Paper

An adaptive LQG TCP congestion controller for the Internet

Langford B. White and Belinda A. Chiera

Abstract— This paper addresses the problem of congestion control for transmission control protocol (TCP) traffic in the Internet. The method proposed builds on the ideas of TCP Vegas, a true feedback control approach to congestion management of TCP traffic. The new method is based on an adaptive linear quadratic Gaussian (LQG) formulation which uses an extended least squares system identification algorithm combined with optimal LQG control. Simulation experiments indicate that the new technique inherits good equilibrium properties from TCP Vegas, but has much superior transient responses which, the paper argues, is important for good dynamic congestion control.

Keywords— TCP, congestion control, LQG, adaptive control.

1. Introduction

The role of congestion control in today's high speed Internet is critical and arguably one of the most essential aspects of traffic management. A significant component of network congestion stems from the fact that over the past two decades there have been no limiting requirements placed on the entry of users onto the network, whilst a simultaneous exponential increase in Internet utilisation has occurred. The resulting effect is one of high levels of congestion in some parts of the network, providing the impetus to improve network efficiency and throughput [1].

The end-to-end transmission control protocol (TCP) [2], designed specifically to avoid and control network congestion, now carries the vast majority ($> 90\%$) of network traffic making it largely responsible for the stability of the Internet to date. However TCP in its original inception is not necessarily well-suited for more current applications. TCP Reno, the most common TCP variant currently in use, has proven effective although shows a decrease in efficacy when multiple packet lossess occur. TCP NewReno, designed specifically to address this issue, is becoming more widely utilised. TCP Vegas, one of the more recent significant proposals, is known to result in substantial improvements in throughput of up to 70% [3]. Performance issues such as fairness has, of late, led to doubt over the suitability of deploying Vegas in a shared environment, however it has recently been demonstrated that the compatibility of Reno and correctly configured Vegas flows results in an improvement in overall network performance [4].

Most TCP algorithms consist of two complementary phases: slow start and congestion avoidance. In slow start the transmission rate – *congestion window* (*cwnd*), is effectively doubled every round trip time (RTT). Once the network has been sufficiently saturated with packets from the source, TCP's congestion avoidance mechanism is invoked. At this point, *cwnd* is conservatively increased so as to gently probe the network until congestion occurs. Reno is what is known as a reactive scheme in that it reacts once congestion has already occurred. TCP Vegas however, is a proactive scheme as it monitors the difference between actual and expected transmission rates and adjusts its *cwnd* accordingly. While Vegas does not further attempt to use any type of model of the relationship between *cwnd* and the measured RTTs, it is clear that there is, at least in principle, the presence of a simple feedback control.

Modelling TCP as a feedback control system has been the subject of recent work (see for example [5–11]). In [5], TCP congestion control is modelled by combining the tools of classical control theory and Smith's principle. However co-operation from intermediate network routers is required, thereby invalidating the implementation of current TCP. Other work of note includes the development of an $\mathscr{H}^{\infty}$ controller for congestion control in communications networks with a capacity predictor [9]. Control theoretic approaches have also been applied to the Vegas mathematical model to address the issues of stability and fairness [10] although this analysis also violates the spirit of current TCP by requiring explicit congestion notification from routers on the network, as does the model of [8]. The delay-based congestion controller of [11] observes current TCP implementation by using measurements of *cwnd* and RTT only. However the model extends only so far as to the system identification of TCP using a simple autoregressive exogenous linear system model.

In this paper, we present a proportional control law for TCP congestion avoidance which we call the linear congestion controller (LCC). The LCC is designed to relate measurements of *cwnd* and logarithmically transformed RTTs to overcome in-built limitations of Vegas whilst possessing the same qualitative behaviour at equilibrium. As LCC uses only information readily available at the source, it is an end-to-end algorithm compliant with today's Internet. We design a model suitable for system identification which we translate to the plant parameters where the plant is the Internet as seen by a given TCP source. Since our model is originally affine rather than linear, we address this issue and synthesise an appropriate linear quadratic Gaussian (LQG) controller followed with the derivation of the corresponding predictor algebraic Riccati equations (ARE) and linear quadratic (LQ) cost function. We are then able to give the adaptive control design for the on-line implementation of LCC which we validate via a series of network simulations.

The outline of the paper is as follows. In Section 2, we derive the LCC model for TCP congestion avoidance. In Section 3, we propose a system model for LCC which is originally affine rather than linear; a property we correct in Subsection 3.1 to make the model suitable for LQG design. We then verify the correctness of the linear model with the standard Matlab command `dh2lqg.m` in Section 4. In Section 5, we give, in detail, the adaptive control design for the LQG component of LCC. In Section 6, we run simulations of the on-line behaviour of LCC and Vegas and compare their respective dynamic performance. Finally, we give our conclusions in Section 7.

2. Control systems model for TCP congestion avoidance

For the purposes of designing a congestion controller, we define the network as being characterised by a set of Q TCP sources $S = \{s_i : i = 1, \ldots, Q\}$ and associated receivers. Each source s_i sets its transmission rate by maintaining a congestion window of length w_i and measures the round trip time r_i (RTT) of a packet, where the RTT denotes the time between the source having sent the packet and the receipt of an acknowledgement caused by its arrival at the receiver. For the sake of clarity, we now drop the i subscript in the following derivation and present a sufficiently generic methodology applicable at each source.

