On queue provisioning, network efficiency and TCP: A framework for Adaptive AIMD Congestion Control

R.N.Shorten*, D.J.Leith†, R. Stanojević, P. Wellstead
Hamilton Institute, NUI Maynooth

November 24, 2004

Abstract

In this paper we study an adaptive *additive-increase multiplicative-decrease* congestion control strategy that supports small network queues while utilising the available network bandwidth efficiently. This is achieved without adversely affecting the manner in which bandwidth is shared amongst competing flows (network fairness), or the rate at which flows acquire their share of available bandwidth (the rate of network convergence). We develop a number of analytic tools to study the dynamic properties of the adaptive algorithm. In particular we give conditions, in both deterministic and stochastic contexts, for the existence of a unique fixed point for such networks and demonstrate that this equilibrium is globally exponentially stable. Finally, simulation and experimental results are presented to illustrate the effectiveness of the proposed algorithm.

1 Introduction

An important issue in the design of internet routers is the issue of buffer provisioning. Router buffers are usually designed with two objectives in mind.

(i) Accommodating bursty packet arrivals. Should packets arrive in a burst the router may be unable to immediately process all of the packets. The first task of the router buffer is to temporarily accommodate these packets in a queue until they can be serviced.

(ii) A second issue in the design of router buffers arises from the desire to keep the link served by the router operating close to 100% utilisation.

Router buffers are designed with both of these objectives in mind: the buffer size should be large enough to accommodate typical packet bursts in the network and should also be chosen to so that the queue does not empty when TCP responds to network congestion. The typical rule of thumb in the design of router buffers is to provision the buffer to be equal to the bandwidth of the link served by the router (measured in packets per second) multiplied by the average round trip time of the

*Email: robert.shorten@may.ie; Joint first author
†Joint first author

flows utilising the router (RTT_{av}.): the *Delay-Bandwidth Product* (DBP). While provisioning network buffers in this manner has served the networking community well in the past, it has already been argued by several authors that provisioning router buffers according to the DBP rule will not be realistic in future network scenarios and several alternative strategies for buffer provisioning have been suggested. Typically, the approach taken by these authors is to exploit statistical multiplexing effects of TCP packets arriving at network buffers to justify arguments in favour of smaller buffer sizes; see the recent paper by Appenzeller *et al.* for such a strategy and a summary of related work [1]. While approaches of this type are of merit and certainly provide key insights into the behaviour of real networks, it is nevertheless not difficult to demonstrate that they are fatally dependent upon the assumption that the buffer of interest serves a large number of TCP flows at any instant of time; if this assumption does not hold provisioning network buffers in this manner will lead to poor utilisation of the bottleneck link bandwidth.

In this paper we consider an alternative approach to the 'buffer-provisioning' problem. Rather than designing buffers to accommodate the TCP congestion control algorithm, we suggest simple modifications to the TCP algorithm itself in order adapt automatically to the level of buffer provisioning in the network. We shall see that such a strategy leads naturally to adaptive AIMD congestion control whereby network sources 'tune' their AIMD parameters to ensure efficient bottleneck link utilisation. each source.

The principle objective of this paper is to present the basic idea underlying Adaptive AIMD Congestion Control and to introduce a number of tools for analysing networks in which this algorithm is deployed. In particular, we focus on the stability and convergence properties in both deterministic and stochastic contexts.

2 Provisioning of queues in AIMD networks

A communication network consists of a number of sources and sinks connected together via links and routers. In this paper we assume that these links can be modelled as a constant propagation delay together with a queue, that the queue is operating according to a drop-tail discipline, and that all of the sources are operating a *Additive-Increase Multiplicative Decrease* (AIMD) -like congestion control algorithm. AIMD congestion control operates a window based congestion control strategy. Each source maintains an internal variable $cwnd_i$ (the window size) which tracks the number of sent unacknowledged packets that can be in transit at any time, i.e. the number of packets in flight. On safe receipt of data packets the destination sends acknowledgement (ACK) packets to inform the source. When the window size is exhausted, the source must wait for an ACK before sending a new packet. Congestion control is achieved by dynamically adapting the window size according to an additive-increase multiplicative-decrease law. Roughly speaking, the basic idea is for a source to probe the network for spare capacity by increasing the rate at which packets are inserted into the network, and to rapidly decrease the number of packets transmitted through the network when congestion is detected through the loss of data packets. In more detail, the source increments $cwnd_i(t)$ by a fixed amount α_i upon receipt of each ACK. On detecting packet loss, the variable $cwnd_i(t)$ is reduced in multiplicative fashion to $\beta_i cwnd_i(t)$.

While the basic function of congestion control is to regulate network congestion, it is also desirable to ensure that the network flows, on aggregate, utilise fully the available network resources. When the network experiences congestion the network bottleneck is necessarily operating at link capacity. The corresponding

data throughput through the bottleneck link is given by

$$R(k)^- = \frac{\sum_i^n w_i(k)}{T_{d_i} + \frac{q_{max}}{B}} = B \quad (1)$$

where k indexes the congestion events and w_i denotes the value of $cwnd_i$ when congestion is detected. B is the link capacity, q_{max} is the bottleneck buffer size, T_{d_i} is the round-trip-time experienced by the $i'th$ source when the bottleneck queue is empty and $T_{d_i} + q_{max}/B$ is the round-trip time when the queue is full. After backoff, the data throughput is given by

$$R(k)^+ = \frac{\sum_i^n \beta_i w_i(k)}{T_{d_i}} \quad (2)$$

under the assumption that the bottleneck buffer empties. If the sources backoff too much, data throughput will suffer as the queue will empty for a period of time and thus the link will operate below its maximum rate. A simple method to ensure maximum throughput is to equate the rates $R(k)^-$ and $R(k)^+$. This can be achieved by enforcing the following constraint,

$$\beta_i \geq \frac{T_{d_i}}{T_{d_i} + \frac{q_{max}}{B}} = \frac{RTT_{min,i}}{RTT_{max,i}}. \quad (3)$$

Comment 1: Responding to congestion in the manner suggested by Equation (3) may be realised in several ways.

(i) For a given choice of β_i one may seek to choose q_{max} such that $R(k)^+ = R(k)^-$ for all k. Evidently, for the case of networks employing standard TCP congestion control with $\beta_i = 0.5$, it follows that we require $q_{max} = BT_{d_i}$. This is the origin of the DBP rule.

(ii) Alternatively, for any given q_{max} one may simply set $\beta_i = \frac{RTT_{min,i}}{RTT_{max,i}}$ for all i; thereby ensuring that $R(k)^+ = R(k)^-$ for all k. The effect of this modification can be seen in Figure 1. In this example the queue provisioning is less than the delay-bandwidth product and the queue empties for a substantial period following backoff by a factor of 0.5 (see the first backoff event in the figure) with an associated reduction in link utilisation. Once the flow adjusts its backoff factor to the level of buffer provisioning we can see that the queue now just empties following a backoff event and the link continues to operate at capacity as desired.

