n\times n}$. We use the symbol $\succ$ to denote that the entries of a matrix (vector) are greater than zero. We say that the matrix A is strictly positive if all entries of the matrix are positive; namely, $A \succ 0$.

Strictly positive matrices will play an important role in developing the results in this paper. We note the following important theorem for strictly positive matrices.

Theorem 2.1 *[6, 7, 8] Let $A \in \mathbb{R}^{n\times n}$ be a strictly positive matrix. Then: (i) there is an eigenvalue $\rho(A)$ that is simple and whose magnitude is greater that any other eigenvalue; (ii) A has a positive eigenvector x_p corresponding to $\rho(A)$ and any non-negative eigenvector of A is a multiple of x_p; and (iii) $\lim_{N\to\infty}(\frac{1}{\rho(A)}A)^N = x_p y_p^T$ where $A^T y_p = \rho(A) y_p$, $x_p \succ 0, y_p \succ 0$, and $x_p^T y_p = 1$.*

Corollary 2.1 *Let $A \in \mathbb{R}^{n\times n}$ with $A \succ 0$. There exists a unique vector x_p such that $Ax_p = \rho(A)x_p, x_p \succ 0$ and $\sum_{i=1}^n x_p = 1$. We refer to x_p as the Perron eigenvector of the matrix A and $\rho(A)$ as the Perron eigenvalue of the matrix A.*

3 A network model

A network consists of sources and sinks connected together via links and routers.

3.1 Links

The path between source-sink pairs consists of wireline or wireless links. We focus here on wireline links, which can be modelled as a constant propagation delay together with a queue to buffer traffic. This is typically a FIFO queue which simply drops packets that arrive when the queue is full. This is the so-called *drop-tail* queueing discipline.

3.2 TCP Sources

The TCP standard defines a variable *cwnd* called the congestion window. This variable determines the number of unacknowledged packets that can be in transit at any time, i.e.

the number of packets in the 'pipe' formed by the links and buffers in a transmission path. When the window size is exhausted, the source must wait for an acknowledgement (ACK) before sending a new packet. Congestion control is achieved by dynamically adapting the window size according to an additive-increase multiplicative-decrease (AIMD) law. The basic idea is for a source to gently probe the network for spare capacity and rapidly back-off its send rate when congestion is detected.

In the congestion avoidance phase, when a source i receives an ACK packet it increments its window size $cwnd_i$ according to the additive increase law

$$cwnd_i \rightarrow cwnd_i + \alpha_i / cwnd_i \tag{1}$$

where $\alpha_i = 1$ for standard TCP. Consequently, the source gradually ramps up its send rate as the number of packets successfully transmitted grows. By keeping track of the ACK packets received, the source can infer when packets have been lost en route to the destination. On detecting a loss in this way, the source enters the fast recovery phase. The lost packets are retransmitted and the window size $cwnd_i$ of source i is reduced according to

$$cwnd_i \rightarrow \beta_i cwnd_i \tag{2}$$

where $\beta_i = 0.5$ for standard TCP . It is assumed that multiple drops within a single round-trip time lead to a single back-off action. When receipt of the retransmitted lost packets is eventually confirmed by the destination, the source re-enters the congestion avoidance phase, adjusting its window size according to (1). In summary, on detecting a dropped packet (which the algorithm assumes is an indication of congestion on the network), the TCP source reduces its send rate. It then begins to gradually increase the send rate again, probing for available bandwidth. A typical window evolution is depicted in Figure 1 ($cwnd_i$ at the time of detecting congestion is denoted by w_i in this figure).

