Efficient Generalized Engset Blocking Calculation — Extended version

Jianan Zhang and Lachlan L. H. Andrew

***Abstract*—Engset's model of resource blocking with a finite population has recently been generalized to allow blocked users to have a recovery time before they re-enter contention for the resources. This can model a bufferless optical packet switch, in which the recovery time equals the duration of packet reception. We propose an algorithm to find the stationary distribution of the resulting level-dependent quasi-birth-and-death (LDQBD) process, and hence the blocking probability. Its running time is linear in the number of resources (wavelengths) and the population size (number of input ports).**

I. Introduction

Engset [1] modelled a telephone system as having a finite population of users compete for a finite pool of resources. Upon becoming idle, a user waits an exponentially distributed amount of time; if at the end of that time a resource is free, the user places a call (i.e., occupies a resource) of exponentially distributed duration. At the end of the call, or upon finding all resources occupied, the user becomes idle again. The model is used to calculate the probability of blocking, i.e., no free resources being available at the end of a users idle time.

This model has recently been generalized [2], [3] to model a bufferless optical cross connect (OXC) used for optical packet switching (OPS) [4] and optical burst switching (OBS) [5], [6]. In this context, a user represents an input port and a resource represents an output wavelength channel that the packet (or burst) can be placed on to reach its next hop destination. When the first bit of a packet arrives, if it finds no free output wavelength channel, then it is discarded. However, the input port does not become idle again immediately. Instead, the remaining bits of the packet must still be received and simply "dumped". Hence, the Generalized Engset model [2] assumes that a blocked user remains in a "dumping" state for an exponentially distributed time before becoming idle again.

This model is related to a model that has nonidentical off-time for sources considered by Cohen [7] and Syski [8].

The system constitutes a level-dependent quasi-birth-and-death (LDQBD) process, in which the phase is the number of busy servers and the level is the number of dumping servers. Matrix geometric methods can be applied to solve the blocking probability in the LDQBD [9], [10]. As in [11], this LDQBD has a very sparse upward transition matrix. It allows the standard technique for rank one upward transitions to be optimized further, yielding an algorithm whose computation is linear in the number of phases. This requires significantly less computation than previously proposed exact solutions, such as directly solving the balance equations of the Markov chain [2] and block LU decomposition [12].

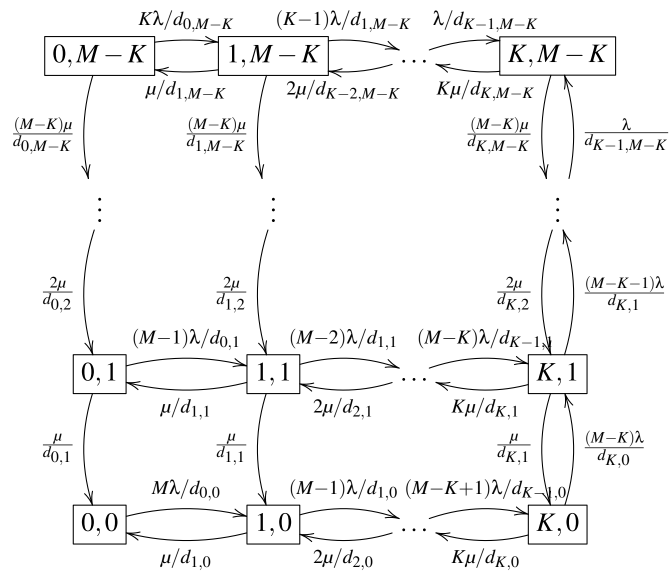

Fig. 1. State transition probabilities of the embedded Markov chain, where $d_{i,j} = (M-i-j)\lambda + (i+j)\mu$.

II. Model and notation

Unlike Engset's model, the Generalized Engset model is not insensitive to the shape of the inter-event distributions. However, numerical results [13] suggest that a Markov approximation gives a good estimate of the blocking probability, and so will be adopted.

Suppose there are M input wavelength channels, and K output wavelength channels available to packets on those inputs. Let (i,j) represent the state where data from i input channels are being transmitted through output channels, and data from j input channels are being dumped. Consider an embedded Markov chain by observing the system at the epochs when state transitions occur. The state space of the Markov chain is $\mathcal{X} = \{(i,j) \in \mathbb{N}^2 | \ i \le K \wedge j \le M-K\}$.

