Numerical Analysis of Queues with Batch Arrivals

G. Wolfner and M. Telek

Department of Telecommunications, Technical University of Budapest
Sztoczek 2, H-1111 Budapest – Hungary
Email: {wolfner,telek}@hit.bme.hu

Abstract

The steady state distribution of quasi-birth-death processes can be efficiently obtained by matrix-geometric (MG) methods. Since a number of telecommunication problems are modelled by processes with batch arrivals, the extension of MG methods for these processes has practical importance. This paper presents an extension of MG methods which is effective for the analysis of quasi-birth-death processes with batch arrivals. The proposed method is compared with one of the well-known methods.

Keywords: Matrix-geometric methods, Quasi-birth-death processes, Batch arrivals.

1 Introduction

With the extremely rapid evolution of communication and computer systems and with the intention of their integration, whose most well-known example is the introduction of the asynchronous transfer mode (ATM), the present and future communication networks are characterised by the coexistence of different transmission/service requirements, communication protocols and transmission speeds. With very simple assumptions on the stochastic behaviour of the network traffic (memoryless or Markov modulated arrival and service) the transfer of data from one part of a network to another results in complex queue behaviour at the transfer point. For example, in one part of the network packets of size 1500 byte are transmitted (IP packet size used in Ethernet LANs) while in an other part cells of size 48+5 byte (the size of ATM

[1] This work was supported by the European Community through COPERNICUS'94 contract n. 1463, by the Hungarian Research Found, OTKA, under contract n. F-23971.

cell payload+header) are the base data units. At the border of these communication protocols the arrival of a 1500 byte packet on the one side requires the transmission of 32 cells on the other side. This phenomenon is commonly referred to as batch arrival.

A queuing system with Markovian arrival and service can be analysed by the underlying Markov chain. When in addition the arrival and the service are queue length independent and there is no batch arrival the underlying Markov chain has a nice block structure and is referred to as a quasi-birth-death (QBD) Markov chain [8]. There are several numerical methods to evaluate the steady state behaviour of QBD processes. The most well-known is the one proposed by Neuts which is often referred to as Matrix Geometric (MG) method [8] and is based on an iterative procedure called Simple Substitution (SS) method. Mitrani and Chakka proposed a one step method based on the spectral expansion of submatrices [2,6]. While Latouche and Ramaswami proposed an other iterative procedure with better numerical properties [5]. Naoumov et al. enhanced this method by reducing the complexity of the iteration steps [7] with a higher memory requirement.

To analyse real communication networks an effective extension of these methods for batch arrivals is necessary. A solution of these queues with batch arrivals is to enlarge the matrix block size in order to obtain a QBD structure. The price of this block size enlargement is the significant increase of computation complexity and computer storage requirement. Spectral expansion is one of the methods has been developed for batch arrivals [2,6] and has shown a good performance compared to the others [4]. According to our experiences close eigenvalues, which more often occurs when applying the method for queues with batch arrivals, may cause numerical problems in solving the boundary equations. There is an extension of the SS algorithm for fixed size batch arrivals that also shows some advantages in performance and storage requirement [11]. In this paper we propose an alternative method that generally has a better performance both in computation complexity and computer storage requirement than the above mentioned ones. Recently a paper that uses a similar approach for a particular multiplexer model with finite state space came to the authors' knowledge [12]. In contrast with [12] we provide a detailed analytical discussion (proofs) about the steps of the proposed method which can be used for infinite state models.

The rest of the paper is organised as follows. The next Section summarises the analysis of QBD processes. In Section 3 the problem of batch arrivals and the proposed algorithm is introduced. Section 4 is devoted to the comparison of the proposed algorithm and the algorithm proposed by Naoumov et al. which seems to be the best methods for QBD process. The paper is concluded in Section 5.

2 Analyses of quasi-birth-death processes

Consider a Discrete Time Markov Chain (DTMC) where the state of system is described by two random variables: $Z_n = \{I_n, J_n\}$; I_n is taking its value from $\{1, 2, \ldots, N\}$ and J_n is taking its value from $\{0, 1, \ldots\}$. This DTMC is called a *quasi-birth-death* process if only the state transitions where $J_{n+1} - J_n \in \{-1, 0, 1\}$ have a positive probability. The set of states with the same value of J_n defines a level, e.g. the states where $J_n = j$ defines level j. According to this definition state transitions are possible inside the levels and between the neighbouring levels.

The non-zero transition probabilities are given by the submatrices

- $A^{(j)}$ (lateral transitions): $A^{(j)}(i,k) = \Pr(I_{n+1} = k, J_{n+1} = j | I_n = i, J_n = j)$ $(i, k \in \{1, 2 \ldots, N\},\ j \in \{0, 1, \ldots\})$, where $\cdot(i,k)$ denotes the kth element of the ith row of a matrix;
- $B^{(j)}$ (upward transitions): $B^{(j)}(i,k) = \Pr(I_{n+1} = k, J_{n+1} = j+1 | I_n = i, J_n = j)$ $(i, k \in \{1, 2, \ldots, N\},\ j \in \{0, 1, \ldots\})$;
- $C^{(j)}$ (downward transitions): $C^{(j)}(i,k) = \Pr(I_{n+1} = k, J_{n+1} = j-1 | I_n = i, J_n = j)$ $(i, k \in \{1, 2, \ldots, N\},\ j \in \{1, 2, \ldots\})$;

and all the other transition probabilities equal to 0.

As it is seen from these definitions $A^{(j)}$, $B^{(j)}$ and $C^{(j)}$ are matrices of size $N \times N$. Assume that an m $(m \geq 1)$ threshold exists such that

- $A^{(j)} = A,\ \forall j \geq m$,
- $B^{(j)} = B,\ \forall j \geq m-1$,
- $C^{(j)} = C,\ \forall j \geq m+1$,

which means that the transition probabilities are level independent if $j \geq m$. The block structure of the transition probability matrix of a QBD process is shown in Figure 1. The upper-left submatrix of size $(Nm) \times (Nm)$ is referred to as the irregular part of the transition probability matrix and the rest as its regular part.

