Matrix Geometric Analysis of Discrete Time Queues with Batch Arrivals and Batch Departures with Applications to B-ISDN [1]

K. Wuyts†, B. Van Houdt‡, R. K. Boel†, and C. Blondia‡

† *University of Ghent*
Vakgroep Elektrische Energietechniek
Technologiepark-Zwijnaarde, 9
B-9052 Ghent
{Kristiaan.Wuyts,Rene.Boel}@rug.ac.be

‡ *University of Antwerp*
Dept. Math. and Computer Science
Universiteitsplein, 1
B-2610 Antwerp - Belgium
{vanhoudt,blondia}@uia.ua.ac.be

Abstract

This paper presents some extensions to the matrix geometric analysis method for performance evaluation of communication systems modeled via discrete time queues. The paper considers the case where the arrival streams generate periodic traffic with a period which varies according to an underlying modulating Markov process. The proposed algorithms evaluate the rate matrices which appear in the definition of the equilibrium distribution. This distribution is evaluated only at times which are multiples of the smallest common multiple of all the periods. This is sufficient for the evaluation of CLR, delay characteristics, and many other performance measures. Evaluating the equilibrium distribution only at this reduced set of times provides a significant reduction in the computational complexity of the performance evaluation for servers with periodic arrival streams, by reducing the required size of the state space of the modulating Markov processes. The method is applicable to ATM systems with periodicity's caused by the higher layer protocols, by traffic shaping, by ABR or UBR control mechanisms, etc.

Keywords : ATM, Traffic Management, Matrix Geometric Analysis, Discrete Time Queueing System

1 Introduction

Performance analysis of ATM systems at the cell level is generally considered computationally intractable, because of the burstiness of the arrival streams combined with the extremely small cell loss ratios to be evaluated. A large body of the literature uses approximate analysis, calculating only the asymptotic decay rate of the equilibrium distribution of the buffer occupation. This is

[1]This work was supported in part by the Commission of the European Union, under project ACTS AC094 "EXPERT" and by Vlaams Actieprogramma Informatietechnologie under project ITA/950214/INTEC "Design and Control of Broadband Networks for Multimedia Applications".

also the asymptotic decay rate of the cell loss ratio as a function of the buffer size. This has led to interesting applications to call acceptance control, estimation of effective bandwidth, etc. These applications are based on the assumption that the coefficient in front of the asymptotic expansion is close to 1. However, it has been shown [5] that in many interesting cases of practical interest this coefficient can be orders of magnitude above or below 1, leading to severe overestimation or underestimation of the cell loss ratio. This in turn leads to large errors in the call admissions policy, which can compromise the economic viability or the reliability of the network operations. Matrix geometric analysis -using the regular block structure of the transition matrices of the Markov process- has been proposed [2] as a possible tool for calculating the equilibrium distribution of the buffer occupation. A particular, but interesting example of this approach is used in a model consisting of a superposition of Markov modulated Bernoulli processes [9, 1] entering an infinite buffer serviced by a constant time server. In [11] this model has been extended to the finite buffer case. This method gives accurate results, provided one can solve the iterative equations for the rate matrix in the matrix geometric method, together with the set of linear equations representing the boundary conditions at the empty and full buffer boundaries, where the regular block structure of the transition matrices is modified. The severe limitation of this method is the growth in the size of the state of the modulating Markov process. This dimension is the product of the sizes of the state spaces of each of the individual arrival streams. The complexity of the problem inevitably grows exponentially with the number of arrival streams under consideration. To alleviate this problem [12, 10] proposed a spectral decomposition method which has been shown very effective in reducing the complexity of the calculation of the rate matrix. The rate matrix is constructed via Kronecker products of blocks of smaller matrices. The calculation of each of the components of these Kronecker products can be decomposed in simpler calculations dealing with one block, referring to one arrival stream, at a time. Once per iteration, a single algebraic equation has to be solved which expresses the interaction between the different arrival streams.

Moreover there are arrival streams which in themselves have a very large modulating state space, such as for example ON-OFF traffic with a periodic cell stream during an ON-period. Such periodic streams occur very often as a result of the higher layer protocol(TCP/IP, video, ...) generating the traffic to be carried by ATM. Shapers also introduce periodicity into the cell streams arriving at an ATM buffer. Using a Markov modulated Bernoulli process as model for these streams with periodicity inevitably leads to a state space of the order of the period of the arrival stream, since the system state has to remember in what phase of the period it is. These large state spaces (for large periods) can be avoided by considering the number of arriving cells per time interval of length one period of the arrival stream.

Take as an example an ON-OFF source, with one arrival every T_1 slots during an ON-period. A cell based Markov modulated model has a state space of size $T_1 + 1$, with a cell being generated in state 1, no cell in all the other states. The modulating Markov process has a very special structure since the T ON-states simply occur in a fixed order, with occasional transitions between state T and the OFF-state. Unfortunately this special structure does not automatically reduce the complexity of the calculation of the rate matrix. The Markov modulated model becomes much simpler if one only tries to calculate the number of cells generated per frame of size T_1 slots. Assume that the duration of ON and OFF periods are independent, each of the form $K.T_1$, with K a geometrically distributed random variable (with parameter p_{ON} for ON-periods, p_{OFF} for OFF-periods). If one counts the number of arrivals during successive intervals of length T_1, then the Markov modulated model is a simple two state model, with 1 arrival per period in state ON, and 0 arrivals in the state OFF. If the length of the ON and OFF periods were of the form $K.T$ with T a multiple of T_1, then a similar model can be built with T/T_1 arrivals per ON-period. Of course in practice the length of ON- and OFF-periods will not be nicely distributed as geometric multiples of the periods T of the arrival stream. However in those cases where the average lengths of the ON- and especially of the OFF-period are large compared to the period T, the above model may represent a very good approximation, especially since it is intuitively clear that short ON- and OFF-periods will not have an important contribution to cell losses.

