RZ 1730 (#62748) 8/30/88
Communications 23 pages

Research Report

Analysis of a Discrete Time Queueing System with Batch Arrivals and its Applications in Packet-Switching Systems

Phuoc Tran-Gia, Member IEEE, and Hamid Ahmadi, Member, IEEE

IBM Research Division
Zurich Research Laboratory
8803 Rueschlikon
Switzerland

ANALYSIS OF A DISCRETE TIME QUEUEING SYSTEM WITH BATCH ARRIVALS AND ITS APPLICATIONS IN PACKET-SWITCHING SYSTEMS

Phuoc Tran-Gia, Member, IEEE, and Hamid Ahmadi, Member, IEEE.
IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract – In this paper, we present and solve a discrete-time $G^{[X]}/D/1-S$ queueing system with a finite queue size and batch arrivals with a general batch-size distribution. The motivation for this model arises from performance modeling of a statistical multiplexer with synchronous transmission of fixed-size data-units in synchronous time slots. The arrival process to the multiplexer, for example, may originate from a number of independent sources with packets of variable lengths. Hence, a packet arrival corresponds to an arrival of a batch of data-units. Different performance measures such as percentage of packet loss and data-unit loss are considered under two different admission policies of packets into the queue.

July 28, 1988

Revised and extended version of a paper presented in the INFOCOM '88 Conference, New Orleans, March 1988.

I. Introduction and Problem Statement

THE overall performance of a packet-switching network depends heavily on the performance of its communication links, their associated statistical multiplexers or buffers, and packet switches. Proper sizing of the buffers and loading of the links for a specific performance are of major concern in the design of any packet-switched network. The purpose of this paper is to present the analysis and applications of the discrete-time $G^{[X]}/D/1-S$ queueing system. The notation used indicates a single-server finite queue (S) with batch arrivals of general interarrival times (G) and batch-size (X) distribution, and a constant service time (D).

The motivation for the discrete-time $G^{[X]}/D/1-S$ queueing model presented here arises from some practical applications in packet-switching systems where the frequency of packet loss is an important performance measure. We deliberately use a discrete-time model because, in many practical applications, systems actually operate in clocked cycles and transfer fixed-length data blocks from buffers. This model is very well suited for a statistical multiplexer with a synchronous output transmission link. Synchronization means that the system clock is maintained, and a single *data-unit* is transmitted at equally spaced time slots. This data-unit may be considered as a character, a byte, or a fixed-size block of data (minipacket). The arrival instants to the queue are also assumed to occur at discrete-time slots. The total number of data-units arriving during a time slot is modeled as a batch with batch interarrival times (in units of slots) having a general distribution. The generality of our model is such that the batch size and the batch interarrival time distribution can be modeled to represent different classes of input sources. There are several applications for this model in which the arrival process to the multiplexer is non-Poisson. For example, as shown in Fig. 1(a), this model may represent a statistical multiplexer fed by a number of independent sources [1]-[3], the common-control sections of an electronic exchange system as modeled in [4], [5], using in addition a renewal approximation. Another example is a statistical multiplexer, like in [6]-[8], with arrivals comprising user packets of variable length composed of many data-units. In this respect, the arrival of a user packet can be considered as arrival of a batch of data-units to the queue. In queueing terminology, the data-units are equivalent to customers and user packets are batches of customers.

Recently, much attention has been paid to high-performance switch fabric architectures which handle different traffic types in a uniform way. Of particular interest is a generic switch fabric structure [Fig. 1(b)] which operates in a packet-switched mode and has buffers at each output. The switch has no internal loss so that packets arriving at any of the inputs can reach any of the outputs. One example of such a switch is the integrated switch fabric proposed in [9]. Other examples are a crossbar switch with output FIFOs [10], or the output stage of the Knockout switch [11]. The performance of these switches can be analyzed exactly by the queueing model presented in this paper.

Another application of this model is in performance modeling of an Asynchronous Transfer Mode (ATM) concept which makes use of statistical multiplexing with an attempt to handle different types of traffic in a uniform manner. Various analytical models have been proposed using a Geo/D/1/K [12] or an approximate M/D/1 [13] queueing model to study the performance of ATM. Each of these models is a special case of the model presented here. There are several factors such as burstiness of the sources and packet length distribution which are ignored in these models. These effects can be captured effectively with our model. In general, queueing models with batch arrivals can be used to approximate the effect of an arrival process which is bursty and an exact analytical solution for it is quite complicated.

Returning to our model, since the queueing model under consideration has limited capacity, overflow can occur. Two performance measures are considered: 1) the probability of batch or packet loss, and 2) the probability of data-unit loss. Depending on the application, each of the loss probabilities has a different merit. For example, when the batches are equivalent to user packets of variable length, then the probability of packet loss is of interest. When batches are comprised of several data-units emanating from many sources, then the probability of data-unit loss is important. It should be noted that in practical applications, the buffer size is usually represented in data-units and not in user packets, because user packets typically have a variable length.

To complete the modeling, two different admission policies are considered when an arriving batch is larger in size than the number of unoccupied storage places in the queue. They are:

1) Blocking policy 1 (BP1): An arriving batch of data-units larger in size than the number of available free spaces in the queue fills the free positions and the remaining data-units of the batch are lost.

2) Blocking policy 2 (BP2): An arriving batch of data-units larger in size than the available free positions in the queue is completely rejected.

It should be emphasized that from an implementation point of view, the two blocking policies have different overhead and trade-offs which are application dependent. For example, in the case of blocking policy 2 and user packets of multi-data-units, a mechanism is needed to sense the available space in the queue before admitting a packet into the FIFO queue. On the other hand, in the case of blocking policy 1 and user packets of multi-data-units, this mechanism is not necessary. However, a different mechanism is needed to disregard the partially admitted packet in the queue. For the multiplexer in which several sources may simultaneously generate fixed-length user packets (i.e., a user packet is equivalent to a data-unit), policy 2 does not make sense.