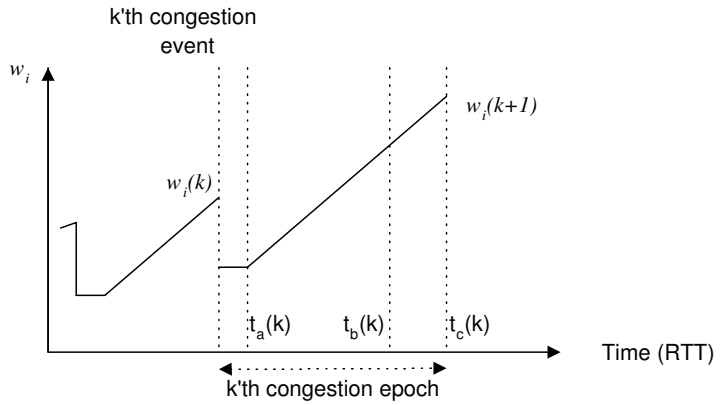


Figure 1: Evolution of window size

Over the kth congestion epoch three important events can be discerned: indicated by $t_a(k), t_b(k)$ and $t_c(k)$ in Figure 1. The time $t_a(k)$ is the time at which the number of unacknowledged packets in the pipe equals $\beta_i w_i(k)$; $t_b(k)$ is the time at which the pipe is full so that any packets subsequently added will be dropped at the congested queue; $t_c(k)$ is the time at which packet drop is detected by the sources. Note that we measure time in units of round-trip time (RTT). RTT is the time taken between a source sending a packet and receiving the corresponding acknowledgement, assuming no packet drop. The update law (1) corresponds to an increase in $cwnd_i$ of α_i packets per RTT.

3.3 Network model under synchronisation

We consider a network of n AIMD sources. Each source is parameterized by an additive increase parameter and a multiplicative decrease factor, denoted α_i and β_i respectively. These parameters satisfy $\alpha_i > 0$ and $0 < \beta_i < 1$ $\forall i \in \{1, ..., n\}$. We assume that the event times t_a, t_b and t_c indicated in Figure 1 are the same for every source i.e. that the sources are synchronised. This synchronisation condition is valid, for example, when sources are constrained by a shared congested link, the round-trip propagation delay between each source and destination is identical and each source transmits at least one extra packet per round-trip time (i.e. $\alpha_i \geq 1$).

Let $w_i(k)$ denote the congestion window size of source i immediately before the kth network congestion event is detected by the sources, see Figure 1. It follows from the definition of the AIMD algorithm that the window evolution is completely defined over all time instants by knowledge of the $w_i(k)$ and the event times $t_a(k), t_b(k)$ and $t_c(k)$ of each congestion epoch. We therefore only need to investigate the behaviour of these quantities.

We have that $t_c(k) - t_b(k) = 1$; namely, each source is informed of congestion exactly one RTT after the first dropped packet was transmitted. Also,

$$w_i \geq 0, \sum_{i=1}^{n} w_i = P + \sum_{i=1}^{n} \alpha_i \tag{3}$$

where P is the maximum number of packets which can be held in the 'pipe'; this is usually equal to $q_{max} + BT$ where q_{max} is the maximum queue length of the congested link, B is the service rate in packets per second and T is the round-trip time. At the $(k+1)$th congestion event

$$w_i(k+1) \quad = \quad \beta_i w_i(k) + \alpha_i [t_c(k) - t_a(k)]. \tag{4}$$

and

$$t_c(k) - t_a(k) \quad = \quad \frac{1}{\sum_{i=1}^{n} \alpha_i} [P - \sum_{i=1}^{n} \beta_i w_i(k)] + 1 \tag{5}$$

Substituting into (5) from (3) yields

$$t_c(k) - t_a(k) \quad = \quad \frac{1}{\sum_{i=1}^{n} \alpha_i} [\sum_{i=1}^{n} (1 - \beta_i) w_i(k)] \tag{6}$$

Hence,

$$w_i(k+1) \quad = \quad \beta_i w_i(k) + \frac{\alpha_i}{\sum_{j=1}^{n} \alpha_i} [\sum_{i=1}^{n} (1 - \beta_i) w_i(k)] \tag{7}$$

The dynamics of the entire network can be described by writing all n equations in matrix form:

$$W(k+1) = AW(k) \tag{8}$$

where $W^T(k) = [w_1(k), \cdots, w_n(k)]$, and

$$\begin{aligned} A \quad &= \quad \begin{bmatrix} \beta_1 & 0 & \cdots & 0 \\ 0 & \beta_2 & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \beta_n \end{bmatrix} + \frac{1}{\sum_{j=1}^{n} \alpha_i} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdots \\ \alpha_n \end{bmatrix} \begin{bmatrix} 1-\beta_1 & 1-\beta_2 & \cdots & 1-\beta_n \end{bmatrix} \\ &= \quad diag[\beta_i] + \frac{1}{\sum_{j=1}^{n} \alpha_i} g^T h, \end{aligned} \tag{9}$$

with $g^T = [\alpha_1, \alpha_2, \cdots, \alpha_n]$ and $h^T = [1-\beta_1, 1-\beta_2, \cdots, 1-\beta_n]$. Note that the initial condition $W(0)$ is subject to constraint (3) (this simply ensures that the window sizes specified by $W(0)$ are non-negative and correspond to a congestion event).

