Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

AN EMPIRICAL VALIDATION OF A DUALITY MODEL OF TCP AND QUEUE MANAGEMENT ALGORITHMS

Sanjeewa Athuraliya

EE Department
California Institute of Technology
Pasadena, CA 91125, U.S.A

Steven H. Low

EE and CS Departments
California Institute of Technology
Pasadena, CA 91125, U.S.A

ABSTRACT

In this paper we validate through simulations a duality model of TCP and active queue management (AQM) proposed earlier. In this model, TCP and AQM are modeled as carrying out a distributed primal-dual algorithm over the Internet to maximize aggregate source utility. TCP congestion avoidance algorithms, such as Reno and Vegas, iterate on source rates, the primal variable. AQM algorithms, such as RED and REM, iterate on marking probability, the dual variable.

1 INTRODUCTION

Congestion control is a distributed algorithm to share network resources among competing sources. An optimal rate allocation problem is formulated in (Kelly 1997) where the goal is to choose source rates so as to maximize aggregate source utility subject to capacity constraints. This problem is solved using a penalty function approach in (Kelly et al. 1998, Kunniyur and Srikant 2000, Golestani et al. 1998), and extended in, e.g., (Mo and Walrand 2000, Massoulie and Roberts 1999, La and Anantharam 2000). It is solved using a duality approach in (Low and Lapsley 1999) leading to a basic algorithm whose convergence has been proved in an asynchronous environment. A practical implementation of this algorithm is studied in (Athuraliya and Low 2000). This set of work leads to abstract congestion control algorithms that can be regarded as distributed computations over a network to solve the optimal rate allocation problem. On the surface, the various TCP and active queue management (AQM) schemes proposed for or deployed on the Internet are not designed to maximize any global objective function. In (Low 2000) a connection between the abstract optimization problem and these practical schemes is proposed. It is shown there that indeed these schemes are distributed algorithms to solve the optimal rate allocation problem with appropriate utility functions and these functions are derived. These characterizations are used to derive performance properties such as throughput, loss, delay and queue length in equilibrium. In this paper we validate these properties through simulations.

In feedback congestion control, sources adjust their rates in response to congestion information on their paths, that is fed back either implicitly through buffer overflow or round trip delay, or explicitly through AQM. Different schemes adopt different measures of congestion, e.g., TCP Reno (Jacobson 1988, Stevens 2000) measures congestion by packet loss, TCP Vegas (Brakmo and Peterson 1995) by queuing (excluding propagation) delay (Low et al. 2001b), RED (Random Early Detection) (Floyd and Jacobson 1993) by queue length, and REM (Random Exponential Marking) (Athuraliya et al. 2001) by a quantity that is decoupled with performance measures such as loss or delay. With RED or REM these quantities get mapped either to a packet dropping or marking probability. These congestion measures in turn evolve in response to the source rates, closing the control loop. The key idea is to regard the source rates as primal variables, the congestion measure (or equivalently the loss/marking probability that the congestion measure gets mapped into) as dual variable, and these TCP/AQM schemes as carrying out Lagrangian methods (Bertsekas 1995) to maximize aggregate source utility.

Specifically consider a network of links (scarce resources) l with finite capacities that is shared by a set of sources. Each source s attains a utility $U_s(x_s)$ when it transmits at rate x_s. Each link updates a congestion measure $p_l(t)$ in response to the aggregate source rate at link l, and each source updates its rate $x_s(t)$ in response to the sum of congestion measures, summed over the links in its path. This can be represented in vector form as:

$$x(t+1) = F(x(t), p(t)) \quad (1)$$

$$p(t+1) = G(x(t), p(t)) \quad (2)$$

Here the function F models source algorithm, such as TCP Reno or Vegas, and the function G models queue management, active or inactive, such as DropTail, RED or REM. We have interpreted in (Low 2000) (F, G) as a