Several models have been proposed which study the behavior of a statistical multiplexer. In [6], [7], a transmission buffer has been modeled as a finite queue with batch Poisson arrivals, geometric batch-size distribution, and a constant synchronous output. In [8], an infinite capacity buffer was used to approximate a finite capacity buffer with a very small packet-loss probability. An infinite queueing model with batch arrival was also used to study the behavior of a common-control switching system [5]. A number of theoretical studies for both infinite [14]-[16] and finite [17] queues with batch arrivals has appeared with different degrees of complexity. However, it is not easily seen how the results in [17] may be used in practical applications to assess the buffer-size requirements. In another paper [18], a queue with a finite capacity storage with exponential service time for individual data-units was analyzed. Although the data-units were all of fixed size, justification for the exponential service time assumption was explained by including other factors such as retransmission time arising from errors on the line in the service time of the data-units.

The major contributions of our paper are: 1) to extend the finite queueing model presented in [6], [15] by incorporating general interarrival times and

batch-size distributions, and 2) to present an efficient, simple and systematic computational method in discrete-time domain based on fast convolution algorithms. It should be emphasized that the model presented here can be formulated theoretically by standard techniques such as state equations and the generating function method. However, solution of the state equations, either directly (by matrix inversion) or indirectly (by inversion of the moment generating function) is in general very complex if not impossible.

The paper is organized as follows: In Section II, we present the basic description of the $G^{[X]}/D/1-S$ queueing system and the analysis. In Section III, we then show, by numerical examples, the effect of various system parameters on the probability of packet and data-unit loss. In addition, in Section IV, we illustrate the application of this model via two examples. Finally, the conclusion is given in Section 5.

II. Analysis of $G^{[X]}/D/1$-S in Discrete-Time Domain

Before proceeding to the analysis, we point out some aspects concerning the methods used. In principle, special forms of the queueing system presented can be solved using standard methods operating in continuous time domain. In those cases, additional simplifying assumptions have to be made, since we have several non-memoryless processes, for which a Markov chain cannot be imbedded [4]. In contrast to this, by observing and analyzing the system in discrete time, we are able to develop algorithms built by a small number of operations. Examples for these operations are the convolution and π -operations as discussed later in this section. In turn, these operations can be enumerated efficiently using powerful algorithms developed in signal-processing theory [e.g., use of Fast Fourier Transform (FFT) based on the Discrete Fourier Transform (DFT) to process the convolution operation]. Due to the finite-state space limited by $S+1$, the convolution can be segmented. Thus, the convolution operations required have just to be enumerated within the finite-state space ($0 \leq k \leq S+1$). For small values of S (e.g., $S < 100$), this can be done directly; for larger S, more efficient algorithms like FFT can be employed. Furthermore, it should be noted that the algorithms in the discrete-time domain developed here are stable for a wide range of system parameters.

A. Random Variables and Notation

As mentioned above, we use methods operating in the discrete-time domain to analyze the general class of queueing systems $G^{[X]}/D/1-S$. In this analysis, we consider the random variables to be of discrete-time nature, i.e., the time axis is conceived to be divided into intervals of unit length Δt, which is the service or transmission time of a single data-unit. As a consequence, samples of these random variables are integer multiples of Δt.

We use the following notation for functions and measures belonging to a discrete-time random variable (r.v.) R:

$r(k) = Pr(R = k)$,	$-\infty < k < +\infty$	distribution (probability mass function) of R
$R(k) = \sum_{i=-\infty}^{k} r(i)$,	$-\infty < k < +\infty$	distribution function of R
ER, c_R		mean and coefficient of variation of R

Further, the following notation is employed:

S — queue capacity in data-units.

A_n — random variable for the generalized interarrival time of the batch input process, which describes the time interval between the arrival epochs of the n-th and the (n+1)-th batch. Since $a_n(0)$ can have a non-zero value, batch-arrival processes with geometrically distributed batch size can also be dealt with (cf. [17]).

X_n — random variable for the size of the n-th batch.

The random variables A_n and X_n can be parameterized individually for each arriving batch so that the analysis derived below can also be applied to investigate the non-stationary behavior of the system.

B. State Analysis

A sample path of the state process development in the $G^{[X]}/D/1-S$ system is shown in Fig. 2. Let U be the amount of unfinished work in the system, which is

the number of data-units to be processed. We define the following random variables (cf. Fig. 2):

U_n random variable for the number of data-units in the systems immediately prior to the arrival instant of the n-th batch.

U_n^+ random variable for the number of data-units in the system immediately after the arrival instant of the n-th batch.

Depending on the two blocking policies defined above, we derive relationships between these random variables and their respective distributions. We then present algorithms to determine the state probabilities and then the blocking probabilities of batches and data-units.

1) Blocking policy 1 (BP1): Based on the definition of BP1, when an arriving batch of size i finds the system with $j < i$ available buffer positions, the buffer will be filled up with (i -j) data-units, and the remainder of the batch will be rejected, i.e., j data-units are accepted and (i -j) data-units are blocked.

Observing the system state prior to and immediately after the arrival epochs of the n-th and $(n+1)$-th batch (cf. Fig. 2), for blocking policy 1, we obtain

$$U_n^+ = \min (U_n + X_n , S + 1) \tag{1}$$

$$U_{n+1} = \max (U_n^+ - A_n , 0). \tag{2}$$

From (1) and (2), their respective distributions are given by

$$u_n^+(k) = \pi^{S+1}(u_n(k) \star x_n(k)), \tag{3}$$

$$u_{n+1}(k) = \pi_0\left(u_n^+(k) \star a_n(-k)\right), \tag{4}$$

where $\pi^{S+1}(.)$ and $\pi_0(.)$ are operators on probability distributions defined by

$$\pi^{m}(r(k)) = \begin{cases} r(k) & k < m \\ \sum_{i=m}^{\infty} r(i) & k = m \\ 0 & k > m \end{cases} \tag{5}$$

$$\pi_{m}(r(k)) = \begin{cases} 0 & k < m \\ \sum_{i=-\infty}^{m} r(i) & k = m \\ r(k) & k > m \end{cases} \tag{6}$$

and the ★-symbol denotes the discrete convolution operation

$$r_3(k) = r_1(k) \star r_2(k) = \sum_{j=-\infty}^{+\infty} r_1(k-j) \cdot r_2(j). \tag{7}$$

Equations (3) and (4) represent a recursive relation between the system states seen upon arrival by two consecutive batches n and $(n+1)$. Using these equations, an algorithm for both stationary and non-stationary cases can be developed to calculate the system-state probability prior to the batch-arrival epochs. The corresponding computational diagram is depicted in Fig. 3.

