PUTTING QUALITY OF SERVICE INTO A NETWORK BY MAKING THE TRAFFIC MARKOVIAN

J.T.Bradley N.Thomas

Department of Computer Science
University of Durham
Durham DH1 3LE, UK.
Email: Jeremy.Bradley@durham.ac.uk

ABSTRACT

In this paper we examine an existing model of composable internet switches from which statistical guarantees of service can be obtained. We then present some small analysis to show how the model can be implemented so that its assumptions, in this case about Markovian traffic streams, are in fact properties of the implementation. In order to achieve this we need to construct a box which makes the traffic streams Markovian.

1 INTRODUCTION

This paper is really about a way of thinking about the modelling process (in this case stochastic modelling) and how it should be applied to an implementation. Usually a model is made of an existing system and the model is analysed to increase the understanding of that system. The model may make many assumptions and simplifications that make it practical to analyse, but in doing so also make the link between the original system and the model rather uncertain. It maybe that simulation and experimentation have to be carried out to justify the model's validity. Here, we propose a slightly different approach—we take the Holyer queueing model [5] for implementing statistical QoS in a network and show how its central assumption can be implemented in reality. This then makes the equivalence between the model and the implementation concrete and gives us the benefit of using the model's QoS predictions for traffic loss, throughput and delay predictable properties of the network—clearly a useful feature for internet-like networks.

1.1 Background

In Davies et al 1999 [4] and Holyer 2000 [5], a framework for managed traffic streams in a composable queueing system was analysed. The papers put forward a type of queueing node and stream characterisation which together allow end-to-end quality guarantees to be made across a network of such nodes (as shown in figure 1). Traffic streams are given a two-dimensional characterisation which determine how real-time a stream is and how much loss it can sustain. So an error corrected live video stream would need a high real-time rating but could sustain a greater rate of loss than say an FTP stream.

As such, each packet in the system is given a different characterisation so that, for instance, a packet from the video stream would go into box c (in figure 1)—indicating a high real-time requirement but an ability to sustain a certain amount of loss. On the other hand an FTP packet will need low loss but could sustain higher delay and might get put in box b. By understanding the interaction of all the loss/delay requirements of all the managed streams, probabilistic predictions can be made about the actual loss, delay and throughput behaviour of all the streams (e.g. a delay of less than 30ms with probability 0.93).

The reader is referred to the papers [4, 5] for the detail behind the system. What needs to be known in the context of this paper is that, for the whole system to operate correctly, incoming traffic streams are required to be Markovian. There is also a requirement that traffic between nodes needs to be reshaped to maintain an exponential distribution.

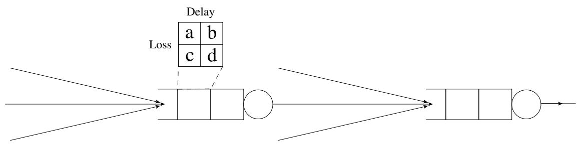

FIGURE 1. The loss/delay model for composable network switches

What is presented in this paper is a method for designing and analysing a traffic shaper box that could be placed at the edges of such a network and thus ensure that the delay-loss-throughput guarantees within the network are actually delivered. Without such a unit, the network will not function and the elegant theory cannot be used.

We suggest a G/M/1/K queueing system for such a shaping unit and present an analysis of G/M/1/K queues in general and a D/M/1/2 queue, as a simple example for this purpose.

2 STOCHASTIC TRANSITION SYSTEMS

2.1 G/M/1/K and M/G/1/K Queues

Most of the theory of finite queues that have one Markovian discipline is of the M/G/1/K variety [7]. This would appear to be because of a certain ambiguity in the operational workings of the G/M/1/K type.

There are two critical moments for a queue: when the buffer is empty and when the buffer is full. For any queue when the buffer empties, the service process is interrupted and only restarts (with re-sampling) when an item enters the queue.

When the buffer is full, it is not clear whether the arrival process is similarly interrupted until a space appears in the buffer, or whether it is allowed to continue concurrently on the off-chance that a service pre-empts the arrival in time to give an arriving item a place in the queue. In the latter case if a space is not available then the item can either be discarded or delayed in an upstream buffer. In this paper, we call the different arrival strategies *blocked* and *non-blocked*, respectively.

The reason that there is an unambiguous description of an M/G/1/K queue is that for a Markovian arrival pattern, there is no difference between blocked and non-blocked disciplines. That is to say, the residual of the arrival distribution after service occurs to give a free space in the buffer is always exponential, by the memoryless property.

We argue that (as a result of the uniform interpretation of behaviour in an empty M/G/1/K queue matching that of a full blocked G/M/1/K queue) M/G/1/K queues are symmetrically related to G/M/1/K blocked queues. This can be shown rigorously by examining the stochastic transition systems of the two queues.