The methods developed for the analysis of these QBD processes, such as the spectral expansion or the MG methods, profit from the structure of the regular part and allow any kind of behaviour in the irregular part including transitions between non-neighbouring levels. Denote the steady state distribution by

$$p_{i,j} = \lim_{n \to \infty} \Pr(I_n = i, J_n = j)$$

$$\Pi = \begin{bmatrix} A^{(0)} & B^{(0)} & & & & & \\ C^{(1)} & A^{(1)} & B^{(1)} & & & & \\ & C^{(2)} & A^{(2)} & B & & & \\ & & C^{(3)} & A & B & & \\ & & & C & A & B & \\ & & & & \ddots & \ddots & \ddots \end{bmatrix}$$

Fig. 1. The non-zero blocks of the transition probability matrix of a QBD process ($m = 3$)

and introduce vectors v_j $(j \geq 0)$ as

$$v_j = [p_{0,j}, \ldots, p_{N,j}].$$

As a consequence of the block structure of the transition probability matrix the steady state distribution of a QBD process can be obtained by solving the following system of vector equations:

$$\begin{array}{llll} (a) & v_0 & = v_0 A^{(0)} + v_1 C^{(1)} & \\ (b) & v_j & = v_{j-1} B^{(j-1)} + v_j A^{(j)} + v_{j+1} C^{(j+1)} & m > j > 0 \\ (c) & v_j & = v_{j-1} B + v_j A + v_{j+1} C & j \geq m \\ (d) & 1 & = \sum_{j=0}^{\infty} v_j e_N^T & \end{array} \qquad (1)$$

where e_N^T is the column vector of size N whose elements equal to 1.

The methods developed for the steady-state analysis are based on the fact [8] that $\forall j \geq m - 1$

$$v_{j+1} = v_j R, \qquad (2)$$

where R is a matrix of size $N \times N$, and it is the minimal non-negative solution of the matrix equation [8]

$$B + RA + R^2 C = R. \qquad (3)$$

A simple way to evaluate R is to apply the SS algorithm whose iteration step is shown in Figure 2 [8]. The sequence of R_n $(n = 0, 1, \ldots)$ is entry-wise non-decreasing and it converges to matrix R [8].

The disadvantage of the method is the slow convergence, especially when the utilisation of the modelled system approaches 1. Latouche and Ramaswami

$$
\begin{aligned}
&R_0 = 0 \\
&n = 0 \\
&DO \\
&\quad R_{n+1} = B + R_n A + R_n^2 C \\
&\quad n = n + 1 \\
&WHILE\ (||R_n - R_{n-1}|| \geq \epsilon)
\end{aligned}
$$

Fig. 2. The simple substitution (SS) algorithm

proposed another iterative method, a Logarithmic Reduction (LR) algorithm, that converges much faster [5].

Recently Naoumov et al. proposed an enhancement of the LR algorithm (Figure 3), that requires less operations per iteration [7]. An iteration step of the LR method is more complicated than a step of the SS algorithm, but the fewer required iteration steps make the LR algorithm faster. The experiences so far have shown that the number of needed iteration steps are usually less than 20 [4,5,11].

$$
\begin{aligned}
&U = A - I \\
&L = B \\
&M = C \\
&W = A - I \\
&DO \\
&\quad X = -U^{-1}L \\
&\quad Y = -U^{-1}M \\
&\quad Z = LY \\
&\quad W = W + Z \\
&\quad U = U + Z + MX \\
&\quad L = LX \\
&\quad M = MY \\
&WHILE\ (||Z|| \geq \epsilon) \\
&R = -BW^{-1}
\end{aligned}
$$

Fig. 3. The logarithmic reduction (LR) algorithm of Naoumov et al.

Mitrani and Chakka proposed a direct method, called spectral expansion [2,6]. In this method the least N eigenvalues and the associated eigenvectors must be obtained from

$$\lambda\phi = \phi\left(B + \lambda A + \lambda^2 C\right),$$

where λ is a complex number and ϕ is a vector of N complex elements.

The advantage of this method is the direct solution and the easy calculation of the state probabilities based on the eigenvalues and the eigenvectors, but numerical problems can arise because of the close eigenvalues and eigenvectors.

The detailed comparison of this method and the algorithm of Naoumov et al. has not been published yet. Since the order of the complexity of these algorithms are the same ($O(N^3)$) their properties have to be investigated through numerical experiences. The previously published results show a better performance of spectral expansion, especially when the utilisation is close to 1 [4].

3 The extension of MG approach for batch arrivals

3.1 Processes with batch arrivals

Consider the DTMC $Z_n = \{I_n, J_n\}$, where I_n is taking its value from $\{1, 2, \ldots, N\}$ and J_n is taking its value from $\{0, 1, \ldots\}$, as in the previous section. Models with batch arrivals and single server differs from QBD processes only by the allowed multi-level upward transitions. Now, upward transitions are allowed from level j to level $j+l$ ($l = 1, 2, \ldots, y$), where y is the maximum batch size.

In this case the non-zero transition probabilities are given by the submatrices

- $A^{(j)}$ (lateral transitions): $A^{(j)}(i,k) = \Pr(I_{n+1} = k, J_{n+1} = j | I_n = i, J_n = j)$ ($i, k \in \{1, 2, \ldots, N\}$, $j \in \{0, 1, \ldots\}$);
- $B_l^{(j)}$ (upward transitions): $B_l^{(j)}(i,k) = \Pr(I_{n+1} = k, J_{n+1} = j+l | I_n = i, J_n = j)$ ($i, k \in \{1, 2, \ldots, N\}$, $j \in \{0, 1, \ldots\}$ and $l \in \{1, 2, \ldots, y\}$);
- $C^{(j)}$ (downward transitions): $C^{(j)}(i,k) = \Pr(I_{n+1} = k, J_{n+1} = j-1 | I_n = i, J_n = j)$ ($i, k \in \{1, 2, \ldots, N\}$, $j \in \{1, 2, \ldots\}$);

and all the other transition probabilities equal to 0.

