An Approximate Analysis of an ATM Multiplexer with Multiple Heterogeneous Bursty Arrivals

Suk W. Hong

Harry G. Perros

Center for Communications and Signal Processing
Department Electrical and Computer Engineering
North Carolina State University

TR-91/13
August 1991

An Approximate Analysis of an ATM Multiplexer with Multiple Heterogeneous Bursty Arrivals

S. Hong and H.G. Perros

Computer Science Department and
Center for Communication and Signal Processing
North Carolina State University
Raleigh, NC 27695-8206, U.S.A.

and

H. Yamashita[2]

Mechanical Engineering Dept.
Sophia University
Kioi-cho 7-1, Chiyoda-ku
Tokyo 102, Japan

abstract

We analyze approximately a discrete-time queue which represents an ATM multiplexer. The queue receives input from N heterogeneous arrival processes, each modelled by an Interrupted Bernoulli Process. The Markov chain associated with these N arrival processes is first aggregated to a small matrix A. This matrix characterizes approximately the superposition process. Subsequently, the queue is analyzed numerically assuming a single arrival process described by matrix A. Comparisons with simulation data showed that the approximation algorithm has a good accuracy.

[1]Supported in part by DARPA under grant no. DAEA18-90-C-0039.

[2]Work done while on a sabbatical leave of absence at the Computer Science Dept. of N. C. State Univ.

Abstract

We analyze approximately a discrete-time queue which represents an ATM multiplexer. The queue receives input from N heterogeneous arrival processes, each modelled by an Interrupted Bernoulli Process. The Markov chain associated with these N arrival processes is first aggregated to a small matrix A. This matrix characterizes approximately the superposition process. Subsequently, the queue is analyzed numerically assuming a single arrival process described by matrix A. Comparisons with simulation data showed that the approximation algorithm has a good accuracy.

1 Introduction

ATM is currently the dorminant technology for broadband ISDN. It can efficiently switch different types of traffic such as voice, data, and video. An ATM multiplexer receives cells(fixed size packets of 53 bytes long) from a number of different incoming lines and then transmits them out onto a single outgoing line. Due to the multiple arrivals of cells, a finite buffer is provided in the multiplexer. Cells are lost when they arrive to find the buffer full. The cell loss probability and cell delay are the two major performance measure of a multiplexer.

An ATM multiplexer is typically modelled by a deterministic single server queue with N arrival streams. The queue has a finite capacity and each arrival stream is assumed to be a bursty process. The deterministic service time is equal to one slot of the outgoing line, which is assumed to be long enough so that a cell can be transmitted out. Each arrival process is also slotted, and typically, one slot is equal to a slot of the outgoing line.

The analysis of such a queueing system is quite complex due to the large number of arrival processes. One way of analyzing approximately this queue is to first obtain the superposition of all the arrival processes, and then analyze the queue with a single arrival process. This approach reduces the dimensionality of the problem. However, it requires the construction of the superposition process which is a fairly complex problem in itself. One approach for obtaining the superposition process focuses on approximating it by a renewal process, see Albin [1], Whitt [18], Sriam and Whitt [15], and also Perros and Onvural [12]. Heffes and Lucantoni [7] proposed a method for superposing approximately voice sources using a Markov Modulated Poisson Process(MMPP). An alternative method for constructing an MMPP approximation to the superposition of identical bursty sources was proposed by Baiocchi et al [3]. Finally, Heffes [6] obtained an MMPP approximation to the superposition of different MMPP arrival processes using a set of simple expressions.

An alternative way to the analysis of the ATM multiplexer queue involves the so-called fluid-flow approximation. In this case, a bursty process in the active state is viewed as a uniform flow of bits into the queue, and cell departures are modelled as a uniform flow out of the queue. This method has been confined to the case of identical bursty sources, see Anick, Mitra, and Sondhi [2], Maglaris et al [9], Norros et al [11], and Tucker [17]. Finally, Nagarajan, Kurose, and Towsley [10] compared three different approximation techniques for superposing voice sources.

ATM multiplexers have also been analyzed assuming continuous-bit-rate arrival streams, see Sengupta [14], Roberts and Virtamo [13]. Typically, all the arrival streams have a deterministic inter-arrival time. Finally, other studies of ATM multiplexers have been reported in Daigle and Langford [5], Chang, Chao, and Pinedo [4], Ide [8], and Stavrakakis [16].

Most of the methodologies reported in the literature for the analysis of a discrete-time ATM multiplexer have been obtained assuming identical arrival processes. However, this assumption is unrealistic as in a real ATM network it is highly unlikely that all patterns of arriving cells will be identical. In this paper, we analyze approximately a discrete-time queue representing an ATM multiplexer assuming different bursty arrival sources. Each source is modelled by an Interrrupted Bernoulli Process(IBP). The analysis can be extended to the case where each arrival source is modelled by a Markov Modulated Bernoulli Process(MMBP). We note that the algorithm presented

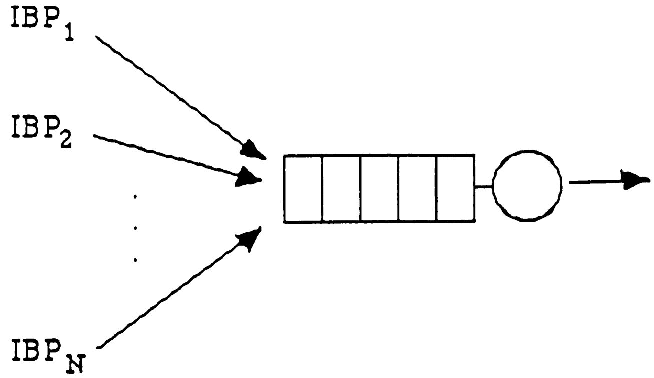

Figure 1: The queueing system under study

in this paper is similar in spirit to the one used in the analysis of a continuous-time queueing model in Yamashita, Perros, and Hong [19]. In the following section, we describe the queueing model analyzed in this paper, and in section 3, we give the approximation algorithm, and derive the cell loss probability and the cell delay distribution. In section 4, we validate the approximation algorithm by comparing it against simulation data. Finally, the conclusions are given in section 5.