Lagrangian method to maximize aggregate source utility $\sum_s U_s(x)$ subject to capacity constraints on the links. Then different TCP congestion controls, such as Reno/DropTail, Reno/RED, Reno/REM, Vegas/DropTail, Vegas/REM, simply correspond to different combinations of (F, G). Equilibrium properties of these algorithms can be derived from the fixed point of (1–2). Moreover, by regarding the fixed point equation $(x, p) = (F(x, p), G(x, p))$ as the Karush-Kuhn-Tucker condition, we can derive the utility functions of these protocols. Hence each TCP/AQM scheme can be characterized by a triple (F, G, U) describing the dynamics of source rates and congestion measures and the utility function that the scheme is implicitly optimizing.

The rest of the paper is organized as follows. We summarize in section 2 several equilibrium properties of the (F, G, U) models for several TCP/AQM algorithms. In Section 3 we present simulation results to verify these properties.

2 TCP/AQM

Consider a network that is modeled as a single link of capacity c. A generalization to a multilink network can be found in (Low et al. 2001a). The network is shared by a set S of sources. Source s attains a utility $U_s(x_s)$ when it transmits at rate $x_s \geq 0$. Our objective is to choose source rates $x = (x_s, s \in S)$ so as to:

$$\max_{x_s \geq 0} \quad \sum_s U_s(x_s) \tag{3}$$

$$\text{subject to} \quad \sum_s x_s \leq c \tag{4}$$

Constraint (4) says that the aggregate source rate does not exceed the capacity. A unique maximizer, called the *optimal* rates, exists if the objective function is strictly concave, since the feasible solution set is compact. Associated with the link is a dual variable p. As will be seen below, the dual variable p for Reno can be regarded as loss probability and that for Vegas can be regarded as queuing delay. A primal-dual method to solve (3–4) takes the form (1–2) where both the primal variable $x(t)$ and the dual variable $p(t)$ are iterated in each step.

We now interpret TCP/AQM within this model. The single link case can be represented pictorially as in Figure 1. Each TCP source algorithm is represented by a F_s that determines how source rate $x_s(t)$ is adjusted based on the information fed back from the link. The queue management algorithm at the link G is driven by the aggregate input rate $y(t) := \sum_s x_s(t)$.

Various TCP/AQM schemes can be modeled as different Lagrangian methods (F, G) to solve (3–4) with different utility functions U_s. The algorithm model (F, G) is derived from description of the protocols. To derive the utility function U, we regard the fixed (equilibrium) point (x, p) of (1–2) as the unique optimal rate vector and Lagrange multiplier pair. The fixed point equation $(x, p) = (F, G)$ is then the Karush-Kuhn-Tucker condition, yielding the utility function.

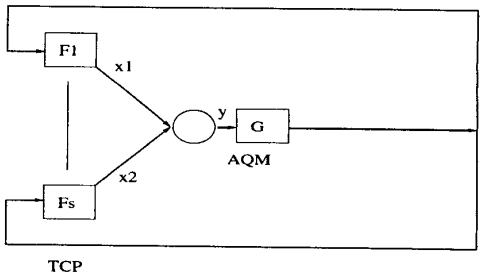

Figure 1: Duality model of TCP/AQM

2.1 TCP RENO

For TCP Reno, we take source rates as the primal variable x and link loss probabilities as the dual variable p. We assume that the round trip time τ_s of source s is constant, and that rate x_s is related to window w_s by

$$x_s(t) = \frac{w_s(t)}{\tau_s} \tag{5}$$

We focus on the additive-increase-multiplicative-decrease (AIMD) algorithm of TCP Reno. At time t, $x_s(t)$ is the rate at which packets are sent and acknowledgments received (ignoring feedback delay). A fraction $(1 - p(t))$ of these acknowledgments are positive, each incrementing the window $w_s(t)$ by $1/w_s(t)$; hence the window $w_s(t)$ increases, on average, at the rate of $x_s(t)(1 - p(t))/w_s(t)$. Similarly negative acknowledgments are returning at an average rate of $x_s(t)p(t)$, each halving the window, and hence the window $w_s(t)$ decreases at a rate of $\frac{2}{3}x_s(t)p(t)w_s(t)$ (see Figure 2). Hence, since $x_s(t) = w_s(t)/\tau_s$, we have for Reno

$$\dot{x}_s = \frac{1 - p(t)}{\tau_s^2} - \frac{2}{3}p(t)x_s^2(t) =: F_s \tag{6}$$