For the case of identical, independent interarrival intervals with random variable A, and batch sizes with random variable X, which are now assumed to be time-independent, (3) and (4) deliver an iterative algorithm to determine the equilibrium state probabilities

$$u(k) = \lim_{n \to \infty} u_n(k). \tag{8}$$

2) Blocking policy 2 (BP2): Based on the definition of BP2, an arriving batch of size i which finds the system with $j<i$ available buffer positions will be entirely rejected. We obtain the following equations for the system-state random variables:

$$U_n^+ = \begin{cases} U_n + X_n & U_n + X_n \le S+1 \\ U_n & U_n + X_n > S+1 \end{cases} \tag{9}$$

$$U_{n+1} = \max(U_n^+ - A_n, 0). \tag{10}$$

Distributions of these random variables are given as

$$u_n^+(k) = \sum_{j=0}^{k} u_n(j)x_n(k-j) + u_n(k) \cdot \sum_{j=S+1-k+1}^{\infty} x_n(j), \quad k = 0,1,\ldots,S+1 \tag{11}$$

$$u_{n+1}(k) = \pi_0\left(u_n^+(k) \star a_n(-k)\right), \qquad k = 0,1,\ldots,S+1. \tag{12}$$

Since the functional relationship between U_{n+1} and U_n^+ is the same for both blocking policies as given in (2) and (10), (4) and (12) are also identical.

Similar to the case of BP1, a recursive relation between the system-state probabilities seen by two consecutively arriving batches is given by (11) and (12). Further steps are analogous to the case of BP1.

3) State Analysis Algorithm: As mentioned above, the state space of the system is limited by $S+1$. Thus, the distributions or probability mass functions of the r.v. U_n, U_n^+ can only have non-zero elements within $[0,S+1]$. Taking into account this property, the convolution and correlation operations in (3), (4) and (12) have to be enumerated only in $[0, S+1]$, irrespective of the lengths of the distributions $x_n(k)$ and $a_n(k)$. For large value of S (e.g., $S > 100$), these operations can be enumerated efficiently in the transform domain, using fast convolution algorithms like the Fast Fourier Transform (FFT). For small values of S, these operations can be implemented in a direct way in the time domain.

The time domain algorithm will be given in more detail in this subsection. For this, it is sufficient to give explicit relations for a recursive calculation of the system states seen upon arrival by two consecutive batches n and (n+1) in the form of $u_{n+1}(k) = \text{function}[u_n(k)]$. To evaluate non-stationary system behavior, this relation can be directly used. To investigate the system state under stationary conditions, this recursive relation is used with an iteration scheme.

STEP 1: $u_n^+(k) = \text{function}[u_n(k)]$

i) *Blocking policy 1 (BP1)*: (3) yields

$$\begin{aligned} u_n^+(k) &= \sum_{j=0}^{k} u_n(j)x_n(k-j) , \qquad k = 0,1,\ldots,S \\ u_n^+(S+1) &= 1 - \sum_{j=0}^{S} u_n^+(j) , \qquad k = S+1 \end{aligned} \tag{13}$$

ii) *Blocking policy 2 (BP2)*: (11) yields

$$u_n^+(k) = \sum_{j=0}^{k} u_n(j)\, x_n(k-j) + u_n(k)\left(1 - \sum_{j=0}^{S+1-k} x_n(j)\right), \quad k = 0,1,\ldots,S+1. \tag{14}$$

STEP 2 : $u_{n+1}(k) = \text{function}[u_n^+(k)]$

for both blocking policies: (4) or (12) can be written as

$$\begin{aligned} u_{n+1}(k) &= \sum_{j=k}^{S+1} u_n^+(j)a_n(j-k) , \qquad k = 1,\ldots,S+1 \\ u_{n+1}(0) &= 1 - \sum_{j=1}^{S+1} u_{n+1}(j) , \qquad k = 0 \end{aligned} \tag{15}$$

It can be clearly seen in (13), (14) and (15) that the analysis algorithm requires only values in the range of $k=0,...,S+1$ of the interarrival time and the batch size distributions a(k) and x(k).

As mentioned, for the parameter range discussed in Section III, the convergence behavior of the algorithm is good. We have used the difference between the mean of two consecutive distributions

$$\text{abs}[EU_{n+1} - EU_n] < \varepsilon \ , \ (\varepsilon = 10^{-6})$$

as convergence criterion. For the range of parameters shown in Section III, convergence has been typically reached after less than 50 iteration cycles.

C. Blocking Probabilities

Using the equilibrium-state probabilities $\{u(k), \ k=0,...,S+1\}$, the blocking probabilities for batches and data-units can be derived for both blocking policies.

1) Batch blocking probability: We first consider the conditional blocking probability for batches defined by

$P_B(k)$: probability for a batch to be rejected, conditioned on the system state $U=k$ seen upon arrival. Under blocking policy 1, a batch is considered as blocked when a part of it is rejected.

It is obvious that

$$P_B(k) = \sum_{j=S+2-k}^{\infty} x(j), \quad k = 0,1,...,S+1. \tag{16}$$

By eliminating the condition, we arrive at the blocking probability for an arbitrary batch:

$$P_B = \sum_{k=0}^{S+1} u(k) \sum_{j=S+2-k}^{\infty} x(j) = \sum_{k=S+2}^{\infty} (u(k) \star x(k)). \tag{17}$$

2) Data-unit blocking probability: In contrast to the batch blocking probability, which can be derived for both blocking policies in the same way, the data-unit blocking probability must be derived separately.

a) Blocking policy 1 (BP1): Observing a test data-unit contained in an arriving batch, we first determine the conditional blocking probability for data-units defined by

$P_{DU}(k)$ = probability for the test data-unit in an arriving batch to be rejected, conditioned on the state $U = k$ observed upon arrival.