The TCP Vegas [3] congestion avoidance algorithm has the update form for congestion window size (in segments) at synchronous clock time k (with sample period T_s) given by

$$w(k+1) = w(k) + \begin{cases} r(k)^{-1} & \tilde{\varepsilon}(k) > \alpha \\ -r(k)^{-1} & \tilde{\varepsilon}(k) < \beta \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where

$$\tilde{\varepsilon}(k) = w(k)\left(\frac{1}{\delta} - \frac{1}{r(k)}\right) \tag{2}$$

and $w(k)$ is the congestion window at time instant k, $r(k)$ the current RTT measurement in sample periods, δ the fixed round trip propagation delay and α, β are throughput parameters. Specifically, α and β are threshold values set at the source which serve as estimates for an under-utilised and over-utilised network, respectively. Here we consider the simplified case and assume $\alpha = \beta$ as also considered in [7].

Thus Eq. (1) becomes

$$w(k+1) = w(k) + \frac{\text{sign}[e(k)]}{r(k)}\ , \tag{3}$$

where the error signal is given by

$$e(k) = t - w(k)\left(1 - \frac{\delta}{r(k)}\right)\ . \tag{4}$$

Here t is the target number of queued segments. The motivation for this form of error signal stems from the desire to have a specified number of segments queued in the system in order to rapidly take up any bandwidth which becomes available. By definition, $w(k)$ is the number of (unacknowledged) segments, and, the term in parentheses in Eq. (4) is the proportion of those which are queued, rather than in transit. Thus the error signal $e(k)$ in Eq. (4) describes a simple feedback control mechanism.

The quantisation imposed on $w(k)$ by the sign() function in Eq. (3) has been observed to limit the effectiveness of Vegas [12]. We propose replacing Eq. (3) with the proportional control form

$$\begin{aligned} u(k) &= K(z)\, y(k), \\ \varepsilon(k) &= \frac{t}{w(k)\left(1 - \frac{\delta}{r(k)}\right)}, \end{aligned} \tag{5}$$

where $u(k) = \log w(k)$, $K(z)$ is a stable, strictly causal transfer function and

$$y(k) = \log\left(1 - \frac{\delta}{r(k)}\right) \tag{6}$$

is the transformed system output. Setting $s = \log t$ and $z(k) = \log \varepsilon(k)$ we have the linear congestion controller $u(k)$ with error term $z(k)$:

$$\begin{aligned} u(k) &= K(z)\, y(k), \\ z(k) &= s - u(k) - y(k). \end{aligned} \tag{7}$$

Comparing the Vegas error term $e(k)$ and the quantity $z(k)$ we observe $e(k) = 0$ if and only if $z(k) = 0$, meaning the equilibrium values of the Vegas controller Eq. (4) and LCC Eq. (7) are identical. Further, $e(k) > 0$ (respectively < 0) if and only if $z(k) > 0$ (respectively < 0), so the control action results in identical qualitative behaviour. Thus *cwnd* is increased when the estimated number of queued segments is less than the target, and decreased when the number of segments is greater than the target. The key difference is that we have removed the quantisation imposed by the sign() function in Eq. (3) and replaced it with a proportional control. Note that although the Vegas controller does not use a model of the system (the Internet TCP layer viewed by a single user), the limitation of the gain imposed on Eq. (3) by the one-bit quantisation of the error aids in ensuring the stability of the resulting closed loop system. In the next section, we shall generalise the control approach and incorporate a model relating $y(k)$ and $u(k)$.

3. Linear system model and LQG control

We propose an ARMAX type model relating the output signal (transformed RTTs) to the input signal (transformed *cwnd*), that is $y(k)$, $u(k)$, respectively,

$$y(k) = G(z)\ u(k) + H(z)\ \xi(k)\ , \tag{8}$$

where $G(z)$ is a strictly proper ($1 \le M \le N$) stable rational transfer function:

$$G(z) = \frac{\sum_{m=1}^{M} b_m\, z^{-m}}{1+\sum_{n=1}^{N} a_n z^{-n}} = \frac{\sum_{m=1}^{M} b_m\, z^{N-m}}{z^N+\sum_{n=1}^{N} a_n z^{N-n}}, \qquad (9)$$

and $H(z)$ is a stable rational transfer function with stable inverse

$$H(z) = \frac{1+\sum_{p=1}^{P} d_p\, z^{-p}}{1+\sum_{p=1}^{P} c_p z^{-p}} = \frac{z^P+\sum_{p=1}^{P} d_p\, z^{P-p}}{z^P+\sum_{np1}^{P} c_p z^{P-p}}\ . \qquad (10)$$