It follows that the synchronised network (8) is a positive linear system and that the matrix A is strictly positive: $A \succ 0$ since $\alpha_i > 0$, $0 < \beta_i < 1$, $\forall\, i \in \{1, ..., n\}$.

Comment : We note that this model incorporates a number of key features of real networks: the hybrid nature of AIMD algorithms (e.g. [9], [10]); time-varying communication delays on links; and drop-tail queueing. The model also encompasses a network 'ecosystem' where sources employ AIMD algorithms with different increase and decrease parameters.

Comment (High-speed TCP) : The foregoing model is derived for AIMD sources where the increase and decrease parameters, α_i and β_i, are constant (although they may differ for each source). A number of recent proposals for high-speed networks envisage varying the rate of increase and decrease as functions of window size etc. The synchronisation model is readily extended to these protocols by extending the model to include time varying parameters $\alpha_i(k)$ and $\beta_i(k)$ and defining the model increase parameter to be an *effective* value i.e. such that $\alpha_i(k) = (w_i(k+1) - \beta_i w_i(k))/(t_c(k) - t_a(k))$.

4 Convergence and Stability

We now present the main mathematical results of the paper.

Theorem 4.1 *Let A be defined as in Equation (9). Then: (i) the Perron eigenvalue of A is given by $\rho(A) = 1$; (ii) the Perron eigenvector of A is given by $x_p^T = \gamma[\frac{\alpha_1}{1-\beta_1}, ..., \frac{\alpha_n}{1-\beta_n}]$, where $\sum_{i=1}^n \gamma x_{pi} = 1$; (iii) the associated eigenvector y_p^T of A^T is $[1, 1, ..., 1]$.*

Proof : Since A is a positive matrix, the Perron eigenvector is the only positive eigenvector. It follows by inspection that x_p is the Perron eigenvector, $\rho(A) = 1$ and $y_p^T = [1, 1, ..., 1]$.

Corollary 4.1 *For a network of synchronised time-invariant AIMD sources: (i) the network has a Perron eigenvector $x_p^T = \gamma[\frac{\alpha_1}{1-\beta_1}, ..., \frac{\alpha_n}{1-\beta_n}]$; and (ii) the Perron eigenvalue is $\rho(A) = 1$. It follows from Theorem 2.1 that all other eigenvalues of A satisfy $|\lambda_i(A)| < \rho(A)$ and therefore the network possesses a stationary point to which it converges asymptotically. The rate of convergence of the network to W_{ss} depends upon the second largest eigenvalue of A.*

Proof : We are interested in the evolution of the system

$$W(k+1) = AW(k), \tag{10}$$

where A is defined by (9). The convergence of this system is determined by

$$\lim_{k\to\infty} W(k) = \lim_{k\to\infty} A^k W(0),$$

From Theorem (2.1) we have

$$\lim_{k\to\infty} (\frac{1}{\rho(A)} A)^k = x_p y_p^T,$$

where x_p is the Perron eigenvector of A, and y_p is the associated eigenvector of A^T. Hence, it follows that

$$\lim_{k\to\infty} W(k) \quad = \quad \lim_{k\to\infty} A^k W(0)$$

$$
\begin{aligned}
&= x_p y_p^T W(0), \\
&= \Theta x_p,\ \Theta = y_p^T W(0) = \sum_{i=1}^{n} w_i(0).
\end{aligned}
$$

The rate of convergence of A^k depends on the second largest eigenvalue of A (see item (j) on page 498 in [7]).

Q.E.D.

Theorem 4.2 *Consider a network of synchronised time-invariant AIMD sources (8) subject to constraint (3). Then: (i) the network dynamics may be written*

$$W(k+1) = \bar{A}W(k) + W_{ss} \tag{11}$$

where $\bar{A} = A - x_p y_p^T$ and W_{ss} is the unique stationary point of (11); (ii) the stationary point W_{ss} is globally exponentially stable.

Proof :

Part (i) : We have,

$$W(k+1) - W_{ss} = AW(k) - W_{ss}. \tag{12}$$

with $W_{ss} = x_p y_p^T W(0)$. Recall that $Ax_p = x_p$ since x_p is an eigenvector of A corresponding to the Perron eigenvalue ($\rho(A) = 1$). Also, $y_p^T W(0) = y_p^T W(k) = \Theta$ is constant owing to constraint (3). Hence, $AW_{ss} = W_{ss}$ and

$$W(k+1) = \bar{A}W(k) + W_{ss} \tag{13}$$

as claimed. It follows that the stationary point W_{ss} is unique since $y_p^T W(0) = y_p^T W(k)$ for all $k \in Z,\ k > 0$.

