The use of games to assess user strategies for differential Quality of Service in the Internet

R. J. Gibbens

Statistical Laboratory, University of Cambridge
16 Mill Lane, Cambridge, CB2 1SB

P. B. Key

Microsoft Research Cambridge
St George House, 1 Guildhall Street, Cambridge, CB2 3NH

November 1999

1 Introduction

Most current approaches to differential QoS in networks rely on the network providing some form of service segregation, where the segregation may be physical (as in separate overlay networks), or virtual (as with virtual paths, VPNs, or service classes). The network then plays a dominant role in determining QoS: here the user behaviour is constrained, and may even be policed.

In the current Internet, users follow a co-operative strategy in that they use a common congestion avoidance strategy (TCP) for the allocation of scarce network resources. Two effects are leading to increasing incentives for users to deviate from this common strategy: the first is the ever more diverse range of applications, and the second the awareness that the strategy may be sub-optimal from an individual user's perspective. In both cases users can improve their net utility by following a more specific strategy. The use of UDP for real time services illustrates some of the tensions with the current approach.

An alternative view is to see the User and Network problems as complementary, and encourage a mechanism for co-operation. One approach is to use an integrated network, relying on congestion signals to encourage co-operation. One way of implementing such a framework is to use an ECN bit [2] to signal congestion via packet marking but where the congestion signals reflect the shadow price of lost packets. A general framework for the User and Network optimisation is given by Kelly et al [5], and specific congestion pricing approaches in Gibbens and Kelly [4]. Related work is by Low and Lapsley [9] and Kunniyar and Srikant [8]. Architectural issues are discussed in [7].

The network problem is then how to appropriately mark packets, whilst the user problem centres on designing good strategies to achieve desired user-specific goals. We concentrate on the User problem for end-to-end connections. Users need not reveal their utilities or preferences, but can instead act selfishly when faced with congestion prices (i.e. shadow prices). An important question is: What User strategies will emerge in such an environment?

Much as the phenotype of an individual animal is a consequence of the interaction between its genotype and the environment in which it exists, the behaviour of flow-control schemes cannot be separated from the environment where they operate. Axelrod's [1] competitions for the repeated Prisoner's Dilemma game show how a simple (and good but sub-optimal) strategy may be more robust in a mixed environment than specialised strategies optimised for a limited one. To this end, we seek strategies that are robust across a broad range of changing environments.

Our approach is a mix of theory and experimental investigation — the later has motivated the construction of a distributed network multi-user game [6].

2 The packet lifecycle

The communications networks considered in this paper comprise three basic components: user end-systems such as desktop workstations, network routers and the associated communication channels which interconnect end-systems to routers and routers to other routers. The information sent between users connected to the network is divided into individual *packets*. A concrete form of this packet network architecture which we shall always assume throughout is the Internet based on the use of IP protocols.

2.1 Users

Users generate packets containing a given amount of user data or information and dispatch them into the network. Limits on the size of the packets are governed by the IP protocols but within these limits users may, in principle, choose to set packet sizes on a packet-by-packet basis. We shall assume throughout this paper that all packets have a given fixed size which we take as 1,000 bytes. We shall concentrate our interest in this paper on when users choose to dispatch their packets into the network. Thus, we shall be concerned with the *rate*, measured in packets per second, at which users send packets.

Packets are sent from one user to another via the communication links and routers. The information is then re-assembled from the individual packets by the receiver. The receiving end-system detects the arrival of a packet and is assumed to send back a positive acknowledgement packet to the sender, containing a sequence number to identify the packet.

2.2 Network resources

Packets traversing the network travel along communication links and pass through network routers where they are forwarded on to the next stage. At any of these stages a packet may be *lost* due to scarce resources. The sending user is able to detect the loss of a packet by means of the lack of a positive acknowledgement being received within some specified timeout period. Thus sending users can sense the *loss rate* of their packets flows.

2.3 Flow control strategies

In the current Internet, users determine their rate of sending packets into the network by observing positive acknowledgements and packet losses. Roughly speaking, in the absence of losses they *increase* their rate of sending so long as packets are positively acknowledged by the receiving end-system but *decrease* their rate whenever lost packets are detected. This adaptive control procedure thereby shares the available network resources over time between the various flows present within the network. Thus, the control signal for a user to slow down is packet loss.