To derive the utility function of TCP Reno, consider the equilibrium of (6):

$$p = \frac{3}{3 + 2\tau_s^2 x_s^2} \tag{7}$$

The Karush-Kuhn-Tucker condition for the constrained optimization (3–4) is

$$U_s'(x_s) = p$$

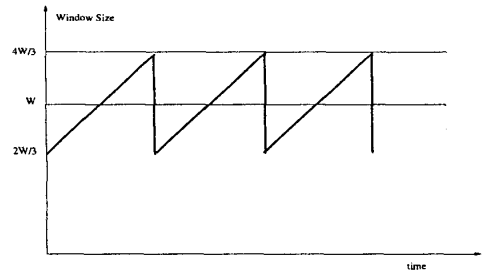

Figure 2: Window

Hence (7) implies that the unique utility function of TCP Reno is

$$U_s(x_s) = \frac{\sqrt{3/2}}{\tau_s} \tan^{-1}\left(\frac{\tau_s x_s}{\sqrt{3/2}}\right) \quad (8)$$

This utility function for TCP Reno seems to appear first in (Kelly 1999, Low 2000).

We make two remarks. First, the relation (7) between equilibrium source rate and loss probability reduces to the well known relation (see e.g. (Lakshman and Madhow 1997, Mathis et al. 1999)):

$$x_s = \frac{a}{\tau_s \sqrt{p}}$$

when the probability p is small, or equivalently, when the window $\tau_s x_s$ is large compared with $\sqrt{3/2}$. The value of the constant a, around 1, has been found empirically to depend on implementation details such as TCP variant (e.g., Reno vs. NewReno vs. SACK) and whether delayed acknowledgment is implemented. Equating $U_s'(x_s)$ with p, the utility function of TCP Reno becomes:

$$U_s(x_s) = -\frac{a^2}{\tau_s^2 x_s}$$

This version is used in (Kunniyar and Srikant 2000, Massoulie and Roberts 1999).

2.2 TCP Vegas

The model (F, G) and utility function U_s of TCP Vegas has been derived and validated in (Low et al. 2001b) We now summarize the results.

The utility function of TCP Vegas is

$$U_s(x_s) = \alpha_s d_s \log x_s \quad (9)$$

where α_s is a protocol parameter and d_s is the round trip propagation delay of source s. In equilibrium, source s buffered $\alpha_s d_s$ packets in the routers in its path.

Table 1: Duality Model of TCP (In the above $\bar{x}_s(t) = U_s'^{-1}(p(t))$)

TCP	Duality Model	
Reno	F_s	$\dot{x}_s = \frac{1-p(t)}{\tau_s^2} - \frac{2}{3}p(t)x_s^2(t)$
	U_s	$\frac{\sqrt{3/2}}{\tau_s} \tan^{-1}\left(\frac{\tau_s x_s}{\sqrt{3/2}}\right)$
Vegas	F_s	$\dot{x}_s = \begin{cases} \frac{1}{\tau_s^2} & \text{if } x_s(t) < \bar{x}_s(t) \\ -\frac{1}{\tau_s^2} & \text{if } x_s(t) > \bar{x}_s(t) \end{cases}$
	U_s	$\alpha_s d_s \log x_s$