The probability for the test data-unit to be in a batch of size i is $i \bullet x(i)/EX$. For a batch of size i, blocking will occur for $i + k > S + 1$, where a fraction of $k + i - (S + 1)$ data-units will be rejected. Accordingly, the probability of the test data-unit being in the fraction rejected is $(k + i - (S + 1))/i$. Thus, the *conditional* data-unit blocking probability is given by

$$\begin{aligned} P_{DU}(k) &= \sum_{i=S-k+2}^{\infty} \frac{k + i - S - 1}{i} \bullet \frac{i \bullet x(i)}{EX} \\ &= \frac{1}{EX} \sum_{i=S-k+2}^{\infty} (k + i - S - 1) \bullet x(i), \quad k = 0,\ldots,S + 1. \end{aligned} \tag{18}$$

By eliminating the condition $U = k$, the data-unit blocking probability is

$$P_{DU} = \sum_{k=0}^{S+1} u(k) P_{DU}(k) = \frac{1}{EX} \sum_{k=0}^{S+1} u(k) \sum_{i=S+2-k}^{\infty} (k + i - S - 1) \bullet x(i) \ . \tag{19}$$

b) Blocking policy 2 (BP2): Again, we observe a test data-unit which arrives in a batch of size i and finds the system in the state U = k. The probability for the test data-unit to be in an arriving batch of size *i* is $i \cdot x(i)/EX$. Blocking will occur for $i+k > S+1$, where the entire batch, i.e., all data-units will be rejected. Hence, the *conditional* data-unit blocking probability is now

$$P_{DU}(k) = \sum_{i=S-k+2}^{\infty} 1 \cdot \frac{i \cdot x(i)}{EX} = \frac{1}{EX} \sum_{i=S-k+2}^{\infty} i \cdot x(i), \quad k = 0,...,S+1. \tag{20}$$

The data-unit blocking probability of a system with blocking policy 2 is given as

$$P_{DU} = \frac{1}{EX} \sum_{k=0}^{S+1} u(k) \sum_{i=S+2-k}^{\infty} i \cdot x(i). \tag{21}$$

III. Numerical Results

In this section, we present numerical results for various classes of input processes and batch-size distributions. It should be noted that the results discussed below will focus on the influence of the variations of the input process and the batch sizes, which are the essential components of the model considered in this study.

For this purpose, with the exception of the deterministic case, we use the negative binomial distribution to obtain a parametric representation of various classes of random processes. We do this by matching the interarrival and batch-size distributions given by their two parameters, namely, the mean and the coefficient of variation. The negative binomial random variable R with mean ER and coefficient of variation c_R, has the distribution

$$r(k) = \binom{y+k-1}{k} p^y (1-p)^k, \quad 0 \le p < 1, \ y \text{ real}, \tag{22}$$

where

$$p = \frac{1}{ER \bullet c_R^2}, \quad y = \frac{ER}{ER \bullet c_R^2 - 1},$$

$$ER \bullet c_R^2 > 1.$$

Since the service time is chosen to be $\Delta t = 1$, the offered traffic intensity is just

$$\rho = \frac{EX}{EA}. \tag{23}$$

For the numerical results given here, the coefficients of variation of the appearing discrete-time processes are chosen to include a wide range of variations.

Figures 4 and 5 show the blocking probabilities (for both batches and data-units) as a function of the buffer size (in data-units) for blocking policies 1 and 2, respectively. These figures include a family of curves for different values of the coefficient of variation of the batch size. The constant parameters for these curves are: $\rho = 0.5$, $c_A = 1.5$, and $EX = 4$. There are several interesting observations. As can be seen in Fig. 4, under blocking policy 1, the blocking probability of a data-unit is smaller than the blocking probability of a batch when the batch size is constant ($c_X = 0$). This is not surprising because when an arriving batch is blocked under policy 1, a fraction of that batch is admitted to the queue, hence the percentage of data-units lost is smaller. As the coefficient of variation of batch size is increased, the blocking probability of a data-unit becomes larger than that of a batch. This is because when the batch-size variation is large, the data-units blocked are more likely to emerge from a large batch than from a small one. For blocking policy 2 (Fig. 5), the batch and data-unit blocking probabilities are the same when batches are all of fixed length ($c_X = 0$). This is obvious, since in this case the whole batch is rejected and as a result, the percentage of loss for batches and data-units must be the same. When $c_X > 0$, the data-unit blocking probability is greater than the batch blocking probability.

Another interesting point is the crossover of the batch blocking-probability curves for both policies when the buffer size is relatively small (as compared to the variance of a batch size), the reason being that when the batch-size variation is

large, some small size batches can still enter the queue when the occupancy of the queue is near to its maximum capacity. For example, if the maximum queue size is ten data-units and the batch size is fixed and equal to four data-units, then any arriving batch will be rejected when the queue contains more than six data-units. Now, if the batch size is uniformly distributed from one to seven data-units (with the same mean $EX = 4$), some batches can still enter the queue up to when the queue is absolutely full.

Figures 6 and 7 depict the batch and data-unit blocking probabilities, respectively, as a function of the batch-size coefficient of variation. These curves are shown for different values of the interarrival coefficient of variation. The other parameters are the same as before, and the buffer size is fixed at 32. As expected, the batch blocking probability in policy 1 is always greater than in policy 2 (Fig. 6), because according to policy 1, the available space in the queue is occupied by the partial admission of a blocked batch. In the case of policy 2, since the available queueing space is not occupied by a fraction of a blocked batch, some small-size batches can still enter the queue, hence causing less overall blocking. For the same reason, as shown in Fig. 7, the data-unit blocking probability is greater in policy 2 than in policy 1.

Figures 8 and 9 show the blocking probabilities for policy 1 and policy 2, respectively, as a function of the interarrival coefficient of variation for different values of the batch-size coefficient of variation. These curves are given for the same parameters as before.