The signal $\xi(k)$ is a Gaussian white noise process with unknown mean μ and unit variance representing the unmeasured effect of all background traffic on the (transformed) RTTs as observed by the modelled user.
Writing Eq. (8) in canonical state-space form gives

$$\begin{aligned} X(k+1) &= A\ X(k)+B_2\ u(k)+B_1\ \xi(k)\ , \\ y(k) &= C_2\ X(k)+D_{21}\ \xi(k)\ , \end{aligned} \qquad (11)$$

with error equation

$$z(k) = s+D_{12}\ u(k)+C_1\ X(k)+D_{11}\ \xi(k)\ , \qquad (12)$$

where $D_{12} = -1$, $C_1 = -C_2$, and $D_{11} = -\sigma$, and

$$A = \left[\begin{array}{cccc|cccc} -a_1 & -a_2 & \cdots & -a_N & 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots & \vdots & & & \vdots \\ 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 & -c_1 & -c_2 & \cdots & -c_P \\ 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \vdots & \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 & 0 \end{array}\right],$$

$$B_2 = \left[\begin{array}{c} 1 \\ 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{array}\right], \qquad B_1 = \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \hline \sigma \\ \vdots \\ 0 \end{array}\right],$$

$$C_2 = \left[\begin{array}{cccc|cccc} b_1 & \cdots & b_M & 0 & \cdots & 0 & d_1-c_1 & \cdots & d_P-c_P \end{array}\right],$$

and $D_{21} = \sigma$, where σ^2 represents the variance of the noise $H(z)\xi(k)$. We thus have the system in the usual 4 block form as used by Matlab's `dh2lqg` function with $D_{22} = 0$.

3.1. Removal of constant values

In the above formulation, we have an affine system rather than linear because of the presence of the non-zero mean noise $\xi(k)$ and the set point s which is non-zero in general. The standard LQG design procedure assumes that all signals are zero mean, so that the resulting controller is linear in nature. Thus we need to modify our model to meet this requirement and then synthesise an appropriate affine controller to ensure that the steady state behaviour is suitable.
Using a symbol ˜ to designate quantities which have had their constant parts removed, we can write

$$\begin{aligned} \tilde{\xi}(k) &= \xi(k)-\mu, \\ u(k) &= \tilde{u}(k)+u_\infty. \end{aligned}$$

Next we consider the steady-state error signal

$$z_\infty = c-y_\infty-u_\infty \qquad (13)$$

which must be zero, otherwise we could reduce its mean square value by subtracting its mean. Thus in implementing our control, we should subtract y_∞ from the measurements $y(k)$, pass this signal to the designed controller, and then add u_∞ to the controller output before closing the loop.

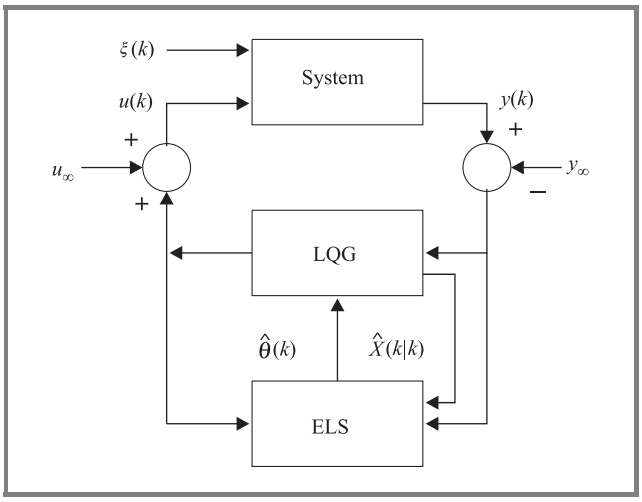

Fig. 1. Adaptive control model.

This process is illustrated in Fig. 1. The design model is now

$$\begin{aligned} X(k+1) &= A\ X(k)+B_2\ \tilde{u}(k)+B_1\ \tilde{\xi}(k), \\ \tilde{y}(k) &= C_2\ X(k)+D_{21}\ \tilde{\xi}(k), \\ \tilde{z}(k) &= D_{12}\ \tilde{u}(k)+C_1\ X(k)+D_{11}\ \tilde{\xi}(k)\ . \end{aligned} \qquad (14)$$

The control design yields the LQG controller $K(z)$ and the control signal is generated according to

$$u(k) = u_\infty+K(z)\,(y(k)-y_\infty)\ . \qquad (15)$$

4. Rapprochement with LQG design tools

To verify the correctness of the Matlab command `dh2lqg.m`, we consider the direct design using control and predictor algebraic Riccati equations (ARE). The LQ part for zero mean signals ($s = 0$) yields a control

$$u(k) = -K_c\ \hat{X}(k|k-1)\ ,$$

where

$$K_c = \left(B_2^T X B_2 + R_c\right)^{-1}\left(B_2^T X A + S_c^T\right)\ .$$

X is the solution to the ARE

$$\begin{aligned} X &= A^T X A - \left(A^T X B_2 + S_c\right)\left(B_2^T X B_2 + R_c\right)^{-1} \\ &\quad \times \left(B_2^T X A + S_c^T\right) + Q_c \end{aligned}$$

and the parameters R_c, S_c and Q_c are the cost matrices in the LQ cost function:

$$J = \mathrm{E}\left\{X(k)^T Q_c X(k) + 2X(k)^T S_c u(k) + u(k)^T R_c u(k)\right\}\ . \tag{16}$$