The dual variable in TCP Vegas is queuing delay, which evolves according to (again ignoring the nonnegativity constraint):

$$\dot{p}_l = \frac{1}{c_l}(y_l(t) - c_l) =: G_l \quad (10)$$

Note that this is similar to the basic algorithm in (Low and Lapsley 1999) with γ replaced by $1/c_l$. To describe the rate adjustment F_s, let

$$\bar{x}_s(t) = U_s'^{-1}(p(t)) = \frac{\alpha_s d_s}{p(t)}$$

be the target rate chosen based on the end-to-end queuing delay $p(t)$ and the marginal utility U_s', as in the basic algorithm (Low and Lapsley 1999). Then Vegas source algorithm is:

$$\dot{x}_s = \begin{cases} \frac{1}{\tau_s^2} & \text{if } x_s(t) < \bar{x}_s(t) \\ -\frac{1}{\tau_s^2} & \text{if } x_s(t) > \bar{x}_s(t) \end{cases} =: F_s \quad (11)$$

Again the Vegas algorithm can be regarded as an approximate version of the basic algorithm, in that it moves the source rate $x_s(t)$ towards the target rate $\bar{x}_s(t)$ at a constant pace of $1/\tau_s^2$.

The duality model of TCP Reno and Vegas is summarized in Table 1. In this paper we do not consider the dynamics of different TCP/AQM schemes. Hence G is not included in the Table.

3 SIMULATION STUDIES

In this section we present simulation studies carried out using ns-2, to empirically validate the duality model of TCP congestion control. We consider the source algorithm TCP Reno with REM, RED and DropTail as queue management algorithms and TCP Vegas with DropTail. In the following experiments we focus in particular on three main results of the duality model. First, through simulation studies we show that congestion control maximizes aggregate source

utility. Then we validate two equilibrium properties: the window size and the loss/marking probability at the link.

3.1 TCP Reno

The first experiment looks at the aggregate utility as a function of time from when sources start till an equilibrium is achieved. The experiment involves a single link with a capacity of four pkts/msec. We use a buffer size of 120 pkts. We have REM, RED and DropTail at the queue. For REM we set $\gamma = 0.001$, $\alpha = 0.1$ and $\phi = 1.001$. The link algorithm is updated every 2msec. For RED we have *min thresh*=10, *max thresh*=60 and w_q=0.002. We used the gentle marking probability function. There are twenty sources starting at time = 0sec with round trip propagation delays equal to 20, 40, 60, ... 400ms. The experiment runs for 20 seconds. We repeat the experiment twenty times thus obtaining twenty different samples of the random process. We take the average of the twenty different experiments. For the Reno/DropTail experiment, in order to introduce some randomness we made the starting times of the sources uniformly distributed between 0 and 0.1 sec. Figure 3 plots the aggregate utility of the twenty sources as a function of time. We have used units of pkts/msec for the source rate and msec for the delay. The initial overshoots present in the plots are due to slow start. During transient, the aggregate utility could exceed its equilibrium value since the capacity constraint can be violated. The result confirms that congestion control maximizes the aggregate source utility given in Table 1.