From states (i,j) $(0 < i < K,\ 0 < j \le M-K)$, the possible transitions to other states include completion of a successful transmission (to state $(i-1,j)$), cessation of dumping (to state $(i,j-1)$), new arrival that will be successfully transmitted (to state $(i+1,j)$). The transition probabilities to enter these states are $i\mu/d_{i,j}$, $j\mu/d_{i,j}$, and $(M-i-j)\lambda/d_{i,j}$, respectively, where $d_{i,j} = (M-i-j)\lambda + (i+j)\mu$. From states (K,j) $(0 < j < M-K)$, when a new packet/burst comes, the system goes to state $(K,j+1)$ because the new packet/burst is being dumped. The transition probability to state $(K,j+1)$ is $(M-i-j)\lambda/d_{K,j}$. The states and transition probabilities are depicted in Fig. 1.

Blocking probability can be derived from the steady state probabilities of the Markov chain. A successful transmission occurs when the Markov chain is in states (i,j) where $0 < i < K$, $0 \le j \le M-K$ and the next state is $(i+1,j)$. A packet/burst is blocked whenever the Markov chain is in states (K,j) where $0 \le j < M-K$ and the next state is $(K,j+1)$.

III. Blocking probability

It will be useful to view the transition process as an LDQBD [14]. In an LDQBD, states can be grouped into levels, indexed by $i = 0,1,\ldots,$ such that all transitions occur either within a single level or between consecutive levels. Any transition from level i to level $i+1$ is called a birth and any transition from level i to level $i-1$ is a death.

To obtain the steady state probabilities of the LDQBD, we use an algorithm inspired by theorems 8.5.2 and 10.1.3 of [9]. Let $\pi_{n,i}$ denote the steady state probability that there are n dumping inputs and i active servers. Let $\pi_n = \{\pi_{n,0},\ \pi_{n,1},\cdots,\ \pi_{n,K}\}$, and $\pi = \{\pi_0,\ \pi_1,\cdots,\ \pi_{M-K}\}$. Let the block tridiagonal P denote the transition matrix, i.e., $\pi = \pi P$. Level n consists of those states that have n dumping inputs. This results in an upward transition matrix that has a single non-zero element, which substantially reduces the complexity of computing the stationary probabilities. The matrix P is given by

$$P = \begin{pmatrix} A_1^{(0)} & A_0^{(0)} & 0 & & & \\ A_2^{(1)} & A_1^{(1)} & A_0^{(1)} & 0 & & \\ 0 & A_2^{(2)} & A_1^{(2)} & A_0^{(2)} & 0 & \\ & 0 & \ddots & \ddots & \ddots & \\ & & 0 & A_2^{(M-K+1)} & A_1^{(M-K+1)} & A_0^{(M-K+1)} \\ & & & 0 & A_2^{(M-K)} & A_1^{(M-K)} \end{pmatrix}$$

Blocks of A_0, A_1 and A_2 are $(K+1)\times(K+1)$ matrices.

$$A_0^{(n)}(i,j) = \begin{cases} \dfrac{(M-n-K)\lambda}{(M-n-K)\lambda + n\mu + K\mu} & i=j=K+1 \\ 0 & \text{otherwise} \end{cases}$$

$$A_1^{(n)}(i,j) = \begin{cases} \dfrac{(i-1)\mu}{(M-n)\lambda + n\mu + (i-1)(\mu-\lambda)} & i=j+1 \in [2,\ldots,K+1] \\ \dfrac{(M-n-(i-1))\lambda}{(M-n)\lambda + n\mu + (i-1)(\mu-\lambda)} & i=j-1 \in [1,\ldots,K] \\ 0 & \text{otherwise} \end{cases}$$

$$A_2^{(n)}(i,j) = \begin{cases} \dfrac{n\mu}{(M-n)\lambda + n\mu + (i-1)(\mu-\lambda)} & i=j \in [1,\ldots,K+1] \\ 0 & \text{otherwise} \end{cases}$$

Introduce rate matrix R [9]:

$$R^{(n)} = A_0^{(n-1)}(I - A_1^{(n)} - R^{(n+1)}A_2^{(n+1)})^{-1},\ 1 \le n \le M-K; \tag{2}$$
$$R^{(M-K+1)} = 0.$$

