17

Markov Chain Animation Technique Applied to ATM Bandwidth Derivation and Tandem Switches

J.M.Griffiths and J.M.Pitts
Queen Mary and Westfield College, University of London
Mile End Road, London, E1 4NS, United Kingdom
Tel. +44 171 975 5596, Fax. +44 181 981 0259
email J.M.Griffiths@qmw.ac.uk

Abstract

The derivation of queue lengths and buffer fills in ATM networks with particular types of cell arrival is known explicitly for some specific cases, but often simulation must be resorted to, which is time consuming and the accuracy is difficult to determine, particularly when one is considering cell losses or CDV around the 10^{-8} level. We describe an alternative technique which involves iteration of the Markov transition and state probabilities for a queue; we have called this Animation. Its principle is simply that of repetitively applying the appropriate Markov chain transition matrix to the current queue length probabilities to produce a new set of probabilities. In the case of a known cell sequence arrival the transition matrix changes for each iteration to reflect the arrival probability. Animation presents some numerical difficulties and techniques are described to overcome these. The specific application described is the derivation of equivalent bandwidth of cell patterns resulting from ATM policing or shaping, and the transit of these cells through networks containing multiple switching stages.

Keywords

ATM, bandwidth, policing, queuing, switching

1. INTRODUCTION

In connection with studies in policing functions for ATM [1] we had to consider their limitations and, in particular, the most adverse patterns (MAP) of full cells which would meet the policing criteria and hence could be applied to the network by users. To give an idea of the problem a typical MAP would be 10 full, 25 empty, 11 full, 26 empty, 11 full, 26 empty, 5 full and 133 empty cells. After this the sender could repeat the pattern. For different system parameters the number of full cells in each burst

D. D. Kouvatsos (ed.), *ATM Networks*

could be much reduced and the number of bursts could be many more; the number of final empty cells may rise to several thousand. In order to assess the impact of a stream with such a MAP we needed to estimate their equivalent bandwidth and the method chosen was to add the stream to a infinite number of random sources with a Poisson load of 0.5. The resulting buffer fill probability would then be compared with a pure Poisson load adjusted to give a similar distribution in the area of probabilities of 10^{-8}. The excess of this second Poisson load over the background Poisson load of 0.5 would be regarded as the equivalent bandwidth of the MAP. To avoid the introduction of an arbitrary parameter the buffer was assumed to be infinite. A Poisson background load seemed appropriate in the absence of any clear understanding of the actual statistics; in the circumstances where there is traffic from a large number of sources ranging from CBR to bursty, central limit considerations suggested this choice.
Although the concept appeared reasonable its evaluation was not obvious. Analytic solutions to the general Poisson + pattern queuing problem are not known; the other alternative was simulation, but to get comparisons at the 10^{-8} level an impractical number of cells would be needed. From this need the technique of Markov Chain "animation" developed.

2. BACKGROUND

The Markov chain is well established as a conceptual device for describing the manner in which state probabilities are modified by events and their probabilities. In ATM terms the classic such chain is the buffer fed by a multiplicity of sources.

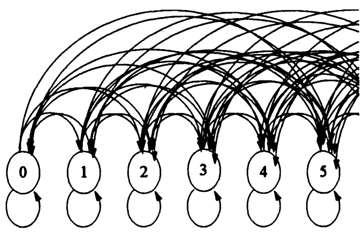

Figure 1 Markov Chain

In figure 1 the circles represent the states - in the case of the buffer they are the fills of the buffer, each with its associated probability s_0, s_1, s_2, etc. The circular arrows represent the probability that after an event the machine remains in that same state s_{00}, s_{11}, s_{22}, etc. The curved arrows represent the probabilities of transition from one state to another. In the case of a buffer with a single server s_{11}, s_{22}, etc. will equal the probability, p_1, that one cell arrives. In the case of s_{00} this equals the probability that one or zero cells arrive. Similarly the probability that the machine moves from state n (n cells in buffer) to state n + 1 (n + 1 cells in buffer), i.e. s_{01}, s_{12}, s_{23}, etc. is the probability, p_2, that 2 cells arrive; the probability that the machine moves from state n to state n - 1, i.e. s_{10}, s_{21}, s_{32}, etc. is the probability, p_0, that no cells arrive.
The customary use for this information is to set up a set of equations to get an explicit solution to the steady state values of the state probabilities. This paper explores the animation method in which the Markov chain is iteratively applied to a set of data to derive solutions to queuing problems.