2 Model Description

We model an ATM multiplexer by a discrete-time queueing system consisting of a FIFO single server queue with N different input arrival processes as shown in figure 1. The server is assumed to be slotted, and the service time is constant equal to 1 slot. Service begins and ends at slot boundaries. For presentation purposes, we shall refer to these slots as service slots. Each arrival stream is also slotted with a slot equal to a service slot. The slot boundaries of the arrival streams are in between the service slot boundaries. If a cell arrives when the system is empty, the cell waits until the beginning of the next service slot, and then begins its service. The cell departs one service slot later.

The capacity of the queue (including the cell in service) is finite and it is equal to M. During each service slot, up to N cells may arrive. Let us assume that during service slot i, c cells arrive. All c cells are admitted to the queue if there are at least c spaces free. If there are only c' free spaces, where $c' < c$, then c' cells are admitted and the remaining $c - c'$ cells are lost. Each arrival process is modelled by an Interrupted Bernoulli Process(IBP). An IBP process may find itself either in the busy state or in the idle state. Arrivals occur in a Bernoulli fashion only when the process is in the busy state. No arrivals occur if the process is in the idle state. Let us assume that at the end of slot i the process is in the idle (or busy) state. Then, in the next slot $i+1$ it will remain in the idle (or busy) state with probability q (or p), or it will change to busy (or idle)

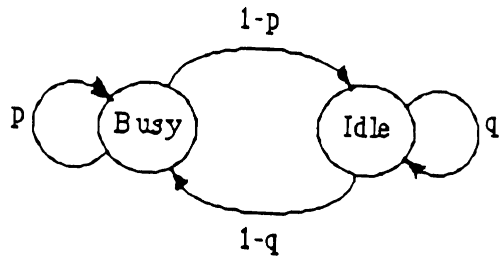

Figure 2: Transition between busy and idle states

with probability $1-q$ (or $1-p$). The transitions between the busy and idle states are shown in figure 2. If during a slot the process is in the busy state, then with probability α a cell will arrive during the slot, and with probability $1-\alpha$ no arrival will occur. In this paper, $\alpha = 1$. That is, at every busy slot there is an arrival.

Let $\bar{t}$ be the interarrival time of a cell. Then, it can be shown that the mean interrival time E[$\bar{t}$] and the squared coefficient of variation of the interarrival time C^2 are as follows:

$$E[\bar{t}] = \frac{2-p-q}{\alpha(1-q)}$$

and

$$\begin{aligned} C^2 &= \frac{Var(\bar{t})}{E[\bar{t}]^2} \\ &= 1+\alpha(\frac{(1-p)(p+q)}{(2-p-q)^2}-1). \end{aligned}$$

The link utilization ρ, i.e., the probability that any slot contains a cell is as follows:

$$\rho = \frac{\alpha(1-q)}{2-p-q}.$$

Assuming that $\alpha = 1$, we have

$$\rho = \frac{1-q}{2-p-q}, \tag{1}$$

and

$$C^2 = \frac{(p+q)(1-p)}{(2-(p+q))^2}. \tag{2}$$

3 The Approximation Algorithm

The state of our model is completely described immediately after the beginning of each service slot by the vector $(\underline{w}; n) = (w_1, \ldots, w_N; n)$, where w_i is the state of the ith IBP arrival process, w_i=0 or 1.

and n is the number of cells in the queue, $n = 0, 1, 2, ..., M$, and M is the buffer size of a multiplexer. If $w_i = 0$, then the IBP arrival process is in the idle state, and if $w_i = 1$, then the process is in the busy state. Based on this state description, we can compute the transition probability P_{ij} from state i to state j, and subsequently the matrix of transition probabilities $P = [P_{ij}]$. Finally, solving the steady state equation $\underline{\pi}P = \underline{\pi}$ where $\underline{\pi}$ is the steady state vector, we can obtain the exact probabilities of all states numerically.

The total number of states of this Markov chain is $2^N(M+1)$. Thus, the above numerical approach may work well if N is very small. We note that in the special case where all the input arrival sources are identical, the state of the system is completely described by the vector (v, n), where $v = 0, 1, 2, ..., N$, is the number of input sources that are in the busy state, and n is the number of cells in the queue. In this case, it is possible to analyze numerically cases where N is large seeing that the total number of states is $(M+1)(N+1)$. For instance, when $N = 100$ and $M = 32$, the total number of states is 3333.

In this paper, we analyze the Markov chain associated with the random variables $(w_1, ..., w_n; n)$ using decomposition. We first consider in isolation the transition matrix of the N heterogeneous arrival processes. The states of this matrix are described by the vector $\underline{w} = (w_1, ..., w_N)$, where $w_i = 1, 2$ and $i = 1, 2, ..., N$. This transition matrix may be very large, seeing that it consists of 2^N states. We aggregate this transition matrix to a considerably smaller transition matrix A involving $N+1$ states. Each state is describled by random variable k which represents the number of the heterogeneous arrival processes which are in the busy state, $k = 0, 1, ..., N$. Finally, the queue-length distribution of the number of cells in the queue is obtained by solving numerically the reduced Markov Chain (k, n). In this way we significantly reduce the state space of the original Markov Chain, while at the same time through the aggregate random variable k we maintain enough information regarding the state of the N arrival processes.

3.1 Aggregation of the N Arrival Processes

Let us consider the N heterogeneous arrival processes in isolation. The state of this system is described by the random variables $\underline{w} = (w_1, w_2, ..., w_N)$. Let $r(\underline{w} \to \underline{w}')$ be the transition probability from state $\underline{w}$ to state $\underline{w}'$. Define S_i to be the set of all states $\underline{w}$ in which there are exactly i arrival processes in the busy state. Using this definition we can lump the state space into $N+1$ sets of states S_i, $i = 0, 1, 2, ..., N$. The transition matrix with the lumped states is shown in figure 3.