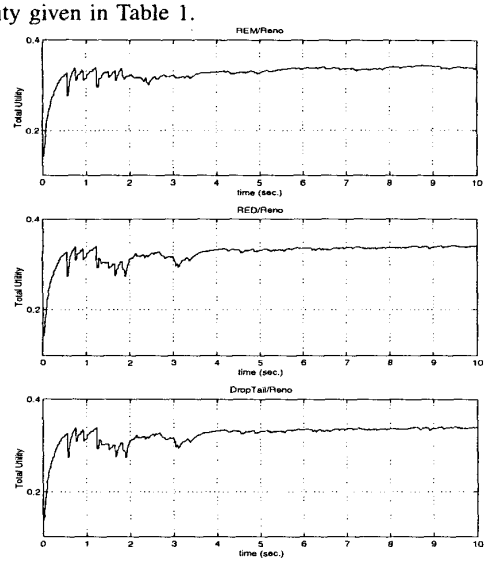

Figure 3: Aggregate Source Utility

The second experiment looks at equilibrium properties that result from the interaction between Reno and the different AQM's. In particular we consider the equilibrium mean window size and the loss/marking probability at the congested link. According to (7), sources that see the same loss probability have equal equilibrium window size, irrespective of their propagation delay. As shown below, this relation is observed for RED and REM, and for DropTail when the buffer capacity is small. With large buffer capacity, sources with the smallest delay always monopolize bandwidth to the detriment of other groups. One possible cause for this discrepancy could be the global synchronization that DropTail creates. We do not completely understand the behavior of DropTail, and an in-depth investigation of this is left for future work. Here we present simulation results with a smaller buffer size that gives results in line with the duality model.

The experiment involves four groups with twenty sources in each group. The round trip propagation delay of the sources within each group is identical but it differs between different groups. Only one group is active at time t=0sec. Every fifty seconds thereafter another group starts transmitting until all the four groups are active at t=150sec. The round trip propagation delays of the groups are 200, 150, 100 and 50 msecs in the order in which they start transmitting. The congested link has a capacity of eight packets per msec. For REM and RED we use a buffer size of 120 pkts. For DropTail we used a buffer size of 40pkts. With REM we have $\gamma = 0.001$, $\alpha = 0.1$ and $\phi = 1.001$. The link algorithm is updated every 1msec. For RED we have *min thresh*=10, *max thresh*=60 and w_q=0.002. We used the gentle marking probability function.

Figure 4 gives the mean window size of each group and Figure 5 gives the loss/marking probability at the link. The mean window size is the average window size over the twenty sources at each time instant. The straight line in Figure 5 gives the loss/marking probability predicted by the model. The loss probability is calculated at every 1sec. The total number of packets dropped during an one sec. interval is measured and then divided by the total number of packets that could have been sent during a period of 1sec. With REM marking the marking probability in the figure is the exact value the link algorithm has computed. The corresponding value for RED marking is not presented due to its severe oscillation.

As predicted by the duality model the mean window size of each group is equal irrespective of its different propagation delays. The loss probability for REM lies just below the predicted. For RED the loss probability lies close to the predicted. It lies above the predicted for DropTail. With all three schemes the deviation of the predicted value from experimental value increases as window size becomes smaller. This is because the model does not capture timeout, which occurs more frequently when congestion becomes severe and window size becomes small.