3. THE ANIMATION TECHNIQUE

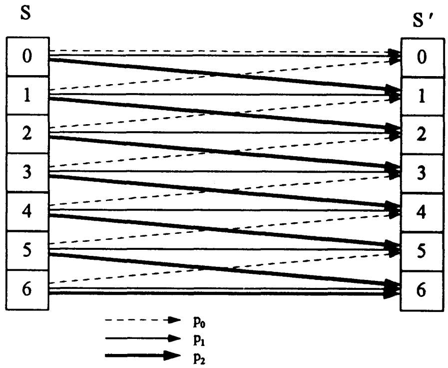

Figure 2 Simple Animation

In a machine with a finite number of states the initial state may be represented by a vector S of the form $[s_0, s_1, s_2, \ldots\ldots\ldots]$. Multiplication by the transition matrix will produce the new vector S′ representing the state probabilities after the first event (e.g. a batch of cells arriving and one being served). Thus one could start with an initial condition where no cells have arrived for a very long time, $s_0 = 1$ and $s_1, s_2, \ldots = 0$. After the first event $s_0 = s_{00}$, $s_1 = s_{01}$, $s_2 = s_{02}$, etc. The next multiplication will establish the state after 2 events and so on. Continuing multiplication will lead S to tend to the steady state condition as calculated in the usual explicit way.

As an example of the process in action let us consider a queue of maximum length 6, being fed by two sources each with probability of full cell = 0.25, giving a total load of 0.5. This simplest of Binomial distributions gives the probability of an event with no full cells arriving, p_0, = 0.5625; with 1 cell, p_1, = 0.375; with 2 cells, p_2, = 0.0625. With a single cell served per event then $s_{11} = s_{22} = s_{33} = s_{44} = s_{55} = 0.375$. $s_{01} = s_{12} = s_{23} = s_{34} = s_{45} = s_{56} = 0.0625$. $s_{65} = s_{54} = s_{43} = s_{32} = s_{21} = s_{10} = 0.5625$. $s_{00} = 0.9375$ (the probability of no cells plus the probability of one cell). $s_{66} = 0.4375$ (the probability of one cell plus the probability of two cells). All other values in the transition matrix are zero as the corresponding transition is not possible. These probabilities can be repetitively applied to a starting state as shown in Figure 2. After 30 iterations stability is achieved to 6 decimal places accuracy. Effectively this process gives the transient state probabilities which are only rarely of interest. However it may be used as a means of finding the steady state probabilities in cases where an explicit solution is obscure. It has further use where the load (and hence the transition matrix) is fluctuating in a specific way, and this will be the subject of the core of this paper.

4. THE INFINITE QUEUE

In the introduction it was mentioned that derivation of the equivalent bandwidth was to be done by considering the probability distributions of an infinite queue loaded with:

a) a background Poisson source + pattern

b) a second purely Poisson source

An infinite queue was chosen to avoid introducing another arbitrary parameter but presents obvious computational difficulties as only finite queues can be handled numerically.

Extending figure 2 to the infinite case (figure 3) shows the difficulty; however large the actual queue length chosen, Z, with only the background source present one always needs one further element, Z^+, to compute the next value of Z. Assuming that Z^+ is zero produces considerable distortion of the values in the upper range of the queue resulting in the need to compute a much extended queue length, with run time penalties. However a simple alternative results from the well known observation that the state probabilities closely approximate to a geometric series for low probabilities. Setting $Z^+ = Z^2/Y$ means that computation becomes apparently error free for practical purposes. This is indicated in figure 3 by the curved arrows.

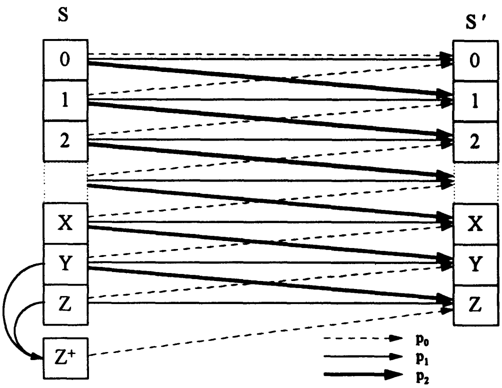

Figure 3 Infinite Queue

5. ROUNDING ERRORS

For a Poisson source of rate 0.5, calculation of the buffer fill probabilities by the explicit method or by continuous iteration results in obvious errors at low probabilities owing to rounding errors, even using double length arithmetic as shown in Table 1. It is clear that the figures are tending to a small (in this case negative) residual constant value rather than the expected value of zero. For the work to be described later it is necessary to continue the iterations considerably more than this initial simple example required. This leads to a residual constant of ever increasing magnitude which, if left uncorrected, would limit the utility of the technique. A solution to this problem also exploits the fact that the values should approximate a geometric series. Simple algebra can determine the value of the residual constant.