We now introduce an algorithm for computing $R^{(n)}$ recursively, from $n = M-K$. Let

$$S^{(n)} = I - A_1^{(n)} - R^{(n+1)}A_2^{(n+1)} \tag{3}$$
$$= \begin{pmatrix} a_{1,1} & a_{1,2} & & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & & \\ & a_{3,2} & a_{3,3} & a_{3,4} & & \\ & & \ddots & \ddots & \ddots & \\ & & & & a_{K,K} & a_{K,K+1} \\ a_{K+1,1} & a_{K+1,2} & \cdots & \cdots & a_{K+1,K} & a_{K+1,K+1} \end{pmatrix}$$

We introduce the following auxiliary variables. Let $q_0 = 0$, $a_1^* = 1$, $b_1^* = a_{K+1,1}$, and $s_{K+1} = 1$, and for $i = 1,\ldots,K$ let

$$q_i = \frac{-a_{i,i+1}}{a_{i,i} + a_{i,i-1}q_{i-1}} \tag{4a}$$
$$b_{i+1}^* = b_i^* q_i + a_{K+1,i+1} \tag{4b}$$
$$s_i = -b_i^*/b_{K+1}^* \tag{4c}$$

In addition

$$a_i^* = \begin{cases} a_{i,i-1}q_{i-1} + a_{i,i} & \text{for } i = 2,\ldots,K \\ b_{K+1}^* & \text{for } i = K+1 \end{cases} \tag{4d}$$

$$t_i = \begin{cases} s_i & \text{for } i = K, K+1 \\ s_i - t_{i+1}a_{i+1,i}/a_{i+1}^* & \text{for } i = 1,\ldots,K-1 \end{cases} \tag{4e}$$

Before we state the algorithm, we state the key property of $R^{(n)}$ that allows efficient computation.

Theorem 1. *The matrix $R^{(n)}$ of (2) is all zeros except for the last row which is $A_0^{(n-1)}(K+1,K+1)$ times the row vector $\mathbf{r}$ where $r_i = t_i/a_i^*$ for $i = 1,\ldots,K+1$ given by (4).*

Proof: The proof is constructive, finding the inverse of $S^{(n)}$ by the Gauss-Jordan algorithm using column operations. The auxiliary variables are intermediates in this process.

We will construct matrices Q, Q^*, Q' and Q'' such that postmultiplying $S^{(n)}$ by Q eliminates the upper diagonal and replaces the diagonal by $(a_i^*)_{i=1}^{K+1}$ and the off-diagonal elements of the bottom row by $(b_i^*)_{i=1}^{K}$; postmultiplying $S^{(n)}Q$ by Q^* and then Q' sets the off-diagonal elements of the bottom row to 0 and then the subdiagonal elements to 0, without changing any other elements; and finally postmultiplying by Q'' yields the identity.

Specifically, Q is the upper triangular matrix $Q_1Q_2\ldots Q_K$, where each Q_i differs from the identity I_{K+1} only in that $Q_i(i,i+1) = q_i$ given by (4a).

Next, Q^* is I_{K+1} with the off-diagonal elements of the last row replaced by $(-b_i^*/b_{K+1}^*)_{i=1}^{K}$.

To cancel the lower diagonal, we proceed from the right, and so Q' is the lower triangular matrix $Q' = Q'_{K-1}Q'_{K-2}\cdots Q'_2Q'_1$ where each Q'_i differs from the identity I_{K+1} only in that $Q'_i(i+1,i) = -a_{i+1,i}/a_{i+1}^*$. Multiplying Q'_i is a column operation of multiplying the $i+1$th column by $Q'_i(i+1,i)$ and adding it to the ith column. (Note that the matrices are multiplied in order of decreasing i, and that there are only $K-1$ factors, since the subdiagonal element of the $K+1$th row was eliminated by Q^*.)

Finally, $Q'' = \operatorname{diag}(1/a_i^*)$ since $S^{(n)}QQ^*Q' = \operatorname{diag}(a_i^*)$.

Hence the inverse of $S^{(n)}$ is $QQ^*Q'Q''$, and $R^{(n)} = A_0^{(n-1)}QQ^*Q'Q''$. Since $A_0^{(n-1)}$ is zero except for element

$A_0^{(n-1)}(K+1,K+1)$, the only non-zero elements of $R^{(n)}$ are the last row, which are $A_0^{n-1}(K+1,K+1)$ times the last row of $QQ^*Q'Q''$. It remains to show that this last row equals $\mathbf{r}$.