We minimise cost function $J = \mathrm{E}|z(k)|^2$, where

$$\begin{aligned} J &= \mathrm{E}\left\{X(k)^T C_1^T C_1 X(k) + 2X(k)^T C_1^T D_{12} u(k)\right. \\ &\quad \left. + u(k)^T D_{12}^T D_{12} u(k)\right\}\ . \end{aligned} \tag{17}$$

Thus minimum error variance control is achieved by setting

$$\begin{aligned} Q_c &= C_1^T C_1\,, \\ S_c &= C_1^T D_{12}\,, \\ R_c &= D_{12}^T D_{12}. \end{aligned} \tag{18}$$

To incorporate a control penalty, we use $R_c = D_{12}^T D_{12} + r$ for some positive quantity r.
The one-step predictions $\hat{X}(k|k-1)$ are produced by a Kalman predictor of the form

$$\hat{X}(k+1|k) = \left(A - K_f C_2\right)\hat{X}(k|k-1) + K_f\ y(k) + B_2\ u(k)\ , \tag{19}$$

where

$$K_f = \left(AYC_2^T + S_o\right)\left(C_2 X C_2^T + R_o\right)^{-1}\ . \tag{20}$$

Y is the solution to the ARE

$$\begin{aligned} Y &= AYA^T - \left(AYC_2^T + S_o\right)\left(C_2 X C_2^T + R_o\right)^{-1} \\ &\quad \times \left(C_2 Y A^T + S_o^T\right) + Q_o \end{aligned} \tag{21}$$

and the noise covariance terms are given by

$$\begin{aligned} Q_o &= B_1\ B_1^T\,, \\ S_o &= B_1\ D_{21}^T\,, \\ R_o &= D_{21}\ D_{21}^T\ . \end{aligned} \tag{22}$$

Substituting from Eq. (4) for $u(k)$, we have the state space description for the LQG controller:

$$\begin{aligned} \hat{X}(k+1|k) &= \left(A - K_f C_2 - B_2 K_c\right)\hat{X}(k|k-1) + K_f y(k)\,, \\ u(k) &= -K_c\ \hat{X}(k|k-1)\,. \end{aligned} \tag{23}$$

It has been verified that the above procedure yields an identical controller to that produced by `dh2lqg.m`.

5. Adaptive control design

Suppose we have measurements $u(k)$ and $y(k)$ for the system in open loop for $k \geq 0$, then we desire to identify the model parameters $a_1, \ldots, a_N$, $b_1, \ldots, b_M$, $c_1, \ldots, c_P$, $d_1, \ldots, d_P$, μ, σ^2 on-line. We firstly address the zero mean case where $s = \mu = 0$. Also, for purposes which will become clear, we remove the noise scaling term (σ) from the model and now assume that the noise process $\xi(k)$ has variance σ^2.
We write the parameter vector θ as

$$\theta = (a_1, \ldots, a_N, b_1, \ldots, b_M, c_1, \ldots, c_P, d_1, \ldots, d_P)^T \tag{24}$$

and let

$$\begin{aligned} \phi(k) &= (y_1(k-1), \ldots, y_1(k-N), u(k-1), \ldots, u(k-M) \\ &\quad , y_2(k-1), \ldots, y_2(k-P), \xi(k-1), \ldots, \xi(k-P))^T\,, \end{aligned}$$

where the observations are given by

$$y(k) = y_1(k) + y_2(k) + \xi(k).$$

Thus we can write

$$y(k) = \phi(k)^T\ \theta(k) + \xi(k)\ . \tag{25}$$

Since y_1 and y_2 cannot be measured separately, we use the extended least-squares (ELS) estimator

$$\begin{aligned} \hat{\phi}(k|k-1) &= \Big(\hat{y}_1(k-1|k-1), \ldots, \hat{y}_1(k-N|k-N), \\ &\quad u(k-1), \ldots, u(k-M), \\ &\quad \hat{y}_2(k-1|k-1), \ldots, \hat{y}_2(k-P|k-P), \\ &\quad \hat{\xi}(k-1|k-1), \ldots, \hat{\xi}(k-P|k-P)\Big)^T, \end{aligned} \tag{26}$$

where

$$\begin{aligned} \hat{y}_1(k-1|k-1) &= C_{21}\hat{X}(k-1|k-1), \\ \hat{y}_2(k-1|k-1) &= C_{22}\hat{X}(k-1|k-1), \\ \hat{\xi}(k-1|k-1) &= y(k-1) - \hat{y}_1(k-1|k-1) \\ &\quad - \hat{y}_2(k-1|k-1) \end{aligned} \tag{27}$$

and $C_{21} = (b_1\ \cdots\ b_M\ 0\ \cdots\ 0 \mid 0)$ and $C_{22} = (0 \mid d_1 - c_1\ \cdots\ d_P - c_P)$.