Assume that an m ($m \geq y$) threshold exists such that

- $A^{(j)} = A, \ \forall j \geq m$;
- $B_l^{(j)} = B_l, \ \forall j \geq m - l, \ \forall l \in \{1, 2, \ldots, y\}$;
- $C^{(j)} = C, \ \forall j \geq m + 1$;

which means that the transition probabilities are level independent if $j \geq m$. The block structure of the transition probability matrix of a process with batch arrivals is shown in Figure 4 (where $y = 3$ and $m = 4$).

$$
\Pi = \left[\begin{array}{cccc|ccc|ccc|cc}
A^{(0)} & B_1^{(0)} & B_2^{(0)} & B_3^{(0)} & & & & & & & & \\
C^{(1)} & A^{(1)} & B_1^{(1)} & B_2^{(1)} & B_3 & & & & & & & \\
 & C^{(2)} & A^{(2)} & B_1^{(2)} & B_2 & B_3 & & & & & & \\
 & & C^{(3)} & A^{(3)} & B_1 & B_2 & B_3 & & & & & \\
\hline
 & & & C^{(4)} & A & B_1 & B_2 & B_3 & & & & \\
 & & & & C & A & B_1 & B_2 & B_3 & & & \\
 & & & & & C & A & B_1 & B_2 & B_3 & & \\
\hline
 & & & & & & C & A & B_1 & B_2 & B_3 & \\
 & & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots
\end{array}\right]
$$

Fig. 4. The non-zero blocks of the transition probability matrix of a process with batch arrivals ($y = 3$, $m = 4$)

As a consequence of the block structure of the transition probability matrix the following system of equations must be solved to obtain the steady state distribution:

$$
\begin{array}{llll}
(a) & v_j = \sum_{i=0}^{j-1} v_i B_{j-i}^{(i)} + v_j A^{(j)} + v_{j+1} C^{(j+1)} & j < y \\
(b) & v_j = \sum_{i=1}^{y} v_{j-i} B_i^{(j-i)} + v_j A^{(j)} + v_{j+1} C^{(j+1)} & m > j \geq y \\
(c) & v_j = \sum_{i=1}^{y} v_{j-i} B_i + v_j A + v_{j+1} C & j \geq m \\
(d) & 1 = \sum_{j=0}^{\infty} v_j e_N^T &
\end{array}
\qquad (4)
$$

A wide range of queuing models of real systems, such as Batch Markovian Arrival Processes (BMAP; a flexible model to capture stochastic properties of real traffic sources) with finite batch size and single server queue, result in a Markov chain with the block structure of Figure 4.

3.2 Analysis by block size enlargement

A possible solution to analyse systems of this kind is the transformation of the regular part of the problem into a QBD process by introducing blocks of size $yN \times yN$ (Figure 4) [4]. The advantage of this block size enlargement is that standard QBD methods can be used, but its disadvantage is the significant increase in computation complexity ($O(y^3N^3)$) and storage requirement

$(O(y^2N^2))$, if the batch size (y) is large.

3.3 Spectral expansion with multilevel jumps

The spectral expansion method has been extended to analyse systems with multi-level jumps in both directions without block size enlargement [2,6]. Although the computation complexity of solving the spectral decomposition is similar to the block enlargement $(O(y^3N^3))$ numerical experiences shows a better performance of this method [4]. But the mentioned numerical problems still remain.

3.4 Condition of stability

The stability of the process can be checked applying the method provided for M/G/1 type models in [9]. Define matrix F as $F = C + A + \sum_{i=1}^{y} B_i$ and vector π as the solution of the linear system $\pi = \pi\ F$ subject to $\pi\ e = 1$, here e is the colomn vector of ones. The considered queue is stable and the associated Markov chain is ergodic if

$$\pi\ C\ e > \pi \sum_{i=1}^{y} i\ B_i\ e. \qquad (5)$$

3.5 A level-block-size method

Here we propose a method, which is more effective than the block size enlargement both in computation complexity and in storage requirement and does not have the numerical problems of spectral expansion. The proposed method is an extension of the SS method for processes with batch arrivals. *In the rest of this section we assume that the process is ergodic, i.e., (5) holds.*

If the MG methods (either the simple substitution or the logarithmic reduction algorithm) with block size enlargement are used then matrices of size $yN \times yN$ are treated. As a result of the MG methods an $\mathbf{R}$ matrix of size $yN \times yN$ is obtained. Denote the $N \times N$ submatrices of this $\mathbf{R}$ as

$$\mathbf{R} = \begin{bmatrix} R_{1,1} & \cdots & R_{1,y} \\ \vdots & & \vdots \\ R_{y,1} & \cdots & R_{y,y} \end{bmatrix}.$$

The extension of the SS method is based on the following important theorem:

Theorem 3.1 *If the steady state distribution of the process with batch arrival exists then*

$$v_j = \sum_{i=0}^{y-1} v_{j-y+i} T_i, \quad \forall j \geq m,$$

where $T_i = R_{i+1,1}$.