Since the last row of Q is $(0,0,\dots,0,1)$, the last row of $QQ^*Q'Q''$ is

$$\left(\frac{-b_1^*}{b_{K+1}^*},\frac{-b_2^*}{b_{K+1}^*},\dots\frac{-b_K^*}{b_{K+1}^*},1\right)Q'Q''.$$

Postmultiplying Q' corresponds to $K-1$ column operations on $\left(\frac{-b_1^*}{b_{K+1}^*},\frac{-b_2^*}{b_{K+1}^*},\dots\frac{-b_K^*}{b_{K+1}^*},1\right)$ in order of decreasing i. The result is $(t_1,t_2,\dots,t_{K+1})$. Finally, after postmultiplying Q'', the result is $(t_1/a_1^*,t_2/a_2^*,\dots,t_{K+1}/a_{K+1}^*)$. ■

Note that this requires $O(K)$ operations for each $R^{(n)}$ (roughly $4K$ multiplications, $4K$ divisions and $3K$ additions)[1], giving a total complexity of $O(MK)$.

We are now ready to calculate the steady state probabilities, which can be done in $O(MK)$ time as follows.

1) Find a solution to

$$\hat{\pi}_0=\hat{\pi}_0(A_1^{(0)}+R^{(1)}A_2^{(1)})=:\hat{\pi}_0A. \tag{5}$$

Specifically, choose $\hat{\pi}_{0,K+1}$ arbitrarily and then

$$\begin{aligned}\hat{\pi}_{0,K}&=\hat{\pi}_{0,K+1}(1-A(K+1,K+1))/A(K,K+1)\\ \hat{\pi}_{0,K-1}&=(\hat{\pi}_{0,K}-\hat{\pi}_{0,K+1}A(K+1,K))/A(K-1,K)\\ \hat{\pi}_{0,i}&=\frac{\hat{\pi}_{0,i+1}-\hat{\pi}_{0,i+2}A(i+2,i+1)-\hat{\pi}_{0,K+1}A(K+1,i+1)}{A(i,i+1)}\end{aligned}$$

for $i=K-1,\dots,1$.

2) Apply Theorem 1 to calculate

$$\hat{\pi}_n=\hat{\pi}_{n-1}R^{(n)},\ n=1,\ 2,\cdots,\ M-K, \tag{6}$$

3) Scale the vectors $\hat{\pi}$ uniformly to achieve

$$\left\|\sum_{n=0}^{M-K}\pi_n\right\|_1=1.$$

4) Calculate the blocking probability as

$$p=\frac{\sum_{n=0}^{M-K}\pi_{n,K}A_0^{(n)}(K+1,K+1)}{\sum_{n=0}^{M-K}(\sum_{i=0}^{K-1}\pi_{n,i}A_1^{(n)}(i+1,i+2)+\pi_{n,K}A_0^{(n)}(K+1,K+1))}.$$

The computational complexity of the algorithm is $O(MK)$, which is a significant improvement over the $O(MK^3)$ of the state-of-the-art block LU decomposition algorithm, which has been shown to be faster than the brute force way of solving the balance equations of the Markov chain [12].

IV. Numerical tractability of the algorithm

Since the algorithm aims to solve large scale problems, numerical tractability of the algorithm should be considered. We only consider overflow, and not rounding errors.

The last row of $R^{(n)}$ is $A_0^{n-1}(K+1,K+1)$ times the row vector $\mathbf{r}$ where $r_i=t_i/a_i^*$. By (4e),

$$t_i=\sum_{j=i}^{K}(s_j\prod_{k=i+1}^{j}(-a_{k,k-1}/a_k^*)) \tag{7}$$

for $i\le K-1$. If $\beta>0$ is a lower bound on $-a_{k+1,k}/a_{k+1}^*$ for $k=K-1,\dots,1$, then the coefficient of s_K in t_i is at least β^{K-i}. If $\beta>1$, this can lead to overflow for large K.

The following result shows that the calculations of $R^{(n)}$ is tractable for $n\ge 1+\lambda/\mu$.