Figure 1: Congestion window and queue occupancy time histories with adaptive TCP backoff. The delay bandwidth product is 85 packets. (NS simulation, bandwidth 10Mbs, RTT 100ms, queue 25 packets).

Comment 2: Note that the DPB-rule discussed in Item (i), and the strategy proposed in Item (ii), may be thought of as duals of one another. They both seek to ensure that $R(k)^+ = R(k)^-$ at each congestion event; (i) proceeds by adjusting the network routers while (ii) makes adjustments at the network edges.

Comment 3: The foregoing discussion considers the problem of ensuring efficient utilisation of a single link shared by a n network flows. While the case of general

networks with multiple bottleneck links is beyond the stated scope of this paper, we briefly note that it is straightforward to extend our arguments to the case networks of multiple congested links; the interested reader is referred to [2] for further details.

We argue that the dual strategy in Item (ii) above offers many advantages over the DBP rule as a means to ensuring efficient bottleneck link utilisation. With this dual approach, network queues are provisioned to accommodate the level of packet burstiness and to meet latency and jitter requirements, rather than to accommodate the TCP AIMD parameters, thereby improving scalability and simplifying network management. It has, however, been well documented that even small modifications to the TCP AIMD algorithm can have a large impact on network fairness and convergence behaviour. In the next section we demonstrate that increasing the AIMD backoff factor to improve link utilisation can negatively impact network responsiveness and, indeed, that in AIMD networks a fundamental trade-off seems to exist between efficiency and responsiveness. This leads naturally to consideration of a novel decentralised switching solution that automatically adjusts the backoff factor to prevailing performance requirements, and it is the analysis and design of this adaptive algorithm that is the main subject of this paper.

3 Performance trade-offs in AIMD congestion control

Before proceeding, we briefly summarise recently obtained analytic results for networks of TCP flows. We begin our discussion by considering communication networks for which the following assumptions are valid: (i) at congestion every source experiences a packet drop; and (ii) each source has the same round-trip-time (RTT)[1]. In this case an exact model of the network dynamics may be found as follows [3].

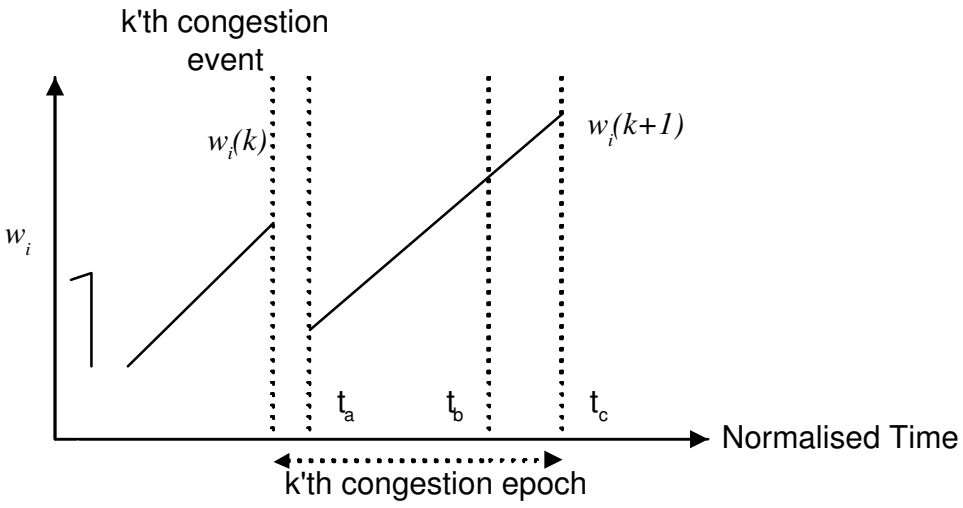

Figure 2: Evolution of window size

Let $w_i(k)$ denote the congestion window size of source i immediately before the k'th network congestion event is detected by the source. Over the k'th congestion epoch

[1]One RTT is the time between sending a packet and receiving the corresponding acknowledgement when there are no packet drops.

three important events can be discerned: $t_a(k), t_b(k)$ and $t_c(k)$; as depicted in Figure 2. The time $t_a(k)$ denotes the instant at which the number of unacknowledged packets in flight equals $\beta_i w_i(k)$; $t_b(k)$ is the time at which the bottleneck queue is full; and $t_c(k)$ is the time at which packet drop is detected by the sources, where time is measured in units of RTT[2]. 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 assume that each source is informed of congestion one RTT after the queue at the bottleneck link becomes full; that is $t_c(k) - t_b(k) = 1$. Also,

$$w_i(k) \geq 0, \sum_{i=1}^{n} w_i(k) = P + \sum_{i=1}^{n} \alpha_i, \ \forall k > 0, \tag{4}$$

where P is the maximum number of packets which can be in transit in the network at any time; P is usually equal to $q_{max} + BT_d$ where q_{max} is the maximum queue length of the congested link, B is the service rate of the congested link in packets per second and T_d is the round-trip time when the queue is empty. At the $(k+1)$th congestion event

$$w_i(k+1) \quad = \quad \beta_i w_i(k) + \alpha_i [t_c(k) - t_a(k)]. \tag{5}$$

and

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

Hence, it follows that

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

and that the dynamics an entire network of such sources is given by

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

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

$$\begin{aligned} A \quad &= \quad \begin{bmatrix} \beta_1 & 0 & \cdots & 0 \\ 0 & \beta_2 & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \beta_n \end{bmatrix} \\ &+ \quad \frac{1}{\sum_{j=1}^{n} \alpha_j} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdots \\ \alpha_n \end{bmatrix} \begin{bmatrix} 1-\beta_1 & 1-\beta_2 & \cdots & 1-\beta_n \end{bmatrix}. \end{aligned} \tag{9}$$

The matrix A is a positive matrix (all the entries are positive real numbers) and it follows that the synchronised network (8) is a positive linear system [4].

The following important theorem follows directly from basic properties of column stochastic matrices.

[2]Note that measuring time in units of RTT results in a linear rate of increase for each of the congestion window variables between congestion events.

Theorem 3.1 *[3, 5] Let A be defined as in Equation (9). Then A is a column stochastic matrix with Perron eigenvector* $x_p^T = [\frac{\alpha_1}{1-\beta_1}, ..., \frac{\alpha_n}{1-\beta_n}]$ *and whose eigenvalues are real and positive. Further, the network converges to a unique stationary point* $W_{ss} = \Theta x_p$*, where* Θ *is a positive constant such that the constraint (4) is satisfied;* $\lim_{k\to\infty} W(k) = W_{ss}$*; convergence is geometric and the rate of convergence of the network to* W_{ss} *is bounded by the second largest eigenvalue of A; the second largest eigenvalue of A lies in the interval* $[\beta_1, \beta_2]$ *where the network backoff factors are ordered as* $\beta_1 \geq \beta_2 \geq .....\beta_n$*.*

In the context of synchronised communication networks, Theorem 3.1 has the following implications for the discussion presented in Section 2.