The parameter estimate is updated using the recursive least squares (RLS) rule:

$$\begin{aligned}\hat{\theta}(k+1) &= \hat{\theta}(k)+G(k)^{-1}\hat{\phi}(k|k-1)\\ &\quad\times\left(y(k)-\hat{\phi}(k|k-1)^T\hat{\theta}(k)\right),\\ G(k+1) &= G(k)+\hat{\phi}(k|k-1)\;\hat{\phi}(k|k-1)^T. \qquad (28)\end{aligned}$$

Thus the identification procedure consists of a Kalman filter (KF) estimating the state $X(k)$ using the current parameter estimates as its model, and an RLS algorithm updating the parameter estimates, using the KF state estimates to construct the regression vector. Figure 1 shows the structure of the adaptive controller.

6. Simulation results

We conducted a series of simulations to compare the behaviour of LCC Eq. (7) and Vegas (as given by Eqs. (3) and (4)). Since the models are control laws for congestion avoidance, we restricted our analysis to this phase of TCP only. We also made the assumption there were no packet losses on the network due to either timeout or the presence of lossy links. Thus having entered congestion avoidance the TCP controller remained in this phase for the duration of the simulation.

The network topology is given in Fig. 2 in which there are two TCP senders and receivers (S_1,R_1), (S_2,R_2) and a third source-sink pair (S_b,R_b) to simulate the aggregate

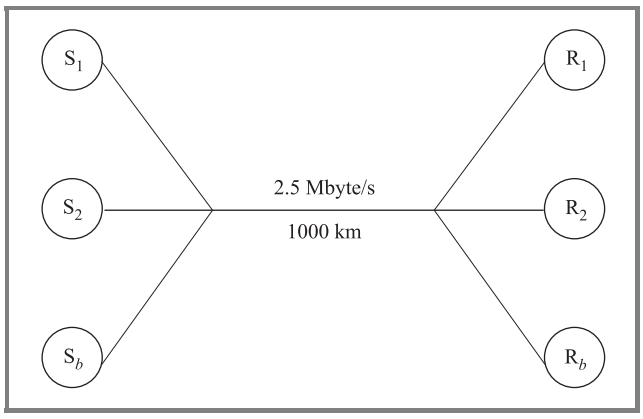

Fig. 2. TCP simulation structure for the two-sender case.

effect of background traffic on the network, generated according to a Poisson distribution. We assumed the presence of a single bottleneck FIFO queue on the network which we used to infer the effects of congestion. We set the network link capacity to 2.5 Mbyte/s and $\delta = 667\ \mu s^{-1}$. The simulation time was 20 s with a sampling rate of 250 samples/s and we staggered the starting time of the sender-receiver pair (S_2,R_2).

Figures 3, 4 and 5 depict the performance of the two models in terms of network link utilisation, the number of times the network queue ran empty and average delay as seen by the two TCP sources. The congestion window target values for S_1 and S_2 were 30 and 80 and the maximum allowed congestion window was 100. Source S_2 commenced sending a quarter-way into the simulation, although it should be noted that the results obtained when starting S_2 half- and then three quarter-way into the simulation are comparable to those given here. For the case where both sources commenced at the same time, the results of LCC were superior to those of Vegas.

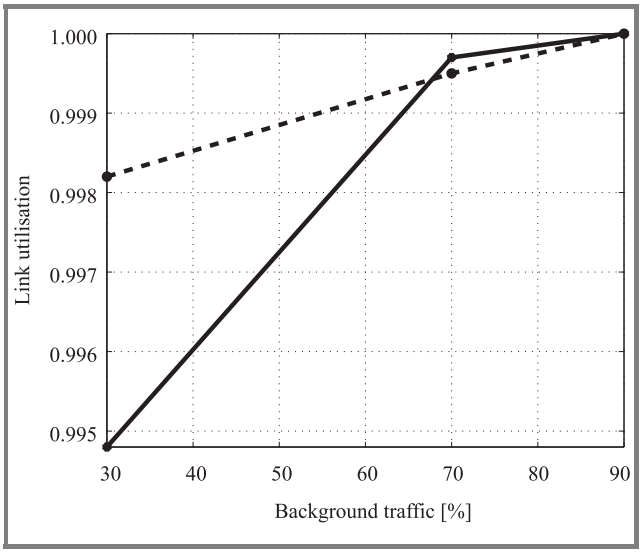

Fig. 3. Network link utilisation of LCC (–) and Vegas (- -). Background traffic loadings are 30, 70 and 90%.

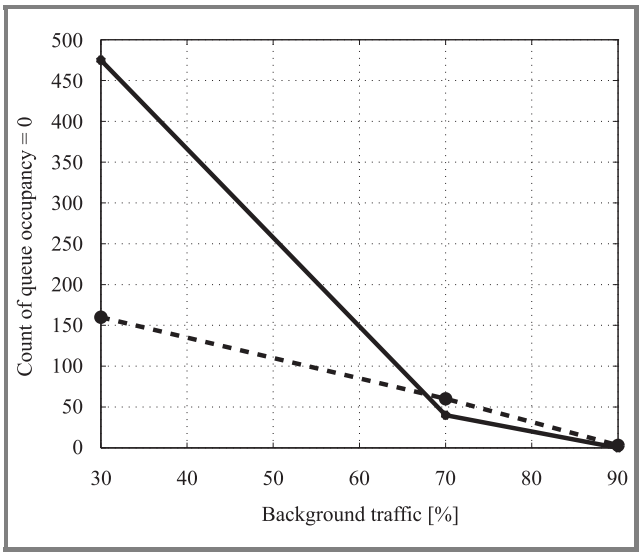

Fig. 4. The number of times the network queue ran empty when running LCC (–) and Vegas (- -). Background traffic loadings are 30, 70 and 90%.