Let R_{ij} be the transition probability from lump i to lump j. Then, we have

$$R_{ij} = \sum_{\underline{w} \in S_i} Pr[\underline{w}|S_i][\sum_{\underline{w}' \in S_j} r(\underline{w} \to \underline{w}')],$$

where

$$Pr[\underline{w}|S_i] = \frac{Pr[\underline{w}]}{P[S_i]}$$

and

$$P[S_i] = \sum_{\underline{w} \in S_i} P[\underline{w}].$$

		$(0,0,\ldots,0)$	$(1,0,\ldots,0)$ $\cdots$ $(0,0,\ldots,1)$		$(1,1,\ldots,1)$
k=0	$(0,0,\ldots,0)$	R_{00}	R_{01}	$\cdots$	R_{0N}
k=1	$(1,0,\ldots,0)$ $(0,1,\ldots,0)$ $\vdots$ $(0,0,\ldots,1)$	R_{10}	R_{11}	$\cdots$	R_{1N}
$\vdots$		$\vdots$	$\vdots$	$\ddots$	$\vdots$
k=N	$(1,1,\ldots,1)$	R_{N0}	R_{N1}	$\cdots$	R_{NN}

Figure 3: The transition matrix with the lumped states

Seeing that the N arrival processes are independent from each other, we have that $Pr[\underline{w}] = Pr[w_1] \cdot Pr[w_2] \cdots Pr[w_N]$, where

$$Pr[w_i = 0] = \frac{1 - p_i}{2 - (p_i + q_i)},$$
$$Pr[w_i = 1] = \frac{1 - q_i}{2 - (p_i + q_i)}.$$

Thus, we have

$$Pr[\underline{w}] = [\prod_{i \in W_1} \frac{1 - p_i}{2 - (p_i + q_i)}][\prod_{i \in W_2} \frac{1 - q_i}{2 - (p_i + q_i)}]$$

where for a given state $\underline{w}$, W_1 and W_2 are the sets of the arrival processes which are in the idle state and the busy state respectively.

The generation of the aggregate transition matrix is the most time consuming operation in the analysis of the ATM multiplexer. This is because we first have to generate the transition matrix associated with the N arrival processes and subsequently calculate the transition probabilities R_{ij}, $i, j = 0, 1, 2, \ldots, N$.

The computational effort for the generation of the aggregate matrix A can be significantly reduced by observing that the parameters p and q of a bursty IBP process are very likely to be close

$$
\begin{array}{c|cccccccccc|}
 & 0 & 1 & 2 & 3 & 4 & 5 & \cdots & N\text{-}2 & N\text{-}1 & N \\
\hline
0 & R_{00} & R_{01} & R_{02} & & & & & & & \\
1 & R_{10} & R_{11} & R_{12} & R_{13} & & & & & & \\
2 & R_{20} & R_{21} & R_{22} & R_{23} & R_{24} & & & & & \\
3 & & R_{31} & R_{32} & R_{33} & R_{34} & R_{35} & & & & \\
\cdot & & & & & & \cdot & \cdot & \cdot & & \\
\cdot & & & & & & & \cdot & \cdot & \cdot & \\
\cdot & & & & & & & & \cdot & \cdot & \cdot \\
N\text{-}2 & & & & & & R_{N-2,N-4} & R_{N-2,N-3} & R_{N-2,N-2} & R_{N-2,N-1} & R_{N-2,N} \\
N\text{-}1 & & & & & & & R_{N-1,N-3} & R_{N-1,N-2} & R_{N-1,N-1} & R_{N-1,N} \\
N & & & & & & & & R_{N,N-2} & R_{N,N-1} & R_{NN} \\
\hline
\end{array}
$$

Figure 4: Aggregate transition matrix A

to 1. For instance, in the case of voice, the squared coefficient of variation of the inter-arrival time C^2 is 18.1(see Heffes and Lucantoni [7]). Assuming a link utilization of 0.1, and using equation(1) and (2), we find that $p = 0.922$ and $q = 0.991$. The higher the value of C^2, the closer to 1 p and q get. It is highly unlikely, therefore, that a bursty source will change its current state at the next slot, seeing that $1 - p \simeq 0$ and $1 - q \simeq 0$. Based on this argument, we neglect all the transitions where more than two arrival processes change their states in the next slot. This results in great savings when it comes to generate the transition matrix of the N arrival processes. As it will be seen in the validation section, neglecting these transtions does not affect the accuracy of the results. The structure of the non-zero elements of the resulting aggregate matrix A ia shown in figure 4.

3.2 The Queue-Length Distribution

The queueing system under study can now be represented approximately by the state (k, n). The system is observed immediately after service ends. If it is found in state (k, n), then during the subsequent slot k arrivals will occur and at the end of the slot one cell will depart (given that $n > 0$). The transition probabilities $P_{(k,n)(k',n')}$ between two states are obtained using the transition probabilities R_{ij} of the aggregate matrix A. we have

$$
\begin{aligned}
P_{(k,n)(k',n')} &= Pr[(k, n) \to (k', max(0, n-1) + k)] \\
&= R_{kk'}
\end{aligned}
$$

We can now solve numerically the system of linear equations $\underline{\pi} P = \underline{\pi}$ where $P = [P_{(k,n)(k',n')}]$, in order to obtain the steady state probabilities $\pi(k,n)$, which subsequently gives the probability distribution of the number of cells in the queue $\pi(n)$, n=0,...,M. In our experiments reported in the validation section, we used the successive overrelaxation method to obtain $\pi(k,n)$.