Finally, in Fig. 10, we show the batch blocking probability as a function of the offered traffic for different values of the batch-size coefficient of variation. The crossing effects of the curves are again apparent here. As the offered load is increased, the batch blocking probability for batches with large variation in size is less than for batches with small variations.

IV. Applications of the Model to Some Practical Problems

In this section, we briefly discuss how this model can be applied to solve some related problems in packet switching. In particular, we look at two examples from the literature in which models have been considered to analyze the

performance of a FIFO buffer in the context of a statistical multiplexer and a packet switch, respectively. The purpose of giving these two examples is to show the power of our model, especially with respect to the arrival process and the batch-size distribution. The intent is to show how our model can be used to solve these two cases, by selecting the interarrival and batch-size distributions appropriately.

The first example is taken from the paper by Karol *et al.* [10], in which they model a crossbar N×N space division switch with output FIFOs, like the one shown in Fig. 1(b). Their assumption is that the crossbar switch operates N times faster than the input and output links so there is no contention within the space switch. Time is slotted and each input generates a fixed-size packet per unit time according to a Bernoulli process with probability p. Each packet has equal probability 1/N of being destined to one of the outputs. From the view of a particular output queue, we can observe that at every time slot, the arrival process is again a batch process. The batch-size distribution is given by a binomial distribution

$$x(k) = \binom{N}{k} Q^k (1-Q)^{N-k}, \tag{24}$$

with Q = p/N. Using our model, one can easily obtain the steady-state buffer length distribution and the probability of packet loss for this system. This model has also been analyzed by Eckberg [19] using generating functions. To obtain the steady-state probabilities the generating functions must be numerically inverted. The method proposed in our paper can, in addition, appropriately model the performance of this class of switches under different traffic scenarios such as variable-length packets and a general interarrival time distribution. Once the arrival process has been correctly modeled, the system performance can be analyzed in a straightforward manner.

The second example is from modeling of the ATM concept (ATM: Asynchronous Transfer Mode) which uses statistical multiplexing to handle different types of traffic in a uniform manner. Some papers [12], [13] have proposed and analyzed a Geo/D/1/K queuing model to study the performance of ATM. This

model is a special case of the class of models presented in this paper, where the interarrival distribution given by (22) is taken for y = 1. In addition, by choosing the proper batch size distribution, one can capture more closely the effect of bursty traffic which may arise partly from the bursty sources themselves and partly from the interdependence between successive arrivals.

V. Conclusion

In this paper, we have presented and analyzed a discrete-time $G^{[X]}/D/1-S$ queueing system with a finite queue size and batch arrivals with general batch-size distribution. By means of numerical examples, we have shown how system performance, namely, the blocking probabilities, depends on the batch-size statistics of the arrival process, not only the mean but the variance of the batch size. We have also shown how batch acceptance policy affects system performance. We used discrete-time analysis for two reasons: 1) many practical systems actually operate in a clocked cycle mode, therefore discrete-time representation is the natural way to capture the behavior of the system, and 2) the discrete-time approach provides a very robust and simple computational method based on fast convolution algorithms. The queueing model presented here is general enough for it to be effectively applied to a wide range of practical problems in packet-switching environments. We have given two examples, modeling of a generic packet switch with output FIFOs, and modeling of a statistical multiplexer with bursty sources.

References

[1] C. Anick, D. Mitra and M. M. Sondhi, "Stochastic theory of a data handling system with multiple sources," Bell Syst. Tech. J., vol. 61, no. 8, pp. 1871-1894, 1982.

[2] B. Gopinath and A. J. Morrison, "Discrete-time single server queues with correlated inputs," Bell Syst. Tech. J., vol. 56, no. 9, pp. 1743-1768, 1977.

[3] S. Halfin, "The blocking of data in buffers with variable input and output rate," in *Performance of Computer-Communication Systems*, H. Rudin and W. Bux, Ed., Amsterdam: North Holland, 1984, pp. 307-319.

[4] G. Nakamura, "Analysis of a discrete-time queueing system with bulk arrival," *Electronics and Communications in Japan*, vol. 51-A, no. 11, pp. 27-32, 1968 Colorado, pp. 41.2.1, 1981.

[5] H. G. Schwaertzel, "Serving strategies of batch arrivals in common control switching systems," in *Proc. 7th Int. Telecomm. Conf.*, Stockholm, Sweden, 1973.

[6] W. W. Chu, "Buffer behavior for batch Poisson arrivals and multiple synchronous constant outputs," *IEEE Trans. Comput.*, vol. 19, pp. 530-534, 1970.

[7] W. W. Chu and L. C. Liang, "Buffer behavior for mixed input traffic and single constant outputs rate," *IEEE Trans. Commun.*, vol. 20, pp. 230-235, Apr. 1972.

[8] W. W. Chu and A. G. Konheim, "On the analysis and modeling of a class of computer communication systems," *IEEE Trans. Commun.*, vol. 20, pp. 645-660, June 1972.

[9] H. Ahmadi, W. E. Denzel, C. A. Murphy and E. Port, "A high-performance switch fabric for integrated circuit and packet switching," in *Proc. INFOCOM'88*, New Orleans, LA, Mar. 1988, pp. 1A.2.1-1A.2.10.

[10] M. J. Karol, M. G. Hluchyj and S. P. Morgan, " Input vs. output queueing on a space division packet switch," in *Proc. GLOBECOM'86*, Houston, TX, 1986, pp. 19.4.1-19.4.7.

[11] K. Y. Eng, M. G. Hluchyj and Y. S. Yeh, "A knockout switch for variable-length packets," in *Proc. ICC'87*, Seattle, WA, 1987, pp. 22.6.1-22.6.6.

[12] P. Boyer, J. Boyer, J. R. Louvion and L. Romoeuf, "Time transparency evaluation of an asynchronous time-division network," ISS'87, Phoenix, AZ, Mar. 87.

[13] M. dePrycker and J. Bauwens, " The ATD concept: one universal bearer service," CEPT/BSLB Seminar on Broadband Switching, Albufeira, Jan. 1987.