Where does this leave us? Clearly given the greater theoretical knowledge for M/G/1/K, we would be better off using that, if possible, than struggling with a harder model. It indicates that in the event we wish to analyse G/M/1/K queues (and we have control over the arrival discipline), we have the ability to subtly alter our operational implementation of the G/M/1/K so that all the M/G/1/K theory can be brought to bear.

2.2 Analysis of D/M/1/2-blocked Queue using Stochastic Aggregation

2.2.1 Introduction

In this section we look at a simple example of a traffic-shaping queue, a finite queue that takes deterministic arrivals (an ATM stream for instance) and shapes it with a Markovian service pattern. We are looking primarily at the steady-state distribution of such a system to see how it varies with load. If such a system is never empty it will endow all leaving traffic with a Markovian inter-departure distribution (the key quality for plugging into the arrival distribution of a downstream queue node). The steady-state distribution will give us a good handle on how close the inter-departure distribution is to exponential while at the same time indicating how much loss occurs.

In figure 2, we show the STS as derived from the blocked D/M/1/2 state space, that affects the steady-state distribution. From earlier work, we also know that $X|X < Y = X$ for $X \sim \delta(d)$, a deterministic event.

In this example, $Y \sim \exp(\mu)$ and so $p_1 = 1 - e^{-d\mu}$.

We divide the analysis into two parts, finding π_0 and

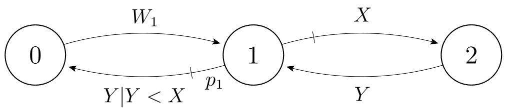

FIGURE 2. The stochastic state space of D/M/1/2 blocked

finding π_2 with $\pi_1 = 1 - (\pi_0 + \pi_2)$.

2.2.2 Finding π_0 and π_2

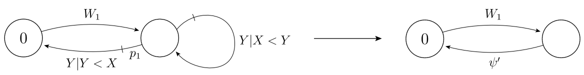

FIGURE 3. Stochastic aggregation around state 1

Figure 3 shows the aggregation of the system (using stochastic aggregation techniques from [2] and [1]), while preserving state 1. We require to find both $\mathbb{E}(W_1)$ and $\mathbb{E}(\psi')$:

$$L_{W_1}(z) = \frac{(1-e^{-d\mu})e^{-dz}}{\frac{\mu}{\mu+z}(1-e^{-d(\mu+z)})} \tag{1}$$

$$\begin{aligned} L_{\psi'}(z) &= \frac{p_1 L_{Y|Y<X}(z)}{1-(1-p_1)L_{Y|X<Y}(z)} \\ &= \frac{\frac{\mu}{\mu+z}(1-e^{-d(\mu+z)})}{1-\frac{\mu}{\mu+z}e^{-d(\mu+z)}} \end{aligned} \tag{2}$$

Giving:

$$\mathbb{E}(W_1) = d - \frac{1}{\mu} + \frac{de^{-d\mu}}{1-e^{-d\mu}} \tag{3}$$

$$\mathbb{E}(\psi') = \frac{1}{\mu(1-e^{-d\mu})} \tag{4}$$

$$\pi_0 = \frac{\mathbb{E}(W_1)}{\mathbb{E}(W_1)+\mathbb{E}(\psi')} = \frac{d\mu-(1-e^{-d\mu})}{d\mu+e^{-d\mu}} \tag{5}$$

Figure 4 shows a similar aggregation of the system, while preserving state 2.

$$\begin{aligned} L_{\psi}(z) &= \frac{(1-p_1)L_X(z)}{1-p_1 L_X(z)} \\ &= \frac{1}{e^{d(\mu+z)}-e^{d\mu}+1} \end{aligned} \tag{6}$$

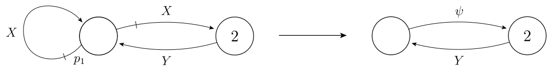

FIGURE 4. Stochastic aggregation around state 2

Giving:

$$\mathbb{E}(\psi) = de^{d\mu} \tag{7}$$

$$\pi_2 = \frac{\mathbb{E}(Y)}{\mathbb{E}(Y)+\mathbb{E}(\psi)} = \frac{e^{-d\mu}}{d\mu+e^{-d\mu}} \tag{8}$$

2.3 Results

Figure 5 presents a 2-D plot of all three steady-state probabilities with μ, the service rate, fixed to 1 and d, the inter-arrival period, allowed to vary. From the left to right on the x-axis, the system is going from infinite overload to underload (as $d > 1$). One of the points this presentation raises, for example, is that even for an arrival rate set at twice the service rate, $d = \frac{1}{2}$, somewhere around 10% of the the time the buffer will still be empty (from π_0). Time spent empty, in the context of a shaper, represents periods of producing non-Markovian shaped traffic, i.e. the shaper not performing its task.