(i) **Fairness:** Window fairness at each congestion event is achieved when the Perron eigenvector x_p is a scalar multiple of the vector $[1, ..., 1]$; that is, when the ratio $\frac{\alpha_i}{1-\beta_i}$ does not depend on i. Further, since it follows for conventional TCP-flows ($\alpha = 1, \beta = 1/2$) that $\alpha = 2(1-\beta)$, any new protocol operating an *AIMD* variant that satisfies $\alpha_i = 2(1-\beta_i)$ will be TCP-friendly (i.e. fair with legacy TCP flows), see Figure 3.

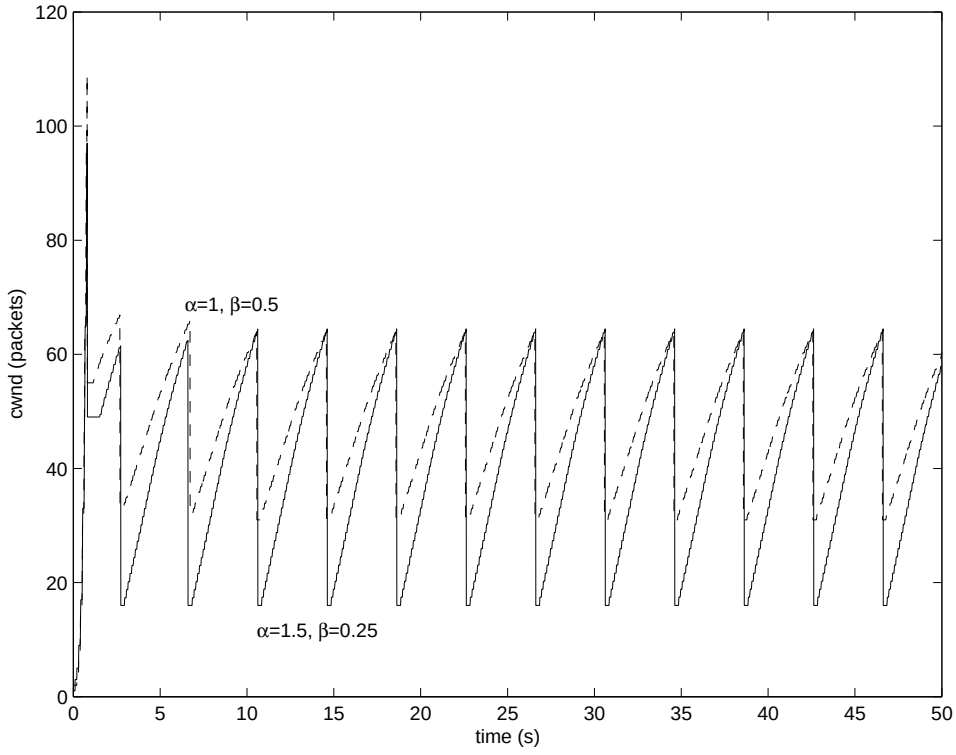

Figure 3: Example of window fairness between two TCP sources with different increase and decrease parameters (NS simulation, network parameters: 10Mb bottleneck link, 100ms delay, queue 40 packets.)

(ii) **Network responsiveness:** The magnitude of the largest backoff factor β_1 bounds the convergence rate of the entire network, with the 95% rise time measured in congestion epochs bounded by $\log 0.05 / \log \beta_1$. Consequently, fast convergence to the equilibrium state (the Perron eigenvector) is guaranteed if the largest backoff factor in the network is small.

3.1 Effect of adaptive TCP on the network dynamics

It follows from the foregoing analysis that congestion control strategies that reduce the network backoff factors to achieve high utilisation of network resources will: (i) result in an unfair network equilibrium unless all sources satisfy $\alpha_i = K\lambda_i(1-\beta_i)$ for some global constant K ($K = 2$ if some of the sources are standard TCP sources); and (ii) can result in slowing of the rate of convergence of the network to its equilibrium. For example, see Figure 4 - the backoff factor here is 0.75 for which the foregoing analysis indicates a 95% rise time of 10 congestion epochs (compared with only 4 congestion epochs when the backoff factor is 0.5) and it can be seen from the figure that this is in good agreement with packet-level simulation results.

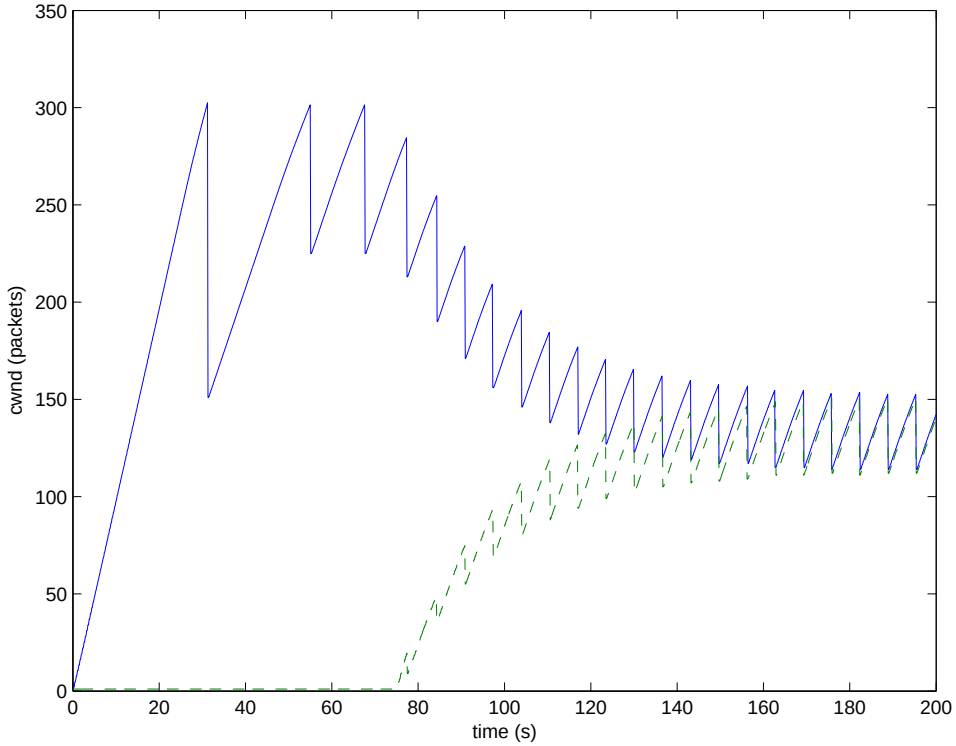

Figure 4: Illustrating poor responsiveness with larger backoff factors using an NS network simulation of a network with a 20Mb bottleneck link, a 150ms delay, a maximum queue size of 50 packets and backoff factor of 0.75.