Part (ii) : Let $J = M^{-1}AM$ denote the real Jordan matrix similar to A, with M the associated modal matrix. The Perron eigenvalue is real ($\rho(A) = 1$) and simple and so the associated Jordan block is scalar. Assume that the rows of J are ordered such that this Jordan block is the (1,1) element of J. Letting x_p denote the Perron eigenvector of A and y_p the associated eigenvector of A^T, we have that $M^{-1}x_p = [1, 0, \cdots, 0]^T$ and $y_p^T M$ is of the form $y_p^T M = [1, *, \cdots, *]^T$ where the $*$'s denote suitable values. Hence,

$$\bar{J} = M^{-1}(A - x_p y_p^T)M = J - \begin{bmatrix} 1 & * & \cdots & * \\ 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{14}$$

Evidently, the matrix $\bar{J}$ is identical to the J apart from the first row, in which element (1,1) is 0. Since J is upper triangular, $\bar{J}$ has the same eigenvalues as J apart from the Perron eigenvalue which is replaced by a zero eigenvalue in $\bar{J}$. That is, the eigenvalues of $\bar{J}$ all lie within the unit circle. Hence, the eigenvalues of $\bar{A}$ also lie within the unit circle and the network dynamics are globally exponentially stable with fixed point W_{ss}.

Q.E.D.

Comment (Fair allocation of network bandwidth) : Let $\alpha_i = \lambda(1-\beta_i)\ \forall i$ and for some $\lambda > 0$. Then $W_{ss}^T = \Theta/n\ [1, 1, ..., 1]$ i.e. $w_1 = w_2 = ... = w_n$. For networks where the queueing delay is small relative to the propagation delay, the send rate is essentially proportional to

the window size. In this case, it can be seen that $\alpha_i = \lambda(1-\beta)\forall i \in \{1,..,n\}$ is a condition for a fair allocation of network bandwidth. For the standard TCP choices of $\alpha = 1$ and $\beta = 0.5$, we have $\lambda = 2$ and the condition for other AIMD flows to co-exist fairly with TCP is that they satisfy $\alpha_i = 2(1-\beta_i)$; see Figure 2 for an example. We can see from Figure 2 that the concepts of fairness and efficient utilization of network bandwidth are distinct.

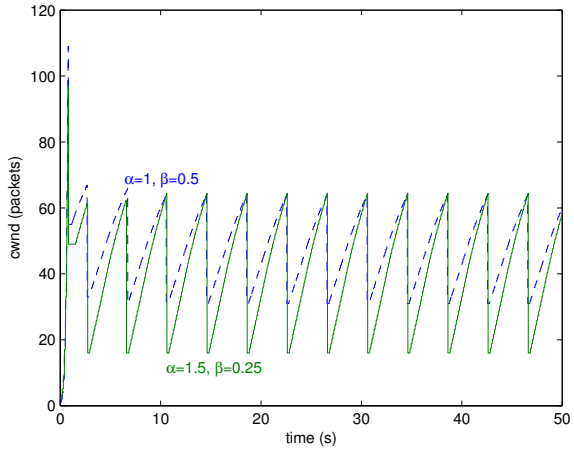


Figure 2: Example of co-existence of two TCP sources with different increase and decrease parameters. (NS simulation, network parameters: 10Mb bottleneck link, 100ms delay, queue 40 packets).

Comment (Convergence) : We note the two following cases where convergence, measured in number of congestion epochs, does not depend on the network α_i.

(i) All of the sources share the same increase parameter: $\alpha_1 = \alpha_2 = \cdots \alpha_n = \alpha$, and

$$A \quad = \quad diag[\beta_i] + \frac{1}{n}\left[\begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array}\right] h \tag{15}$$

Under these conditions the network dynamics (and so the rate of convergence) depends solely on the decrease parameters β_i.

(ii) All of the sources share the same decrease parameter (the α_i need not be the same for all sources): $\beta_1 = \beta_2 = \cdots = \beta_n = \beta$, and the eigenvalues of A (other than the Perron eigenvalue) have value β. Thus, the rate of convergence to a fixed point is β^k where k is the congestion epoch and using this it follows, for example, that the 95% rise time is $\log 0.05/\log\beta$ (giving a rise time of 4 congestion epochs when $\beta = 0.5$). Note that the ratio of the α_i to β determines the duration of the congestion epochs according to the relation (6).