If s_X, s_Y, s_Z are the values of the last three probabilities calculated and they are in a geometric progression with a constant error, ε, then

$$s_Z = s + \varepsilon, \quad s_Y = \kappa s + \varepsilon, \quad s_X = \kappa^2 s + \varepsilon$$

Eliminating s and κ, the constant error is given by:

$$\varepsilon = (s_X.s_Z - s_Y^2)/(s_Z - 2.s_Y + s_X)$$

Queue Position	Uncorrected Probability	Corrected Probability
20	2.03E-11	
21	5.77E-12	
22	1.64E-12	
23	4.68E-13	
24	1.33E-13	
25	3.79E-14	
26	1.08E-14	
27	3.07E-15	
28	8.73E-16	8.75E-16
29	2.47E-16	2.49E-16
30	6.88E-17	7.09E-17
31	1.81E-17	2.02E-17
32	3.67E-18	5.74E-18
33	-4.35E-19	1.64E-18
34	-1.60E-18	4.65E-19
35	-1.94E-18	1.33E-19
36	-2.03E-18	3.77E-20
37	-2.06E-18	1.07E-20
38	-2.07E-18	3.06E-21
39	-2.07E-18	8.70E-22
40	-2.07E-18	2.48E-22

Table 1 Effect of Rounding Errors

The 3rd column in table 1 shows the effect if this is subtracted. The process works well until the denominator in the above expression becomes too small (values of s around 10^{-26}). In the case demonstrated the IEEE 64-bit floating point format was used with sign (1 bit), exponent (11 bits) and mantissa (52 bits). It is convenient to incorporate this correction into a normalisation procedure which is applied after every iteration; the procedure can also sum the queue probabilities to ensure that rounding errors do not cause the overall total to deviate from unity, neglecting the very small probability that the queue will exceed the length computed. Using this technique reduces computation time when low probability events are of interest as it avoids the use of extended length arithmetic.

6. ADDING STREAM CONSISTING OF AN ARBITRARY PATTERN

Figure 3 showed the transitions when the background load was present. If an extra full cell is present in the additional stream then the transitions are modified to those shown in figure 4. This is obviously not a stable situation as queue length probabilities will increase without limit. Interspersed empty cells are required in the additional stream to allow the queue length to reduce. Note that for simplicity of explanation we are still working with the case of 2 random sources to form the background load; the Poisson case works in exactly the same manner but of course the probabilities that more than two background cells arrive are

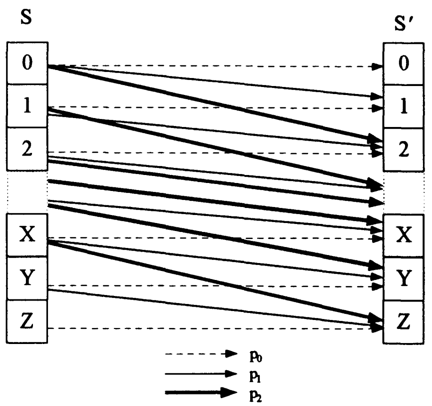

Figure 4 Background+Extra Cell

non - zero and so in the case of no extra full cell arriving $s'_n = \sum_{i=0}^{n+1} s_{n+1-i}.p_i$ and if there is an additional full cell $s'_n = \sum_{i=0}^{n} s_{n-i}.p_i$ assuming that an arriving cell to an empty queue can be served in the same time slot.

We now have all the components necessary to animate the buffer state probabilities. Starting from any initial set of state probabilities the appropriate transitions will be applied - in the rather simplified case we have been considering they will be as shown in figure 4 for every occurrence of a full cell in the additional stream, and figure 3 when there is no extra full cell in the additional stream. We will then end up with a new set of state probabilities. This can then be repeated using those final set of state probabilities as the new starting point. Eventually the final state probability vector will be the same as the starting vector. How many iterations this takes depends on how near the final answer is to the starting condition. An obvious choice is to start with the fill probabilities due to the background stream alone either derived iteratively as shown in section 3 above, or by an explicit solution.

Taking the simple example considered previously together with an additional stream of 1 full cell, 2 empty cells, 2 full cells and, finally, 4 empty cells, the process is as shown in table 2A. For clarity the animations of full cells are shaded. The first column is the fill probabilities due to the background load calculated explicitly. This is then operated on by the transitions appropriate to an additional full cell arriving. This is followed by the two empty cells and so on. The derived parameter Z^+ is shown where it is calculated for the empty extra cell case.

Slot→	1	2	3	4	5	6	7	8	9
Add.Patt→	Full	Empty	Empty	Full	Full	Empty	Empty	Empty	Empty
Initial ↓									
8.89E-01	5.00E-01	6.87E-01	7.75E-01	4.36E-01	2.45E-01	4.44E-01	5.84E-01	6.80E-01	7.45E-01
9.88E-02	3.89E-01	2.33E-01	1.68E-01	3.85E-01	3.80E-01	2.98E-01	2.35E-01	1.92E-01	1.62E-01
1.10E-02	9.88E-02	6.75E-02	4.60E-02	1.37E-01	2.49E-01	1.70E-01	1.19E-01	8.47E-02	6.16E-02
1.22E-03	1.10E-02	1.10E-02	9.02E-03	3.28E-02	9.41E-02	6.48E-02	4.51E-02	3.17E-02	2.25E-02
1.35E-04	1.22E-03	1.22E-03	1.22E-03	6.95E-03	2.48E-02	1.82E-02	1.31E-02	9.48E-03	6.85E-03
1.50E-05	1.35E-04	1.35E-04	1.35E-04	1.10E-03	5.27E-03	4.05E-03	3.08E-03	2.33E-03	1.74E-03
1.67E-06	1.50E-05	1.50E-05	1.50E-05	1.35E-04	9.22E-04	7.66E-04	6.22E-04	4.96E-04	3.91E-04
Z^+→	1.66E-06	1.67E-06			1.61E-04	1.45E-04	1.25E-04	1.06E-04	

Table 2A Initial Animation of Background + Pattern

The final column indicates the fill probabilities after the pattern has passed through. These probabilities may be substituted as a new set of initial conditions and the process repeated. Eventually (after 10 cycles in this case) the initial and final states are the same, as shown in table 2B.