4 Switched Adaptive AIMD congestion control

An immediate consequence of the previous discussion is that for synchronised networks in which the source AIMD parameters are fixed, high network utilisation can in general only be achieved at the expense of fast network convergence. This fundamental constraint suggests that we must design a number of controllers to address the different performance requirements (a controller to ensure high network utilisation, a controller to ensure rapid convergence) and switch between these as network conditions change[3]. Adaptive algorithms that involve mode switching are

[3]Note that TCP currently includes a slow start mode to accelerate convergence at startup of a new flow. However, slow start action is essentially confined to the first congestion epoch following

known to be difficult to design and to analyse [6]. In the case of designing adaptive algorithms to control network congestion, we must ensure that any algorithm results in a network that is (i) stable, (ii) converge rapidly, (iii) are fair and TCP-friendly, (iv) can be realised in a decentralised manner. In this section we consider ways in which these issues can be resolved in the context of AIMD congestion control.

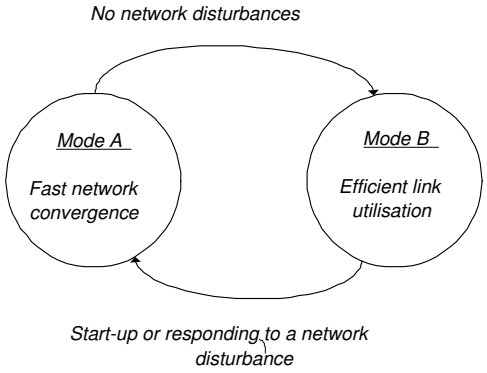

Figure 5: Switched adaptive congestion control algorithm.

4.1 Mode switch design

The following theorem is key to the design of a decentralised mode switch.

Theorem 4.1 *Consider a network of the form described in (8). Let* $w_i(0) = w_i(\infty) + \delta_i$ $\delta_i \in \mathbb{R}$ *denote the initial condition of the i'th flow* $w_i(k)$*, and* $w_i(\infty)$ *denote the asymptotic value. Let the network backoff factors be ordered according to* $\beta_1 \geq \beta_2 \geq .. \geq \beta_n$*, with* $\beta_i = \underline{\beta}$ *for all* $i \in [m+1, n]$*. Suppose that* $\| \frac{\beta - \beta_j}{1-\beta_j} \| \leq 1$ *for all* $j \in [1, m]$*. Then the following statements are true.*

(P1) Invariance.
Let $\delta = sup\{\frac{|w_1(0)-w_1(\infty)|}{w_1(\infty)}, ..., \frac{|w_m(0)-w_m(\infty)|}{w_m(\infty)}\}$*. Then* $\frac{|w_i(k)-w_i(\infty)|}{w_i(\infty)} \leq \delta$ *for all* $k > 0$ *and for* $i \in [1, m]$*.*

(P2) Convergence.
$\frac{|w_i(k)-w_i(\infty)|}{w_i(\infty)} \leq |\underline{\beta}^k \delta_i| + |\delta|$ *for all* $k > 0$*,* $i \in [m+1, n]$*.*

Proof 4.1 *We prove each of our claims in turn. For convenience we assume without any loss of generality that* $w_i(\infty) = \frac{\alpha_i}{1-\beta_i}$*.*

Property P1 : *Denote the i'th component of the vector* $AW(0)$ *by* v_i *where* $1 \leq i \leq m$*. We have:*

$$\begin{aligned} \frac{v_i - w_i(\infty)}{w(\infty)} &= \beta_i \delta_+ (1-\beta_i) \sum_{j=1}^{m} \frac{\beta - \beta_j}{1-\beta_j} \alpha_j \delta_j \\ &= \beta_i \delta_i + (1-\beta_i)\Delta \end{aligned}$$

flow startup and cannot be invoked to deal with the effect of network disturbances during the 'congestion avoidance' phase of TCP.

Since $\beta_i \in [0,1]$ it follows from convexity that

$$|\frac{v_i - w_i(\infty)}{w(\infty)}| \leq max\{|\delta_i|, |\Delta|\},$$

for all $1 \leq i \leq m$. Now assume that $\underline{\beta}$ and β_j are chosen such that $\| \frac{\underline{\beta}-\beta_j}{1-\beta_j} \| \leq 1$ for all $1 \leq j \leq m$. One can see that this is always possible since $\underline{\beta}$ is fixed, and the other β_j are upper bounded by β_1. Then

$$|\sum_{j=1}^{m} \frac{\underline{\beta} - \beta_j}{1-\beta_j} \alpha_j \delta_j| \leq max\{|\delta_j|\}, \tag{10}$$

for all $1 \leq j \leq m$. Hence, it follows that

$$|\frac{v_i - w_i(\infty)}{w(\infty)}| \leq |\delta|,$$

for all $1 \leq i \leq m$.

Property P2 : *In the spirit of $P1$ we have that*

$$|\frac{w_i(k) - w_i(\infty)}{w(\infty)}| \leq |\underline{\beta}^k \delta_i| + (1 - \underline{\beta}^k)|\delta|,$$

for all $m+1 \leq i \leq n$ and for all $k > 0$. It follows that

$$|\frac{w_i(k) - w_i(\infty)}{w(\infty)}| \quad \leq \quad |\underline{\beta}^k \delta_i| + |\delta|,$$

as claimed.

To see how we might use Theorem 4.1, let us partition the network flows into two classes (i) flows with congestion window w_i lying within a δ neighbourhood of the equilibrium value $w_i(\infty)$ i.e. with $\frac{|w_i - w_i(\infty)|}{w_i(\infty)} \leq \delta$ and (ii) other flows. Select the backoff factors of class (i) flows to be $\beta_i \leq \overline{\beta}$. Select the backoff factors of class (ii) flows to be a small value $\underline{\beta}$ such that $|\frac{\underline{\beta}-\overline{\beta}}{1-\overline{\beta}}| \leq 1$. Then we have from property (P1) that the class (i) flows will remain within a δ neighbourhood of the equilibrium for all time while from property (P2) we have that class (ii) flows will converge geometrically to a δ neighbourhood of the equilibrium at rate $\underline{\beta}$. Notice that the latter convergence rate is determined solely by the backoff factor $\underline{\beta}$ of the class (ii) flows; choosing $\underline{\beta} = 0.5$ we recover the convergence rate of standard TCP. In other words, we may select a $\overline{\beta}$ and $\underline{\beta}$ (largest and smallest backoff factor) and specify a disturbance threshold δ. Any source whose perturbation from the equilibrium is within this threshold will remain in this bounded region and can choose their backoff factors freely (in particular, they can choose their backoff factors to maximise link utilisation). Sources that are perturbed outside of this region can ensure rapid convergence to the region as k increases by selecting a small backoff factor.

This yields the following decentralised switching strategy,

$$\beta_i(k+1) \quad = \quad \begin{cases} \beta_i & |\delta_i| \leq \delta \\ \underline{\beta} & \text{otherwise.} \end{cases} \tag{11}$$

where $\beta_i = min[\frac{RTT_{min,i}}{RTT_{max,i}}, \overline{\beta}]$, $\delta_i = \frac{|w_i - w_i(\infty)|}{w_i(\infty)}$ and $\underline{\beta}$, $\overline{\beta}$ are selected to satisfy $|\frac{\underline{\beta}-\overline{\beta}}{1-\overline{\beta}}| \leq 1$.