We shall next consider the situation where routers can explicitly *mark* packets to indicate that they have passed through a congested resource.

2.4 Marking strategies

In this paper we shall consider the situation where routers are permitted to mark packets which may then be subsequently conveyed back to the sending user as part of the packet's positive acknowledgement.

Specifically, we shall consider the marking strategies of the form discussed in [3, 4]. Suppose that a resource has a capacity of C packets per second and a buffer size of b packets. Recall that all packets are here assumed to be of the same size, namely 1,000 bytes. Packets are lost at the resource if the buffer has insufficient room to store the packet.

In addition, suppose that the resource maintains a *virtual resource* shadowing the real resource consisting of a server operating at θC packets per second together with a buffer of length θb packets. In our studies within this paper we have taken the value $\theta = 0.9$ so that congestion within the virtual resource can be thought of as giving *early warning* of congestion within the real resource. We mark packets passing through the real resource according to whether the virtual resource is within a congested period. This congested period is defined to be from when the first packet (virtual) loss within a busy period of the virtual resource occurs until when the virtual resource is next empty of packets. Further discussion together with probabilistic modelling quantifying the behaviour of this marking strategy can be found in [3].

A variant of the shadow-queue marking is to mark in the vitual queue if and only if the shadow queue exceeds the threshold θb, where the shadow queue has no constraint on queue size.

The marked packets are related to the shadow price of lost packets, and in our experiments we shall assume that the users react to lost packets as though they had been marked.

2.5 Congestion control

In this section we describe how users may share the available resources based not on lost packet signals but according to the receipt of marked packet acknowledgements. A comprehensive theoretical introduction to such congestion control strategies is given in [5]. In order to align the user's interests with those of the network we may further suppose that users are charged a fixed (small) amount for each marked acknowledgement. Their interests will then include such charges when taking account of the overall benefit they receive from using the network.

An example of such a strategy for a user in a network with congestion marks will now be described, see also [4, 5].

In this strategy users seek to hold the rate, w, of marked packet acknowledgements fixed over time. Thus, under congestion pricing, they have a fixed *willingness to pay* per unit time for congestion marks. Users do this by adjusting their packet sending rate $x(t)$ as a function of time in the following manner. Suppose that κ is a small positive constant. The function $x(t)$ increases linearly with a slope of κw until the next marked acknowledgement when it instanteously drops by an amount κ. Thus in steady-state marks occur on average every w^{-1} units of time and the wilingness to pay is indeed w marks per unit time. The convergence to steady steady will depend *crucially* on the value of κ. Roughly speaking, κ has to be less than some fraction of the reciprocal of the round-trip time, the exact mutliplier depending on the topology and marking function [10].

Given the trajectory of $x(\cdot)$, the user determines the epoch, t', at which to send the next unit-sized packet in terms of the time, t, of the most recently sent packet according to the identity

$$\int_{t}^{t'} x(s)\, ds = 1\,. \tag{1}$$

We omit here details of how such a strategy may be implemented, except to note that receipt of a marked packet acknowledgement can only delay further the sending of the next packet by the user.

3 Network games

3.1 User's objective

Several objectives can be envisaged for users. The one considered here is stated as follows.

> *Suppose the user has a file of F packets to transfer within a period of T seconds. The user's objective is to transfer the file at least cost, that is, to minimize the resulting number of marked acknowledgements packets.*

3.2 User strategies

We now consider a variety of strategies to transfer F packets within a time period $[T_0, T_0 + T]$ of length T seconds.

3.2.1 Constant rate strategy

For the *Constant-rate* strategy, the user ignores the marks but spreads out the packets uniformly over the interval $[T_0, T_0 + T]$ by sending each successive packet after a delay of $T/(F - 1)$ seconds, with the first packet sent at time T_0. Such a strategy will generate a variable number of marked acknowledgements, depending on the congestion levels within the network during the transfer.

This is a baseline strategy, and it is possible to show that such a strategy is optimal if the probability of marking is a convex function of the load for duration of transfer, and if in addition the marking probability seen by a user is independent between packets (an i.i.d. signal) or a martingale.