Proof: Consider the process as a QBD process with block size enlargement. Equation (2) holds for the first yN size vector of the regular part:

$$[v_m, \ldots, v_{m+y-1}] = [v_{m-y}, \ldots, v_{m-1}]\mathbf{R}. \tag{6}$$

As a consequence of Equation (6)

$$v_m = \sum_{i=0}^{y-1} v_{m-y+i} R_{i+1,1}.$$

Now, assuming that the size of the irregular part is $m' \geq m$ the regular part of the process remains the same, and so does $\mathbf{R}$. In this case Equation (2) gives:

$$[v_{m'}, \ldots, v_{m'+y-1}] = [v_{m'-y}, \ldots, v_{m'-1}]\mathbf{R},$$

hence

$$v_{m'} = \sum_{i=0}^{y-1} v_{m'-y+i} R_{i+1,1}$$

holds $\forall m' \geq m$. □

$$\Pi = \left[\begin{array}{ccccc|ccc|ccc|c}
A^{(0)} & B_1^{(0)} & B_2^{(0)} & B_3^{(0)} & & & & & & & & \\
C^{(1)} & A^{(1)} & B_1^{(1)} & B_2^{(1)} & B_3 & & & & & & & \\
 & C^{(2)} & A^{(2)} & B_1^{(2)} & B_2 & B_3 & & & & & & \\
 & & C^{(3)} & A^{(3)} & B_1 & B_2 & B_3 & & & & & \\
 & & & C^{(4)} & A & B_1 & B_2 & B_3 & & & & \\
\hline
 & & & & C & A & B_1 & B_2 & B_3 & & & \\
 & & & & & C & A & B_1 & B_2 & B_3 & & \\
 & & & & & & C & A & B_1 & B_2 & B_3 & \\
\hline
 & & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots
\end{array}\right]$$

Fig. 5. The non-zero blocks of the transition probability matrix of a process with batch arrivals ($y = 3$, $m' = 5$)

Figure 5 depicts the same process as Figure 4 with an enlarged irregular part $m' = 5$. It can be seen that the regular part of the matrix maintains the same block structure.

The main consequence of Theorem 3.1 is that the first N columns of $\mathbf{R}$ ($T_i,\ i = 0, \dots y-1$) contains sufficient information to determine the steady state distribution of the process. The following two theorems allow to obtain the T_i ($i = 0, \dots y-1$) matrices.

Theorem 3.2 *The $T_i, i = 0, \dots, y-1$ matrices are the minimal nonnegative solutions of the following system of matrix equations:*

$$
\begin{aligned}
T_0 &= B_y + T_0(A + T_{y-1}C) \\
T_i &= B_{y-i} + T_i(A + T_{y-1}C) + T_{i-1}C \quad i = 1, \dots, y-1
\end{aligned}
\tag{7}
$$

Proof: Applying Theorem 3.1 for the left hand side of Equations (4c) we have:

$$
v_j \; = \; \textstyle\sum_{i=0}^{y-1} v_{j-y+i} T_i, \tag{8}
$$

and applying it for the right hand side of Equations (4c) we have:

$$
\begin{aligned}
&\textstyle\sum_{i=1}^{y} v_{j-i}B_i + v_jA + v_{j+1}C = \sum_{i=1}^{y} v_{j-i}B_i + v_jA + \left(\sum_{i=0}^{y-1} v_{j+1-y+i}T_i\right) C = \\
&\textstyle\sum_{i=1}^{y} v_{j-i}B_i + v_jA + \left(\sum_{i=0}^{y-2} v_{j+1-y+i}T_i\right) C + v_jT_{y-1}C = \\
&\textstyle\sum_{i=1}^{y} v_{j-i}B_i + v_j(A + T_{y-1}C) + \left(\sum_{i=0}^{y-2} v_{j+1-y+i}T_i\right) C = \\
&\textstyle\sum_{i=0}^{y-1} v_{j-y+i}B_{y-i} + \left(\sum_{i=0}^{y-1} v_{j-y+i}T_i\right)(A + T_{y-1}C) + \left(\sum_{i=1}^{y-1} v_{j-y+i}T_{i-1}\right) C = \\
&\textstyle v_{j-y}\left(B_y + T_0(A + T_{y-1}C)\right) + \sum_{i=1}^{y-1} v_{j-y+i}\left(B_{y-i} + T_i(A + T_{y-1}C) + T_{i-1}C\right)
\end{aligned}
\tag{9}
$$

The theorem comes from the equality of the coefficients of $v_{j-y+i},\ i = 0, 1, \dots, y-1$ in Equation (8) and (9). □

Theorem 3.3 *If $T_i^{(0)} = 0,\ i = 0, 1, \dots, y-1$ then the iteration*

$$
\begin{aligned}
T_0^{(n+1)} &= B_y + T_0^{(n)}(A + T_{y-1}^{(n)}C) \\
T_i^{(n+1)} &= B_{y-i} + T_i^{(n)}(A + T_{y-1}^{(n)}C) + T_{i-1}^{(n)}C \quad i = 1, \dots, y-1
\end{aligned}
$$

converges on the minimal non-negative solutions of Equation (7).

Proof: This algorithm is the adaptation of the SS algorithm for our case. The convergence of the algorithm can be established in the same way as it is in [8]:

first it is proved that the sequences of $T_i^{(n)}$ are entry-wise non-decreasing then it is proved that the sequence is entry-wise upper bounded by the minimal nonnegative solution. The iteration starts from zero matrices therefore the theorem is a consequence of the above simple statements.