Comment 4: Typically we might use $\underline{\beta} = 0.5$ and $\overline{\beta} = 0.75$.

Comment 5: The switching threshold δ is a design parameter that determines the performance trade-off between efficiency and responsiveness. When δ is large, the mode switch is rarely invoked and the switching strategy reduces to the previously discussed adaptive backoff strategy. When $\delta = 0$, the switching strategy corresponds to the standard TCP strategy with backoff factor $\underline{\beta}$.

Comment 6: This decentralised switching strategy requires that the ith flow can measure or infer distance from the equilibrium congestion window $w_i(\infty)$. In current networks $w_i(\infty)$ generally cannot be measured and so this distance must be estimated. There are many ways in which such estimation might be carried out. One simple approach is to estimate the distance from the magnitude of $w_i(k+1) - w_i(k)$.

4.2 Convergence to unique fixed point

Adaptation of some or all of the (α_i, β_i), $i \in \{1, ..., n\}$ via the above decentralised mode switch results in a network whose dynamics are now time-varying:

$$W(k+1) = A(k)W(k),\ A(k) \in \mathcal{M} = \{A_1, .., A_m\}, \tag{12}$$

Time-varying systems are well known to be difficult to analyse and it remains to show that networks of this type converge to a unique fixed point. We exploit the fact that all of the matrices in the set $\mathcal{M}$ are column stochastic and use this to show that all of the matrices in the set have a common $n-$ dimenional invariant subspace. Conditions for convergence to a unique fixed point then follow from this observation.

We begin formally by denoting for any $T \in \mathbb{R}^{n\times n}$, T^* as the restriction of T to $n-1$ dimensional subspace S that is orthogonal to vector $y^T = [1, 1 \ldots, 1]$. Recall that if T is column stochastic and positive then T^* is contraction on S and therefore $\|T^*\| < 1$ [7].

Theorem 4.2 *Let $\{A(k)\}_{k\in\mathbb{N}}$ be a stochastic process which corresponds to synchronized, time-varying, network (this means that $A(k)$ is a positive matrix and is am element of a finite set of positive matrices $\mathcal{M}$). If all matrices in $\mathcal{M}$ have common right Perron eigenvector x_p then $W(k)$ converges to $x_p y^T W(0)$. Moreover convergence is geometrical with convergence rate not larger then*

$$\mu = max\{\|M^*\| :\ M \in \mathcal{M}\}$$

Proof 4.2 *Note first, that since that all $A(k)$ are positive products $A(k)^* A(k-1)^* \cdots A(1)^*$ converge to zero. Let $z \in \mathbb{R}^n$ be arbitrary. Then $z = tx_p + z'$ for some $t \in \mathbb{R}$ and $z' \in S$ and*

$$\lim_{k\to\infty} A(k)\cdots A(1)z = \lim_{k\to\infty} A(k)\cdots A(1)(tx_p + z') =$$

$$= tx_p + \lim_{k\to\infty} A(k)A(k-1)\cdots A(1)z' = tx_p.$$

Thus $\lim_{k\to\infty} A(k)A(k-1)\cdots A(1) = x_p y^T$. Speed of convergence is determined by speed of convergence of $A(k)A(k-1)\cdots A(1)$. But

$$\|A(k)\cdots A(1)\| \le \|A(k)\|\cdots\|A(1)\| \le \mu^k.$$

Theorem 4.2 states that the network (12) will converge to a unique fixed point if each of the matrices in the set $\mathcal{M}$ has the same Perron eigenvector. Consider the family $\Sigma(A), A \in \mathcal{M}$ of time-invariant systems $\Sigma(A) : W(k+1) = AW(k)$. The equilibrium point of $\Sigma(A)$ is determined by the Perron eigenvector of A. The requirement that the A share the same Perron eigenvector is equivalent to the requirement that the $\Sigma(A)$ share the same equilibrium point.

4.3 Fairness and Friendliness

It follows from the discussion in Section 3 that the $A \in \mathcal{M}$ matrices share a common Perron eigenvector $[x_1 ... x_n]^T$ when

$$\frac{\alpha_i(A)}{1-\beta_i(A)} = x_i \tag{13}$$

where $\alpha_i(A), \beta_i(A)$ denote the AIMD parameters used in matrix $A \in \mathcal{M}$ for the ith flow. Equation (13) states that to ensure a common Perron eigenvector we require that variations in the AIMD parameters α_i, β_i of the ith flow be constrained such that the ratio $\alpha_i/(1-\beta_i)$ remains constant. Observe that we have that this ratio is 2 for standard TCP. We therefore have immediately that constraining the ratio $\alpha_i(A)/(1-\beta_i(A))$ to have the value 2 for all flows simultaneously yields (i) a common Perron eigenvector that ensures a unique fixed point that is fair and (ii) ensures backward compatibility and TCP friendliness.

The complete switched adaptive AIMD algorithm is therefore

$$\beta_i(k+1) = \begin{cases} \beta_i & |\delta_i| \leq \delta \\ \underline{\beta} & \text{otherwise.} \end{cases} \tag{14}$$

$$\alpha_i(k+1) = 2(1-\beta_i(k+1)) \tag{15}$$

where $\beta_i = min[\frac{RTT_{min,i}}{RTT_{max,i}}, \overline{\beta}]$, $\delta_i = \frac{|w_i - w_i(\infty)|}{w_i(\infty)}$ and $\underline{\beta}$, $\overline{\beta}$ are selected to satisfy $|\frac{\underline{\beta}-\overline{\beta}}{1-\overline{\beta}}| \leq 1$.

4.4 Example

The impact on responsiveness of introducing a mode switch is illustrated in Figure 6. The network conditions are identical to those in Figure 4, with a backoff factor β of 0.75 for efficient link utilisation. The additive increase parameter α is adjusted with β such that $\alpha/(1-\beta) = 2$ and so α reduces when β increases, as is evident in the figure. When a second flow starts at 75s, each flow decreases its backoff factor to 0.5, reverting to 0.75 when the congestion window is within 10% of its equilibrium value. The 95% rise time to this boundary region is 4 congestion epochs owing to the 0.5 backoff factor used, compared to 11 congestion epochs when the backoff factor is 0.75. Note that while in this example both flows switch to small backoff factors in response to the change in network conditions, in general only those flows that are perturbed outside the boundary region need adjust their backoff factors to ensure fast convergence. This is illustrated, for example, in Figure 7 where a network of 10 flows subject to a cross-flow disturbance between 200s and 205s is considered. After the cross-flow disturbance ends at 205s, it can be seen that the network rapidly converges back to equilibrium. In this example only 5 out of the 10 flows move outside the boundary region and reduce their backoff factors to 0.5.