3.2.2 Last-one strategy

In the *Last-one* strategy, and several related variants described below, the user has two states of operation. Packets are either being sent periodically according to a high rate of x_h packets per second or alternatively at a low rate of x_ℓ packets per second.

The user modifies which state it is in upon receipt of a packet acknowledgement. If the latest packet acknowledgement carries a mark the rate is set to its low value x_ℓ otherwise it is set to the high value x_h. As the rate switches between states the time for the next packet to be sent is revised accordingly.

In this way, the user adjusts between two rates according to feedback on congestion levels experienced within the network. Marked acknowledgements cause it to slow down while unmarked acknowledgements allow it to proceed more rapidly.

Suppose that the first packet is sent at time T_0 and that the user is initially in the state with low sending rate x_ℓ. In order to meet the precise objective of completing within a period of length T we must also allow the user to speed up if necessary towards the end of the transfer. Let $F(t)$ denote the number of packets still remaining to be transfered at time $t \in [T_0, T_0 + T]$. Then if $R = F(t)/(T_0 + T - t)$ exceeds the high rate x_h the strategy switches to the high rate state.

This is a very simple strategy that attempts to exploit dependence or correlation in the marking function.

3.2.3 Last-two strategy

The *Last-two* differs from the *Last-one* strategy above in that the last two acknowledgements are stored and the low rate, x_ℓ, is used unless the last two acknowledgements were *both* unmarked. This is a simple way of detecting a 'non-marking' period.

3.2.4 Congestion estimator strategy

For the *Congestion estimator* strategy the user attempts to determine a short-run estimator of the current congestion levels within the network by examining the proportion of packet acknowledgements that have been marked. Specifically, we suppose that the current rate of sending is adjusted according to the number, m, of packet acknowledgements that were marked within the n most recently received acknowledgements. The rate of sending is then given by the expression

$$x_h \left(1 - \frac{\alpha + m}{\alpha + \beta + n}\right) \tag{2}$$

where α and β are two positive constants. Typical values might be $n = 5$ and $\alpha = \beta = 1$.

The estimate can be thought of as a Bayesian estimator using the last n measurements to modify prior beliefs on the probability of marking, where prior beliefs are represented by a Beta distribution with parameters α and β.

Again the above mentioned adjustment to the rate must be made to ensure that the file transfer completes within the required time period $[T_0, T_0 + T]$.

4 Experiments

To compare the different user strategies for transferring a file within a given time the following experiments were investigated. The network consisted of a single resource of capacity $C = 600$ packets per second (equivalent to a rate of 600 KB/s) with a buffer of size 10 packets. As would be the case in practice, delays were introduced between packets being sent by the user and packets being received at the resource. Similarly, there were delays between both acknowledgements and lost packet indications being generated by the resource and received at the sender. In the experiments reported here all such delays were set at 0.01 seconds.

The experiment then consisted of a background load on the resource together with one additional user seeking to undertake the file transfer.

4.1 Background load

The background load consisted of a collection of users each following the willingness to pay strategy. The users were divided into 20 categories according to their willingness to pay w. Each such category contained 5 users giving a total of 100 background users. The $i = 1, \ldots, 20$ categories of users had w_i values equally spaced in the range 0.1 to 9.6. The κ value was taken as $\min\{w_i, 1\}$. Thus $5 \times \sum_{i=1}^{20} w_i$ gives a measure of the total demand on the resource.

In order to make the total demand vary slowly over time the background users were further controlled by independent alternating on/off processes. In the *on* state the user operated as usual but in the *off* state no packets were sent. Once the user reverted to the on state it continued with exactly the rate $x(\cdot)$ at which it had switched off. We suppose that the on and off time periods for each user formed a sequence of independent exponential random variables. The mean on period was μ_{on} and the mean off period was μ_{off}. Typical values were $\mu_{\text{on}} = 10$ seconds and $\mu_{\text{off}} = 30$ seconds or 90 seconds. Thus the demand fluctuated over time with on average $1/4$ or $1/10$ of the background users in the on state.

A background load was simulated for a period of 300 seconds. Figure 4.1 shows a trace of the fluctuating demand over time on the resource.