3.3 The Cell Loss Probability

Once we have obtained the stationary vector $\pi(k,n)$, we can easily compute the cell loss probability by dividing the expected number of lost cells by the expected number of arriving cells. We have

$$p_{loss} = \frac{\sum (k+n-1-M)\pi(k,n)}{\sum_{k=1}^{N} k \sum_{n=1}^{M} \pi(k,n)},$$

where the summation in the numerator runs for all (k,n) such that $k+n-1 > M$.

3.4 The Cell Delay Distribution

The probability distribution of the number of the time slots that a cell waits in the queue before transmission can be also computed. If we exclude the transmission time, every cell waits for a minimum of 0 slots and a maximum of $M-1$ slots.

Let C_r denote the expected number of cells which wait exactly r slots and C_E denote the expected number of cells that entered the queue. If the current state at slot i is (k,n), then k cells would arrive in the next slot $i+1$. The number of cells in slot $i+1$ is equal to the sum of the remaining cells and the new arriving cells, i.e., $(n-1)+k$. If this sum is less than the buffer size M, then all k cells can enter the queue. However, if it is greater than M, only the cells which the buffer can accomodate, i.e., $M-(n-1)$, can enter the queue, and the others are lost. Thus, C_E is expressed as

$$C_E = \sum_{n=0}^{M} \sum_{k=1}^{N} \tilde{k}\pi(k,n)$$

where $\tilde{k} = min(k, M-(n-1))$.

Now, let us consider C_r. If the queue is empty or it contains one cell and the number of arriving cells at the next slot is more than r $(k > r)$, then exactly one cell among the arriving cells will wait for r slots in the queue before transmission. Likewise, if there are already two cells in the queue, one cell among arriving cells at the next slot will wait for r slots only when more than $r-1$ cells arrive. If there are already $r+1$ cells in the queue, one cell among the arriving cells will always wait for r slots regardless of the number of arriving cells k. Thus, C_r can be obtainsed as follows:

$$C_r = \sum_{n=0}^{r+1} \sum_{k=r+1-n}^{N} \pi(k,n)$$

where $\tilde{n} = max(0, n-1)$. The probability $P(r)$ that a cell waits exactly r slots before transmission can now be obtained by dividing C_r by C_E. (Note that in this calculation we do not include the offset of the arrival boundary to the next service boundary.)

4 Validation

The approximation algorithm was validated by comparing it against simulation data for eight soures ($N = 8$) and sixteen sources ($N = 16$). In figures 5 and 7 we give the approximate and simulation results for the queue-length distribution $\pi(n)$ and the cell delay distribution $P(r)$ respectively for the case of $N = 8$. (Note that the confidence intervals for the simulation results were not plotted because they were too small.) The absolute errors (*simulation – approximation*) are given in figures 6 and 8. The results were obtained assuming a buffer capacity M equal to 24. The 8 arrival processes were set as follows: $\rho_i = 0.12$, i=1,2,...,8, and $C_1^2 = C_2^2 = 50$, $C_3^2 = C_4^2 = 100$, $C_5^2 = C_6^2 = 200$, $C_7^2 = C_8^2 = 500$. The parameters p and q of each arrival process were then obtained from expressions (1) and (2). We observe that there is a good agreement between the approximation and simulation results. We note that the queue-length distribution in figure 5 is bi-modal. The two peaks correspond to the case where the queue is empty and the case where the queue is full. This pattern is due to the fact that the overall arrival process to the queue is highly bursty.

In figure 9 and 11 we give the approximate and simulation results for the queue-length distribution $\pi(n)$ and the cell delay distribution $P(r)$ respectively for the case of $N = 16$. The absolute errors are given in figures 10 and 12. The results were obtained assuming a buffer capacity of 32. The arrival processes were set as follows: $\rho_i = 0.03125$, i=1,2,...,16, and $C_1^2 = C_2^2 = C_3^2 = C_4^2 = 50$, $C_5^2 = C_6^2 = C_7^2 = C_8^2 = 100$, $C_9^2 = C_{10}^2 = C_{11}^2 = C_{12}^2 = 200$, $C_{13}^2 = C_{14}^2 = C_{15}^2 = C_{16}^2 = 500$. As in the case of $N = 8$, we observe that there is a good agreement between the approximate and simulation data. In general, it has been observed that the algorithm has a good accuracy. The algorithm's accuracy seems to be affected by the variability in the C^2 values of the arrival processes. (The squared coefficient of variation of the interarrival time C^2 is used in this paper as a measure of burstiness of an arrival process). For example, the accuracy of the approximate results in figures 5 and 7 will not be as good if the C^2 values of the arrival processes range from 1 to 500, rather than from 50 to 500. Depending on the number of arrival processes, the construction of the aggregate matrix may be CPU intensive. This is because it requires the generation of the Markov chain of the N arrival processes, which may involve a large number of states. CPU complexity problems were encountered when $N > 20$. The case of $N = 8$ can run efficiently on any small workstation. The case of $N = 16$ required a mainframe system.

The effect of the buffer capacity M on the cell loss probability and mean cell delay is shown in figures 13 and 14 for an ATM multiplexer with 8 arrival processes. Plots are given for various values of the total link utilization λ where $\lambda = \sum_{i=1}^{8} \rho_i$. The arrival sources have been set as follows: $\rho_i = \frac{1}{8}\lambda$, i=1,2,...,8, and $C_1^2 = C_2^2 = 50$, $C_3^2 = C_4^2 = 100$, $C_5^2 = C_6^2 = 200$, $C_7^2 = C_8^2 = 500$. We note that as M increases, the cell loss probability decreases and the mean cell delay increases.

The approximation algorithm described in this paper can be adapted to analyze a continuous-time version of the ATM multiplexer, where each arrival process is represented by a different Interrupted Poisson Process(IPP), and the server provides an exponentially distributed service. In this case, the construction of the aggregate matrix A is not as time-consuming as in the discrete-time case. Such a queueing system with heterogeneous arrival processes can be also analyzed using the results by Heffes [6]. In particular, Heffes constructs an MMPP approximation to the superposition of any number of different MMPP sources using a set of very simple expressions.