Slot→	1	2	3	4	5	6	7	8	9
Add.Patt→	Full	Empty	Empty	Full	Full	Empty	Empty	Empty	Empty
Initial ↓									
6.85E-01	3.86E-01	5.59E-01	6.60E-01	3.71E-01	2.09E-01	3.86E-01	5.18E-01	6.15E-01	6.85E-01
1.68E-01	3.52E-01	2.40E-01	1.89E-01	3.53E-01	3.38E-01	2.78E-01	2.29E-01	1.94E-01	1.68E-01
7.84E-02	1.50E-01	1.13E-01	8.46E-02	1.60E-01	2.45E-01	1.80E-01	1.34E-01	1.02E-01	7.84E-02
3.79E-02	6.12E-02	4.86E-02	3.77E-02	6.47E-02	1.18E-01	8.81E-02	6.62E-02	4.99E-02	3.79E-02
1.73E-02	2.88E-02	2.21E-02	1.70E-02	2.90E-02	5.05E-02	3.89E-02	2.97E-02	2.27E-02	1.73E-02
7.76E-03	1.32E-02	1.01E-02	7.73E-03	1.31E-02	2.23E-02	1.72E-02	1.32E-02	1.01E-02	7.77E-03
3.47E-03	5.94E-03	4.56E-03	3.50E-03	5.93E-03	1.01E-02	7.71E-03	5.92E-03	4.53E-03	3.47E-03
Z^+→	2.67E-03	2.06E-03			4.54E-03	3.47E-03	2.65E-03	2.03E-03	

Table 2B Animation to Stability of Background + Pattern

We have now reached a stable set of state probabilities. A more graphical illustration of the process can be seen in figure 5. This shows how the state probabilities vary during the iterative process, starting from the initial condition until stability is reached.

State	Probability
0	4.88E-01
1	2.60E-01
2	1.39E-01
3	6.36E-02
4	2.85E-02
5	1.27E-02
6	5.74E-03

Table 3 Mean State Probabilities

All that is now necessary is to calculate the mean probability for each state by taking the average value of each row in the stable state in table 2B; of course the initial and final conditions should only be included once. The result is shown in table 3. In order to estimate the equivalent bandwidth it is only necessary to find the random load which approximates to these buffer fill probabilities. Any excess over 0.5 (the background load) will be the equivalent bandwidth.

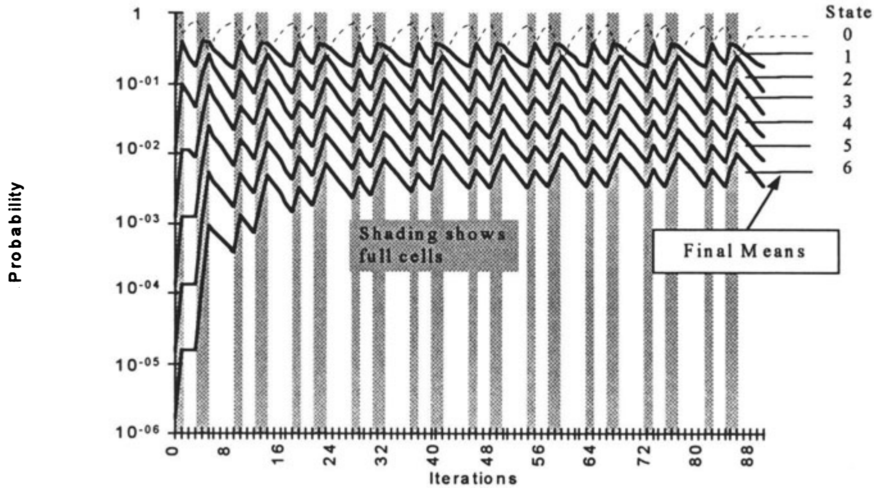

Figure 5 Animation to Stability - Background + Pattern

7. PRACTICAL RESULTS

Let us have a look at results from the MAP mentioned right at the beginning (i.e. 10 full, 25 empty, 11 full, 26 empty, 11 full, 26 empty, 5 full and 133 empty cells). Note that in this case the large number of final empty cells in the pattern will mean that the queue length probabilities will have returned to the level of the background load and a second cycle will not be needed. In fact with other patterns we considered ending with several thousand empty cells, steps were taken to discontinue the animation once the probabilities had reached a stable state.