Based on the above considerations this paper proposes a contribution to the calculation of the rate matrices which appear in the equilibrium distribution as evaluated at times which are multiples of the frame length T_1. The proposed method reduces significantly the computational complexity of the calculation of these rate matrices, allowing the matrix geometric evaluation of performance measures for larger, more realistic models with periodic arrival streams of the type described above. After presenting these models, we show that they lead to a computationally tractable matrix geometric method for performance evaluation for ATM buffers with several periodic sources. Rather than studying the buffer occupation at the end of each slot, we only try to calculate the distribution of the buffer occupation Q_t just prior to the end of a fixed length frame. Again we obtain a transition matrix with a block structure. But now, because of the frame structure, we have up to U arrivals and up to L departures per frame, there are U blocks above the diagonal, and L blocks below the diagonal. Again, except at the empty and full buffer boundaries, the blocks at the same distance from the diagonal are identical. This matrix geometric block structure in the Kolmogorov equations can be exploited as shown later to efficiently obtain the equilibrium distribution at times which are a multiple of the frame length $T1$.

This paper illustrates the proposed method through its application in some examples of calculation of cell loss ratios (CLR) for systems which combine a few periodic ON-OFF traffic sources with many Markov modulated Bernoulli background sources. As a particular example, we will study how the cell loss ratio depends on the number of cells per frame which a multiplexer removes from a buffer for a particular priority class (e.g. in flow control for ABR traffic this number could be a slowly varying flow control variable). This scheme can also be used to evaluate the ER congestion control mechanism used to control the behaviour of the ABR traffic streams, where this computation scheme lead to some useful reductions in the computation time. The results on this application can be found in [3, 4]. Finally the method can also be applied - provided the boundary equations are solved efficiently - to many interesting flow control strategies with thresholds, such a packet discard strategies in UBR (or, to provide a totally different example, for evaluating delays of cars in front of traffic lights in an urban traffic model).

The structure of the paper is as follows. Sections 2 and 3 describe the model class for which our approach is applicable. In sections 4 and 5 we describe the proposed analysis tool for the infinite and the finite buffer case. An easy and computationally efficient implementation of the method is presented so that it can be applied easily by all users. Finally in section 6 some numerical examples are given which illustrate our method. The probabilistic interpretations which justify the proposed method are discussed in the appendix.

2 System Description and Model

Consider a buffer in an ATM system, with N arrival streams (or input sources). The arrival process for each source is modeled as follows. Source $1 \leq n \leq N$ can be in either one of M_n different states, denoted as $z_{m,n}, m = 1, \ldots, M_n$. We define the following variables

- $Z_{t,n}$ is the state of the n-th source during the t-th frame
- $K_{k,n}$ is the number of cells generated by the n-th source during an interval of length $T_{k,n}$ while this source is in state $k \in \{1, \ldots, M_n\}$.

Note that we only require that the cells are generated in a periodic fashion during the frame, but that we do not use the exact arrival times of the cells in the frame. This is sufficient for calculating

the rate matrices, since the detailed structure of the arrival times in a frame only become relevant when the buffer is almost empty or almost full, i.e. when evaluating the boundary conditions of the Kolmogorov equations. For calculating the rate matrices one only needs the values of $T_{k,n}$ and $K_{k,n}$. Note that the proposed solution is also valid if $K_{k,n}$ is a random number with a distribution which only depends on the state k the source is in during the t-th frame. We now define the frame length T such that T is a multiple of all periods $T_{k,n}$ involved in the description of all traffic streams. Notice that when a source n is in state k during a frame, then it will generate $K_{k,n}.T/T_{k,n}$ cells during this frame. Classical Markov modulated Bernoulli sources are included in this model by taking $T_{k,n} = 1$ for some states k, with the corresponding $K_{k,n}$ a Bernoulli random number with parameter $p_{k,n}$, taking only the values 0 and 1.
As mentioned before for ON-OFF processes with periodic traffic during ON-periods, the assumption that sources remain in the same state for a length of time which is a multiple of the frame length T represents an approximation to realistic source models, which will lead to errors in the calculated cell loss ratios which one expects to be small on intuitive grounds. Moreover the discussion on computational efficiency in section 4 indicates that one should not consider too large a period. Together these observations indicate that it will often be necessary to make a compromise by choosing a value T for the frame size which is not a multiple of all the periods appearing in all the traffic streams. Rather one should approximate periods that are too long by assigning a distribution on the number of cells generated during a fraction of a period. Consider as an example the case where a frame size T is chosen, and there is a source of low intensity which generates cells periodically every $2.T$ slots. Then one can approximate this behaviour by assigning the number of arrivals 0 and 1 during a frame of size T the probabilities 0.5 each. It is easy to see how this can be generalized for more complicated periodicities. In the same way one can approximate the fact that durations of ON- and OFF-periods are not exact multiples of the frame length T by including the probability of a few cells more or less arriving in a frame for each modulating state. Of course one could also model this at the price of a small increase in size of the state space (small in comparison with the size of the state space which would be needed for a slot-based model), by introducing additional transition states expressing the possibility of two different modulating states occurring in the same frame.

For the server we assume the following. During the t-th frame (of length T slots) a random number $S_t \leq K_s$ cells will be transmitted by the server connected to the buffer. If all states are periodic, source n can generate at most $U_n = max_k K_{k,n}.T/T_{k,n}$ cells per frame of length T. At most $U = \sum_n U_n$ cells can arrive per frame of length T slots, while at most $L = K_s - \sum_n U_{min}(n)$ cells will leave the buffer during the same interval, where the minimum number of cells $U_{min}(n)$ which source n generates per frame may be strictly positive for some periodic sources. Clearly the above modeling assumptions specify completely the transition matrix of the discrete time Markov process $(Q_{k.T}, Z_{k,1}, \ldots, Z_{k,N})$. Here Q_t represents the buffer length at time/slot t.