In order to compare our method against Heffes' method, we considered an ATM multiplexer in continuous-time as described above with 8 arrival processes and a buffer capacity of 16. Each arrival process had an average arrival rate of 0.5, and a peak arrival rate of 1. The squared coefficient of variation of the interarrival time of the arrival processes was set as follows: $C_1^2 = 200$, $C_2^2 = 100$, $C_3^2 = 80$, $C_4^2 = 50$, $C_5^2 = 40$, $C_6^2 = 20$, $C_7^2 = 10$, $C_8^2 = 2$. The total offered average traffic was $\sum_{i=1}^{8} \lambda_i = 4$. The service rate μ was obtained from the ratio $\nu = \sum_{i=1}^{8} \lambda_i / \mu$, which in our experiments was varied from 0.1 to 0.96.

Using Heffes' approximation, we obtained an MMPP approximation to the superposition of the 8 arrival processes. Subsequently, we analyzed the multiplexer numerically as an MMPP/M/1 finite capacity queue. The results for the grade-of-service(GOS), expressed as $100(1 - Pr\{cell\ loss\})$, and the mean queue-length as a function of ν are given in figures 15 and 16 respectively. In the same figures, we give results obtained using the approximation method described in this paper, but adapted for the continuous-time case, and we also give simulation results. We note that our approximation algorithm gives better estimates for GOS as ν increases. This is probably due to the fact that the aggregate matrix A contains more information regarding the superposition process than Heffes' MMPP. However, our approximation method is computationally more intensive.

5 Conclusion

We presented an approximation algorithm for the analysis of a discrete-time queue representing an ATM multiplexer. The queue receives input from N different arrival processes, each modelled by an Interrupted Bernoulli Process(IBP). The Markov chain associated with these N arrival processes is first aggregated to a small matrix A, which characterizes approximately the superposition process. Subsequently, the queue was analyzed numerically assuming a single arrival stream described by matrix A. Comparisions with simulation data showed that the approximation algorithm has a good accuracy. The algorithm's accuracy is affected by the range of C^2 values of the arrival processes. CPU complexity problems were encountered in the calculation of the aggregate matrix A, when the number of arrival processes was greater than 20.

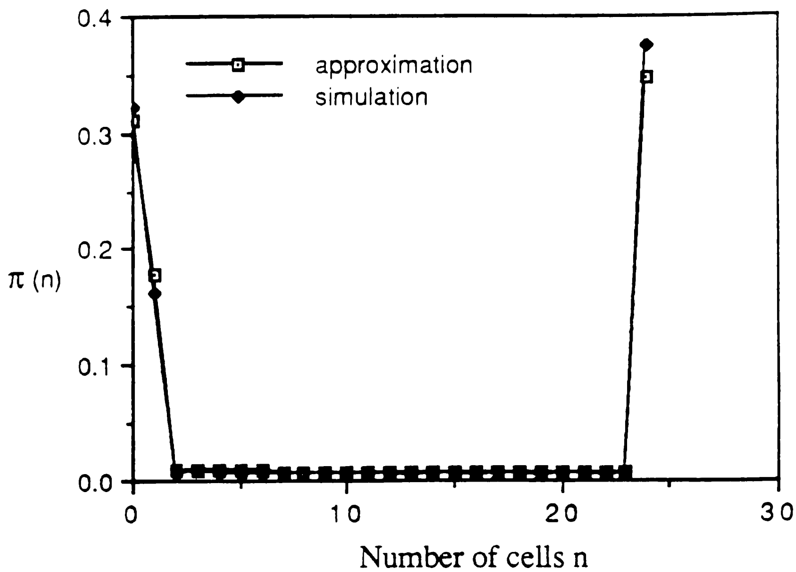

Figure 5: Queue length distribution (N=8)