[14] U. N. Bhat, " Imbedded Markov chain analysis of single server bulk queues," *J. Aust. Math. Soc.*, vol. 4, pp. 244-263, 1964.

[15] P. J. Burke, "Delays in single-server queues with batch input," *Operation Research*, vol. 23, pp. 830-833, 1975.

[16] I. W. Kabak, "Blocking and delays in $M^{[X]}/M/c$ bulk arrival queueing systems," *Management Science*, vol. 17(1), pp. 112-115, 1970.

[17] T. P. Bagchi, and J. G. C. Templeton, "Finite waiting bulk queuing systems," *J. Eng. Math.*, vol. 7, pp. 313-317, 1973.

[18] D. R. Manfield and P. Tran-Gia, "Analysis of a finite storage system with batch input arising out of message packetization," *IEEE Trans. Commun.*, vol. COM-30, pp. 456-462, 1982.

[20] P. Tran-Gia, "Discrete time analysis for the interdeparture distribution of a GI/G/1 queue," in *Teletrafic Analysis and Computer Performance Evaluation*,

[19] A. E. Eckberg and T.-C. Hou, "Effects of output buffer sharing on buffer requirements in an ATDM packet switch," in *Proc. INFOCOM'88*, New Orleans, LA, pp. 5A.4.1-5A.4.7, 1988.

[20] P. Tran-Gia, "Discrete time analysis for the interdeparture distribution of a GI/G/1 queue," in *Teletraffic Analysis and Computer Performance Evaluation*, O. J. Boxma, J. W. Cohen and H. C. Tijms, Ed., Amsterdam: North Holland, pp. 341-357, 1986.

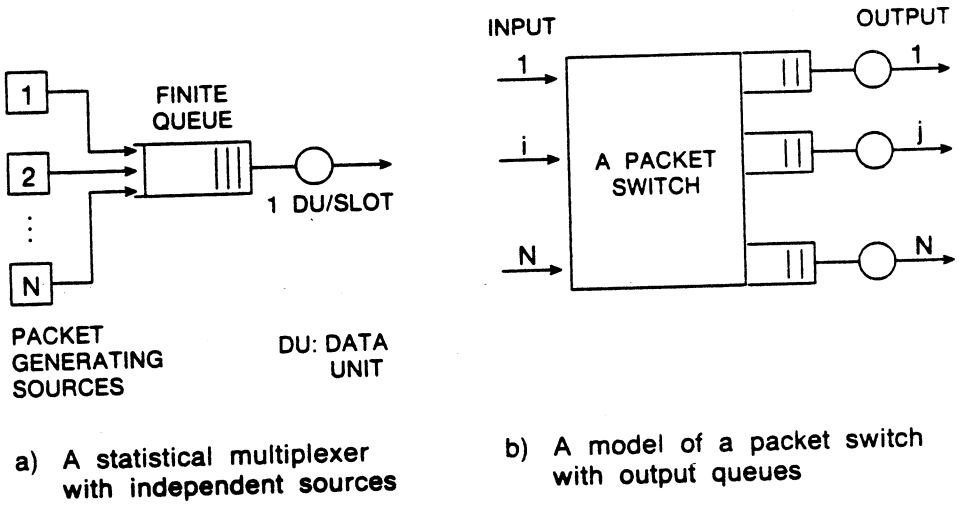

Fig. 1. Modeling examples.

(a) A statistical multiplexer with independent sources.

(b) A model of a packet switch with output queues.

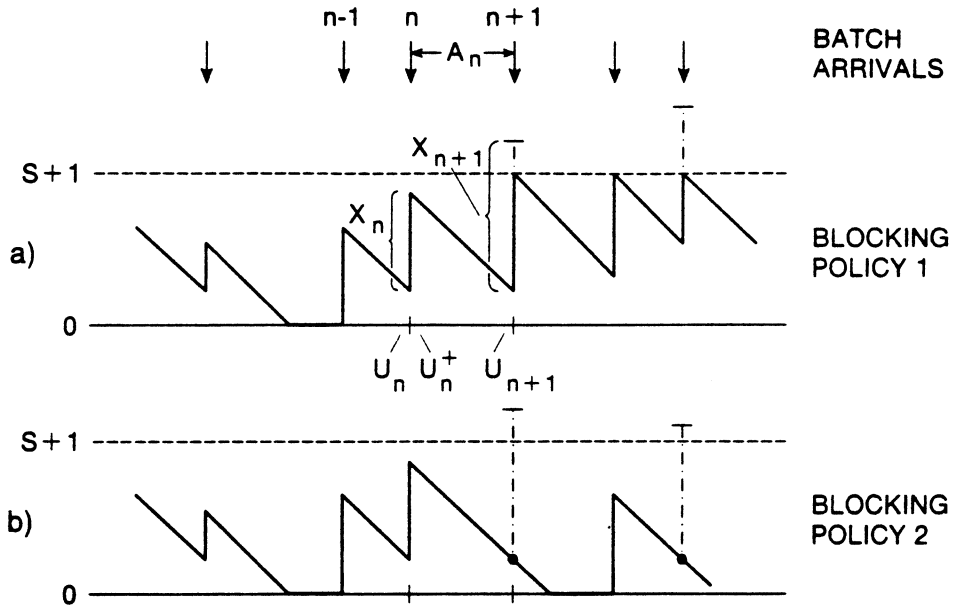

Fig. 2. Sample path of the state process.

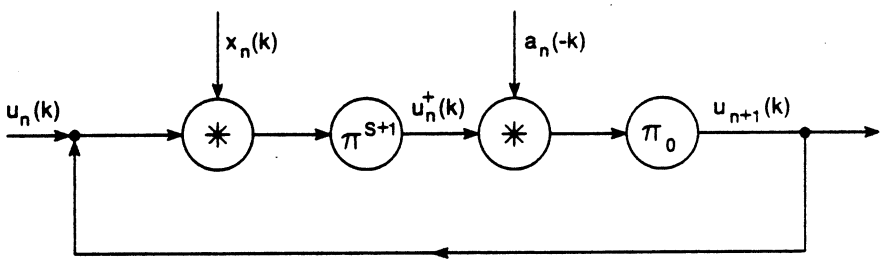

Fig. 3. Computational diagram of state probabilities. Blocking policy 1.