3 The Model

In this section some extensions of the matrix geometric algorithm, presented in [11], are introduced. Our aim is to find the equilibrium distribution of the Markov process represented by the following transition matrix in the infinite buffer case:

$$P^{(\infty)} = \begin{pmatrix} \sum_{i=-L}^{0} C_i & C_1 & \cdots & C_U & 0 & \cdots & 0 & 0 & \cdots \\ \sum_{i=-L}^{-1} C_i & C_0 & \cdots & C_{U-1} & C_U & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ C_{-L} & C_{-L+1} & \cdots & C_{-L+U} & C_{-L+U+1} & \cdots & C_U & 0 & \cdots \\ 0 & C_{-L} & \cdots & C_{-L+U-1} & C_{-L+U} & \cdots & C_{U-1} & C_U & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{pmatrix} \tag{1}$$

The submatrices C_i have dimension dC. In the finite buffer case the transition matrix is denoted by $P^{(B)}$ with dimension $(B+1)dC$. The first B columns of submatrices of $P^{(B)}$ equal the first B columns of $P^{(\infty)}$ and its last column is given by $[0 \ldots 0\ C_U\ \sum_{i=U-1}^{U} C_i \ldots \sum_{i=0}^{U} C_i]^T$.

4 The infinite buffer system

By grouping blocks of $U \times U$ matrices C_i if $U > L$, or blocks of $L \times L$ matrices C_i if $L > U$ a Quasi-Birth-and-Death (QBD) process is obtained, which can be solved by a classical algorithm e.g. folding algorithm ([13]), the logarithmic reduction algorithm ([7]) and many others ([6, 2]). Many of these algorithms use a well known rate matrix R (or the related G matrix). For the specific model described in the preceding section(s) the computational complexity of the rate matrix can be reduced considerably by using the internal structure of $P^{(\infty)}$. We consider the system for $U \geq L$, without loss of generality. When $U \geq L$ then the rate matrix R, corresponding to the QBD, is also a block of $U \times U$ matrices R_{ij} $(1 \leq i, j \leq U)$ with dimension dC.

In [8] it is shown that in the case of a positive recurrent chain the matrices R_{ij} are finite and have a useful probabilistic interpretation:

- for any integer number n, $(R_{ij})_{kl}$ is the expected number of visits to the state $(n+U-i+j, l)$ until the first return to the level $n+U-i$ or below, under the condition that the process starts in state (n, k).

Let us now define $R_i = R_{U+1-i,1}$. Thus for any integer number n, $(R_i)_{kl}$ gives the expected number of visits to the state $(n+U, l)$ until the first return to state $n+U-1$ or below, under the condition that we started in state $(n+U-i, k)$. Using the probabilistic interpretation, we can easily prove that R can be written as:

$$R = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & R_U \\ I & 0 & 0 & \cdots & 0 & R_{U-1} \\ 0 & I & 0 & \cdots & 0 & R_{U-2} \\ 0 & 0 & I & \cdots & 0 & R_{U-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & I & R_1 \end{pmatrix}^U \tag{2}$$

If $\pi = (\pi_n)_{n \geq 0}$ is the steady state vector of $P^{(\infty)}$ (with the dimension dC of C_i equal to the one of π_n), it follows from the substitution of (2) in the matrix geometric method of [8] that the steady state vector must satisfy:

$$\pi_n = \pi_{n-1} R_1 + \ldots + \pi_{n-U} R_U, \forall n \geq U. \tag{3}$$

Substituting (2) in the equation which computes R in [8] proves that the rate matrices R_i must satisfy:

$$R_i = C_i + R_i C_0 + \sum_{l=1}^{L} \sum_{k=1}^{l+1} \left(\sum_{\substack{i_1+\ldots+i_k=l+i \\ i \le i_1 \le min(U,L+i)\ ,\ i_1,\ldots,i_k \ge 1}} \prod_{j=1}^{k} R_{i_j} \right) C_{-l}\ ,\ 1 \le i \le U. \tag{4}$$

One way of solving this set of matrix equations is to use an iterative scheme, starting with the values of the rate matrices R_i set to zero. To obtain the new values $R_1^{(n+1)}, \ldots, R_U^{(n+1)}$, we use the method below:

$$R_i^{(n+1)} = C_i + R_i^{(n)} C_0 + \sum_{l=1}^{L} \sum_{k=1}^{l+1} \left(\sum_{\substack{i_1+\ldots+i_k=l+i \\ i \le i_1 \le min(U,L+i)\ ,\ i_1,\ldots,i_k \ge 1}} \prod_{j=1}^{k} R_{i_j}^{(n)} \right) C_{-l}\ ,\ 1 \le i \le U. \tag{5}$$

The validity of this iterative scheme is proven in a probabilistic manner in proposition 2 of Appendix A. In the following subsection we describe an efficient implementation of this iterative scheme and compare its complexity with that of some other, well known algorithms for QBD which do not use the internal structure of P in the same way as we do.

4.1 Implementation and Complexity of the Repetitive Scheme

In this section we present a fast method for obtaining the matrices $R_i^{(n+1)}$ having found $R_i^{(n)}$, the idea is the following. We start by creating a set of matrices X_i as follows

$$\begin{array}{l} X_1 = R_1^{(n)} \\ X_i = R_i^{(n)} + \sum_{j=1}^{i-1} R_{i-j}^{(n)} X_j \quad 2 \le i \le L \end{array} \tag{6}$$

This step takes about $(\mathrm{d}C)^3 * L^2$ flops. Next we define the matrices Y_i^U as

$$Y_i^U = R_U^{(n)} * X_i \tag{7}$$

with $1 \le i \le L$ and the number of flops is negligible compared with the previous step. Having done this we find the matrix $R_U^{(n+1)}$ as follows

$$R_U^{(n+1)} = \sum_{i=1}^{L} Y_i^U * C_{-i} + C_U + R_U^{(n)} C_0 \tag{8}$$

keeping the matrices X_i as before, we now show how to obtain $R_{i-1}^{(n+1)}$ having found $R_i^{(n+1)}$

$$(Y_1^{i-1}, Y_2^{i-1}, \ldots, Y_L^{i-1}) = (R_i^{(n)}, Y_1^i, \ldots, Y_{L-1}^i) + (R_{i-1}^{(n)} * X_1, \ldots, R_{i-1}^{(n)} * X_L) \tag{9}$$

and we find $R_{i-1}^{(n+1)}$ as

$$R_{i-1}^{(n+1)} = \sum_{j=1}^{L} Y_j^{i-1} * C_{-j} + C_{i-1} + R_{i-1}^{(n)} C_0 \tag{10}$$

which indeed results in a scheme of order $(\mathrm{d}C)^3 UL$ times the number of iterations. Comparing this with the complexity of some well-known general purpose QBD algorithms we find that the extra structure in P allows us to reduce the computational complexity as follows