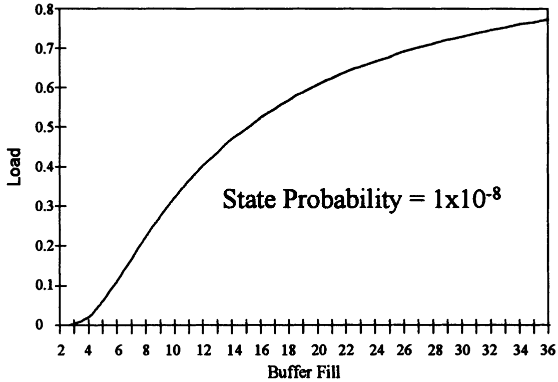

Figure 6 Poisson Load

The tabulation of the resulting probability distribution shows that the buffer fill of 28 occurs with probability of about 10^{-8}. From Figure 6 this is equivalent to a Poisson load just over 0.7. More careful examination comes up with a figure of 0.714. Figure 7 shows the queue length probability distribution due to a Poisson load of 0.5 together with the distribution from the animation with the extra stream. It can be seen that the distribution due to the combined load is approximately that from a purely Poisson load. Thus it can be argued that the additional pattern contributes a load equivalent to a Poisson load of 0.714 - 0.5 = 0.214, representing the equivalent bandwidth.

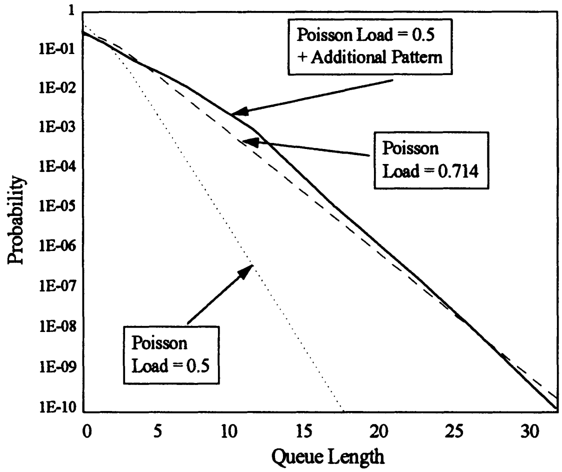

Figure 7 Comparison of Probabilities

8. EQUIVALENT BANDWIDTH: SENSITIVITY TO DERIVATION

In deriving a figure for the equivalent bandwidth two plausible but arbitrary parameters were included. They were the background level of 0.5 and the comparison value of 10^{-8}. Inspection of figure 7 shows that had a higher comparison value been chosen then a higher value for the equivalent bandwidth would have resulted. A summary of the variation for different values of the two parameters is shown in Figure 8 using the

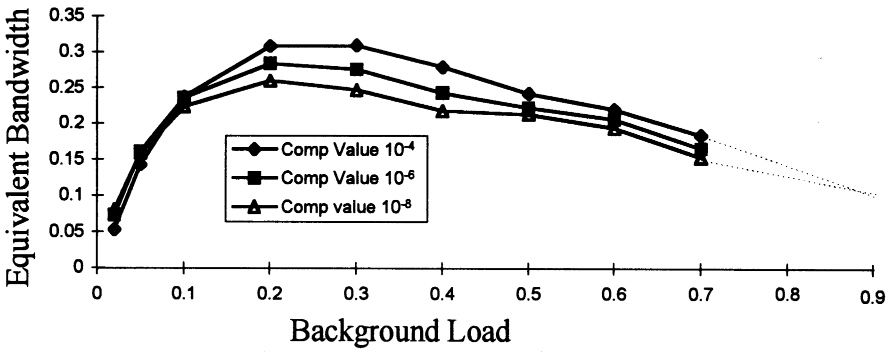

Figure 8 Equivalent Bandwidth: derivation sensitivity

MAP mentioned above. It can be seen that higher values of the comparison value lead to higher estimates of the equivalent bandwidth, implying that the probability distribution function continues to be of the form shown in figure 7. At higher values of the background load the apparent equivalent bandwidth falls. This is inevitable as, in the limit, queues become very large, virtually all cells are full, and the details of the distribution are of little consequence. As the combined load must be less than 1 and the pattern in question contributes 0.1 the trend is indicated by the dotted lines extending to the right. At low background levels the queuing tends to that due to the MAP alone and, as this is a single source, there will be no queuing and hence zero equivalent bandwidth. It can be seen that to ascribe an exact figure to the equivalent bandwidth derived by these means is not possible. However working in the chosen region results in a figure that gives a good indication of the impact of the pattern on the network buffers.

9. PATTERN CELL POSITION PROBABILITY DISTRIBUTION

Let us once again revert to the simple example shown in Table 2B. The columns indicate the queue length probabilities but there is no indication where in the queue the pattern cell might be. If for some reason the pattern cell is always served after the background cells then the first pattern cell will have to wait for an extra time slot before it is served with a probability (from Table 2B) of 0.352, and for 2 extra slots with probability 0.150 and so on. Similarly the other pattern cells will be delayed in the same way. There was always a probability that there was no background cell and the pattern cell was the only one in the queue in which case it would be served immediately; this arises with probability 0.386. The whole process can be tabulated as shown in Table 3.