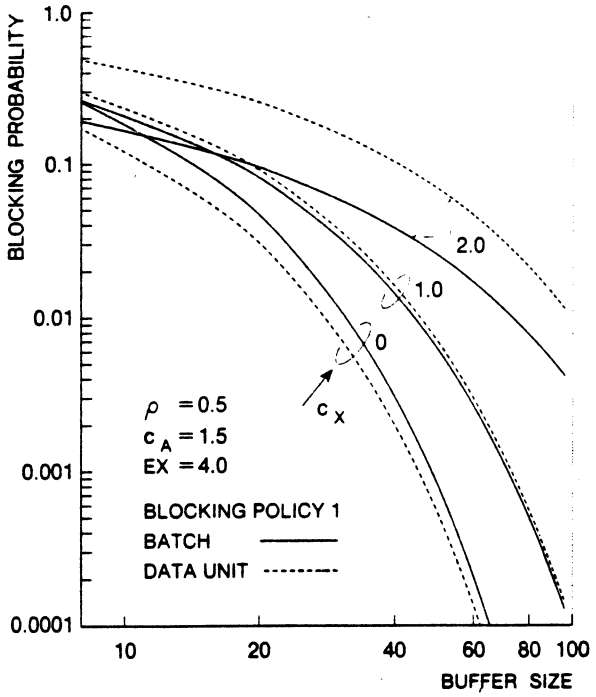

Fig. 4. Blocking probabilities vs buffer size: impact of batch statistics on the first blocking policy (BP1).

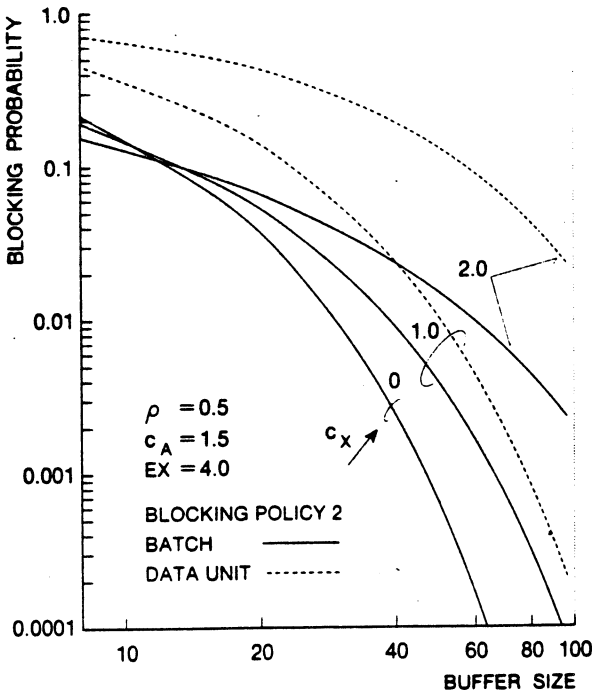

Fig. 5. Blocking probabilities vs buffer size: impact of batch statistics on the second blocking policy (BP2).

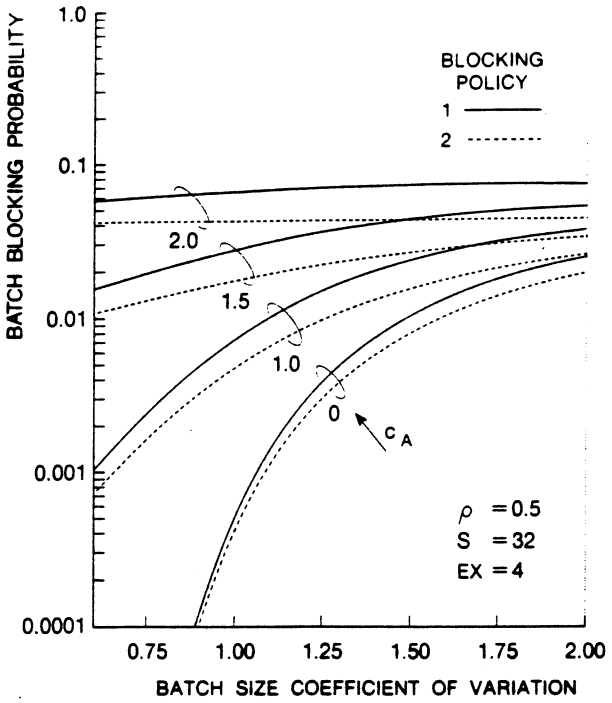

Fig. 6. Batch blocking vs batch variation: comparison of blocking policies.

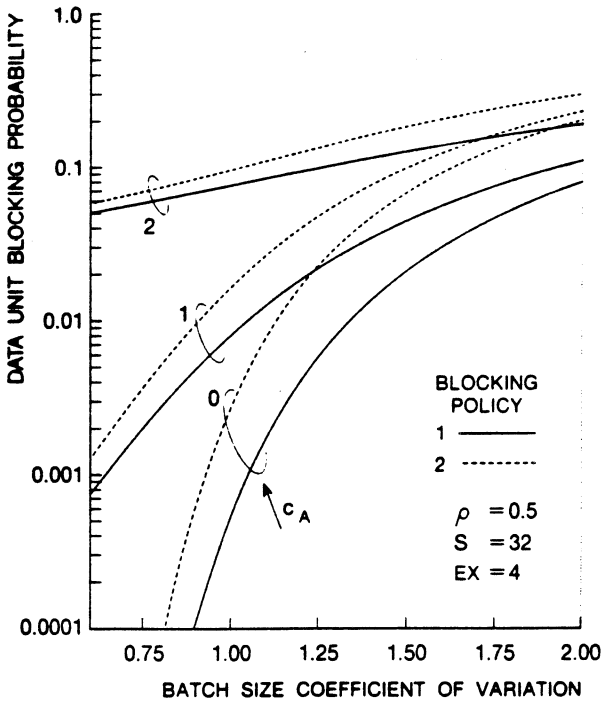

Fig. 7. Data-unit blocking vs batch variation: comparison of blocking policies.