Since the B_y, A and C matrices consist of nonnegative elements, $T_i^{(1)} \geq T_i^{(0)} = 0,\ i = 0, \ldots, y-1$ entry-wise. The increase of $T_i^{(n+1)},\ i = 0, \ldots, y-1,\ n \geq 1$ can be proved by induction:

$$T_0^{(n+1)} = B_y + T_0^{(n)}(A + T_{y-1}^{(n-1)}C) \ \geq\ B_y + T_0^{(n-1)}(A + T_{y-1}^{(n-1)}C) = T_0^{(n)}$$

and

$$\begin{aligned} T_i^{(n+1)} &= B_{y-i} + T_i^{(n)}(A + T_{y-1}^{(n)}C) + T_{i-1}^{(n)}C \ \geq \\ &\geq\ B_{y-i} + T_i^{(n-1)}(A + T_{y-1}^{(n-1)}C) + T_{i-1}^{(n-1)}C = T_i^{(n)} \quad i = 1, \ldots, y-1 \end{aligned}$$

On the other hand, $T_i \geq T_i^{(0)} = 0,\ i = 0, \ldots, y-1$ entry-wise and $T_i \geq T_i^{(n)},\ i = 0, \ldots, y-1,\ n \geq 0$ can be verified by induction as well:

$$T_0^{(n+1)} = B_y + T_0^{(n)}(A + T_{y-1}^{(n)}C) \ \leq\ B_y + T_0(A + T_{y-1}C) = T_0$$

and

$$\begin{aligned} T_i^{(n+1)} &= B_{y-i} + T_i^{(n)}(A + T_{y-1}^{(n)}C) + T_{i-1}^{(n)}C \ \leq \\ &\leq\ B_{y-i} + T_i(A + T_{y-1}C) + T_{i-1}C = T_i \qquad i = 1, \ldots, y-1 \end{aligned}$$

Since an upper-bounded monotone increasing sequence converges, the sequences $T_i^{(n)},\ i = 0, \ldots, y-1$ converge entry-wise. The limit matrices satisfy Equation (7) and they are not greater than the minimal nonnegative solutions, thus sequences of $T_i^{(n)},\ i = 0, \ldots, y-1$ converge on the minimal non-negative solutions of Equation (7). □

According to Theorem 3.3 the algorithm in Figure 6 can be used to obtain the $T_i,\ i = 0, \ldots, y-1$ matrices. The complexity of one iteration step of the algorithm is $O(y\ N^3)$ that is significantly better than the complexity of the other mentioned methods.

3.6 The steady state distribution

When the $T_i,\ i = 0, \ldots, y-1$ matrices are known only the vectors $v_0, v_1, \ldots, v_{m-1}$ miss to determine the steady state distribution of the process.

$FOR\ i = 0\ TO\ y-1$

$T_i^{(0)} = 0$

$ENDFOR$

$n = 0$

DO

$T_0^{(n+1)} = B_y + T_0^{(n)}(A + T_{y-1}^{(n)}C)$

$FOR\ i = 1\ TO\ y-1$

$T_i^{(n+1)} = B_{y-i} + T_i^{(n)}(A + T_{y-1}^{(n)}C) + T_{i-1}^{(n)}C$

$ENDFOR$

$n = n+1$

$WHILE\ (\max_i(||T_i^{(n)} - T_i^{(n-1)}|| \geq \epsilon)$

Fig. 6. The proposed numerical method to obtain the T_i matrices

Equations (4a), (4b) and (4d) can be used to obtain these unknowns. The number of unknowns is mN and the number of linearly independent equations is the same. Since $v_m = \sum_{i=0}^{y-1} v_{m-y+i}T_i$ (Theorem 3.1) the unknowns in Equation (4a) and (4b) are $v_0, v_1, \ldots, v_{m-1}$. The infinite sum in Equation (4d) can be resolved by the following theorem:

Theorem 3.4 *The normalizing equation of ergodic systems is solved by the following equality:*

$$\sum_{j=m}^{\infty} v_j = \sum_{i=0}^{y-1} v_{m-y+i} \left(\sum_{n=0}^{i} T_n \left(I - \sum_{l=0}^{y-1} T_l \right)^{-1} \right) ,$$

where I is the identity matrix with the appropriate dimension.

Proof: Let $s = \sum_{j=m}^{\infty} v_j$ and make the following transformations:

$$s = \sum_{j=m}^{\infty} v_j = \sum_{i=0}^{y-1} \sum_{j=0}^{\infty} v_{m+jy+i} = \sum_{i=0}^{y-1} \sum_{j=0}^{\infty} \sum_{n=0}^{y-1} v_{m+(j-1)y+i+n} T_n = \sum_{n=0}^{y-1} \left(\sum_{i=0}^{y-1} \sum_{j=0}^{\infty} v_{m+(j-1)y+i+n} \right) T_n = \sum_{n=0}^{y-1} \left(\sum_{i=n}^{y-1} v_{m-y+i} + s \right) T_n ,$$

Note that s is finite and entry-wise upper bounded by 1. Using the obtained expression of s the Theorem comes as:

$$s = \sum_{n=0}^{y-1} \left(\sum_{i=n}^{y-1} v_{m-y+i} + s \right) T_n$$
$$s = \sum_{i=0}^{y-1} v_{m-y+i} \left(\sum_{n=0}^{i} T_n \right) + s \sum_{l=0}^{y-1} T_l$$
$$s = \sum_{i=0}^{y-1} v_{m-y+i} \left(\sum_{n=0}^{i} T_n \left(I - \sum_{l=0}^{y-1} T_l \right)^{-1} \right)$$

In the second equation the following inequality holds entry-wise $0 \leq s\sum_{l=0}^{y-1} T_l,\ \sum_{i=0}^{y-1} v_{m-y+i}\left(\sum_{n=0}^{i} T_n\right),\ s \leq 1$. □

By this theorem we have a system of equations with the same number unknowns and independent equations. The result of the system of equations is the $v_0, v_1, \ldots, v_{m-1}$ vectors, thus all information is available to obtain the steady state distribution.

3.7 Continuous time processes

So far the discrete time Markov chains has been discussed, but the results can be applied to the steady state analysis of continuous time Markov chains (CTMC) as well. A simple way to do so is the application of the method of randomization, which produces a DTMC from a CTMC with the same steady state distribution.

Consider a CTMC with generator matrix Q. Let $q = \max_{i,j} |Q(i,j)|$. The DTMC with transition probability matrix $\Pi = Q/q + I$, has the same steady state distribution [3]. Note that neither the entry-wise division by q nor the addition of the identity matrix modify the block structure of Q, hence all the above results can be applied for CTMCs if the structure of Q is as in Section 3.1.