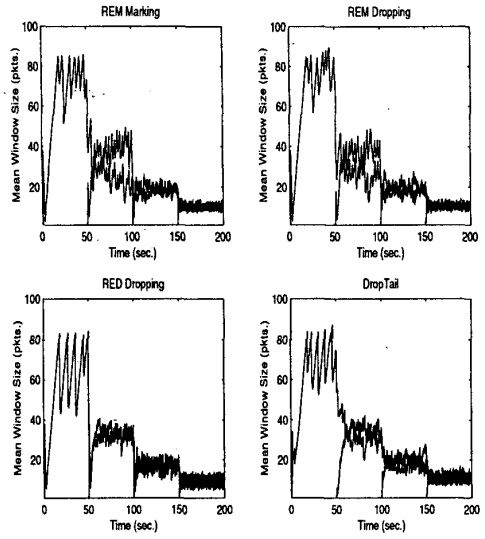

Figure 4: Mean Window Size

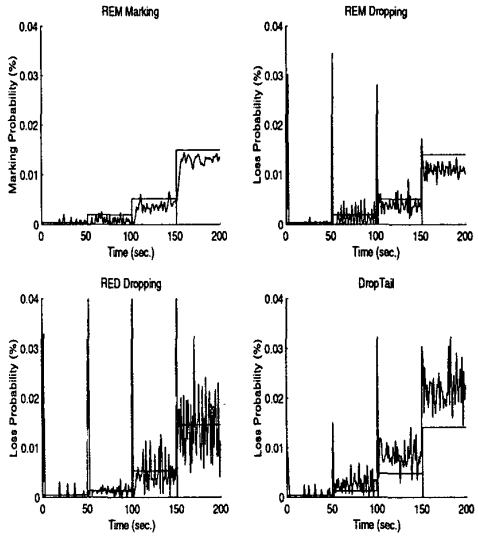

Figure 5: Loss/Mark Probability

3.2 Vegas/DropTail

In this section we present simulation results for the same experiments as in the last section but with TCP Vegas as the source algorithm and DropTail at the queue. We set α_s and β_s to be 3 pkts/rtt. The queue size is set to 400pkts thus enabling the network to reach equilibrium. The equilibrium properties of the interaction between Vegas and DropTail is quite different to that encountered with Reno as the source algorithm. Figure 6 plots the aggregate utility as a function of time. Here the increase in the Total utility is more apparent and more gradual.

Figures 7 and 8 plot the mean window size and queue length respectively. The equilibrium mean window sizes of the groups are not equal but with the choice of equal values of α_s and β_s we expect a equal source rate for all the sources. The simulation results confirm that. The round trip delays seen by the sources, in particular the ones in the group staring at last are distorted because of the nonzero queue length present at the start of the last group. This has the effect of increasing the baseRTT value (estimated propagation delay) of that group by the queuing delay. This explains the larger than expected mean window size of the last group; see (Low et al. 2001b) for more details.

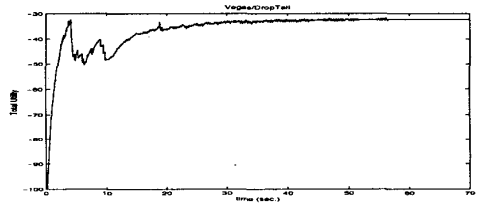

Figure 6: Aggregate Source Utility

ACKNOWLEDGMENT

This work is supported by the Australian Research Council under grant A49930405, the Caltech Lee Center for Advanced Networking, and the Yuen Research Fund.

REFERENCES

Athuraliya, S., V. H. Li, S. H. Low, and Q. Yin. 2001. REM: active queue management. *May/June, IEEE Network*. `http://netlab.caltech.edu`.

Athuraliya, S., and S. H. Low. 2000, May. Optimization flow control, II: Implementation. Submitted for publication, `http://netlab.caltech.edu`.

Bertsekas, D. 1995. *Nonlinear programming*. Athena Scientific.

Brakmo, L. S., and L. L. Peterson. 1995, October. TCP Vegas: end to end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications* 13 (8). `http://cs.princeton.edu/nsg/papers/jsac-vegas.ps`.

Floyd, S., and V. Jacobson. 1993, August. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking* 1 (4): 397–413. `ftp://ftp.ee.lbl.gov/papers/early.ps.gz`.

Golestani, J., and S. Bhattacharyya. 1998, October. End-to-end congestion control for the Internet: A global optimization framework. In *Proceedings of International Conf. on Network Protocols (ICNP)*.

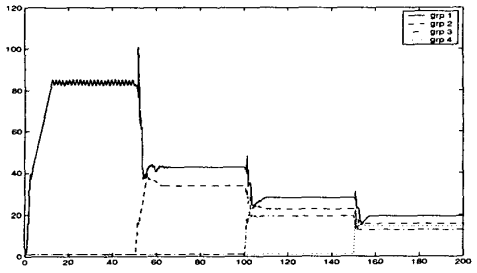

Figure 7: Mean Window Size.

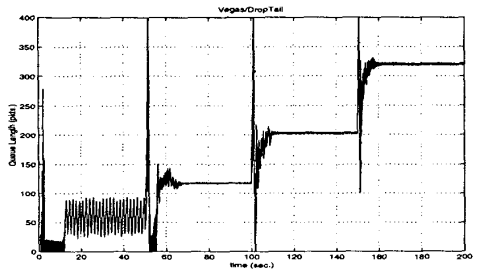

Figure 8: Queue Length

Jacobson, V. 1988, August. Congestion avoidance and control. *Proceedings of SIGCOMM'88, ACM*. An updated version is available via `ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z`.