1. the logarithmic reduction scheme of Latouche and Ramaswami has a complexity of $\frac{25}{3} * (\mathrm{d}C)^3 * U^3 * I_1$ with I_1 the number of iterations for obtaining the matrix R,
2. the U-algorithm uses $\frac{7}{3} * (\mathrm{d}C)^3 * U^3 * I_2$ flops where I_2 is the number of iterations and $I_1 = \log(I_2)$,
3. the new algorithm computes R in $3 * (\mathrm{d}C)^3 * U * L * I_3$ flops with again I_3 the number of iterations,

clearly both schemes are outperformed by the new one in case the traffic has no bursty character i.e. the number of iterations is small. For more bursty traffic the performance will depend on the number of iterations I_3 which can be shown smaller than or equal to I_2 (we get an equality if $U = L = 1$) but is expected to be (much) larger than I_1. Thus in case of bursty traffic the choice between the different schemes depends on the exact values of U and L.

Perhaps it's good to explain why I_2 and I_3 are different from each other although they are based on the same formula. When the U-algorithm is used all the entries of the matrix R (of dimension $\mathrm{d}C \times U$) are calculated using the classical formula (see [7]). In our case we compute the first $\mathrm{d}C$ rows using this formula and then calculate the rest as a function of the first row. As a consequence we get a different result for the other components of $R^{(n)}$. Let us demonstrate this by means of an example.

Example Suppose that $L = U = 2$. Then $R_c^{(0)}$ and $R_n^{(0)}$ are both equal to zero, where the subscripts refer to the classic and the new scheme. Using the formula of [7]

$$R = A_0 + RA_1 + R^2 A_2. \tag{11}$$

Where A_0, A_1 and A_2 are the matrices corresponding to the QBD process that is found by grouping the matrices C_i. We find that

$$R_c^{(1)} = \begin{pmatrix} C_2 & 0 \\ C_1 & C_2 \end{pmatrix}. \tag{12}$$

If we now look at $R_n^{(1)}$ we find the same first column but a different second one

$$R_n^{(1)} = \begin{pmatrix} 0 & C_2 \\ I & C_1 \end{pmatrix}^2 = \begin{pmatrix} C_2 & C_2 C_1 \\ C_1 & C_2 + (C_1)^2 \end{pmatrix}. \tag{13}$$

And thus as $R_c^{(1)}$ and $R_n^{(1)}$ are used to find $R_c^{(2)}$ and $R_n^{(2)}$ we get different results after n steps and thus a different rate of convergence. This is why a new probabilistic proof was necessary.

When looking at our application, the computation of the efficiency above does not yet take into account the advantage of computing the equilibrium distribution for periodic sources on frame level instead of on slot level.

Example Let us look at the complexity of the computation of matrix R for 2 on/off-sources. If source i is "on", this means that the source behaves as a periodic source, sending 1 cell per T_i slots. Source i being "off" means that the source sends no cells. Let us consider a "periodic" server, taking 1 cell from the buffer per T_s slots.

For performance analysis on slot level, source i has thus T_i states describing its on behaviour and 1 state for the off behaviour, resulting in a total of T_i+1 states. The server has T_s states. Since we have to take into account in which state every source and the server remain, the dimension of de matrices C_i equals $dC = (T_1+1)(T_2+1)T_s$. L is the maximum number the buffer occupancy can decrease (per slot): this happens if both source 1 and 2 are in the off-state (no arrivals) and if the server takes 1 cell from the buffer, so $L = 1$. U is the maximum number the buffer occupancy can increase (per slot): this happens if source 1 and 2 transmit a cell during the same slot and if the server takes no cells from the buffer, so $U = 2$. As shown above, the complexity for computing R equals $3 * (\mathrm{d}C)^3 * U * L * I_3$, which gives after substitution $6 * (T_1+1)^3 * (T_2+1)^3 * T_s^3 * I_s$, where I_s denotes the number of iterations on slot level.

For performance analysis on frame level, let us consider the worst case where the least common multiple of T_1, T_2 and T_s is their product. Thus the frame must be $T_1T_2T_s$ slots long in order to contain an integer number of the periods T_1, T_2 and T_s. The server is now described by 1 state and it takes T_1T_2 cells from the buffer per frame. Source i is described by 2 states: an on-state and an off-state. If source 1 remains in the on-state it transmits T_sT_2 cells per frame and if it remains in the off-state it transmits no cells at all (per frame). If source 2 remains in the on-state it transmits T_sT_1 cells per frame and if it remains in the off-state it transmits no cells at all (per frame). Again the dimension of the matrices C_i is given by the product of the dimensions of the transition matrices of the sources and the server, which equal respectively 2, 2 and 1, so $dC = 4$. For the computation of L, we have to consider the case where both sources are in the off-state and since the server always removes T_1T_2 cells from the buffer $L = T_1T_2$. The buffer can only increase a maximum number of cells U (per frame) if both source are in the on-state (arrival of $T_sT_1 + T_sT_2$ cells) and since the server always takes T_1T_2 cells from the buffer $U = T_sT_1 + T_sT_2 - T_1T_2$. After substitution in the complexity expression, we find that the complexity is now given by $192 * T_1 * T_2 * (T_sT_1 + T_sT_2 - T_1T_2) * I_f$, where I_f denotes the number of iterations on frame level.