4 Performance comparison

4.1 The system model

A simple queuing system has been evaluated to investigate the performance of the proposed method. A system with a Markov modulated source is considered. The source transmits packets to an output link. The output link works in a slotted manner: there are fixed size time slots and in every time slot at most one data unit can be transmitted. The transmission of a data unit begins at the beginning of a time slot. We refer to data units transmitted through the output link as cells below. An infinite buffer is assumed at the output link.

The source submits at most one packet at the end of the time slots and all of these packets have the same size. The probability of a packet arrival in a time slot depends on the phase of the Markov modulated source. The source may change its phase at the end of the time slots independent of packet arrivals.

These assumptions are realistic considering a file server where TCP/IP over

ATM is used. The slotted output link has the properties of ATM and packets consisting of a fix number of cells is a possible model for large file transfers since most of the IP packets has the size of maximum transfer unit (MTU) during bulk transfer [10]. For example in Ethernet-based networks the MTU of an IP datagram is 1500 bytes, therefore the maximum packet size is 32 cells. The default MTU value in IP over ATM environment is chosen to be 9180 byte, and thus the MTU size is 192 cells [1]. The Markov modulated source represents a phase dependent arrival, e.g., the phase refers to the number of simultaneously active connections or arrivals are according to a renewal process with phase-type distributed interarrival times.

The system behaviour at the end of the nth time slot is characterised by

- the number of cells in the buffer of the output link (J_n) and
- the phase of the source (I_n).

The system has the following parameters:

- C: the number of phases of the source;
- r: the number of cells in a pac ket;
- $D_0(i,k) = \Pr(I_{n+1} = k,$ no message arrives $|I_n = i)$, D_0 is a matrix of size $C \times C$
- $D_1(i,k) = \Pr(I_{n+1} = k,$ message arrives $|I_n = i)$, D_1 is a matrix of size $C \times C$

The stochastic process $\{I_n, J_n\}$ is a DTMC. ¿From the steady state distribution of this DTMC the queue length distribution and the packet delay distribution can be obtained. The state transitions of the system are as follows:

- If no packet arrives then a cell leaves the buffer, if it was not empty at the beginning of the time slot, and the source has a phase transition from phase i to k:

$$(i,j) \to (k, \max(j-1,0)) \tag{10}$$

  The probability of this state transition is $D_0(i,k)$.
- If a packet arrives then it is stored in the buffer and a cell leaves the buffer, if it was not empty at the beginning of the time slot, and the source has a phase transition from phase i to k:

$$(i,j) \to (k, \max(j-1,0)+r) \tag{11}$$

  The probability of this state transition is $D_1(i,k)$.

As a consequence of (10) and (11) the block structure of the transition probability matrix is as in Figure 7. This structure corresponds to the problem presented in Section 3.

$$
\begin{bmatrix}
D_0 & 0 & 0 & 0 & D_1 & & \\
D_0 & 0 & 0 & 0 & D_1 & & \\
 & D_0 & 0 & 0 & 0 & D_1 & \\
 & & \ddots & & & & \ddots
\end{bmatrix}
$$

Fig. 7. The block structure of transition probability matrix ($r = 4$)

4.2 Numerical results

We have compared the proposed method (referred to as Level-Block-Size (LBS) method) to the method proposed by Naoumov et al. with block size enlargement (see Figure 3, referred to as LR method), since the LR method is one of the best among the published general methods for the steady state analysis of level independent QBD processes.

Both methods have been implemented in C using the Meschach library[2] for matrix operation. The CPU time measurements have been performed on a PC with Intel Pentium processor, using MSDOS and GNU C compiler. The boundary equations are solved with Meschach library equation system solving method, which is based on LU factorisation (Gaussian elimination) with implicit scaled partial pivoting. The reported CPU time includes only the time needed to obtain the $\mathbf{R}$ matrix in the LR algorithm and the T_i matrices in the proposed LBS method. In the experiments the required relative accuracy (ϵ in Figure 3 and 6) was set to 10^{-10} in both algorithms. The results show that the differences in the obtained steady state distribution by the two methods are marginal at this relative accuracy.

First the influence of the batch size, that is related to the packet length ($y = r - 1$), on the computation time has been investigated (Figure 8). In these experiments the number of phases of the source (C) was 5. We have compared the performance of these methods with two different system utilisation parameters ($\rho = 40\%$ and $\rho = 80\%$). The packet length has only a little influence on the computation time with the proposed method, but it has a significant impact with the LR method which uses block size enlargement. This difference can be explained by the complexity of the algorithms, the complexity of LR algorithm is $O(r^3N^3)$, while the complexity of the proposed method is $O(rN^3)$.

In Figure 8, it can be observed that the system utilisation influences the computation time as well, thus we investigated the effect of system utilisation.

[2] Meschach library for matrix computation is developed at the School of Mathematical Sciences, Australian National University by David E. Stewart and Zbigniew Leyk and it is available via netlib (ftp.netlib.org/c/meschach).