Theorem 2. *In the calculation of $R^{(n)}$ for $1\le n\le M-K$ we have $q_i\in(0,1)$ for all $1\le i\le K$, and if $n\ge 1+\lambda/\mu$ then for all $1\le i\le K$:*

$$b_i^*\in\left(-\sum_{k=1}^{i}|a_{K+1,k}|,\,a_{K+1,i}\right);\qquad t_i\in\left(0,\sum_{j=i}^{K}s_j\right).$$

Proof: First, note the signs of the variables. For $i=1,\dots,K$,

- $R^{(n+1)}$ is non-negative by Theorem 12.1.1 of [9].
- $a_{i,i+1}\in(-1,0)$, since the only contribution is from $A_1^{(n)}$.
- $a_{i+1,i}<0$. For $i\le K-1$, $a_{i+1,i}\in(-1,0)$ since the only contribution is from $A_1^{(n)}$. For $a_{K+1,K}$, there is also a non-positive contribution from $R^{(n+1)}A_2^{(n+1)}$.
- $a_{i,i}=1$ for $i\le K$, since the only contribution is from I.
- $a_{K+1,i}\le 0$, since the only contribution is from $R^{(n+1)}A_2^{(n+1)}$.
- $q_i\in(0,1)$. This is shown inductively in Lemma 1 below, using only the signs of the $a_{j,j\pm1}$ not including $a_{K+1,K}$.
- $a_i^*\in(0,1)$ by (4d), because $q_i\in(0,1)$, $a_{i+1,i}\in(-1,0)$ and $a_{i,i}=1$.
- $b_i^*\in(-i,0)$ by induction on (4b) since $q_i\in(0,1)$ and $a_{K+1,i}\in(-1,0)$. Note that b_{K+1}^* need not be, since $a_{K+1,K+1}$ need not be negative.
- $t_i\ge 0$ since $R^{(n+1)}$ and a_i^* are non-negative and $r_i=t_i/a_i^*$.
- $s_i\ge 0$ since $s_K=t_K\ge 0$ by (4e) and all s_i have the same sign by (4c).
- $a_{K+1}^*=b_{K+1}^*\in(0,1)$. Positivity follows by (4c), since $b_i^*<0$. The upper bound comes from (4b) since the first term is negative and the second is 1 minus a term from $R^{(n+1)}A_2^{(n+1)}$.
- $a_{K+1,K+1}\in(0,1)$; the lower bound is because $b_{K+1}^*>0$.

Using the fact that $a_{i,i}=1$ and $a_{i,i\pm1}<0$ for $i=1,\dots,K$, it is shown inductively in Lemma 1 below that $0\le q_k\le\frac{(M-n-k+1)\lambda}{(M-n-k+1)\lambda+n\mu}<1$ for all $1\le k\le K$. By (4b), this gives the bound on b_{i+1}^*. Next, it is shown in Lemma 2 below that $p_i:=-a_{i,i-1}/a^*$ satisfies the recursion

$$p_{i+1}=\frac{-a_{i+1,i}}{p_ia_{i+1,i}a_{i,i+1}/a_{i,i-1}+a_{i+1,i+1}}. \tag{8}$$

whence it is inductively shown that $p_i\in(0,(i-1)/i]$ for $1+\lambda/\mu\le n\le M-K$. The bound on t_i follows by substituting this into (7) and noting that $a_{i+1,i}<0$. ■

The following are the two lemmas used in the proof of Theorem 2.

Lemma 1. *Variables $0\le q_k\le\frac{(M-n-k+1)\lambda}{(M-n-k+1)\lambda+n\mu}$ for $1\le k\le K$ in the calculations of $R^{(n)}$ for $1\le n\le M-K$.*

Proof: The proof is by induction. By (4a),

$$q_1=\frac{(M-n)\lambda}{(M-n)\lambda+n\mu}.$$

[1] Note also that an alternative $O(K)$ algorithm to calculate $R^{(n)}$ would be to express $S^{(n)}=ET$ where E is the transpose of an elementary matrix [15] and T is tridiagonal. The last row of E can be found by the Thomas Algorithm [16], and the last row of T^{-1} using [17]. However this approach seems to take roughly $10K$ multiplications, $2K$ divisions and $7K$ additions.