If one compares the exponents of T_1, T_2 and T_s in the complexity expressions for the slot level case and for the frame level case, one can see that even in this worst case scenario performance analysis on frame level is more efficient than on slot level. Even in a small example such as $T_1 = 3$, $T_2 = 4$ and $T_s = 5$ the number of flops becomes $52992 I_f$ for frames, while it takes 6000000 I_s flops for a slot level calculation.

5 The finite buffer system

Let us now consider a buffer with a finite capacity , say B, and denote the stationary distribution of P as $\pi = [\pi_0 \dots \pi_B]$. Notice that the last column of P now consists of sums, in such a way that P is stochastic. Without loss of generality we assume that $U \geq L$. Throughout this section we'll follow the lines of reasoning maintained in [11]. Let us assume that the steady state vector π can be written as:

$$\pi_n = \alpha_n + \beta_n, 0 \leq n \leq B$$

in such a way that

$$\begin{aligned} \alpha_n = &\quad \alpha_{n-1}R_1 + \alpha_{n-2}R_2 + \dots + \alpha_{n-U}R_U \quad n \geq U, & (14)\\ \beta_n = &\quad \beta_{n+1}S_1 + \beta_{n+2}S_2 + \dots + \beta_{n+L}S_L \quad n \leq B - L. & (15) \end{aligned}$$

The dimensions of the vectors α_n, β_n and π_n are all equal to $\mathrm{d}C$, the dimension of the matrices C_i, which is also the order of R_i and S_i. By definition we know that $(\pi_n)_n$ obeys the steady-state equations. And above we assumed that $\pi_n = \alpha_n + \beta_n$. Thus the steady state equations hold if they both hold for $(\alpha_n)_n$ and $(\beta_n)_n$. Thus if

$$\begin{aligned} \alpha_n &= \alpha_{n+L}C_{-L} + \dots + \alpha_nC_0 + \dots + \alpha_{n-U}C_U \quad U \leq n \leq B - L, & (16)\\ \beta_n &= \beta_{n+L}C_{-L} + \dots + \beta_nC_0 + \dots + \beta_{n-U}C_U \quad U \leq n \leq B - L. & (17) \end{aligned}$$

This allows us to obtain a set of non-linear equations for the matrices R_i and S_i by substituting (14) resp. (15) repeatedly in equation (16) resp. (17) until the only α_{n-u}'s and β_{n+l}'s remaining are those with $1 \leq u \leq U$ and $1 \leq l \leq L$. This results in a set of $U + L$ non-linear equations by matching the coefficients of these α's and β's. We have the following expression for S_i

$$S_i = C_{-i} + S_i C_0 + \sum_{l=1}^{U} \sum_{k=1}^{l+1} \left(\sum_{\substack{i_1+\ldots+i_k=l+i \\ i \leq i_1 \leq L \ , \ i_1,\ldots,i_k \geq 1}} \prod_{j=1}^{k} S_{i_j} \right) C_l \ , \ 1 \leq i \leq U. \tag{18}$$

and the condition on R_i is the same as in (4). We have already shown that there exists a solution for (4) and because of symmetry reasons we are able to solve (18) in a similar manner.

Thus once we know the values of $\alpha_0, \ldots, \alpha_{U-1}$ and $\beta_{B-L+1}, \ldots, \beta_B$, we can compute the equilibrium distribution using (14) and (15). As in [11], the remaining steady state equations yield a homogeneous set of linear equations but this time for $\alpha_0, \ldots, \alpha_{U-1}$ and $\beta_{B-L+1}, \ldots, \beta_B$:

$$[\ \alpha_0 \quad \ldots \quad \alpha_{U-1} \quad \beta_{B-L+1} \quad \ldots \quad \beta_B\ ] \begin{pmatrix} T & V \\ U & W \end{pmatrix} = 0. \tag{19}$$

We will describe the structure of the matrices T and U in Appendix B, the other two are analogue.

6 Numerical results

Let us consider a single finite buffer in an ATM buffer system. Several sources (say N) are sending ATM cells to this buffer and a single server takes cells from the buffer if the buffer is not empty at a rate which can be less than or equal to 1 cell/slot. Although the method described in the previous section can be used to find the equilibrium distribution for sources described by general stochastic processes and for any server behaviour (as long as the transition matrix of the resulting Markov process for the buffer occupancy can be written as $P^{(B)}$), the method is especially interesting for "periodic" on/off sources: i.e. source i transmits K_i cells per frame of T_f slots if that source remains in the on-state; if source i remains in the off-state, the source sends no cell during a frame. Only at the end of a frame, source i can change states according to transition matrix Q_i:

$$Q_i = \begin{pmatrix} p_{on}(i) & 1 - p_{on}(i) \\ 1 - p_{off}(i) & p_{off}(i) \end{pmatrix} \tag{20}$$

We also assume that the server is a deterministic server, taking K_s cells from the buffer per frame. The transition matrix of the process describing the buffer occupancy then has indeed the same structure as $P^{(B)}$, with $L = K_s$ and $U = \sum_{i=1}^{N} K_i - K_s$.

Figures 1 and 2 show results for $N = 5$ identical sources, the frame has length $T_f = 10$ slots, $K_i = 5$, $p_{on} = 0.91$ and $p_{off} = 0.99$. The average load is $\rho = 2.5/K_s$. Figure 1 shows the curves for the CLR: decreasing the server rate means actually increasing the average load because of a decrease in available bandwidth in the network. This model describes how ABR traffic is influenced by flow control when the load of other traffic (with higher priority) increases in the network such that less bandwidth is left for the ABR traffic. Figure 2 shows the probability that there are b cells in the buffer for a buffer with size $B = 128$. Notice that the oscillations are caused by the periodicity of the sources. An extreme case is $K_s = 10$, since the number of cells arriving at or leaving the buffer is always a multiple of 5. Starting from an empty or full buffer, there can only be 0, 5, 10, ... or 128, 123, 118, ... cells in the buffer, i.e. $P(1) = 0$. Starting from a buffer which is not empty and not full, after a finite time the full buffer or the empty buffer state is reached and thus the equilibrium distribution will still have only non-zero probabilities at 0, 3, 5, 8, ... The effect of the oscillations can be decreased by adding some Bernoulli traffic in the background, although it will not disappear entirely and adding Bernoulli traffic means that more rate matrices have to be computed, because U will increase by T_f.