From Figs. 3 and 4 it can be seen that apart from the case of low background traffic (30%), LCC makes better use of the network resources than Vegas. This is particularly true on a more congested network with background traffic

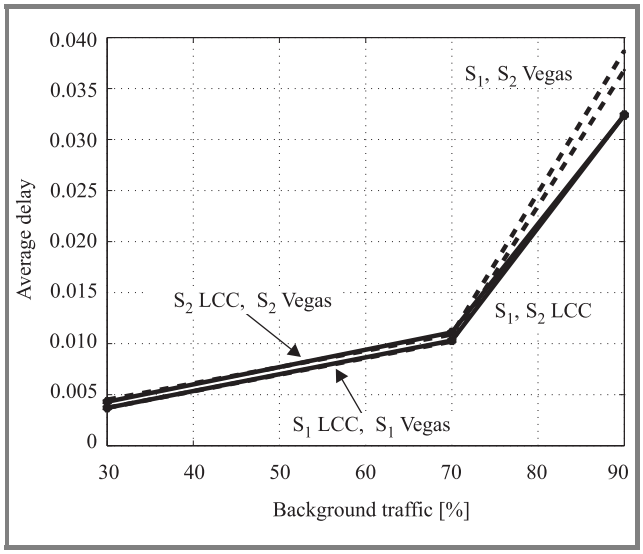

Fig. 5. Average network delay as seen by the two sources when running LCC (–) and Vegas (- -). Background traffic loadings are 30, 70 and 90%.

at 70% and 90%, in which case LCC better utilises the network resources while at the same time reducing the average delay experienced by each of the TCP sources. Moreover, from a snapshot of the congestion window in Figs. 6 and 7

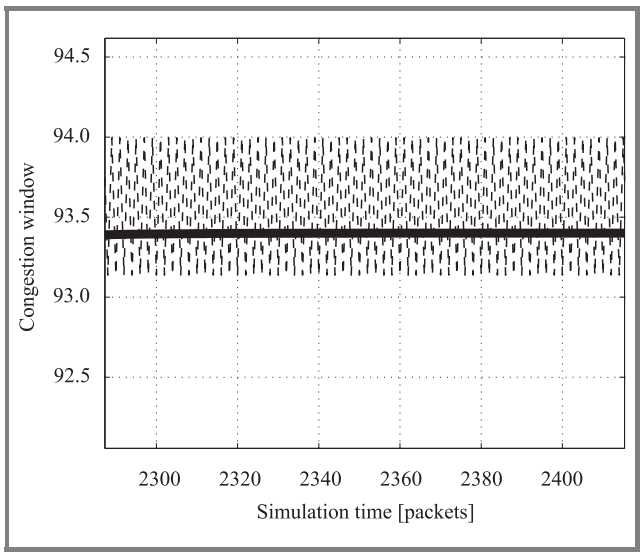

Fig. 6. A close up of the dynamic behaviour of the network measured by the response of the congestion window to S_2 joining the network. The results are for LCC (–) and Vegas (- -) with background traffic at 30%.

as well as the full set of results in Fig. 8, the dynamic behaviour of LCC is more stable than that of Vegas, irrespective of the level of background traffic. In particular, LCC responds more quickly than Vegas to (S_2,R_2) entering the network and becomes relatively stable quite rapidly. In contrast, Vegas takes longer to respond after which time the congestion window oscillates for the remainder of the simulation. This behaviour worsens when the network is more heavily loaded (Fig. 8) in which case the quality of S_1's congestion window is severely compromised. Note also the corresponding controller error as shown in Fig. 9. It can be easily seen that while the S_2 Vegas error eventually stabilises to zero, the S_1 Vegas error is destabilised after S_2 enters the network and continues to oscillate. The LCC controller error rapidly stabilises to zero.

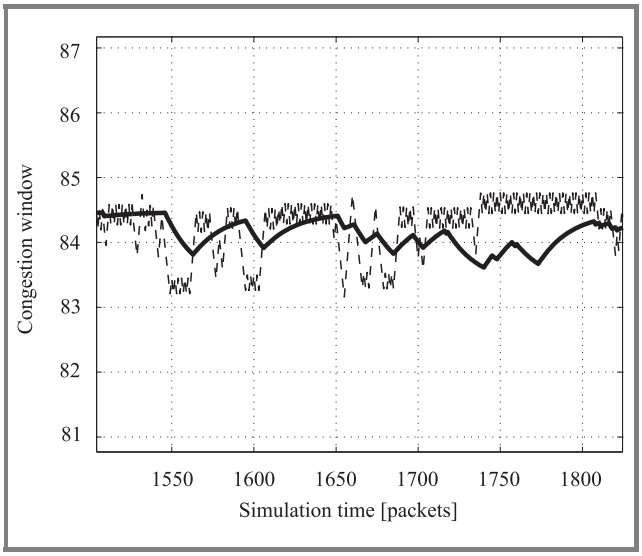

Fig. 7. A close up of the dynamic behaviour of the network measured by the response of the congestion window to S_2 joining the network. The results are for LCC (–) and Vegas (- -) with background traffic at 70%.