4.2 Results

Each experiment consisted of adding a single file transfer user to the background load at time T_0. The experiment was repeated 25 times with starting times T_0 chosen at random uniformly from the interval $[50, 200]$. For each experiment the random number seeds used by the background processes (for the on and off state changes) were held fixed.

Each of the user strategies was considered. The parameters of the file transfer objective were $F = 1,000$ packets (equivalent to 1 MB) and either $T = 100$ seconds or $T = 10$ seconds. The two time durations were chosen to examine the degree to which the file transfer strategies could learn the background levels of congestion from the delayed feedback of packet marks. For the *Last-one* and *Last-two* strategies the parameters $x_h = 20$ or $x_h = 200$ packets per second

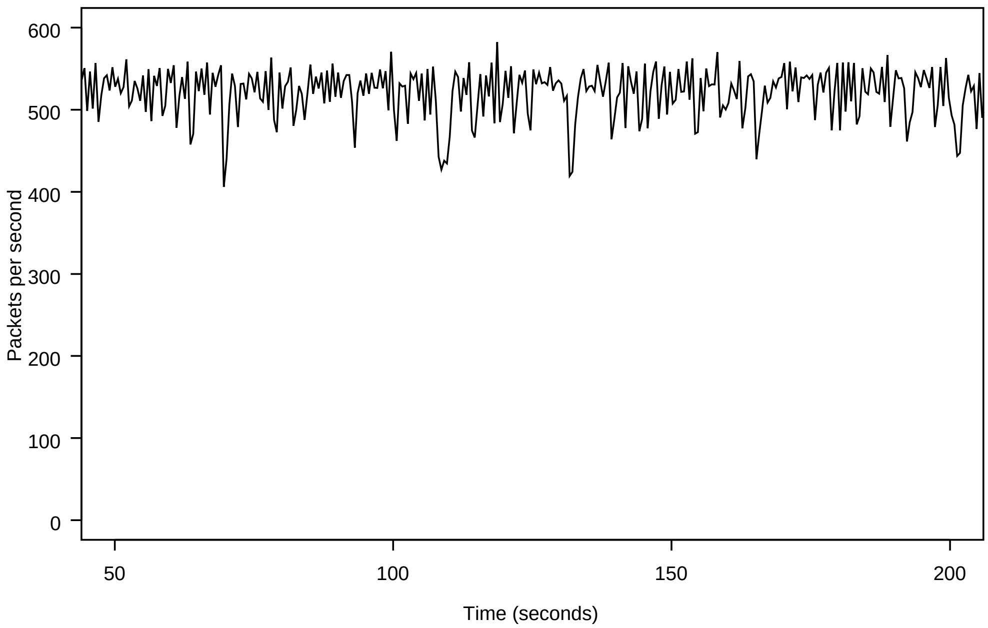

Figure 1: Fluctuating demand over time on the resource

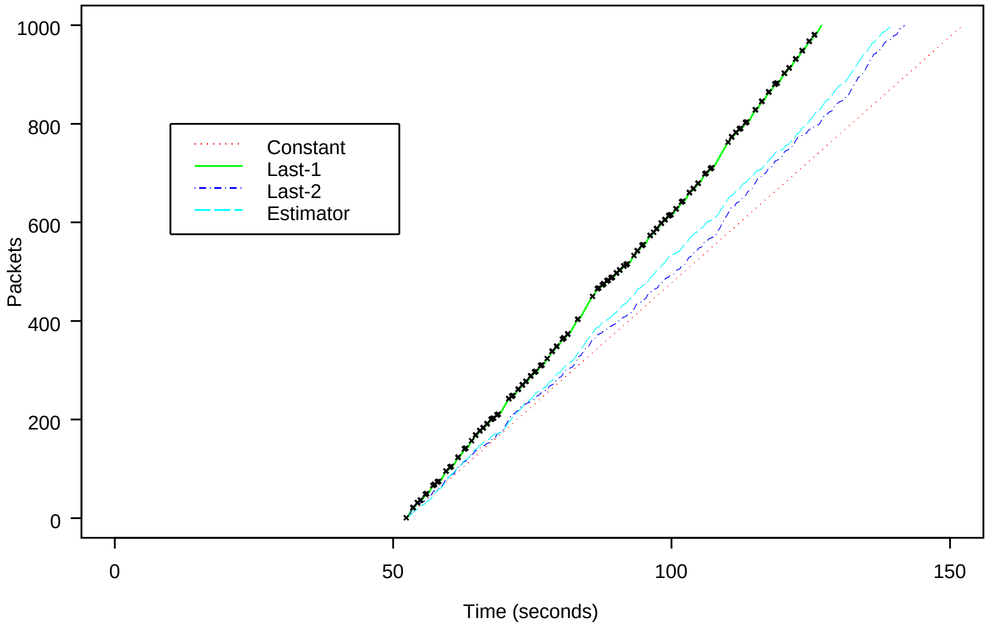