5 High-speed networks

While the current TCP congestion control algorithm has proved remarkably durable, it is often inefficient on modern high-speed networks [11, 12]. On such links the window sizes can be very large (perhaps tens of thousands of packets). Following a congestion event, the window size is halved and subsequently only increased by one packet per round-trip time. Thus, it can take a substantial time for the window size to recover, during which time the send rate is well below that of the link. One possible solution is to simply make

the TCP increase parameter α larger, thereby decreasing the time to recover following a congestion event and improving the responsiveness of the TCP flow. Unfortunately, this direct solution is inadmissable because of the requirement on lower speed networks for backward compatibility and fairness with existing TCP traffic (this requirement is relaxed in high-speed networks [12]). The requirement is thus for α to be large in high-speed networks but unity in low-speed ones, naturally leading to consideration of some form of mode switch. However, mode switching creates the potential for introducing undesirable dynamic behaviours in otherwise well behaved systems and any re-design of TCP therefore needs to be carried out with due regard to such issues.

In this section we consider amending TCP to include a high-speed and a low-speed mode. In the high-speed mode the increase parameter of source i is α_i^H and in the low-speed mode α_i^L. On congestion, we back-off to $\beta w_i(k) - \delta_i$, with $\delta_i = 0$ in low-speed mode and $\delta_i = \beta(\alpha_i^H - \alpha_i^L)$ in high-speed mode[1]. The mode switch is governed by:

$$\alpha_i \quad = \quad \begin{cases} \alpha_i^L & cwnd_i - (\beta w_i(k) - \delta_i) \leq \Delta^L \\ \alpha_i^H & cwnd_i - (\beta w_i(k) - \delta_i) > \Delta^L \end{cases} \tag{16}$$

where $cwnd_i$ is the current congestion window size of the ith TCP source, $\beta w_i(k) - \delta_i$ is the size of the congestion window immediately after the last congestion event, α_i^L is the increase parameter for the low-speed regime (unity for backward compatibility), α_i^H is the increase parameter for the high-speed regime, β is the decrease parameter as usual and Δ^L is the threshold for switching from the low to high speed regimes[2]. We refer to this strategy as H-TCP and a typical congestion epoch is illustrated in Figure 3.

Comment : This switching strategy differs from that proposed in [11, 12] in two key respects. Firstly, for high-speed networks a mode switch takes place in every congestion epoch. Secondly, the strategy (16) leads to a symmetric network; that is, one where the *effective* α_i and β_i are the same for all H-TCP sources experiencing the same duration of congestion epoch.

Our strategy is motivated by the following design criteria.

(i) Sources deploying H-TCP should behave as a normal TCP-source when operating on low-speed communication links. Such behaviour is *guaranteed* by (16) since it tests the low/high speed status of the network every congestion epoch.

(ii) Normal AIMD sources competing for bandwidth should be guaranteed some (small) share of the available bandwidth.

(iii) H-TCP sources competing against each other should receive a fair share of the bandwidth. This is guaranteed using symmetry and Theorem 4.1.

(iv) H-TCP sources should be responsive. Again, this is guaranteed using symmetry and an appropriate value of β combined with a value of α that ensures that the congestion epochs are of suitably short duration.

A model for the dynamics of this high-speed variant of TCP is derived, under the assumption of synchronisation, as follows. Consider n flows, with the ith flow having window increase parameters α_i^L and α_i^H (with $\alpha_i^H \geq \alpha_i^L > 0$), decrease parameter $0 < \beta < 1$ and high-speed threshold Δ^L. Note that when $\alpha_i^H = \alpha_i^L$ we recover the standard AIMD algorithm for the ith flow. The window sizes evolve according to

$$w_i(k+1) \quad = \quad \beta w_i(k) - \delta_i + \alpha_i^L(t_b(k) - t_a(k)) + \alpha_i^H(t_d(k) - t_b(k)) \tag{17}$$

[1]This ensures that the combined initial window size $\sum_{i=1}^n (\beta w_i(k) - \delta_i)$ following a congestion event is the same regardless of the source modes before congestion