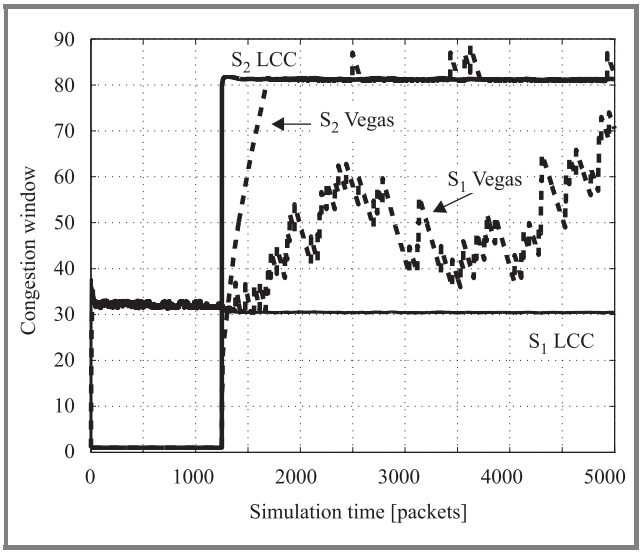

Fig. 8. Dynamic behaviour of the network measured by the response of the congestion window to S_2 joining the network. The results are for LCC (–) and Vegas (- -) with background traffic at 90%.

A secondary issue of importance is that of fairness in the allocation of network resources to the two TCP sources.

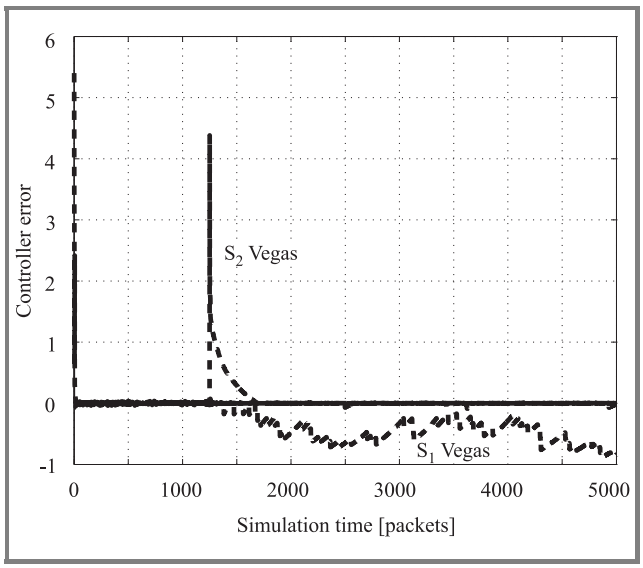

Fig. 9. The controller error of LCC (–) and Vegas (- -). The individual source errors for Vegas are as marked. Background traffic is at 90%.

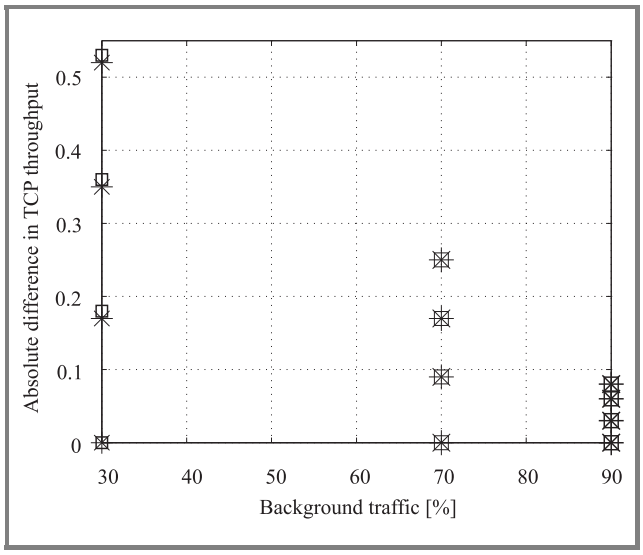

Fig. 10. Absolute difference of throughput experienced by TCP sources S_1 and S_2. The results of LCC are denoted by an asterisk ($*$) and those of Vegas by a square ($\square$). The starting times of S_2 are staggered at each level of background traffic with starts at the same time and then quarter-, half- and three quarter-way through the simulation.

In order to determine the fairness of LCC and Vegas, we set identical target values for sources S_1 and S_2 and staggered S_2's starting times as before, as well as altering network traffic levels. We then computed the absolute difference of the average dynamic throughput of each source, for each controller. The results, given in Fig. 10, show that LCC and Vegas almost entirely identically allocated network resources to both TCP sources S_1 and S_2, which is as expected since it was shown in Section 2 that LCC and Vegas will have the same equilibrium value.