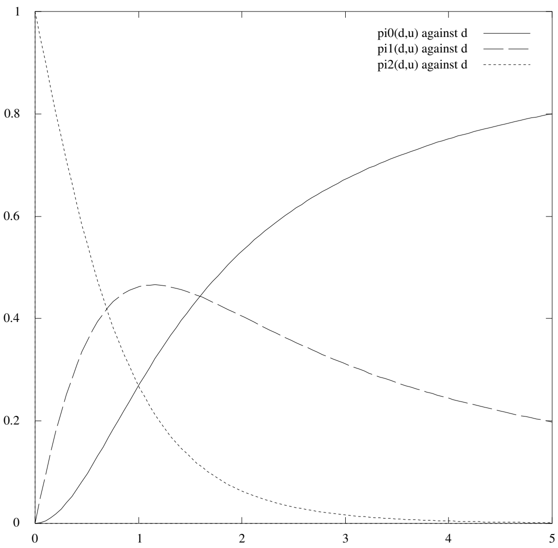

FIGURE 5. Plot of steady-state distribution for fixed service $\mu = 1$ and d varying from overload to underload

Similarly, by observing the steady-state π_2 we can ascertain that a high level of loss is sustained for high arrival rates. This would especially be a problem in overload conditions which are unfortunately also those most conducive to good shaping. As we discuss in the conclusion, this can be solved through use of a larger buffer. More usefully, this loss can be quantified approximately by using a model with one larger buffer than actually used in order to observe *overflow* traffic.

Figures 6–8 depict 3-D plots against a d–μ parameter space and are of interest because of the existence of a maximal ridge in π_1. This line on the surface represents conditions which would necessarily minimise both loss and non-shaping activity within the system.

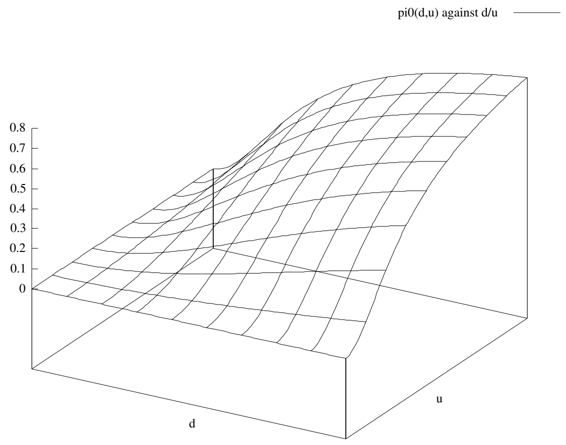

FIGURE 6. A 3-D plot of π_0 for values of d and μ

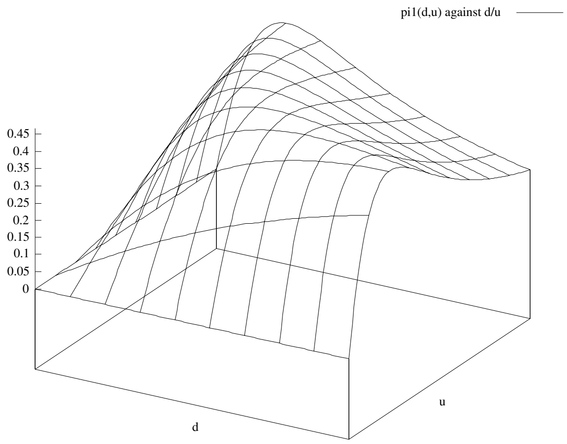

FIGURE 7. A 3-D plot of π_1 for values of d and μ

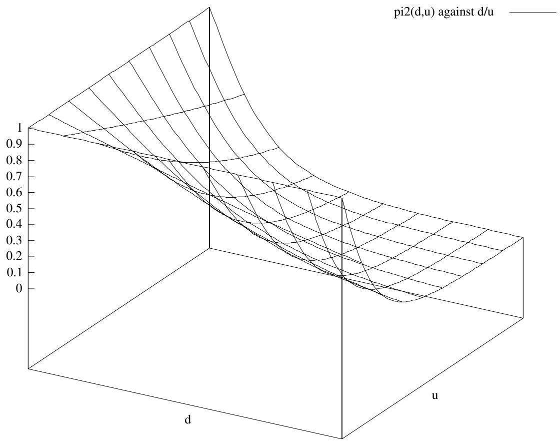

FIGURE 8. A 3-D plot of π_2 for values of d and μ

2.4 A Second Solution: Null Packets

Clearly, we have a dilemma: to have full buffers creates perfectly shaped traffic but has associated large loss; to have only partially full buffers solves the loss problem but means we are more likely to lose our Markovian shaping. We can partially solve the loss problem through having larger buffers (although even here we have to be careful not to introduce huge latencies into the traffic shaper). To solve the shaping problem, we can redesign our service engine.