Slot =	1	2	3	4	5	6	7	8	9
1st cell	3.86E-01	3.52E-01	1.50E-01	6.12E-02	2.88E-02	1.32E-02	5.94E-03	0	0
2nd cell	5.93E-03	0	0	3.71E-01	3.53E-01	1.60E-01	6.47E-02	2.90E-02	1.31E-02
3rd cell	2.23E-02	1.01E-02	0	0	2.09E-01	3.38E-01	2.45E-01	1.18E-01	5.05E-02
Total	4.14E-01	3.62E-01	1.50E-01	4.32E-01	5.91E-01	5.11E-01	3.16E-01	1.47E-01	6.36E-02

Table 3 Cell Delay Distribution - Pattern cell served last

The table shows that the original pattern cell that had a probability of 1 of being in a particular slot may now occur in one of several slots with appropriate probabilities. The shading indicates the position of the original pattern cells. This is a sort of convolution but the delay probability distribution varies from cell to cell. As the sequence is repetitive probabilities wrap round to the start. The bottom row shows the probability of there being a pattern cell in any slot. Figure 9 illustrates the effect graphically; the clear columns are the original pattern cell positions and the shaded area represents the pattern cell position probability. The different delay variations for the different cells can clearly be seen.

It may be argued that the assumption that the pattern cells are served last is unrealistic. The converse assumption that the pattern cell is served before any background cells arriving in that slot is also easy to evaluate. In that case the first pattern cell will be confronted by the queue length probability distribution due to the cells that have arrived previously. Thus the first pattern cell will not be delayed with probability 0.685, be delayed by one slot with probability 0.168 and so on. Table 4 gives the corresponding figures.

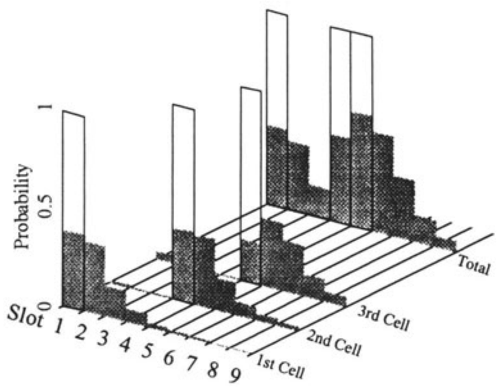

Figure 9 Cell Probabilities

Slot=	1	2	3	4	5	6	7	8	9
1st cell	6.85E-01	1.68E-01	7.84E-02	3.79E-02	1.73E-02	7.76E-03	3.47E-03	0	0
2nd cell	3.50E-03	0	0	6.60E-01	1.89E-01	8.46E-02	3.77E-02	1.70E-02	7.73E-03
3rd cell	1.31E-02	5.93E-03	0	0	3.71E-01	3.53E-01	1.60E-01	6.47E-02	2.90E-02
Total	7.01E-01	1.73E-01	7.84E-02	6.98E-01	5.77E-01	4.45E-01	2.02E-01	8.17E-02	3.67E-02

Table 4 Cell Delay Distribution - Pattern cell served first

The more realistic case of the pattern cell being served at random amongst the background cells is more complex. With N background cells arriving there are N+1 places that a pattern cell might find itself and all are equally likely. Hence if N background cells arrive with probability p_N, then the probability that exactly n background cells precede a pattern cell is $p_N/(N+1)$ $0 \leq n \leq N$, Hence the total probability of exactly n background cells before the pattern cell, Þn, is found by summing for all N. Hence $Þ_n = \sum_{N=n}^{\infty} \frac{1}{N+1} p_N$

Taking the simple case we have been pursuing, and the figures from section 3:

$Þ_0 = p_0 + p_1/2 + p_2/3 = 0.7708$
$Þ_1 = p_1/2 + p_2/3 = 0.2083$
$Þ_2 = p_2/3 = 0.0208$

These probabilities may then be applied in the manner of Figure 4 to the queuing probabilities before the Pattern Cell arrival to give the queue length as seen by the arriving pattern cell. The results are shown in Table 5.

initial →		Slot 3 →		Slot 4 →	
6.85E-01	5.28E-01	6.60E-01	5.09E-01	3.71E-01	2.86E-01
1.68E-01	2.72E-01	1.89E-01	2.83E-01	3.53E-01	3.49E-01
7.84E-02	1.10E-01	8.46E-02	1.18E-01	1.60E-01	2.05E-01
3.79E-02	4.90E-02	3.77E-02	5.06E-02	6.47E-02	9.05E-02
1.73E-02	2.29E-02	1.70E-02	2.27E-02	2.90E-02	3.92E-02
7.76E-03	1.04E-02	7.73E-03	1.03E-02	1.31E-02	1.75E-02
3.47E-03	4.65E-03	3.50E-03	4.66E-03	5.93E-03	7.90E-03

Table 5 Figures of table 2B Modified to Case where Pattern Cell is Queued at Random

These may be rearranged in the same manners as Tables 3 and 4 to get the Cell delay distributions. Figure 10 compares the results of the 3 environments. The tall columns show the original positions of the 3 cells. The other columns show the Pattern Cell position probabilities for the first in the queue, last in the queue and random position environment. Not surprisingly the random environment produces a result between the first and the last in magnitude.