7. Conclusion

In this paper we have described a linear congestion controller for TCP. The proposed controller is based on an adaptive LQG design with extended least squares system identification. A novel transformation of the TCP congestion window size (the control signal) and measured segment round-trip times, was proposed together with an ARMAX type model of the transformed signals. The proposed system design has the same equilibrium behaviour as TCP Vegas, which has been shown to be an improvement over current TCP variants. The proposed controller offers substantially better transient behaviour than TCP Vegas, which we argue is an important factor in practice as there are always TCP users joining and departing the system (Internet) on many different time scales. We have used a bottleneck network queue model to simulate the behaviour of our controller compared to TCP Vegas. Improved transient properties were observed.

In future work we remove the explicit ARMAX model and instead apply modern subspace based LQG approaches. Initial results encourage the use of the subspace-based equivalent [13]. We also present NS-2 based network simulations assuming a fully functioning network in which the simulation is not restricted to the congestion avoidance phase only while also allowing for packet loss. These results will provide a more realistic characterisation of the performance of the new method.

References

[1] V. Firoiu and M. Borden, "A study of active queue management for congestion control", in *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, 2000, vol. 3, pp. 1435–1444.

[2] V. Jacobson, "Congestion avoidance and control", in *Proc. SIGCOMM '88*, Stanford, USA, 1988.

[3] L. S. Brakmo and L. L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet", *IEEE J. Select. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, 1995.

[4] W. Feng and S. Vanichpun, "Enabling compatibility between TCP Reno and TCP Vegas", in *Proc. IEEE 2003 Symp. Appl. Internet*, Orlando, USA, 2003, pp. 301–308.

[5] S. Mascolo, "Smith's principle for congestion control in high-speed data networks", *IEEE Trans. Autom. Contr.*, vol. 45, no. 2, pp. 358–364, 2000.

[6] S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control", *IEEE Contr. Syst. Mag.*, vol. 22, no. 1, pp. 28–43, 2002.

[7] S. H. Low, L. L. Peterson, and L. Wang, "Understanding Vegas: a duality model", *J. ACM*, vol. 49, no. 2, pp. 207–235, 2002.

[8] Y. Gao and J. C. Hou, "A state feedback control approach to stabilizing queues for ECN-enabled TCP connections", in *IEEE INFOCOM 2003*, San Francisco, USA, 2003.

[9] P. F. Quet, S. Ramakrishnan, H. Özbay, and S. Kalyanaraman, "On the $\mathscr{H}^\infty$ controller design for congestion control in communications networks with a capacity predictor", in *Proc. 40th IEEE Conf. Decis. Contr.*, Orlando, USA, 2001.

[10] H. Ohsaki, M. Murata, T. Ushio, and H. Miyahara, "A control theoretical approach to a window-based flow control mechanism with explicit congestion notification", in *Proc. 38th IEEE Conf. Decis. Contr.*, Phoenix, USA, 1999.

[11] H. Ohsaki, M. Morita, and M. Murata, "On modeling round-trip time dynamics of the Internet using system identification", in *Proc. 16th International Conference on Information Networking ICOIN-16, Lecture Notes in Computer Science*. Springer, 2002, vol. 2343, pp. 359–371.

[12] L. B. White and B. A. Chiera, "LQG congestion control for TCP", in *IEEE Worksh. Internet, Telecommun. Sig. Proces.*, Adelaide, Australia, 2004, pp. 70–75.

[13] B. A. Chiera and L. B. White, "A subspace predictive controller for end-to-end tcp congestion control", in *Proc. 6th Austr. Commun. Theory Worksh.*, Brisbane, Australia, 2005, pp. 39–45.

Langford B. White graduated from the University of Queensland, Brisbane, Australia, with the degrees of B.Sc. (maths), B.E. (hons) and Ph.D. (electrical eng.) in 1984, 85 and 89, respectively. From 1986–1999, he worked for the Defence Science and Technology Organisation, Salisbury, South Australia. He received the Australian Telecommunications and Electronics Research Board Prize in 1994 for outstanding young investigator. Since 1999, he has been Professor in the School of Electrical and Electronic Engineering, The University of Adelaide, where he is also Director of the Centre for Internet Research. His research interests include automated planning, signal processing, control, telecommunications and Internet engineering. Professor White is a National ICT Australia Fellow.

e-mail: Lang.White@adelaide.edu.au
Centre for Internet Research (CIR)
School of Electrical and Electronic Engineering
The University of Adelaide
Adelaide, SA 5005, Australia

Belinda A. Chiera graduated from the University of South Australia (Mawson Lakes), Australia, with the degrees of B.App.Sc. (applied mathematics), B.App.Sc. (hons) industrial and applied mathematics and Ph.D. (applied mathematics) in 1993, 1994 and 2000, respectively. From 1999–2001 she was a postdoctoral visitor at The Delft University of Technology, The Netherlands, before returning to Australia to hold research positions at The University of Adelaide and The University of Melbourne. In 2003 Doctor Chiera joined the Centre for Internet Research (CIR) as a research fellow and has since become as a visiting fellow at CIR. Her research interests include telecommunications and internet engineering, networks and protocols, markov decision processes, control theory and signal processing.

e-mail: bchiera@eleceng.adelaide.edu.au
Centre for Internet Research (CIR)
School of Electrical and Electronic Engineering
The University of Adelaide
Adelaide, SA 5005, Australia