7 Conclusions

This paper has presented an efficient algorithm for calculating the rate matrices of a matrix geometric representation of the equilibrium distribution of the Markov process of a multiplexer with periodic sources. The reduction in computational complexity is achieved by using the special structure, resulting from the periodicity of the arrival streams, of the transition matrix of the Markov processes. This mehod can make the matrix geometric method attractive for performance evaluation for more realisitic models of ATM multiplexers with many periodic arrival streams. It can thus help in dimensioning of buffers and in optimising parameter values for systems involving shapers and flow controlled ABR and UBR traffic.

A Appendix A

Proposition 1:

If we define the matrices $(X_i^{(n)})_{1 \leq i \leq U, n \geq 0}$ as:

$$X_i^{(0)} = 0 \tag{21}$$

$$X_i^{(n+1)} = C_i + X_i^{(n)} C_0 + \sum_{l=1}^{L} \sum_{k=1}^{l+1} \left(\sum_{\substack{i_1+\ldots+i_k=l+i \\ i \leq i_1 \leq min(U,L+i)\ ,\ i_1,\ldots,i_k \geq 1}} \prod_{j=1}^{k} X_{i_j}^{(n)} \right) C_{-l} \ \ ,1 \leq i \leq U. \tag{22}$$

then they form non-decreasing sequences in n, which converge to R_i.

Proof:

Since $X_i^{(0)} = 0$ and $X_i^{(1)} = C_i$ it is obvious that $X_i^{(0)} \leq X_i^{(1)}$. Through induction it is easy to show that then $X_i^{(n)} \leq X_i^{(n+1)}$. Since $R_i \geq X_i^{(0)} = 0$. It follows again through induction that $X_i^{(n)} \leq R_i$ for any n. This proves that

$$X_i^* = \lim_{n \to \infty} X_i^{(n)} \leq R_i \tag{23}$$

It now remains to show that $R_i \leq X_i^*$. Therefore we define the taboo probability ${}_mP_{(i,j)(k,l)}^{(n)}$ as the probability that, starting in the state (i,j) at time 0, the chain reaches (k,l) at time n without returning to level m (or below) in between. We also define the matrix $R_{U+1-k}(N,\kappa)$ as

$$(R_{U+1-k}(N,\kappa))_{j,\nu} = \sum_{n=1}^{N} {}_{l+U}P_{(l+k,j)(l+U+1+\kappa,\nu)}^{(n)} \tag{24}$$

for $\kappa \geq 0, 1 \leq j \leq dC, 1 \leq \nu \leq dC, 1 \leq k \leq U$. It follows from the probabilistic interpretation of R_i that

$$R_i = \lim_{N \to \infty} R_i(N,0) \tag{25}$$

Based upon their definition, a recursive expression for the taboo probabilities can be constructed:

$$ {}_{l+U}P^{(1)}_{(l+k,j)(l+U+1,\nu)} = (C_{U+1-k})_{j,\nu} \tag{26} $$

and for $n \geq 2$:

$$ {}_{l+U}P^{(n)}_{(l+k,j)(l+U+1,\nu)} = \sum_{h=1}^{dC} \sum_{\kappa=0}^{L} {}_{l+U}P^{(n-1)}_{(l+k,j)(l+U+1+\kappa,h)} (C_{-\kappa})_{h,\nu} \tag{27} $$

Adding the equations (26) and (27) for n ranging from 1 to N and substituting $U+1-k$ by i yields in matrix-notation:

$$ R_i(N,0) = C_i + \sum_{\kappa=0}^{L} R_i(N-1,\kappa) C_{-\kappa} \tag{28} $$

We now derive the an inequality for the matrices $R_i(N,\kappa)$:

$$ \begin{aligned} & (R_{U+1-k}(N,\kappa))_{j,\nu} \\ = & \sum_{n=1}^{N} {}_{l+U}P^{(n)}_{(l+k,j)(l+U+1+\kappa,\nu)} \\ = & \sum_{p=1}^{\kappa+1} \sum_{\substack{0 \leq \kappa_1 < \ldots < \kappa_p = \kappa \\ \kappa_i - \kappa_{i-1} \leq U, \kappa_0 = k-U-1}} \sum_{\substack{h_1,\ldots,h_{p-1}=1 \\ h_0 = j, h_p = \nu}}^{dC} \sum_{n=1}^{N} \sum_{\substack{r_1+\ldots+r_p = n \\ 1 \leq r_i \leq N}} \prod_{j=1}^{p} {}_{l+U+\kappa_j}P^{(r_j)}_{(l+U+1+\kappa_{j-1},h_{j-1})(l+U+1+\kappa_j,h_j)} \\ \leq & \sum_{p=1}^{\kappa+1} \sum_{\substack{0 \leq \kappa_1 < \ldots < \kappa_p = \kappa \\ \kappa_i - \kappa_{i-1} \leq U, \kappa_0 = k-U-1}} \sum_{\substack{h_1,\ldots,h_{p-1}=1 \\ h_0 = j, h_p = \nu}}^{dC} \prod_{j=1}^{p} \sum_{r_j=1}^{N} {}_{l+U+\kappa_j}P^{(r_j)}_{(l+U+1+\kappa_{j-1},h_{j-1})(l+U+1+\kappa_j,h_j)} \\ = & \sum_{p=1}^{\kappa+1} \sum_{\substack{0 \leq \kappa_1 < \ldots < \kappa_p = \kappa \\ \kappa_i - \kappa_{i-1} \leq U, \kappa_0 = k-U-1}} \sum_{\substack{h_1,\ldots,h_{p-1}=1 \\ h_0 = j, h_p = \nu}}^{dC} \prod_{j=1}^{p} \sum_{r_j=1}^{N} {}_{l+U}P^{(r_j)}_{(l+U+1+\kappa_{j-1}-\kappa_j,h_{j-1})(l+U+1,h_j)} \end{aligned} $$