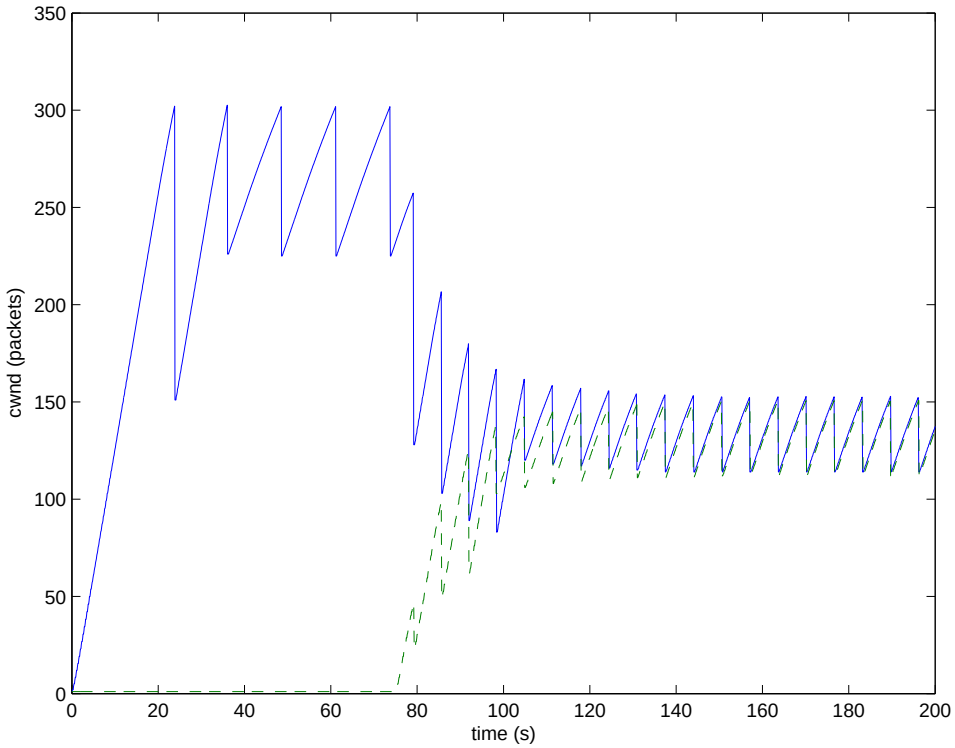

Figure 6: Improvement in responsiveness with adaptive mode switch compared with Figure 4. (NS simulation, 20Mb bottleneck link, 150ms delay, queue size 50 packets).

5 Unsynchronised networks

The foregoing discussion relates to so-called synchronised networks. Unfortunately, the assumptions of source synchronisation and uniform RTT, are quite restrictive (although they may, for example, be valid in many long-distance networks [8]). It is therefore of interest to extend our approach to more general network conditions. As we will see, much of the analysis for synchronised networks carries over directly to the unsynchronised case provided we now work in terms of the average congestion window. To distinguish variables, we will from now on denote the nominal parameters of the sources used in the previous section by α_i^s, β_i^s, $i = 1, \ldots, n$. Here the index s may remind the reader that these parameters describe the *synchronized case*, as well as that these are the parameters that are chosen by each *source*.

It is shown in [9] under the assumption of a single bottleneck link that unsynchronised networks of AIMD flows can be modelled as a switched linear system of the form

$$W(k+1) = A(k)W(k), \tag{16}$$

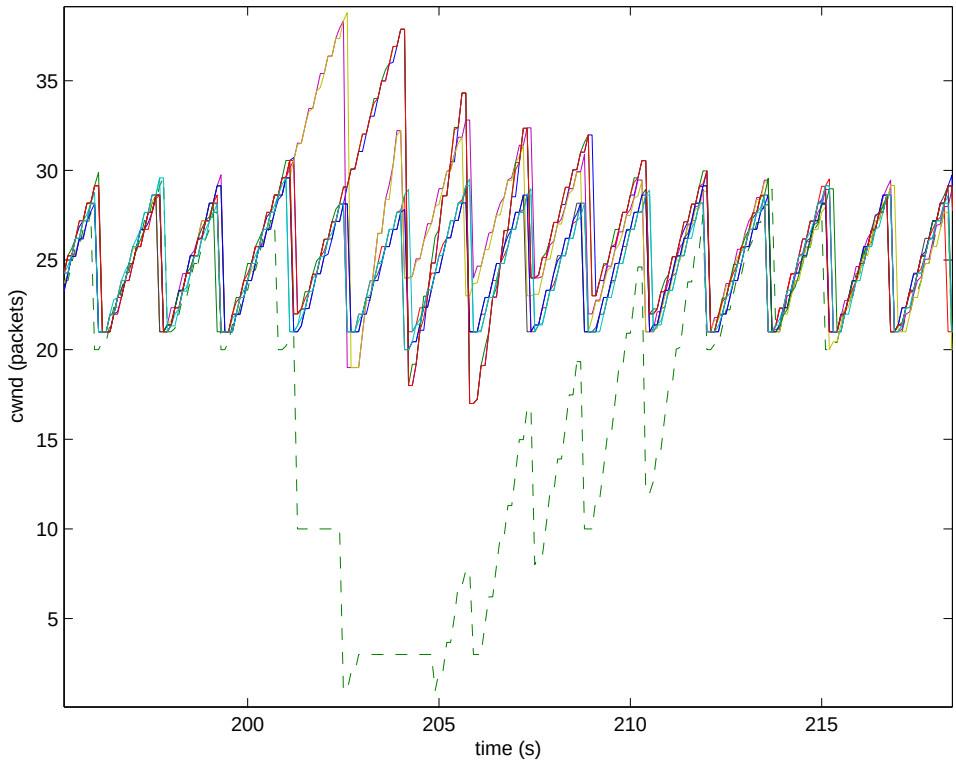

Figure 7: Adaptive mode switch for multiple flows subject to a cross-flow disturbance from 200s to 205s. (NS simulation, 10 flows, 20Mb bottleneck link, 150ms delay, queue size 50 packets).

where

$$A(k) = \begin{bmatrix} \beta_1(k) & 0 & \cdots & 0 \\ 0 & \beta_2(k) & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \beta_n(k) \end{bmatrix} + \frac{1}{\sum_{j=1}^{n} \gamma_j \alpha_j} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdots \\ \alpha_n \end{bmatrix} [\gamma_1(1-\beta_1(k)), \ldots, \gamma_n(1-\beta_n(k))]$$

where k denotes the k'th congestion event and where, depending respectively on whether flow i experiences a packet drop or not, $\beta_i(k)$ is either β_i^s (the backoff factor of the i'th source) or 1; α_i now denotes the rate at which the congestion window of flow i increases per second (rather than per RTT as previously). Since the $\beta_i(k)$ can take either of two values, we have $m = 2^n - 1$ possible matrices associated with the system (16). These correspond to the different combinations of source drops that are possible and we denote the set of these matrices by $\mathcal{A}$.

Evidently, unsynchronised networks may exhibit complex behaviour. Nevertheless, it is possible to analyse the dynamic behaviour of such networks from a probabilistic viewpoint. We proceed by making the following assumptions.