Figure 2: Packets sent and marks received during a sample transfer with start time $T_0 = 52.3$. The crosses indicate the times when marks were received for the Last-1 strategy.

according to $T = 100$ or $T = 10$ seconds respectively. The values of x_ℓ were 5 and 50 accordingly. For the *Congestion estimator* strategy the same values were used for x_h. The parameter values of $\alpha = \beta = 1$ and $n = 5$ were used for the estimator.

Figure 2 shows the accumulated number of packets sent and marks received during a sample file transfer. Figure 3 shows the queue length for the real and virtual queues during the course of Experiment 1.

Table 2 and Figures 4-7 show the results of these experiments using parameters given in 4.2.

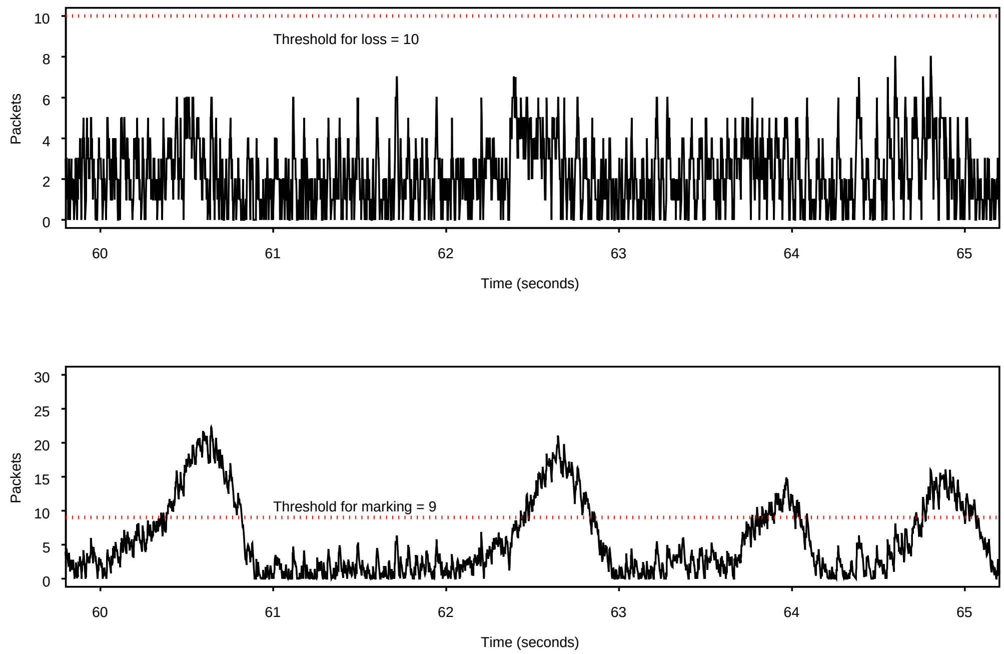

Figure 3: The top panel shows the queue length of the real queue. The buffer length is 10 packets. The bottom panel shows the queue length of the virtual queue with a marking threshold of 9.

Experiment	F (packets)	T (seconds)	μ_{off} (seconds)
1	1000	100	30
2	1000	100	90
3	1000	10	30
4	1000	10	90

Table 1: Experiments and parameters.