[2]This strategy can be developed to include several mode switches

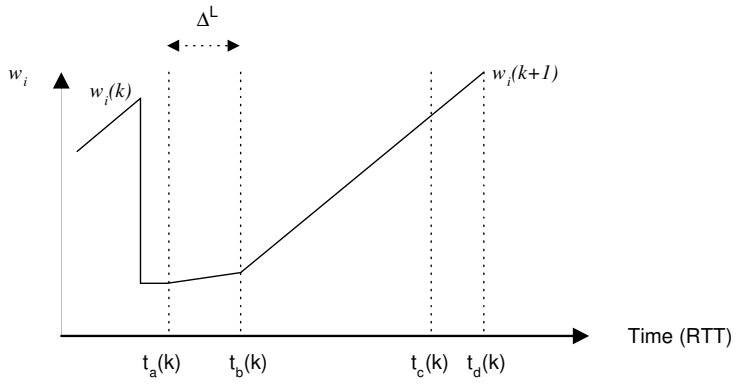


Figure 3: Evolution of window size with high-speed H-TCP algorithm

with

$$t_b(k) - t_a(k) \quad = \quad min[\Delta^L, \frac{1}{\sum_{i=1}^n \alpha_i^L} \sum_{i=1}^n (1-\beta) w_i(k) - \delta_i] \tag{18}$$

and

$$t_d(k) - t_b(k) \quad = \quad \begin{cases} 0 & t_b(k) - t_a(k) \leq \Delta^L \\ \frac{1}{\sum_{i=1}^n \alpha_i^H} \sum_{i=1}^n [(1-\beta) w_i(k) - \delta_i - \alpha_i^L \Delta^L] & otherwise \end{cases} \tag{19}$$

As usual the initial condition is subject to the constraint that $\sum_{i=1}^n w_i(0) = P + \sum_{i=1}^n \alpha_i^H$ to ensure that the window sizes specified correspond to a congestion event.

While the dynamics involve a mode switch, it is straightforward to verify that switching sequences with more than a single switch do not exist as solutions to these dynamics. That is, repeated switching is not possible. We proceed as follows. When the sources operate in high-speed mode, the dynamics of the network can be described in the matrix form

$$W(k+1) \quad = \quad \bar{A} W(k) + W_{ss} + V \tag{20}$$

where $\bar{A}$ and W_{ss} are defined as in Theorem 4.2 and

$$A \quad = \quad \beta I + \frac{1}{\sum_1^n \alpha_i^H} g h^T \tag{21}$$

with $g^T = [\alpha_1^H, ..., \alpha_n^H], h^T = [1-\beta, ..., 1-\beta]$, and V is an n-vector whose jth element is given by $v_j = \alpha_j^L \Delta^L - \delta_j - \frac{\alpha_j^H}{\sum_{i=1}^n \alpha_i^H} \sum_{i=1}^n [\delta_i + \alpha_i^L \Delta^L]$.

The convergence, fairness and stability results of the previous section apply directly. That is, any network possesses a unique stationary point and this point is globally exponentially stable; and the stationary point is fair for sources with matching increase and decrease parameters.

The performance of this high-speed algorithm is illustrated in Figure 4 using an NS packet-level simulation. Two high-speed flows with the same increase and decrease parameters are shown. As expected, the stationary solution is fair. It can be seen that convergence is rapid,

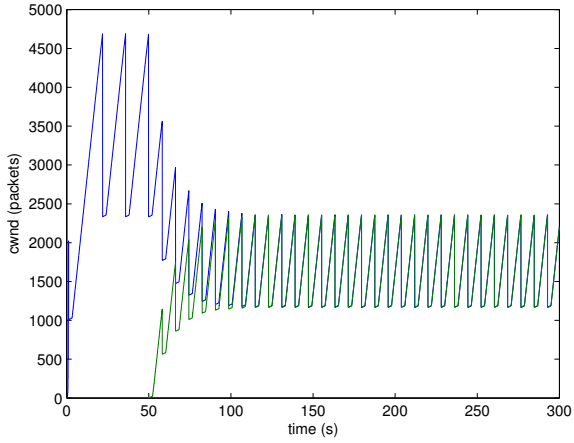


Figure 4: Example of two H-TCP flows illustrating rapid convergence to fairness - the second flow experiences a drop early in slow-start focussing attention on the responsiveness of the congestion avoidance algorithm (NS simulation, network parameters: 500Mb bottleneck link, 100ms delay, queue 500 packets; TCP parameters: $\alpha^L = 1, \alpha^H = 20, \beta = 0.5, \Delta^L = 19$ corresponding to a window size threshold of 38 packets).

taking approximately 4 congestion epochs which is in agreement with the rise time analysis for $\beta = 0.5$.