In matrix-notation this is equivalent with:

$$ R_{U+1-k}(N,\kappa) \leq \sum_{p=1}^{\kappa+1} \sum_{\substack{0 \leq \kappa_1 < \ldots < \kappa_p = \kappa \\ \kappa_i - \kappa_{i-1} \leq U, \kappa_0 = k-U-1}} \prod_{j=1}^{p} R_{\kappa_j - \kappa_{j-1}}(N,0) \tag{29} $$

Substitution of (29) in (28) yields:

$$ \begin{aligned} R_i(N,0) \leq\ & C_i + R_i(N-1,0)C_0 \\ & + \sum_{l=1}^{L} \sum_{k=1}^{l+1} \left(\sum_{\substack{i_1+\ldots+i_k = l+i \\ i \leq i_1 \leq min(U,L+i)\ ,\ i_1,\ldots,i_k \geq 1}} \prod_{j=1}^{k} R_{i_j}(N-1,0) \right) C_{-l} \quad , 1 \leq i \leq U. \end{aligned} \tag{30} $$

We now have that $R_i(1,0) = C_i = X_i^{(1)}$, such that $R_i(2,0) \leq X_i^{(2)}$. By induction (30) yields that $R_i(N,0) \leq X_i^{(N)}$ for $N \geq 1$. The sequence of matrices $R_i(N,0)$ is non-decreasing and tends to R_i, such that the preceding inequality implies that $R_i \leq X_i^*$ and therefore $R_i = X_i^*$.

B Appendix B

In this section we show the construction of the matrices T and U which are part of the linear set of equations in (19). An important remark about this appendix is that if we talk about the j-th column (row) of a matrix, we'll actually mean the columns (rows) $(j-1) * \mathrm{d}C + 1$ until $j * \mathrm{d}C$, thus we think in terms of matrices of order $\mathrm{d}C$. Let us start with the matrix T:

$$T = I - S$$

where I is the unity matrix of dimension $\mathrm{d}C$ times U and S has the following form:

$$\begin{array}{lll}
S_{i,1} & = \left(\sum_{k=-L}^{-(i-1)} C_k\right) + 1_{\{L=U\}} R_{U-(i-1)} C_{-U} & U \geq i \geq 1 \\
S_{1,2} & = C_1 + R_U X_1 & \\
S_{i,2} & = C_{-i+2} + R_{U-(i-1)} X_1 + 1_{\{U=L\}} R_{U-(i-2)} C_{-U} & U \geq i \geq 1, i \geq 2 \\
S_{i,j} & = R_{U-(i-1)} X_{j-1} + S_{i-1,j-1} & U \geq i \geq 2, j \geq 3 \\
S_{1,j} & = C_{j-1} + R_U X_{j-1} & j \geq 3
\end{array}$$

where the values X_j for $1 \leq j \leq P-1$ represent the matrices below:

$$X_j = \sum_{m=0}^{j-(U-L)} \left(\left(\sum_{k=1}^{m} \sum_{\substack{i_1+\ldots+i_k=m \\ i_1,\ldots,i_k \geq 1}} \prod_{l=1}^{k} R_{i_l} \right) C_{-U+j-m} \right).$$

As can be seen from the formula above we need a smaller computational effort as the difference between U and L increases. This concludes the construction of T.

Before we can descibe the matrix U we'll introduce two matrices Z and F, this to make the structure of U more transparent:

$$Z = \begin{pmatrix}
S_1 & I & 0 & 0 & \cdots & 0 \\
S_2 & 0 & I & 0 & \cdots & 0 \\
S_3 & 0 & 0 & I & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
S_{L-1} & 0 & 0 & 0 & \cdots & I \\
S_L & 0 & 0 & 0 & \cdots & 0
\end{pmatrix}, F = \begin{pmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix},$$

where I denotes the unity matrix of dimension $\mathrm{d}C$. With this in mind U equals the following definition $U = [U_1 U_2 \ldots U_U]$:

$$\begin{array}{ll}
U_1 = & Z^{B-2L+1} \left[Z^L F - Z^L F \left(\sum_{i=-L}^{0} C_i\right) - Z^{L-1} F \left(\sum_{i=-L}^{1} C_i\right) - \ldots - F C_{-L} \right], \\
U_j = & Z^{B-2L-(j-2)} \left[Z^L F - Z^{L+j-1} F C_{j-1} - Z^{L+j-2} F C_{j-2} - \ldots - F C_{-L} \right], 2 \leq j.
\end{array}$$

The two remaining matrices V and W are very similar because of the symmetric nature of the system.

References

[1] C. Blondia. A discrete-time batch markovian arrival process as b-isdn traffic model. *Belgian Journal of Oper. Res., Statistics and Computer Science 32 (3,4) 3–23*, 1993.

[2] C. Blondia and O. Casals. Statistical multiplexing of vbr sources: A matrix-analytical approach. *Performance Evaluation 16 5 - 20*, 1992.

[3] C. Blondia O. Casals and B. Van Houdt. Buffer analysis of the ER congestion control mechanism for the ABR service category in ATM networks. *Proc. of PICS '98, Lund, May 1998*, 1998.

[4] O. Casals C. Blondia L. Cerda and B. Van Houdt. Congestion control and charging for the ABR service category in ATM networks. *Proc. of SPIE, Performance and Control of Network Systems II, Boston*, 1998.

[5] G. Choudhury, D. Lucantoni, and W. Whitt. Squeezing the most out of ATM. *AT&T Bell Laboratories Report, May*, 1993.

[6] W. Grassmann and D. Heyman. Computation of steady-state probabilities for infinite markov chains with repeating rows. *Bellcore Technical Memorandum*, 1991.

[7] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for quasi-birth-death processes. *Journal of Applied Probability 30 (650-674)*, 1993.

[8] M. F. Neuts. Matrix-geometric solutions in stochastic models. *The Johns Hopkins University Press, Baltimore*, 1981.

[9] M. F. Neuts. Algorithmic probability. *Chapman and Hall, London*, 1995.