If $0 \leq q_{i-1} \leq \frac{(M-n-i+2)\lambda}{(M-n-i+2)\lambda+n\mu}$ for some $i \in [2,K]$, then from (4a),

$$0 \leq \frac{-a_{i,i+1}}{a_{i,i}} \leq q_i = \frac{-a_{i,i+1}}{a_{i,i}+a_{i,i-1}q_{i-1}} \leq \frac{-a_{i,i+1}}{a_{i,i}+a_{i,i-1}}$$
$$= \frac{(M-n-i+1)\lambda}{(M-n-i+1)\lambda+n\mu}.$$

Therefore, $0 \leq q_k \leq \frac{(M-n-k+1)\lambda}{(M-n-k+1)\lambda+n\mu}$ for all $1 \leq k \leq K$. ■

Lemma 2. *In the calculation of t_k for $n \in [1+\lambda/\mu, M-K]$ and $2 \leq k \leq K$, we have $0 < -a_{k,k-1}/a_k^* \leq (k-1)/k \leq 1$.*

Proof: Let $p_i = -a_{i,i-1}/a_i^*$. This gives the recursion:

$$p_i = \frac{-a_{i,i-1}}{a_i^*} = \frac{-a_{i,i-1}}{a_{i,i-1}q_{i-1}+a_{i,i}};$$
$$p_{i+1} = \frac{-a_{i+1,i}}{a_{i+1}^*} = \frac{-a_{i+1,i}}{a_{i+1,i}q_i + a_{i+1,i+1}}$$
$$= \frac{-a_{i+1,i}}{a_{i+1,i}\left(\frac{-a_{i,i+1}}{a_{i,i}+a_{i,i-1}q_{i-1}}\right)+a_{i+1,i+1}}$$
$$= \frac{-a_{i+1,i}}{p_i a_{i+1,i}a_{i,i+1}/a_{i,i-1}+a_{i+1,i+1}}. \quad (9)$$

We next inductively prove $p_k \leq (k-1)/k$ in the calculations of $R^{(n)}$ for $n \in [1+\lambda/\mu, M-K]$. Since $q_1 = -a_{1,2}$,

$$p_2 = \frac{-a_{2,1}}{a_2^*} = \frac{-a_{2,1}}{-a_{2,1}a_{1,2}+a_{2,2}}$$
$$= \frac{\frac{\mu}{(M-n-1)\lambda+n\mu+\mu}}{1-\frac{\mu}{(M-n-1)\lambda+n\mu+\mu}\frac{(M-n)\lambda}{(M-n)\lambda+n\mu}}$$
$$= \frac{\mu}{n\mu+(M-n-1)\lambda+\mu\frac{n\mu}{(M-n)\lambda+n\mu}} \leq \frac{1}{n} \leq \frac{1}{2}.$$

Make the inductive assumption $p_i \leq (i-1)/i$ for some $i \geq 2$. Substituting into (9) gives

$$p_{i+1} = \frac{\frac{i\mu}{[M-n-i]\lambda+n\mu+i\mu}}{1-\frac{i\mu}{[M-n-i]\lambda+n\mu+i\mu}\frac{(M-n-(i-1))\lambda}{(i-1)\mu}p_i}$$
$$= \frac{i\mu}{(M-n-i)\lambda+i\mu+n\mu-(M-n-(i-1))\lambda p_i i/(i-1)}$$
$$\leq \frac{i\mu}{i\mu+n\mu-\lambda} \leq \frac{i}{i+1}$$

where the first inequality uses the inductive hypothesis, and the last uses $n \geq 1+\lambda/\mu$. This establishes the upper bound.

To see that $p_k > 0$, substitute $a_{k,k-1} \in (-1,0)$, $q_{k-1} \in (0,1)$, and $a_{k,k} = 1$ (since $k \leq K$) into (4d). ■

Theorem 2 does not guarantee that s_i will remain small, since b_{K+1}^* may become small. However, if s_i nearly overflows then t_i and r_i will be large, since $a_i^* < 1$. Hence $\hat{\pi}_{n-1}$ will be negligible compared with $\hat{\pi}_n$ by (6), and its exact value is unimportant since π_{n-1} will be rounded to 0 by the following procedure.

Overflow can also arise when for some n and i the ratio of $\pi_{0,K+1}$ to $\pi_{n,i}$ is less than the ratio of the smallest to largest positive values the machine can represent. In this case, no initial choice of $\hat{\pi}_{0,K+1}$ can prevent overflow. To avoid overflow, the partially computed vector $\hat{\pi}$ can be rescaled at any stage. Even if this results in some values such as $\hat{\pi}_{0,K+1}$ being rounded down to 0, this will not affect p substantially unless p is itself close to the smallest positive value that can be represented

TABLE I
COMPUTATION TIME (SECONDS) FOR THE PROPOSED ALGORITHM (LDQBD) AND A BENCHMARK (LU [12]). $\lambda = \mu = 1$.