An important consideration in the design of H-TCP is backward compatibility; namely when deployed in low-speed networks H-TCP sources should co-exist fairly with sources deploying standard TCP ($\alpha = 1, \beta = 0.5$). This requirement introduces the constraint that $\alpha^L = 1, \beta = 0.5$. When the duration of the congestion epochs is less than Δ^L, the effective increase parameter for high-speed sources is unity and the fixed point is fair when a mixture of standard and high-speed flows co-exist. When the duration of the congestion epochs exceeds Δ^L, the network stationary point may be unfair. The degree of unfairness depends on the amount by which the congestion epochs exceeds Δ^L, with a gradual degradation of network fairness as the congestion epoch increases; see, for example, Figure 5.

For comparison, the behaviour of the High-Speed TCP (HS-TCP) algorithm proposed in [12] is shown in Figure 6. The rate of convergence is evidently rather slow. In this simulation, the second flow experiences a packet drop early in slow-start and this focuses attention on the responsiveness of the congestion avoidance algorithm. While slow-start might be amended to improve start-up performance in this particular case, this does not remove the difficulties associated with the unresponsiveness of the congestion avoidance algorithm. A second simulation is shown in Figure 7 where a cross-flow disturbs two HS-TCP flows. Slow convergence back to fairness is evident and this would be unaffected by changes to slow-start. For comparison, the corresponding time histories for H-TCP are shown in Figure 8

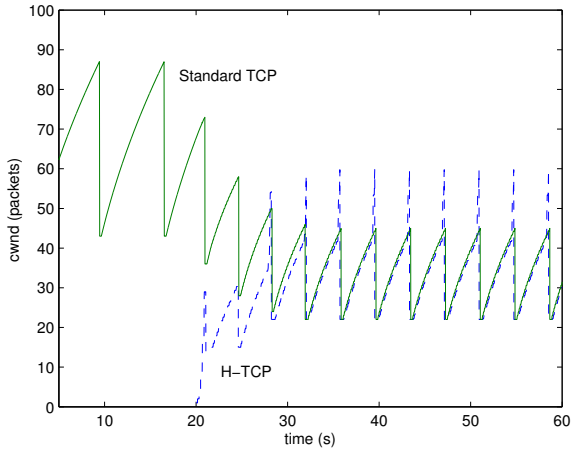


Figure 5: Example of standard TCP and H-TCP flows co-existing on a low speed link (NS simulation, network parameters: 5Mb bottleneck link, 100ms delay, queue 44 packets; H-TCP parameters: $\alpha^L = 1, \alpha^H = 20, \beta = 0.5, \Delta^L = 19$).

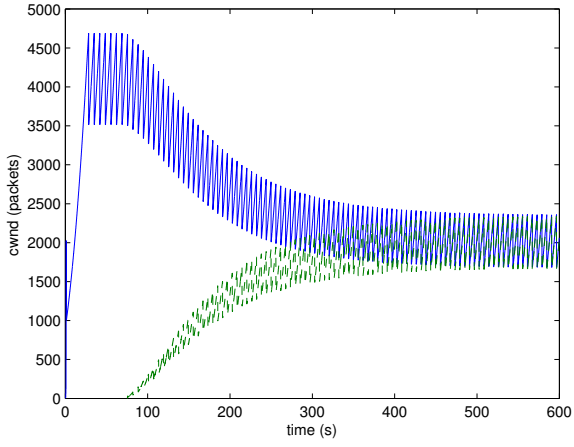
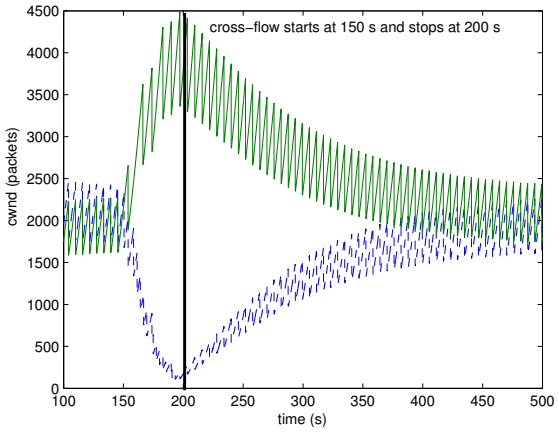
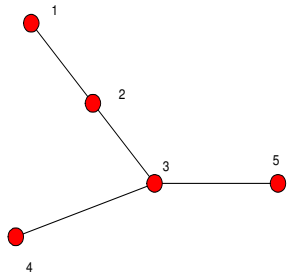


Figure 6: Example of two HS-TCP flows - the second flow experiences a drop early in slow-start focussing attention on the responsiveness of the congestion avoidance algorithm (NS simulation, network parameters: 500Mb bottleneck link, 100ms delay, queue 500 packets)



(a) Window evolution



(b) Network topology

Figure 7: Example of two HS-TCP flows with a cross-flow active between 150-200 seconds. The sluggish response of the HS-TCP flows is evident (NS simulation, bottleneck link is between nodes 3 and 5: 500Mb, 100ms delay, queue 500 packets; HS-TCP cross-flow is between nodes 1 and 2).