Kelly, F. P. 1997. Charging and rate control for elastic traffic. *European Transactions on Telecommunications* 8:33–37. `http://www.statslab.cam.ac.uk/~frank/elastic.html`.

Kelly, F. P. 1999, July. Mathematical modelling of the Internet. In *Proc. 4th International Congress on Industrial and Applied Mathematics*. `http://www.statslab.cam.ac.uk/~frank/mmi.html`.

Kelly, F. P., A. Maulloo, and D. Tan. 1998, March. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of Operations Research Society* 49 (3): 237–252.

Kunniyur, S., and R. Srikant. 2000, March. End–to–end congestion control schemes: utility functions, random losses and ECN marks. In *Proceedings of IEEE Infocom*. `http://www.ieee-infocom.org/2000/papers/401.ps`.

La, R., and V. Anantharam. 2000, March. Charge-sensitive TCP and rate control in the Internet. In *Proceedings of IEEE Infocom*. `http://www.ieee-infocom.org/2000/papers/401.ps`.

Lakshman, T. V., and U. Madhow. 1997, June. The performance of TCP/IP for networks with high bandwidth–delay products and random loss. *IEEE/ACM Transactions on Networking* 5 (3): 336–350. `http://www.ece.ucsb.edu/Faculty/Madhow/Publications/ton97.ps`.

Low, S. H. 2000, September 18-20. A duality model of TCP flow controls. In *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*. `http://netlab.caltech.edu`.

Low, S. H., and D. E. Lapsley. 1999, December. Optimization flow control, I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking* 7 (6): 861–874. `http://netlab.caltech.edu`.

Low, S. H., F. Paganini, and J. C. Doyle. 2001a, December. Internet congestion control: an analytical perspective. To appear *IEEE Control Systems Magazine*. `http://netlab.caltech.edu`.

Low, S. H., L. Peterson, and L. Wang. 2001b, June. Understanding Vegas: a duality model. In *Proceedings of ACM Sigmetrics*. `http://netlab.caltech.edu/pub.html`.

Massoulie, L., and J. Roberts. 1999, March. Bandwidth sharing: objectives and algorithms. In *Infocom'99*. `http://www.dmi.ens.fr/\%7Emistral/tcpworkshop.html`.

Mathis, M., J. Semke, J. Mahdavi, and T. Ott. 1997, July. The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM Computer Communication Review* 27 (3). `http://www.psc.edu/networking/papers/model_ccr97.ps`.

Jeonghoon Mo and Jean Walrand. 2000, October. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567.

Stevens, W. 1999. *TCP/IP illustrated: the protocols*, Volume 1. Addison–Wesley. 15th printing.

AUTHOR BIOGRAPHIES

SANJEEWA ATHURALIYA Sanjeewa Athuraliya received his BEng and MEngSci Degrees from the University of Melbourne, Australia in 1998 and 2000 respectively. He is currently completing the requirements for a PhD at the California Institute of Technology, U.S.A. His research interests are in the field of Internet Congestion Control. His e-mail address is <sanjeewa@caltech.edu>.

STEVEN H. LOW Steven H. Low received his Ph.D. from University of California, Berkeley and has been with Bell Labs, Murray Hill, NJ and the University of Melbourne, Australia. He is now an Associate Professor of the California Institute of Technology, Pasadena. He is a co-recipient of the IEEE Bennett Best Paper Award in 1997 and is on the editorial board of IEEE/ACM Transactions on Networking. His research is in the control and optimization of networks and protocols His e-mail address is <slow@caltech.edu>, and his web page is <http://netlab.caltech.edu/slow>.