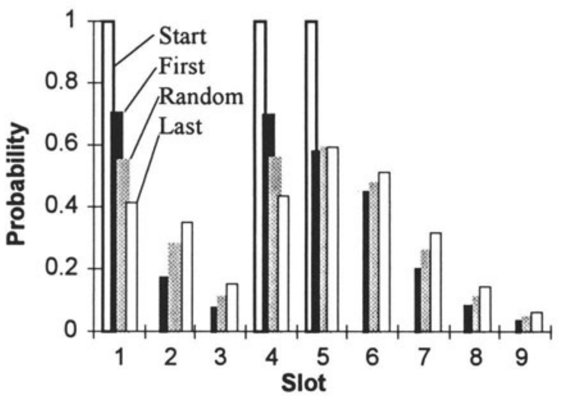

Figure 10 Comparison of Effect of Pattern Cell Priority

10. EQUIVALENT BANDWIDTH OF DISTRIBUTED CELL PROBABILITIES

The technique described in section 6 above assumes that there either 'is' or 'is not' a cell present. Extending this to the environment where a pattern cell is present with a given probability presents no difficulty.

Let c be the probability of a pattern cell. If the probability of n background cells is p_n then the probability of n cells from the pattern and the background together, $pc_n = p_n * (1 - c) + p_{n-1} * c$. It is found easiest to separate the queuing into two processes - arrival and serving. Given a vector representing the first $N+1$ terms ($s_0 \rightarrow s_N$) of the queue length probability distribution, the first process is to represent the arrival of cells in the slot: $s'_n = \sum_{i=0}^{N} s_i \, pc_{n-i}$.

To avoid the problem of rounding errors the correction process described in section 5 may be applied at this point.

The next process is to represent the serving of the queue. That is to say $s''_0 = s'_0 + s'_1$; for $n = 1 \rightarrow N-1$ $s''_n = s'_{n+1}$; and using the result in section 4, $s''_N = s'^2_N / s'_{N-1}$. This new vector S'' represents the new queue length probability distribution, S. Derivation of mean buffer fill and the equivalent bandwidth may then proceed exactly as described in section 6. A new cell position probability distribution and equivalent bandwidth may also be calculated. If c_m was the relevant cell probability in slot m, with a total number of slots M ($0 \rightarrow M-1$) then the new probability distribution is found:

$$c'_m = \sum_{j=0}^{N} s_j c_{(m-j)^*}$$

The * attached to the (m-j) term is to indicate that if the term becomes negative then M should be added to it to take into account the wrapping effect of the repetitive pattern. If s_j is used then it assumes the pattern cell is last in the queue; s'_j can be used if the first in the queue result is required. If the random case is required then a special version of s must be calculated after Þ is calculated as discussed in section 8.

Remembering that s is different for each m in practical calculations it is simplest to follow the practice of Figure 11, accumulating the new cell position probabilities for every value of m.

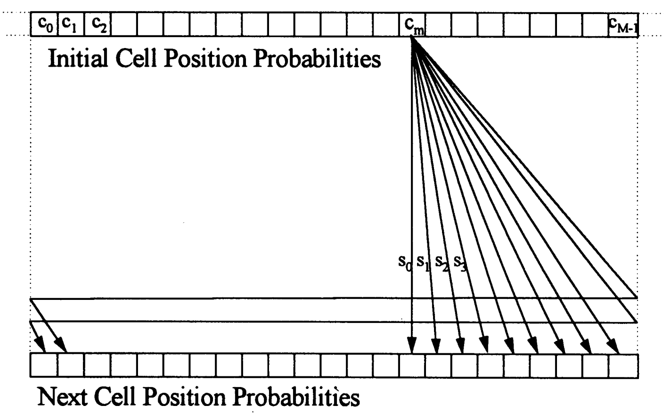

Figure 11 Queuing with Distributed Cell Probabilities

All results quoted here are based on the 'last in the queue' result. However the 'first in the queue' results are similar except that passage through a queue has less effect due to the smaller position probability spreading. By repeatedly applying the whole process the cell distribution may be calculated as the pattern of cells passes through a series of queues as shown in Figure 12. From simulation, and also intuitively, it is known that the cells will be distributed about their initial starting position.