Experiment	Constant	Last-1	Last-2	Estimator
1	223.3 (16.9)	152.6 (15.3)	175.9 (19.4)	235.5 (23.0)
2	87.2 (6.2)	50.1 (7.4)	52.0 (8.8)	72.8 (9.1)
3	290.3 (36.2)	290.6 (42.9)	274.4 (45.1)	262.2 (39.0)
4	126.1 (72.9)	140.0 (70.1)	132.8 (67.1)	123.6 (65.7)

Table 2: Results of Experiments: means and standard deviations of the numbers of marks received.

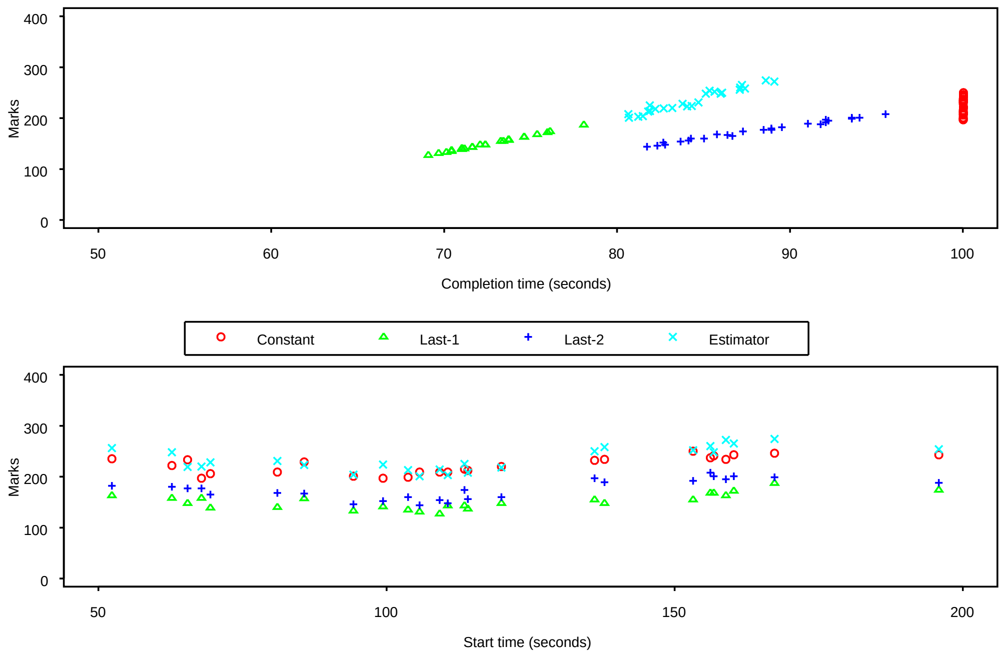

Figure 4: Results for experiment 1.

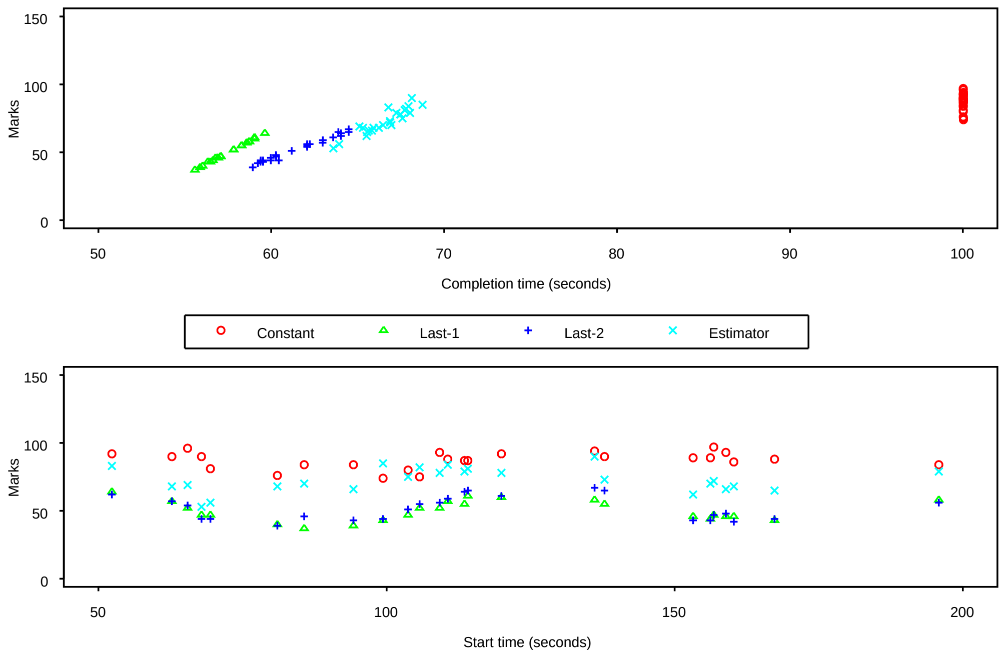

Figure 5: Results for experiment 2.