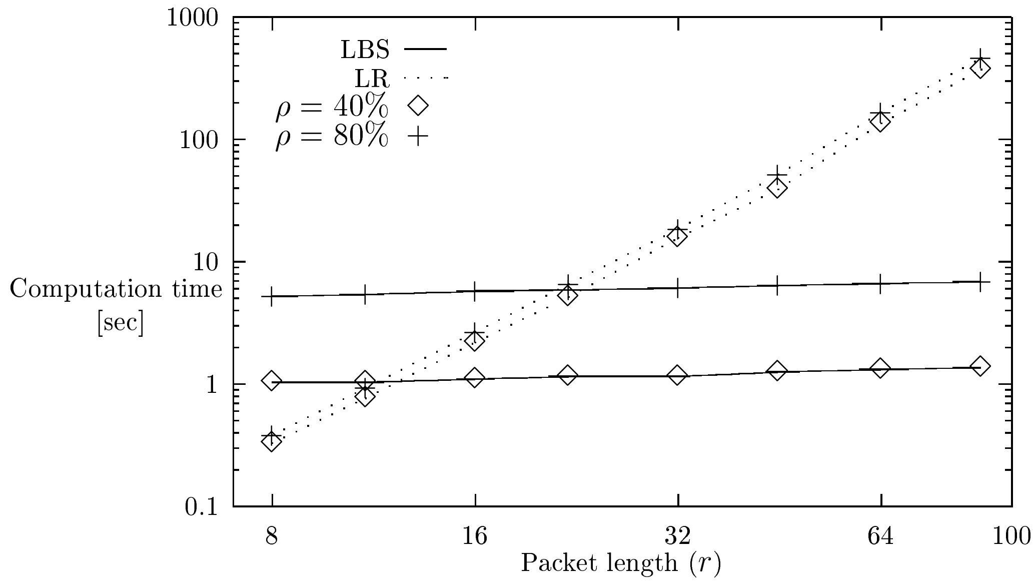

Fig. 8. Computation time versus the packet length ($C = 5$ and $\rho = 40,\ 80\%$)

Utilisation	Number of iteration steps ($C = 5,\ r = 16$)		Number of iteration steps ($C = 5,\ r = 32$)	
	LR	LBS	LR	LBS
20%	10	440	9	241
40%	11	1203	10	642
60%	13	2601	12	1374
80%	14	6341	13	3345
90%	15	13087	14	6917
95%	16	25370	15	13445
97%	17	40471	16	21499

Table 1
Number of iterations

We have found that the proposed algorithm is very sensitive to the utilisation and becomes inefficient comparing to the LR algorithm when the utilisation converges on 1 (Figure 9). This fact is due to the high number of iteration steps of the LBS algorithm (Table 1) which corresponds to the known properties of the SS algorithm [2,6]. The utilisation level at which the LR algorithm becomes more efficient than the LBS algorithm strongly depends on the maximum batch size.

The impact of the number of phases of the source (C) on the computation time is depicted in Figure 10. For this experiment the message length (r) was 16, i.e., the batch size was 15. The behaviour of the algorithms is quite

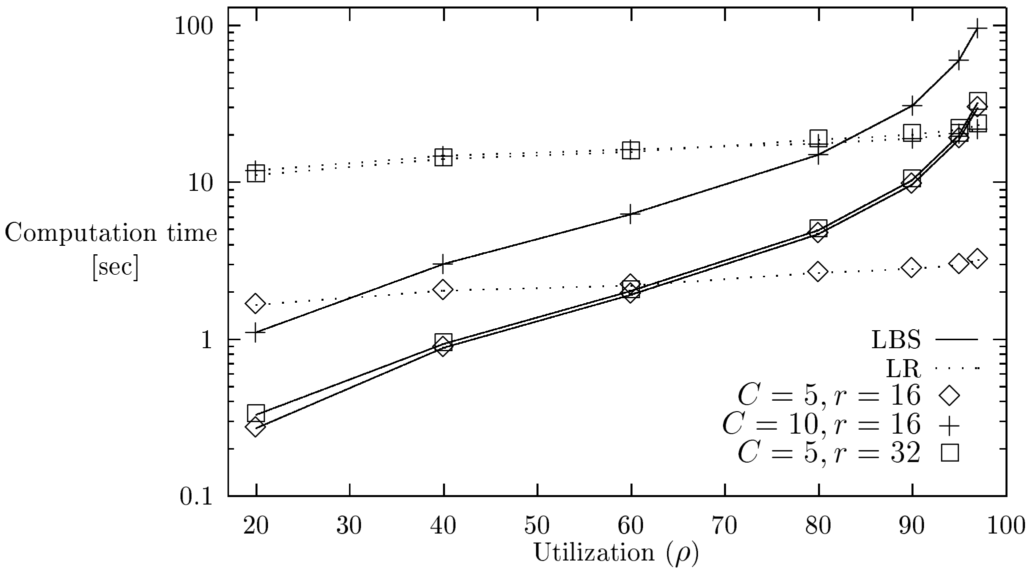

Fig. 9. Computation time versus the utilisation ($C = 5$ and $r = 16,\ 32$)

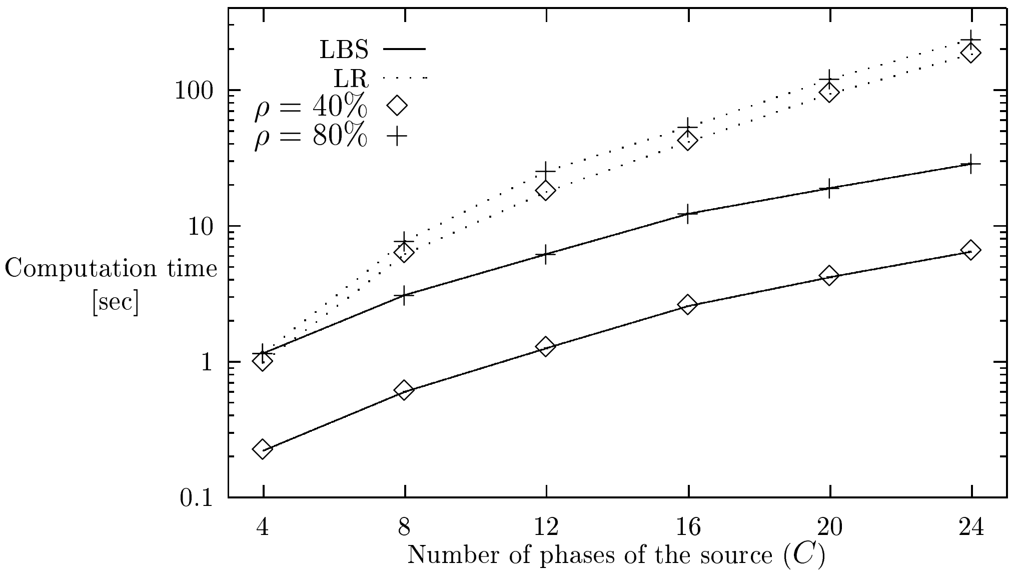

Fig. 10. Computation time versus the number of phases of the source

similar as before, although the proposed algorithm seems more efficient when the number of phases of the source is higher. The larger memory requirement of the LR algorithm may cause this phenomenon.