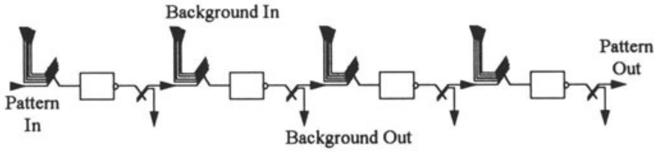

Figure 12 Multiple Queuing Stages

One interesting result is that if the cells start fairly clumped initially their equivalent bandwidth will not decrease until they have been through many queues. In fact taking the MAP mentioned several times already the effect of passing through the queue is initially to increase the equivalent bandwidth. Several adjacent slots with fractional probabilities of being occupied by a cell have greater bandwidth than an occupied cell surrounded by empty slots during which extra cells may be served. The initial effect of the series of queues is to fill in the short gaps between the short bursts and this increases the equivalent bandwidth With more passages through queues the now smoothed burst of bursts begins to spread giving a reduction in equivalent bandwidth. This qualitative description is quantified in figure 13. The equivalent bandwidth has

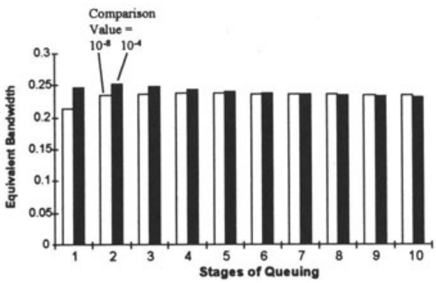

Figure 13 Effect of Several Switching Stages

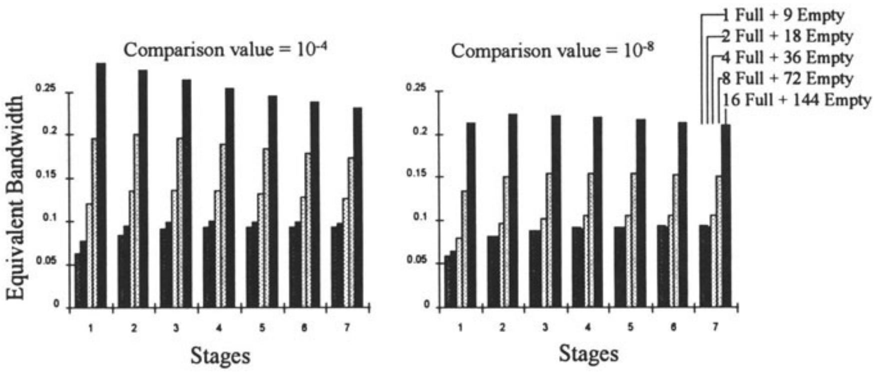

Figure 14 Effect of Bursts Passing Through Several Switching Stages

been calculated against a Poisson background of 0.5 and comparison value of 10^{-4} and 10^{-8}. It is interesting that in this case there is not much difference between the two comparison values; presumably the pattern is "random in nature" and hence does not distort the probability distribution due to the background. Figure 14 also shows the effect of the passage of a bursts of cells, passing through the series of queues. In all cases the mean rate is 0.1 but the cells are in bursts of 1, 2, 4, 8 and 16. The calculation of the equivalent bandwidth has been done at two comparison values - 10^{-4} and 10^{-8}. It can be seen that the fall off in the equivalent bandwidth with passage through several queues is less marked at the lower probability level and that the difference between the two comparison values is much more marked.

11. CONCLUSIONS

It is relatively easy to derive the equivalent bandwidth of a stream consisting of an arbitrary sequence of full and empty cells using the technique of animation described above. This can be extended to the impact of transit through several switching stages. The figure for equivalent bandwidth is not an absolute figure but gives a guide to the impact of a data stream. The value of the equivalent bandwidth, and hence the impact, of the stream is fairly independent of the background parameters and the passage of the stream through the network. Computational requirements are well within the resources of a Personal Computer, with computation times of seconds or minutes.

REFERENCE

[1] J M Griffiths, L G Cuthbert "Very Selective ATM Statistical Filter" Electronics Letters, vol 32, No 7 28 March 96

ACKNOWLEDGEMENT

This work is funded by EPSRC Contract GR/J41635

BIOGRAPHY

JOHN GRIFFITHS graduated from Manchester University in 1966 with a degree in Electrical Engineering. He spent 27 years at BT (formerly the PO) laboratories working on early PCM systems, local network digitalisation, ISDN and, latterly, leading a Division including Submarine Systems, BISDN and LAN interconnection. He has published 2 books: "Local Telecommunications" and "ISDN Explained". He is a Fellow of the IEE, and is a Research Fellow at Queen Mary and Westfield College of the University of London

JONATHAN PITTS graduated from Queen Mary College, University of London, in 1987 with an M.Eng degree in Communications Engineering. Since then he has worked in the Queen Mary College Department of Electronic Engineering on several European RACE projects and, more recently, on UK EPSRC projects concerned with ATM traffic engineering and resource management issues. He completed his Ph.D. on cell-rate accelerated simulation for ATM in 1993, and became a lecturer in the department in 1994.