$(M,\ K)$	LDQBD	LU
(200,50)	0.0048	0.0468
(200,150)	0.0033	0.2340
(600,150)	0.0238	1.321
(600,450)	0.0200	13.011

V. NUMERICAL RESULTS

Table I compares the running time of this method and block LU decomposition algorithm. All the results are obtained using MATLAB software executed on a desktop PC with Intel® Xeon®, 2.67 GHz CPU, 4 GB RAM and 64-bit operating system.

We observe considerable improvement in the computation time of the LDQBD algorithm compared with that of the block LU decomposition algorithm. Moreover, the computation time is much less variable, differing by less than a factor of 10, compared with a factor of over 100 for LU decomposition. As expected, the blocking probabilities obtained by the LDQBD algorithm match the results obtained by block LU decomposition algorithm for the cases in Table I.

For larger values of M and K, it is computationally prohibitive to calculate the blocking probability using block LU decomposition. Instead, we validate the algorithm by comparing the blocking probability results with simulation results. Table II demonstrates the accuracy of our algorithm for a wide range of parameters. The cases include underload, critical load, and overload conditions, and evaluate the blocking probability in the range $10^{-6} \sim 10^{-1}$.

To further illustrate the computational efficiency of the algorithm, Fig. 2 shows the computation time for different K when $M = 200,\ 20000$.

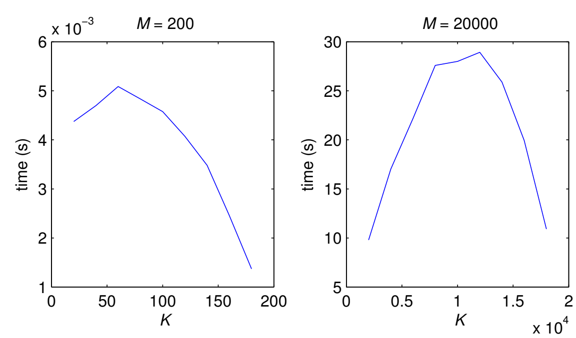

Fig. 2. Computation time of the blocking probabilities when $M = 200,\ 20000$.

TABLE II
BLOCKING PROBABILITIES OF THE PROPOSED ALGORITHM (LDQBD) AND 95% CONFIDENCE INTERVALS OF SIMULATIONS. $\mu = 1$.

$(M,\ K,\ \lambda)$	LDQBD	simulation
(1000,100,0.07)	8.18×10^{-6}	$(8.20 \pm 0.85) \times 10^{-6}$
(1000,100,0.11)	6.889×10^{-2}	$(6.886 \pm 0.009) \times 10^{-2}$
(1000,100,0.12)	1.173×10^{-1}	$(1.174 \pm 0.001) \times 10^{-1}$
(1000,500,0.75)	9.85×10^{-6}	$(9.94 \pm 1.15) \times 10^{-6}$
(1000,500,1)	3.045×10^{-2}	$(3.049 \pm 0.005) \times 10^{-2}$
(1000,500,1.3)	1.2564×10^{-1}	$(1.2562 \pm 0.0007) \times 10^{-1}$
(1000,900,6)	7.25×10^{-6}	$(7.78 \pm 0.85) \times 10^{-6}$
(1000,900,9)	1.590×10^{-2}	$(1.589 \pm 0.002) \times 10^{-2}$
(1000,900,100)	9.430×10^{-2}	$(9.428 \pm 0.004) \times 10^{-2}$
(10000,1000,0.097)	4.35×10^{-6}	$(4.68 \pm 0.74) \times 10^{-6}$
(10000,1000,0.11)	1.895×10^{-2}	$(1.897 \pm 0.007) \times 10^{-2}$
(10000,1000,0.14)	1.891×10^{-1}	$(1.891 \pm 0.001) \times 10^{-1}$
(10000,5000,0.92)	1.25×10^{-6}	$(1.05 \pm 0.37) \times 10^{-6}$
(10000,5000,1)	9.83×10^{-3}	$(9.77 \pm 0.07) \times 10^{-3}$
(10000,5000,1.3)	1.167×10^{-1}	$(1.167 \pm 0.001) \times 10^{-1}$
(10000,9000,7.9)	4.03×10^{-6}	$(3.78 \pm 0.74) \times 10^{-6}$
(10000,9000,9)	5.19×10^{-3}	$(5.19 \pm 0.01) \times 10^{-3}$
(10000,9000,100)	9.167×10^{-2}	$(9.169 \pm 0.003) \times 10^{-2}$

VI. DISCUSSION AND CONCLUSION

We have proposed an $O(MK)$ algorithm to calculate the steady state distribution, and hence blocking probability, of a generalized Engset model that arises in optical packet switching and optical burst switching. The proposed algorithm depends only on the sparsity structure of the transition matrix. This structure arises in many other applications, such as two-class priority queues and overflow queues. Those applications have a more regular structure, and we hope that the techniques introduced here may yield analytic insights into the performance of those applications. QBDs without level dependence were successfully applied to the analysis of priority queues with baulking in [18], and finding tail asymptotics of priority queues in [19].