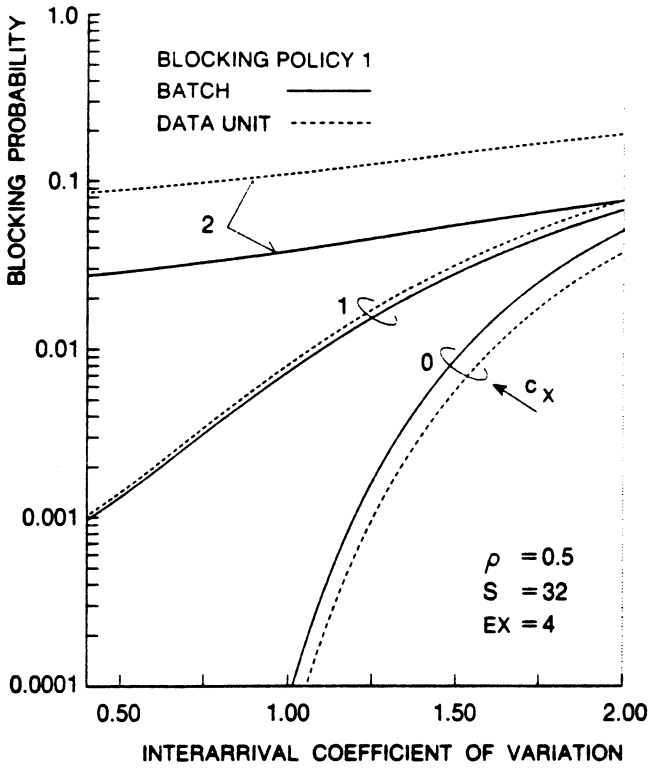

Fig. 8. Blocking probabilities vs interarrival variation: comparison of blocking policies.

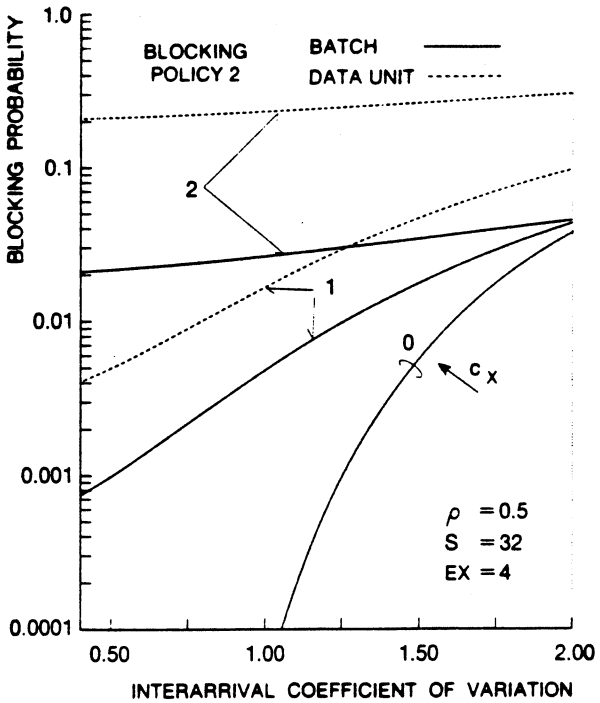

Fig. 9. Blocking probabilities vs interarrival variation: comparison of blocking policies.

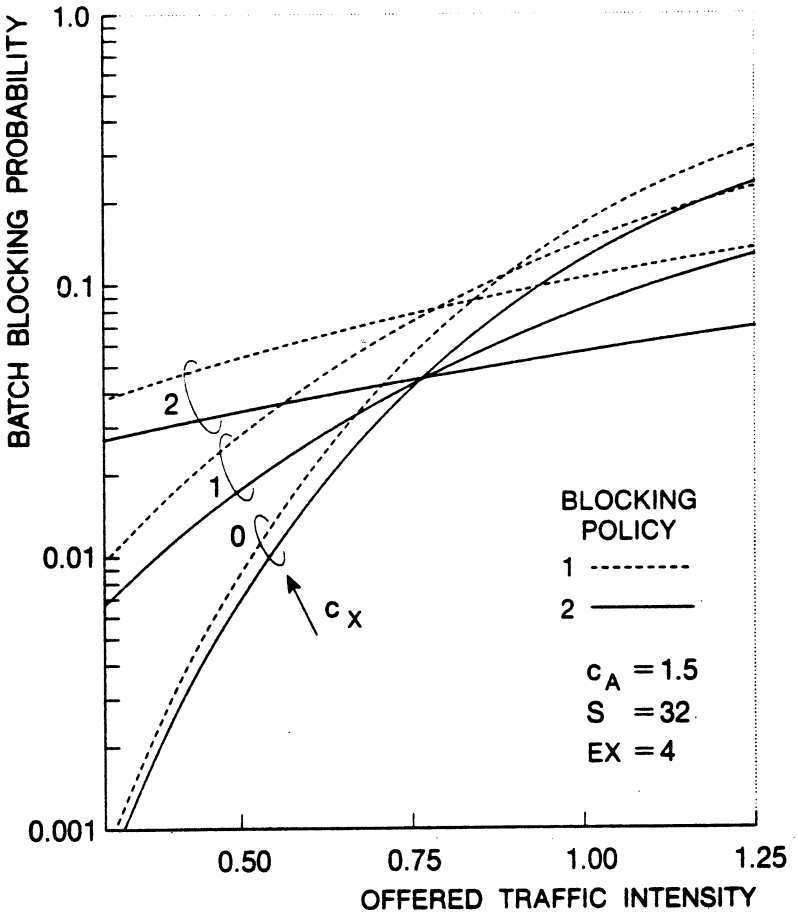

Fig. 10. Batch blocking vs offered traffic intensity: impact of batch variation.