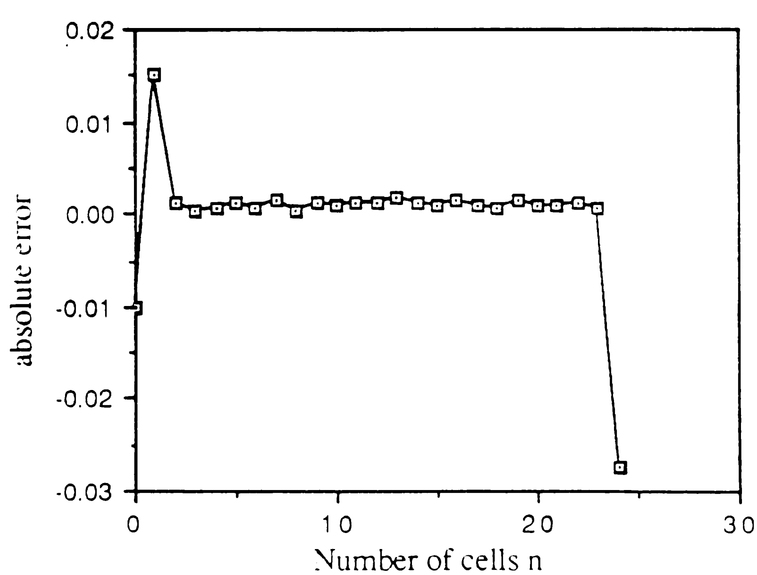

Figure 6: Absolute errors for the results in figure 5

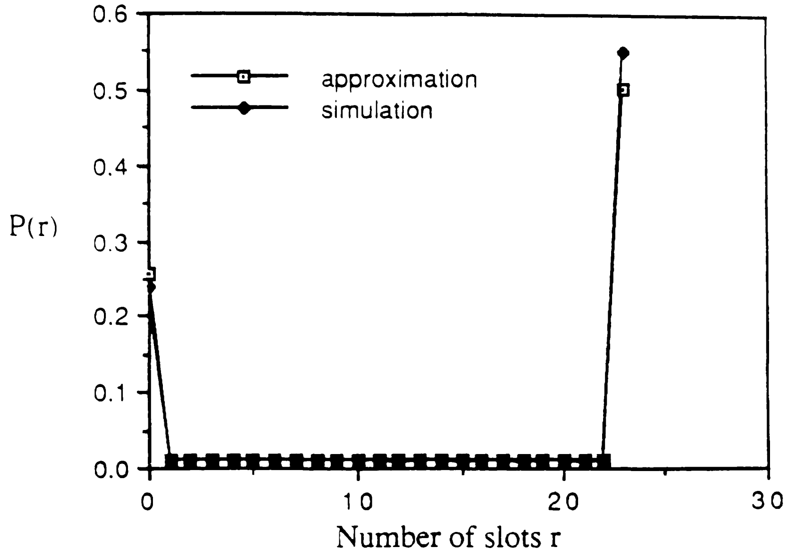

Figure 7: Cell delay distribution

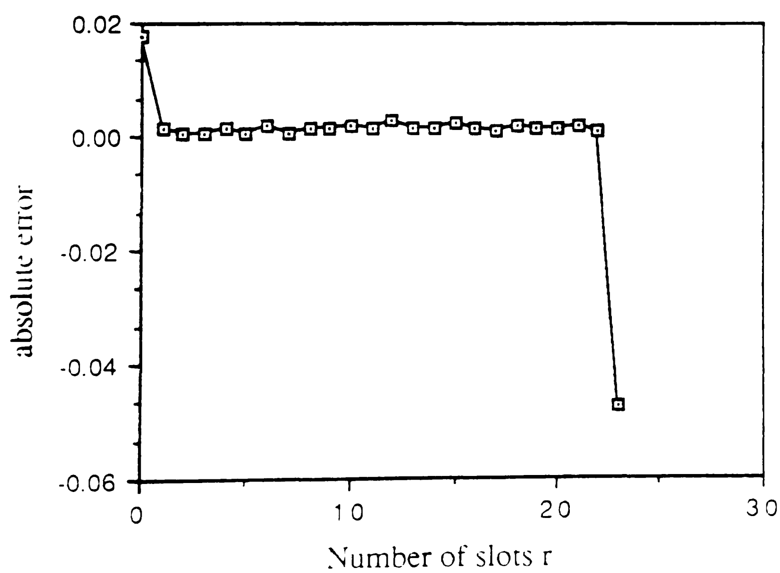

Figure 8: Absolute errors for the results in figure 7

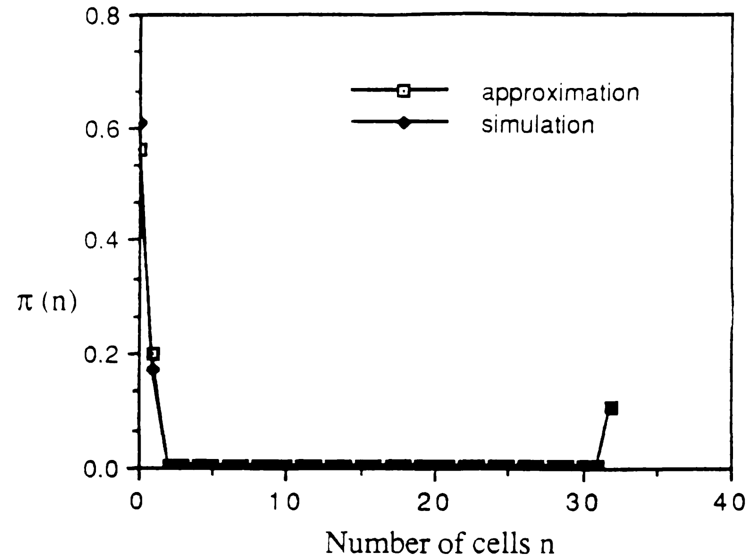

Figure 9: Queue-length distribution (N=16)

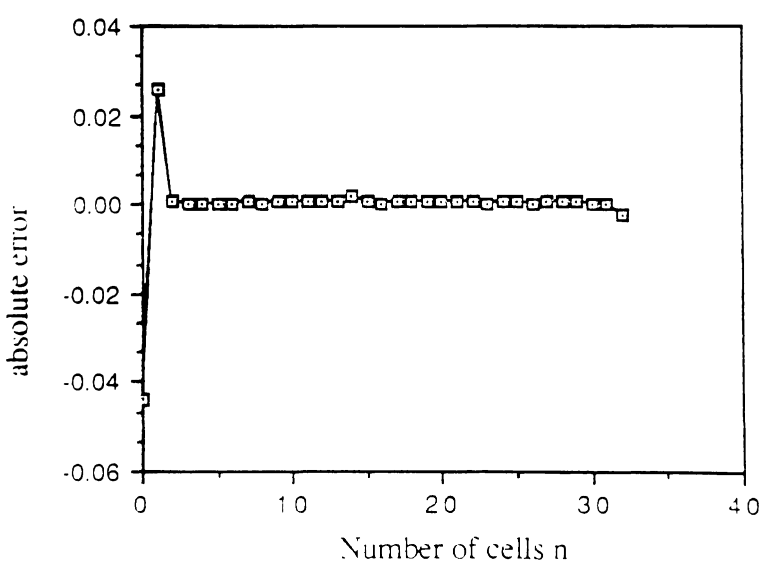

Figure 10: Absolute errors for the results in figure 9

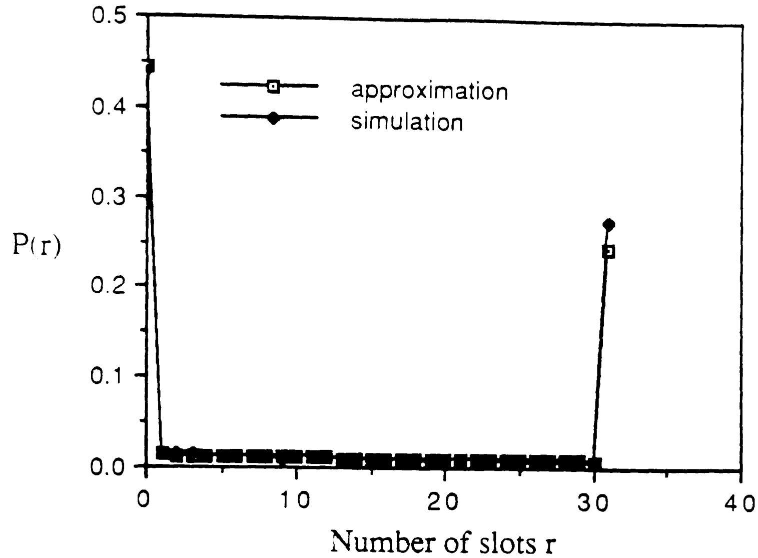

Figure 11: Cell delay distribution (N=16)