Our last investigation was the sensitivity of the algorithms on the required accuracy (stoping criteria, ϵ). In the previously presented experiments we used a strict required accuracy ($\epsilon = 10^{-10}$). Beyond this accuracy the obtained steady state distribution does not change significantly. The curves in Figure 11 show that the required accuracy has only a little influence of the computation time of the LR algorithm, while computation time of the LBS method is

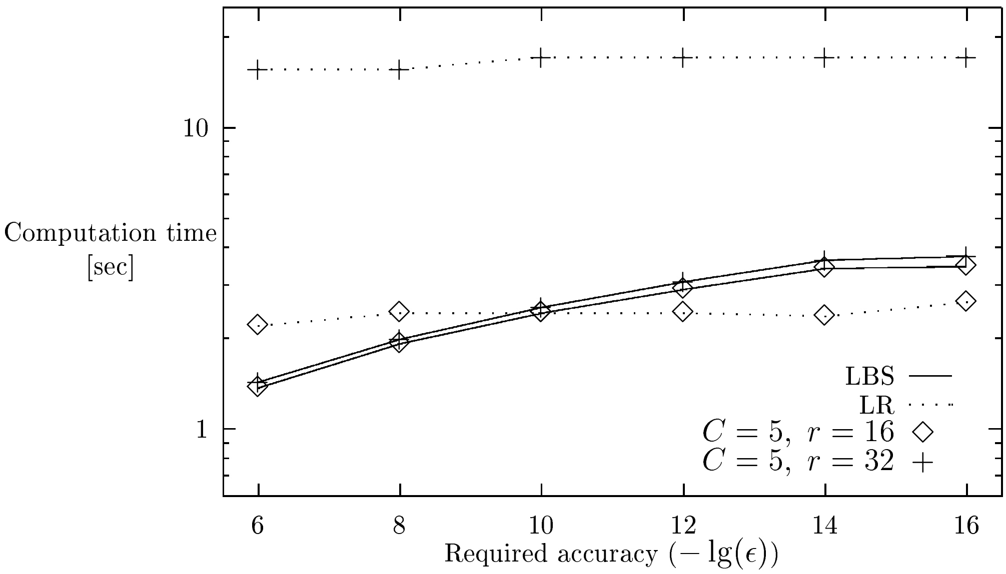

Fig. 11. Computation time versus the required accuracy ($\rho = 60\%$)

strongly depends on it. The network utilisation in this experiment was chosen to be 60%. This result implicates that the use of a less accuracy would show a better performance of the LBS algorithm.

5 Conclusions

In this paper an extension of the MG approach for pr ocesses with batch arrivals has been presented and a numerical method has been proposed to obtain the steady state distribution of these models. A process of this kind can be analysed as a QBD process with a larger block size, thus the algorithms available for the analysis of QBD processes are inefficient. We proposed a method that performs the computation without block size enlargement. The proposed approach reduces the computation complexity and the memory requirement of the numerical analysis.

A performance comparison of the proposed method with an efficient general method (LR) has also been presented. The results show that the proposed method is efficient in case of large batch size, but it becomes inefficient if the system utilisation converges on 1.

References

[1] R. Atkinson. *Default IP MTU for use over ATM AAL5, Internet RFC 1626.* Naval Research Laboratory, 1994.

[2] Ram Chakka. *Performance and Reliability Modelling of Computing Systems Using Spectral Expansion.* PhD thesis, University of Newcastle upon Tyne, 1995.

[3] W. Feller. *An Introduction to Probability Theory and Its Application*, volume 2. John Wiley & Sons, 1971. 2nd edition.

[4] B. Haverkort and A. Ost. Steady state analyses of infinite stochastic petri nets: A comparing between the Spectral Expansion and the Matrix Geometric method. In *Proceedings of 7th International Workshop on Petri Nets and Performance Models*, pages 335–346, Saint Malo, France, 1997.

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

[6] Isi Mitrani and Ram Chakka. Spectral expansion solution for a class of Markov models: application and comparison with the matrix-geometric method. *Performance Evaluation*, 23:241–260, 1995.

[7] Naoumov, Krieger, and Wagner. Analysis of a multi-server delay-loss system with a general Markovian arrival process. *Marix-analytic methods in stochastic models*, 1997.

[8] M. F. Neuts. *Matrix Geometric Solutions in Stochastic Model.* Johns Hopkins University Press, Baltimore, 1981.

[9] M. F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Theeir Matrices.* Marcel Dekker, Inc, 1989.

[10] G. Wolfner, T.V. Do, and M. Telek. A source model for file transfer applications in local ATM networks. In *Proceedings of 5th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, Ilkley (UK), 1997.

[11] G. Wolfner and M. Telek. A numerical analysis method of queues with batch arrivals. In *Proceedings of Fifth International Conference on Advanced Computing*, Chennai (Madras), India, 1997.

[12] K. Wuyts and R.K. Boel. A matrix geometric algorithm for finite buffer systems with B-ISDN applications. In *Proceedings of the ITC Specialists Seminar on Control in Communications*, pages 265–276, Lund, Sweden, September 1996.