[10] K. Wuyts and R. Boel. Efficient matrix-geometric methods for b-isdn by using the spectral decomposition of the rate matrices. *Advances in matrix-Analytic Methods for Stochastic Models, A. Alfa and S. Chakravarthy, eds., pp.341-359, Notable Publications*, 1998.

[11] K. Wuyts and R. K. Boel. A matrix geometric algorithm for finite buffer systems with B-ISDN applications. *Proc. ITC Specialists Seminar on Control in Communications (Lund, Sweden) 265-276*, 1996.

[12] K. Wuyts and R.K. Boel. Efficient performance analysis of ATM buffer systems by using the spectral analysis of rate matrices. *Proc. ATM '96, Ilkley*, 1996.

[13] Jingdong Ye and San qi Li. Folding algorithm, a computational method for finite QDB processes with level-dependent transitions. *IEEE Trans on Comm. 42(2/3/4) 625–639*, 1994.

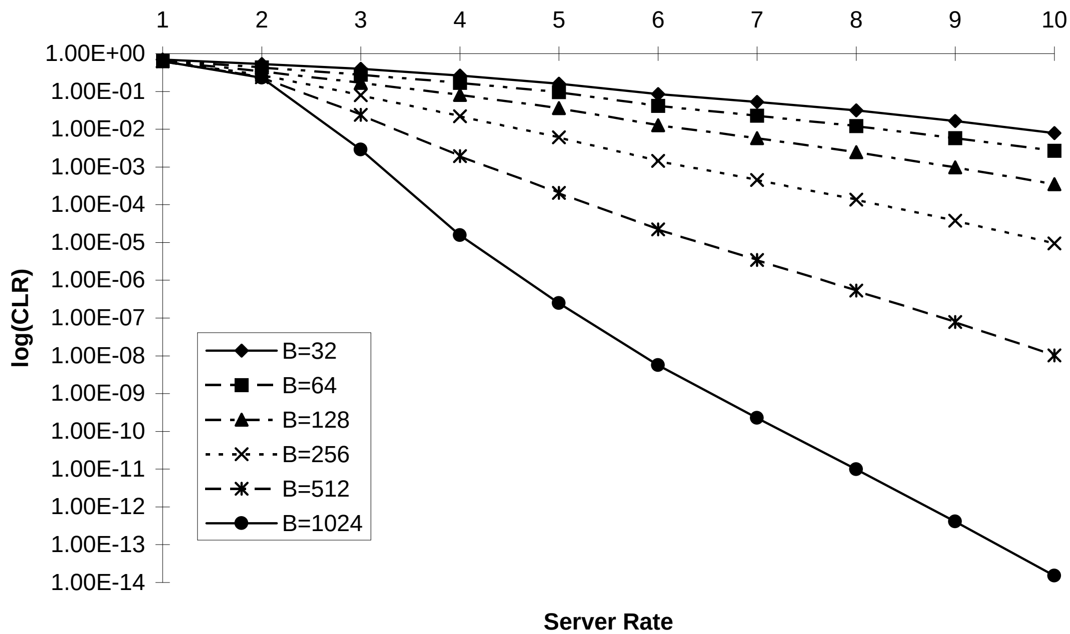

Figure 1: *CLR as a function of the server rate K_s for several buffer sizes; there are 5 identical sources, $T_f = 10$, $K_i = 5$, $p_{on} = 0.91$ and $p_{off} = 0.99$.*

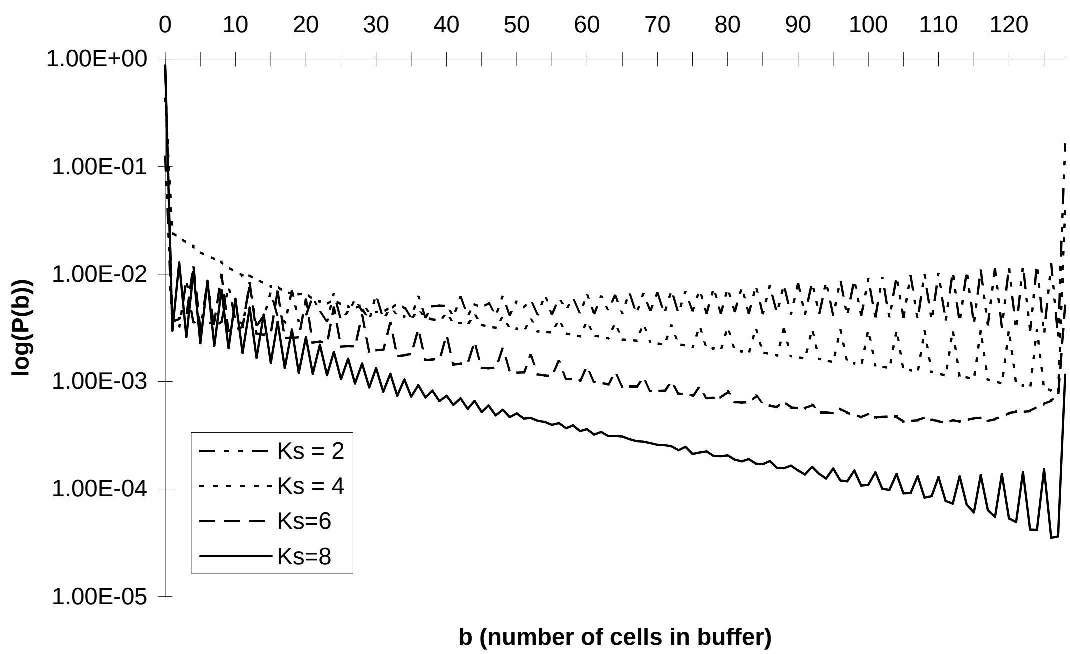

Figure 2: *The probability $P(b)$ that there are b cells in the buffer as a function of b for several server rates and for a buffer of size $B = 128$; there are 5 identical sources, $T_f = 10$, $K_i = 5$, $p_{on} = 0.91$ and $p_{off} = 0.99$.*