For the specific case motivated by OPS/OBS networks, we have also investigated the numerical tractability of the algorithm, and shown that most of the intermediate values in the computation can be guaranteed not to cause numeric overflow. This allows the proposed technique to be applied to very large switches, including all those that will be developed for the foreseeable future.

ACKNOWLEDGEMENTS

This work was supported by ARC grants FT0991594 and DP130100156 and NSF grant CNS-1017800.

REFERENCES

[1] T. O. Engset, "The probability calculation to determine the number of switches in automatic telephone exchanges," *Telektronikk*, pp. 1–5, Jun. 1991, english translation by Mr. Eliot Jensen.

[2] M. Zukerman, E. Wong, Z. Rosberg, G. M. Lee, and H. L. Vu, "On teletraffic applications to OBS," *IEEE Communications Letters*, vol. 8, no. 2, pp. 116–118, Feb. 2004.

[3] H. Øverby, "Performance modelling of optical packet switched networks with the Engset traffic model," *Optics Express*, vol. 13, pp. 1685–1695, 2005.

[4] M. J. O'Mahony, D. Simeonidou, D. Hunter, and A. Tzanakaki, "The application of optical packet switching in future communication networks," *IEEE Communications Magazine*, vol. 39, no. 3, pp. 128–135, 2002.

[5] C. Qiao and M. Yoo, "Optical burst switching (OBS) - a new paradigm for an optical Internet," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 69–84, Mar. 1999.

[6] J. S. Turner, "Terabit burst switching," *J. High Speed Netw.*, vol. 8, no. 1, pp. 3–16, Mar. 1999.

[7] J. W. Cohen, "The generalized Engset formulae," *Philips Telecommunication Review*, vol. 18, pp. 158–170, 1957.

[8] R. Syski, *Introduction to Congestion Theory in Telephone Systems*. North Holland, 1959.

[9] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, ser. ASA-SIAM Series on Statistics and Applied Probability. SIAM, 1999.

[10] J. P. Kharoufeh, "Level-dependent quasi-birth-and-death processes," *Wiley Encyclopedia of Operations Research and Management Science*, Feb. 2011.

[11] J. F. Pérez and B. Van Houdt, "Quasi-birth-and-death processes with restricted transitions and its applications," *Performance Evaluation*, vol. 68, no. 2, pp. 126–141, 2011.

[12] N. Akar and Y. Gunalay, "Stochastic analysis of finite population bufferless multiplexing in optical packet/burst switching systems," *IEICE Transactions on Communications*, vol. E90-B, no. 2, pp. 342–345, 2007.

[13] J. Zhang, Y. Peng, E. W. M. Wong, and M. Zukerman, "Sensitivity of blocking probability in the generalized engset model for OBS," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1243–1245, Nov. 2011.

[14] L. Bright and P. G. Taylor, "Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes," *Stochastic Models*, vol. 11, no. 3, pp. 497–525, 1995.

[15] R. Brayton, F. Gustavson, and R. Willoughby, "Some results on sparse matrices," *Mathematics of Computation*, vol. 24, no. 112, pp. 937–954, 1970.

[16] E. Süli and D. Mayers, *An introduction to numerical analysis*. Cambridge University Press, 2003.

[17] R. Usmani, "Inversion of a tridiagonal Jacobi matrix," *Linear Algebra and Its Applications*, vol. 212, pp. 413–414, 1994.

[18] S. Drekic and D. G. Woolford, "A preemptive priority queue with balking," *European Journal of Operational Research*, vol. 164, no. 2, pp. 387–401, 2005.

[19] H. Li and Y. Q. Zhao, "Exact tail asymptotics in a priority queuecharacterizations of the preemptive model," *Queueing Systems*, vol. 63, no. 1-4, pp. 355–381, 2009.