Assumption 5.1 *Let $\mathcal{A} = \{A_1, .., A_m\}$. Then, we assume that the probability that $A(k) = A_i$ in (16), is independent of k and equals ρ_i.*

Given the probabilities ρ_i for $A_i \in \mathcal{A}$, one may then define λ_j, the proportional of

congestion events at which source j experiences a backoff, as follows:

$$\lambda_j = \sum \rho_i \,,$$

where the summation is taken over those i which correspond to a matrix in which the j'th source sees a drop. Or to put it another way, the summation is over those indices i for which the matrix A_i is defined with a value of $\beta_j \neq 1$.

Assumption 5.2 *We assume that $\lambda_j > 0$ for all $j \in \{1, ..., n\}$.*

Simply stated, this assumption requires that almost surely all flows must see a drop at some time (provided that they are of long enough duration).

It is shown in [9] that the stochastic behaviour of (16) under these assumptions accurately captures the behaviour of networks carrying a mix of FTP and web traffic. Given these assumptions the following result follows directly.

Theorem 5.1 *[9] Consider the stochastic system defined in the preamble. Let $\Pi(k)$ be the random matrix product arising from the evolution of the first k steps of this system:*

$$\Pi(k) = A(k)A(k-1)....A(0).$$

Then, the expectation of $\Pi(k)$ is given by

$$E(\Pi(k)) = (\sum_{i=1}^{m} \rho_i A_i)^k; \tag{17}$$

and the asymptotic behaviour of $E(\Pi(k))$ satisfies

$$\lim_{k\to\infty} E(\Pi(k)) \quad = \quad x_p y_p^T, \tag{18}$$

with $x_p = \Theta(\frac{\alpha_1}{\lambda_1(1-\beta_1^s)}, ..., \frac{\alpha_n}{\lambda_n(1-\beta_n^s)})$, $y_p^T = (\gamma_1, ..., \gamma_n)$ where $\Theta \in \mathbb{R}$ is some constant. Furthermore, the rate of convergence of the network to its stochastic equilibrium is bounded by the second largest eigenvalue γ_2 of the positive matrix $\sum_{i=1}^m \rho_i A_i$.

It can be seen from Theorem 5.1 that, provided we work in terms of the average congestion window and the associated average matrix $\sum_{i=1}^m \rho_i A_i$, a close link exists between the analysis of unsynchronised networks and that of synchronised networks. Indeed, under the transformation $\Gamma = \text{diag}[\gamma_1, ..., \gamma_n]$ we have for $A \in \mathcal{A}$, determined by a choice of parameters $\beta_1(A), \ldots, \beta_n(A)$, that

$$\Gamma A \Gamma^{-1} = \begin{bmatrix} \beta_1(A) & 0 & \cdots & 0 \\ 0 & \beta_2(A) & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \beta_n(A) \end{bmatrix}$$

$$+\frac{1}{\sum_{j=1}^n \hat{\alpha}_j} \begin{bmatrix} \hat{\alpha}_1 \\ \hat{\alpha}_2 \\ \cdots \\ \hat{\alpha}_n \end{bmatrix} \begin{bmatrix} (1-\beta_1(A)), & \cdots, & (1-\beta_n(A)) \end{bmatrix}$$

where we have used $\hat{\alpha}_j := \gamma_j \alpha_j, j = 1, \ldots, n$. It follows immediately that $\sum_{i=1}^m \rho_i \Gamma A_i \Gamma^{-1}$ is of the same form as the synchronised matrix A in equation (9) provided we use as the increase parameters $\hat{\alpha}_j$ and as backoff factors $\hat{\beta}_j = \sum_{i=1}^m \rho_i \beta_1^i$. Hence, working in terms of the transformed state co-ordinates $\Gamma W(k)$ and in terms of average quantities almost all the analysis in the preceding sections for synchronised networks carries over directly to unsynchronised networks. Specifically, it follows that

(i) **Fairness:** Window fairness at each congestion event is achieved when the Perron eigenvector x_p is a scalar multiple of the vector $[1, ..., 1]^T$; that is, when the ratio $\frac{\alpha_i}{\lambda_i(1-\beta_i^s)}$ does not depend on i.

(ii) **Network responsiveness:** The magnitude of the second largest eigenvalue of the matrix $\sum_{i=1}^{m} \rho_i A_i$ bounds the convergence properties of the stochastic network. This eigenvalue depends on the backoff factors β_i^s of the network.

(iii) **Invariant set** Theorem 4.1 carries over directly to the unsynchronised case. This implies that the rationale underlying the proposed adaptive mode switch in the context of synchronised networks remains unchanged when we move to unsynchronised networks.

As in the case of synchronised networks, the price of efficient utilisation is slower network convergence. However, owing to the invariance and partitioned convergence properties of the network the adaptive mode switch proposed in Section 4 can be used to recover fast convergence in the face of disturbances.

The impact on responsiveness of introducing a mode switch is illustrated in Figure 8. The network topology is identical to that in Figure 4, with a backoff factor β of 0.75 for efficient link ulitisation. The results show convergence following the startup of a second FTP flow. In addition to the two FTP flows, there are 10 HTTP sessions running in the network. After the startup of the second FTP flow, it can be seen that the network rapidly converges to equilibrium. For comparison, Figure 9 shows the corresponding results for adaptive backoff without the mode switch.

6 Related work

The algorithm presented in this paper is related to several other AIMD algorithms that have been proposed in the TCP literature. In particular, the fact that we tune the AIMD parameters to reflect prevailing network conditions, suggests similarities with adaptive TCP algorithms such as TCP-Westwood [10]. The basic idea in TCP-Westwood is that each network source regulates its congestion window based upon its estimated share of bottleneck link bandwidth. More specifically, in TCP-Westwood, the bandwidth of a TCP connection is continuously estimated as the share of the available bandwidth of a link that is being used by a particular source. If the bandwidth estimate for the i^th flow is denoted (BWE_i), then the TCP-Westwood strategy is to reset the congestion window after a congestion event to $w_i(k) \rightarrow BWE_i \times RTT_{min}$. To see how this algorithm relates to our proposed strategy, consider replacing BWE_i with an estimate of the throughput for source i at the $k'th$ congestion event: $\frac{w_i(k)}{RTT_{max,i}}$. Then, using the expression $BWE_i = \frac{w_i(k)}{RRT_{max,i}}$, and using the TCP-Westwood formula, the congestion window immediately after a packet drop is:

$$w_i(k) \quad \rightarrow \quad w_i(k)\frac{RTT_{min,i}}{RTT_{max,i}}, \tag{19}$$

and our backoff method arises. There are however a number of crucial differences between TCP Westwood and our proposed scheme. While network fairness and friendliness properties are guaranteed in our scheme by controlling the Perron eigenvector of the network matrix (e.g. adjusting the parameters according to $\alpha_i = 2(1 - \beta_i)$), controlling these network properties in a satisfactory manner using Westwood has proved problematic to-date [10] using only the network backoff factors β_i. Further, TCP-Westwood is concerned solely with link utilisation and makes no attempt to control the network convergence rate.