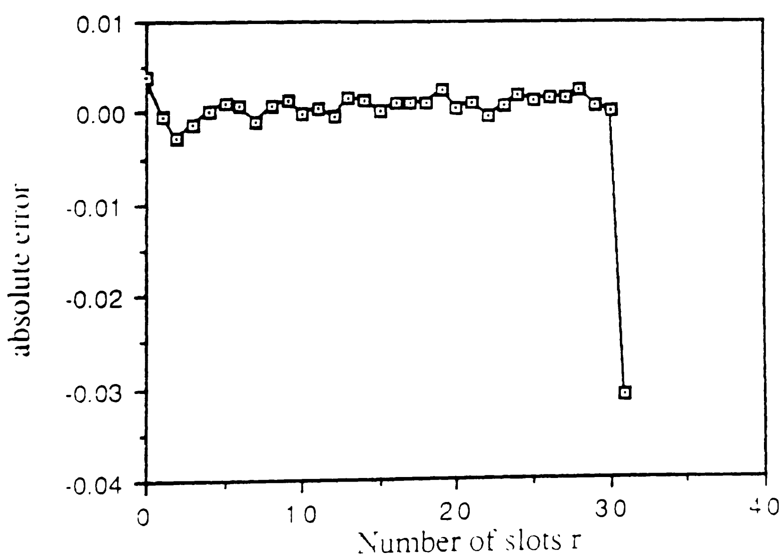

Figure 12: Absolute errors for the results in figure 11

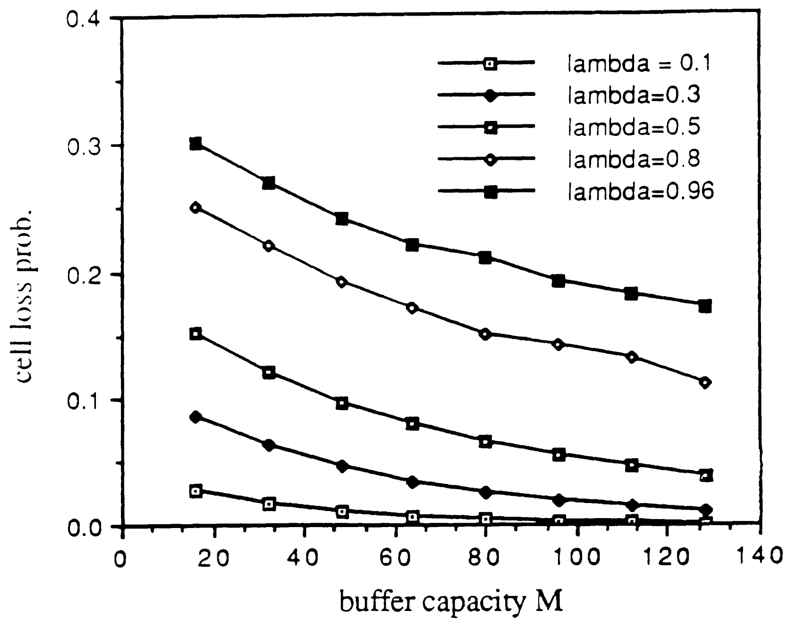

Figure 13: Cell loss probability vs. buffer size for various values of λ

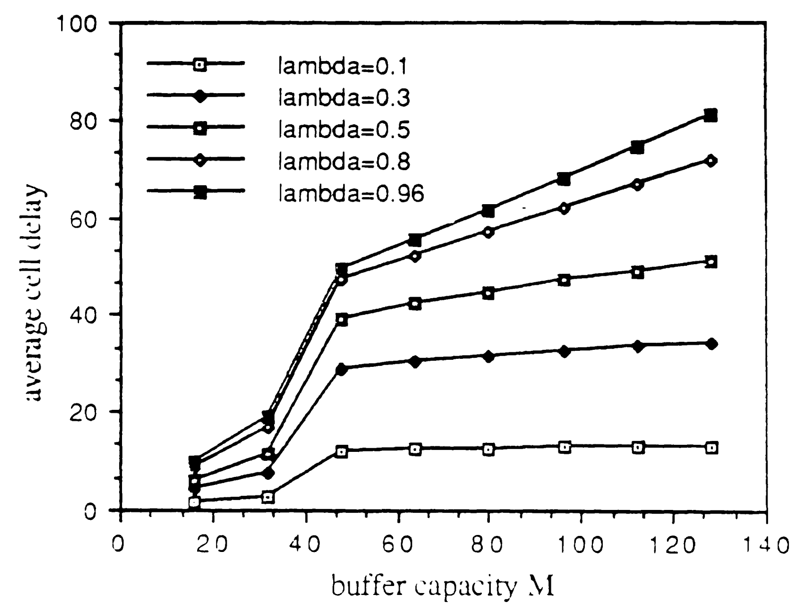

Figure 14: Average cell delay vs. buffer size for various values of λ

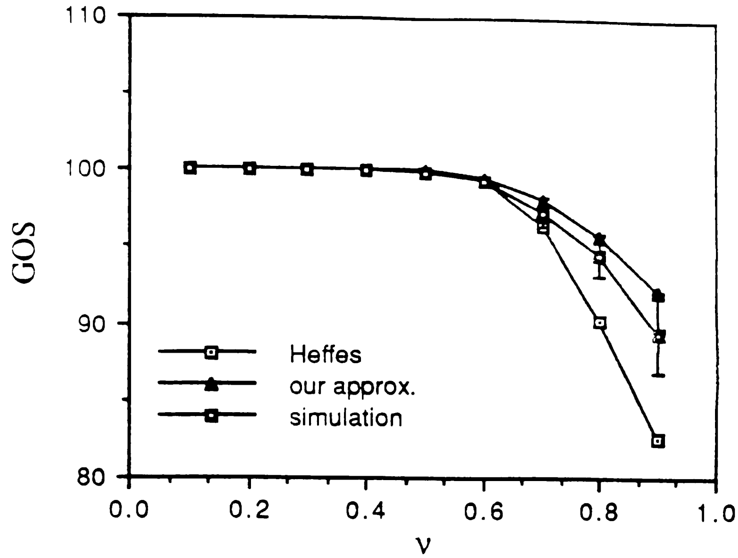

Figure 15: Comparisons for grade-of-service(GOS)

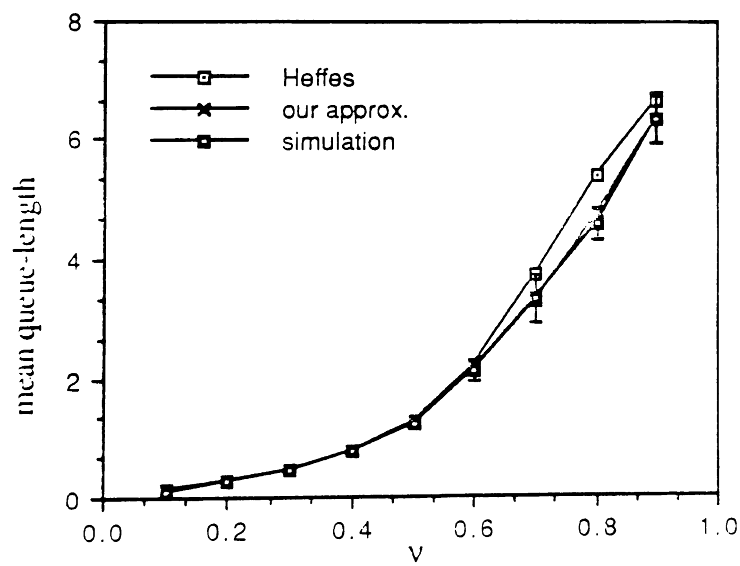

Figure 16: Comparisons for the mean queue-length

References

[1] S. L. Albin, "Approximating a point process by a renewal process, 2: Superposition arrival processes to queues," *Oper. Res.*, 32(5):1133–1162, 1984.

[2] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *Bell Sys. Tech. J.*, 61:1871–1894, October 1982.

[3] A. Baiocchi, N. Melazzi, M. Listanti, A. Roveri, and R. Winkler, "Loss performance analysis of an ATM multiplexer loaded with high-speed on-off sources," *IEEE J. SAC*, 9:388–393, 1991.

[4] C.-S. Chang, X. Chao, and M. Pinedo, "Integration of discrete-time correlated Markov processes in a TDM system," *Prob. Engin. and Inform. Sci.*, 4:29–56, 1990.

[5] J. N. Daigle and J. D. Langford, "Models for analysis of packet voice communications systems," *IEEE J. SAC*, 4:847–855, September 1986.

[6] H. Heffes, "A class of data traffic processes-covariance function characterization and related queueing results," *Bell Sys. Tech. J.*, 59:897–929, July-August 1980.

[7] H. Heffes and D. M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. SAC*, 4:856–868, September 1986.

[8] I. Ide, "Superposition of interrupted poisson processes and its application to packetized voice multiplexers," *ITC 12*, pp. 3.1B.2.1–3.1B.2.7, 1988.