The non-Markovian aspect of the traffic arises when the buffer empties, waits for an arrival and immediately starts processing a packet once one has arrived. The delay in processing the next packet is the problem. Clearly the server cannot force the arrival before it is due, however it can pretend that it has a queued item and process it in a Markovian fashion and carry on doing so until a packet actually does arrive. In practice, a protocol would need a null packet that the server could throw out at the culmination of each dummy service, so that the Markovian stream is maintained. When a packet does arrive during a null service, the server can throw away its current service time and commence processing of the newly arrived packet with a fresh Markovian service (uniquely in the G/M shaper case, as a result of memorylessness).

This null service discipline is harder to analyse as there are potentially an indefinite number of null services that might occur before a real packet arrives. We note that an arrival at a point in a sequence of null services will reset the buffer to state 1 (as shown in figure 9), so this gives us an infinite queueing structure that is not like our previous G/M/1/K models.

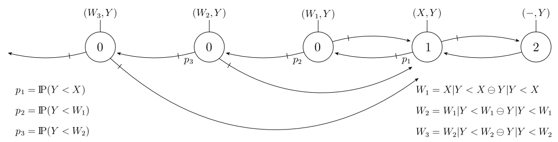

FIGURE 9. Stochastic transition system of null-packet-enabled D/M/1/2 queue—the tuples represent the current state of the arrival and service processes

The null packet solution is clearly a preferable solution to the the standard blocked G/M/1/K model for the reasons stated and its analysis is the subject of future research.

3 CONCLUSION

In this example, our goal is that in order to gain predictability through model accuracy we need to shape incoming traffic. We discuss a stochastic transition system decomposition of two different flavours of G/M/1/K queue. We propose that using a G/M/1/K queue would be a good way of shaping non-Markovian traffic in a data stream, so that it subsequently has a Markovian shape. Such a shaped stream would now be suitable for pushing into a queueing network which requires Markovian streams—such as that originally proposed in [4]. The ability to be able to reason about traffic streams accurately and quantitatively is an important and outstanding problem in Quality-of-Service management in networks.

We have identified two types of G/M/1/K queue. We have suggested that the blocked G/M/1/K queue is an easier model to deal with as it can be analysed as an M/G/1/K system, by symmetric arguments. Again an implementation would have to reproduce the correct blocking strategy for arrivals for this analysis to hold true. We have analysed a basic D/M/1/2-blocked system using stochastic aggregation methods—since this gives us extra aggregate distribution information—and plotted the steady-state probabilities in a 1-D and 2-D parameter space. We have shown that this type of analysis can show how much of the traffic is exponentially shaped and how much loss occurs as a result.

Larger systems need to be analysed to study more realistic buffer sizes and an ability to extract other metrics would be useful. Stochastic aggregation provides the ability to obtain, for instance, response-time, return and transient distributions [2]. As solely the steady-state values were required for our simple analysis, we suggest using a blocked queue G/M/1/K queue and exploiting the symmetry with standard M/G/1/K analysis [3, 6].

Finally, as future work, we suggest analysing a null packet service discipline which would provide perfect Markovian shaping. The buffer size would then only be used for moderating the packet loss under overload conditions.

References

[1] BRADLEY, J. T. *Towards Reliable Modelling with Stochastic Process Algebras*. PhD thesis, Department of Computer Science, University of Bristol, Bristol BS8 1UB, UK, October 1999.

[2] BRADLEY, J. T., AND DAVIES, N. J. A matrix-based method for analysing stochastic process algebras. In *ICALP Workshops 2000, Process Algebra and Performance Modelling Workshop* (Geneva, July 2000), J. D. P. Rolim, Ed., University of Waterloo, Ontario, Canada, Carleton Scientific, pp. 579–590.

[3] COX, D. R., AND MILLER, H. D. *The Theory of Stochastic Processes*. Methuen, 1965.

[4] DAVIES, N., HOLYER, J., AND THOMPSON, P. End-to-end management of mixed applications across networks. In *WIAPP'99, Proceedings of the IEEE Workshop on Internet Applications* (San Jose, California, July 1999), IEEE, pp. 12–19.

[5] HOLYER, J. A queueing theory model for real data networks. In *UKPEW'00, Proceedings of the 16th Annual UK Performance Engineering Workshop* (Durham, July 2000), N. Thomas and J. T. Bradley, Eds., Department of Computer Science, University of Durham, pp. 59–70.

[6] KEILSON, J. The ergodic queue length distribution for queueing systems with finite capacity. *Journal of the Royal Statistical Society B*, 28 (1966), 190–201.

[7] KLEINROCK, L. *Queueing Systems, Volume I: Theory*. John Wiley & Sons, 1975.