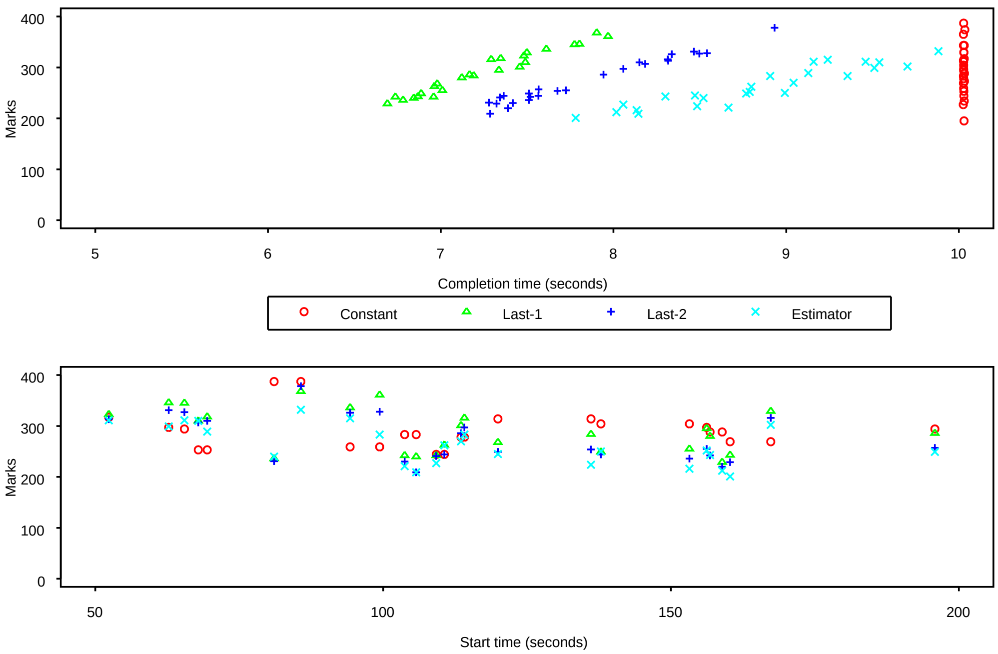

Figure 6: Results for experiment 3.