[9] B. Maglaris, S. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance models of statistical multiplexing in packet video communications," *IEEE Tran. Comm.*, 36:834–844, July 1988.

[10] R. Nagarajan, J. F. Kurose, and D. Towsley, "Approximation techniques for computing packet loss in finite-buffered voice multiplexers," *IEEE J. SAC*, 9:368–377, 1991.

[11] I. Norros, J. W. Roberts, A. Simmonian, and J. T. Virtamo, "Loss performance analysis of an ATM multiplexer loaded with high-speed on-off sources," *IEEE J. SAC*, 9:378–387, 1991.

[12] H. G. Perros and R. Onvural, "On the superposition of arrival processes for voice and data," *Fourth Int. Conf. on Data Communication Systems and Their Performance*, pp. 341–357, Barcelona, June 1990.

[13] J. W. Roberts and J. T. Virtamo, "The superposition of periodic cell arrival streams in an ATM multiplexer," *IEEE Tran. Comm.*, 39:298–303, February 1991.

[14] B. Sengupta, "A queue with superposition of arrival streams with and application to packet voice technology," King, Mitrani, and Pooley (Ed.), *Performance '90*, pp. 53–59 North Holland, 1990.

[15] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE J. SAC*, 4:833–846, September 1986.

[16] I. Stavrakakis, "Analysis of a statistical multiplexer under a general input traffic model," *INFOCOM '90*, pp. 1220–1225, 1990.

[17] R. C. F. Tucker, "Accurate method for analysis of a packet-speech multiplexer with limited delay," *IEEE Tran. Comm.*, 36:479–483, April 1988.

[18] W. Whitt, "Approximation a point process by a renewal process, I: Two basic methods," *Oper. Res.*, 30(1):125–147, 1982.

[19] H. Yamashita, H. G. Perros, and S.-W. Hong, "Performance modelling of a shared buffer ATM switch architecture," *ITC 13*, Copenhagen, June 1991.