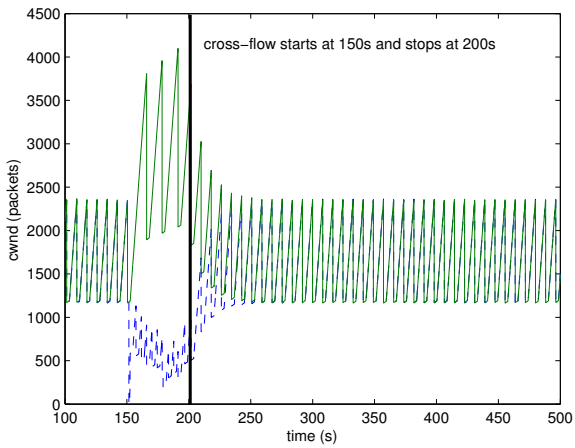


Figure 8: Example of two H-TCP flows with a cross-flow active between 150-200 seconds. (NS simulation, bottleneck link is between nodes 3 and 5: 500Mb, 100ms delay, queue 500 packets; H-TCP parameters: $\alpha^L = 1, \alpha^H = 20, \beta = 0.5, \Delta^L = 19$).

6 Concluding remarks

In this paper we present a simple model of a network of sources competing for a shared bandwidth. This model incorporates the hybrid nature of the TCP algorithm, time-varying delays on links, and drop-tail queueing (all features of networks of TCP sources). We show that networks satisfying our assumptions are very well behaved; namely, such networks always have a unique, globally exponentially stable stationary point. Using this model, conditions are established for a fair allocation of network bandwidth in networks employing a mix of AIMD algorithms, and for the convergence of the global network to its stationary point. Finally, these ideas are extended to develop and study an example protocol for high-speed networks. This protocol is demonstrated to possess an attractive combination of responsiveness and fairness on high-speed networks as well as fairness with standard TCP in low-speed environments. Future work will include the extension of the model and analysis to situations where the synchronisation assumption is relaxed.

Acknowledgements

This work was supported by Science Foundation Ireland grant 00/PI.1/C067. This work was also partially supported by the European Union funded research training network *Multi-Agent Control*, HPRN-CT-1999-00107[3] and by the Enterprise Ireland grant SC/2000/084/Y. Neither the European Union or Enterprise Ireland is responsible for any use of data appearing in this publication.

References

[1] V. Jacobson, "Congestion avoidance and control," in *Proceedings of SIGCOMM*, 1988.

[2] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, 1999.

[3] Various authors, "Special issue on tcp performance in future networking environments," *IEEE Commuications magazine*, vol. 39, no. 4, pp. –, 2001.

[4] S. Low, F. Paganini, and J. Doyle, "Internet congestion control," *IEEE Control Systems Magazine*, vol. 32, no. 1, pp. 28–43, 2002.

[5] Various authors, "Special issue on internet technology and convergence of communication services," *Proceedings of the IEEE*, vol. 90, no. 9, pp. –, 2002.

[6] A. Berman and R. Plemmons, *Nonnegative matrices in the mathematical sciences*. SIAM, 1979.

[7] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[8] L. Farina and S. Rinaldi, *Positive linear systems: Theory and applications*. Wiley, 2000.

[9] J. Hespanha, S. Hohacek, K. Obrarzka, and J. Lee, "Hybrid model of tcp congestion control," in *Hybrid Systems: Computation and Control*, pp. 291–304, 2001.

[10] M. Branicky, V. Borkar, and S. Mitter, "A unified framework for hybrid control: model and optimal control theory," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 31–45, 1998.

[3]This work is the sole responsibility of the authors and does not reflect the European Union's opinion

[11] T. Kelly, "On engineering a stable and scalable tcp variant," tech. rep., Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR.435, 2002.

[12] S. Floyd, "High speed tcp for large congestion windows," tech. rep., Internet draft draft-floyd-tcp-highspeed-02.txt, work in progres, February 2003.