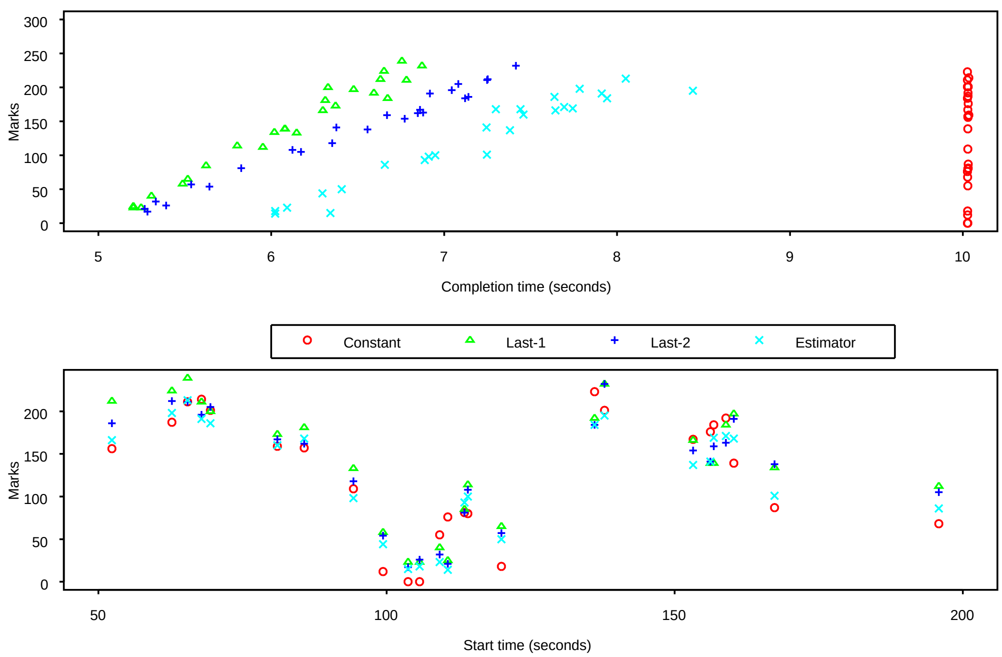

Figure 7: Results for experiment 4.

5 Conclusions and further work

It is hard to draw firm conclusions from this preliminary study. The simple strategies considered here vary quite widely in their performance. However, several relationships can be seen. The more time the users have to observe the feedback relative to the busy hours in the congestion levels the better they can be expected to perform. A strategy with little time to observe the congestion level relative to the timescale over which it varies need only spread out its packets uniformly and behave independently of the feedback marks.

In broad terms, for experiments 1, and 2, it is possible to do better than a constant rate strategy, and the Last-1 strategy, which is the simplest, performs best. This has some similarity with the 'Tit-for-Tat' strategy of the Prisoner's dilemma, or with a 'weather-forecast' strategy: predict tomorrow's weather from today's. It attempts to exploit short term stochastic fluctuations in the marking rate. Note that for our experiments, the round-trip time is short compared to the marking interval, and a more complex strategy may be needed if this is not true.

In experiments 3 and 4, where there is little time to send, all the strategies perform similarly. However even those which complete quickly do better than a single instance of TCP connection, which was observed to transfer the file in 5 seconds but at a cost of 410. Notice how much more variable the number of marks is in these cases: this is caused the individual stream acting as a price-setter not just a price-taker by as they can form a significant portion of the load. For experiment 4, the background load fluctuates widely, and in two cases the constant rate strategy saw no marks, since it hit a low load period and its effect on the load was not enough to cause congestion: in contrast, for the same period, the other strategies sent at a high rate with caused some congestion, so they incurred some tens of marks in the same period.

It is clear that further work is needed here. The current simulator machinery would permit far more elaborate experiments using multiple resources and possibly several simultaneously competing users.

6 Acknowledgements

Richard Gibbens was supported by a Royal Society University Research Fellowship with additional computing equipment provided by the EPSRC under grant GR/M09551. Grateful thanks to Paul Barham for wrestling with the simulator code.

References

[1] R. Axelrod. *The Evolution of Co-operation*. Penguin, London, 1984.

[2] S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communications Review*, 24:10–23, 1994. http://www-nrg.ee.lbl.gov/floyd/ecn.html.

[3] R. J. Gibbens and F. P. Kelly. Distributed congestion acceptance control for a connectionless network. In *Proceedings of the 16th International Teletraffic Congress — ITC 16*, 1999.

[4] R. J. Gibbens and F. P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35:1969–1985, 1999.

[5] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998. http://www.statslab.cam.ac.uk/∼frank/rate.html.

[6] P. B. Key and D. R. McAuley. Differential QoS and pricing in networks: where flow control meets game theory. *IEE Proc Software*, 146, 1999.

[7] Peter Key, Derek McAuley, Paul Barham, and Koenraad Laevens. Congestion pricing for congestion avoidance. Microsoft Research Technical Report MSR-TR-99-15, MSR, 1999.

[8] S. Kunniyar and R. Srikant. End-to-end congestion control schemes: utility functions, random losses and ECN marks. Technical report, University of Illinois, 1999.

[9] S. H. Low and D. E. Lapsley. Optimization flow control, I: basic algorithms and convergence. *IEEE/ACM Transactions on networking*, 1999. To appear, http://www.ee.mu.oz.au/staff/slow/.

[10] D. K. H. Tan. *Mathematical models of rate control for communication networks*. PhD thesis, University of Cambridge, 1999.