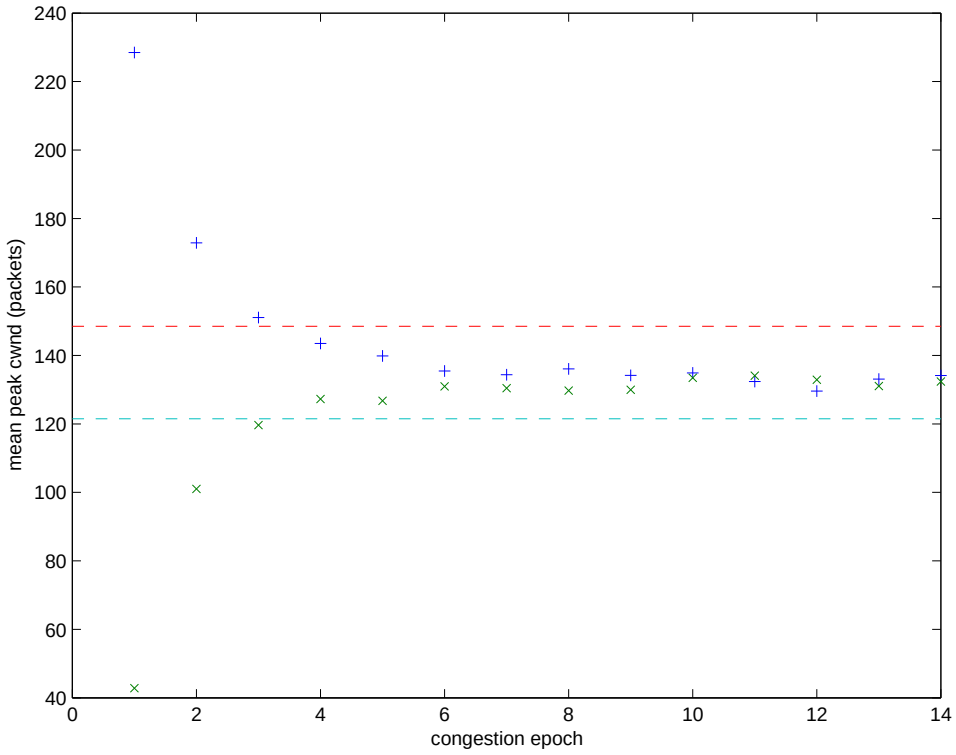

Figure 8: Ensemble average convergence with adaptive mode switch. The dashed lines mark a 10% region about the network equilibrium. (NS simulation, 2 FTP flows, 10 HTTP flows, 20Mb bottleneck link, 150ms delay, queue size 50 packets).

7 Concluding remarks

In this paper we have presented and analysed an adaptive congestion control algorithm that is suitable for deployment in communication networks that carry AIMD traffic. We have shown that adaptation can be used to achieve high aggregate throughput through the bottleneck link, rapid network convergence, and maintain the network fairness properties for networks of ling lived flows. Further, a number of analytic results have been presented to characterise the existence and qualitative properties of network equilibria, as well as a number of simulation results to demonstrate the efficacy of the design.

Acknowledgements

This work was supported by Science Foundation Ireland grant 00/PI.1/C067.

References

[1] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *In Proceedings of ACM SIGCOMM '04, August /September 2004.*, 2004.

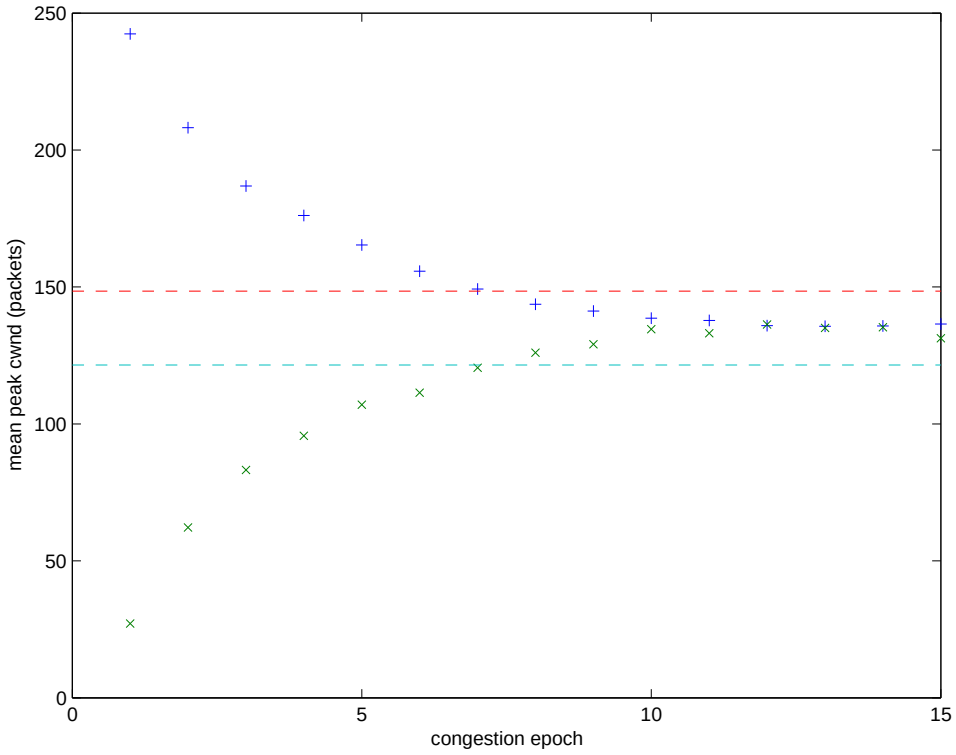

Figure 9: Ensemble average convergence with adaptive backoff but no mode switch. The dashed lines mark a 10% region about the network equilibrium. (NS simulation, 2 FTP flows, 10 HTTP sessions, 20Mb bottleneck link, 150ms delay, queue size 50 packets).

[2] D. Leith and R. Shorten, "On queue provisioning, network efficiency, and tcp: Towards network with small buffers." Submitted to SIGCOMM05, 2005.

[3] R. Shorten, D. Leith, J. Foy, and R. Kilduff, "Analysis and design of synchronised communication networks." Accepted for publication by Automatica, 2004.

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

[5] A. Berman, R. Shorten, and D. Leith, "Positive matrices associated with synchronised communication networks." Accepted for publication in Linear Algebra and its Applications, 2003.

[6] S. Morse, *Control Using Logic Based Switching*. Springer Verlag, 1997.

[7] D. Hartfiel, *Nonhomogeneous matrix products*. World Scientific, 2002.

[8] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast long-distance networks." To appear in Proceedings of IEEE INFOCOM, 2004.

[9] R. Shorten, F. Wirth, and D. Leith, "Positive matrices and the internet: Asymptotic results." Accepted for publication in IEEE Transactions on Networking, 2004.

[10] L. Massoulie, "Stability of distributed congestion control with heterogeneous feedback delays," *IEEE Transactions on Automatic Control*, vol. 47, no. 6, pp. 895–902, 2002.