

RZ 1632 (#59035) 10/13/87  
Communications 20 pages


# Research Report

Analysis of Discrete-Time  $G X / D / 1-S$  Queueing  
System with Applications in Packet-Switching Systems

Phuoc Tran-Gia and Hamid Ahmadi

IBM Research Division  
Zurich Research Laboratory  
8803 Rueschlikon  
Switzerland

LIMITED DISTRIBUTION NOTICE: This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties).

 Research Division  
Almaden • Yorktown • Zurich

**ANALYSIS OF A DISCRETE-TIME  $G^{[X]}/D/1 - S$  QUEUEING SYSTEM WITH APPLICATIONS IN PACKET-SWITCHING SYSTEMS**

Phuoc Tran-Gia and Hamid Ahmadi

IBM Research, 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.

August 18, 1987

## 1. 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 inter-arrival 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 (mini-packet). 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 inter-arrival times (in units of slots) having general distribution. The generality of our model is such that the batch size and the batch inter-arrival 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 has non-Poissonian statistics. For example, as shown in Fig. 1a, this model may represent a statistical multiplexer fed by a number of independent sources [1,9,10,14] which oscillate between "on" and "off" states and emit data-units at some fixed rate during an "on" state, or a switch fabric (Fig. 1b) which operates in a packet-switched mode and has buffers at each output. Examples of the latter case are a crossbar switch with output FIFOs [12], or the output stage of the Knockout switch [8]. Another example is a statistical multiplexer, like in [5-7], 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.

Because of the limited queue size, 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 [5,6], a transmission buffer has been modeled as a finite queue with batch Poisson arrivals, geometric batch-size distribution, and a constant synchronous output. In [7], 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 [15]. A number of theoretical studies for both infinite [3,4,11] and finite [2] queues with batch arrival has appeared with different degrees of complexity. However, it is not easily seen how the results in [2] may be used in practical applications to assess the buffer-size requirements. In another paper [13], 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. An algorithmic technique was also proposed in [14], which analyzes a discrete-time D/D/1 queue with both infinite and finite queue size with multiple arrivals per unit time (slot) by using a bivariate Markov chain which describes the whole system. The major contribution of our paper is: 1) to extend the finite queueing model presented in [4,6,14] by incorporating general inter-arrival 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 2, we present the basic description of the  $G^{[X]}/D/1 - S$  queueing system, and present the analysis. In section 3, 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 4, we illustrate the application of this model via two examples. Finally, the conclusion is given in section 5.

## 2. 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, the queueing system presented can be solved using standard methods operating in continuous time domain. In this case, additional simplifying assumptions have to be made, since we have several non-memoryless processes, for which a Markov chain cannot be imbedded [14]. 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 pi-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]. Apart from the case of finite-state space (limited by  $S + 1$ , where  $S$  is the maximum queue size) assumed in the following, the convolution can be segmented, since 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 here that the algorithms in discrete-time domain developed here are stable for a wide range of system parameters.

### 2.1 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  $\Delta t$ , which is the service or

transmission time of a single data-unit. As a consequence, samples of these random variables are integer multiples of  $\Delta t$ .

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

$$r(k) = \Pr(R = k), -\infty < k < +\infty \quad \text{distribution of } R$$

$$R(k) = \sum_{i=-\infty}^k r(i), -\infty < k < +\infty \quad \text{distribution function of } R$$

$$ER, c_R \quad \text{mean and coefficient of variation of } R$$

Further, the following notation is employed:

$S$  queue capacity in data-units.

$A_n$  random variable for the generalized inter-arrival 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. [16]).

$X_n$  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.

## 2.2 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 variable (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 consecutively the blocking probabilities of batches and data-units.

### a) 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,

$$U_n^+ = \min(U_n + X_n, S + 1) \quad (1)$$

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



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

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

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

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

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

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

and the  $\star$ -symbol denotes the discrete convolution operation

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

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 inter-arrival intervals with random variable  $A$ , and batch sizes with random variable  $X$ , which are now assumed to be time-independent, eqs. (3) and (4) deliver an iterative algorithm to determine the equilibrium state probabilities

$$u(k) = \lim_{n \rightarrow \infty} u_n(k). \quad (8)$$

*b) 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:

$$U_n^+ = \begin{cases} U_n + X_n & U_n + X_n \leq S + 1 \\ U_n & U_n + X_n > S + 1 \end{cases} \quad (9)$$

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

Distribution of these random variables are given as

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

$$u_{n+1}(k) = \pi_0(u_n^+(k) \star a_n(-k)), \quad k = 0, 1, \dots, S+1. \quad (12)$$

Since the functional relationship between  $U_{n+1}$  and  $U_n^+$  is the same for both blocking policies as given in eqs. (2) and (10), eqs. (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 eqs. (11) and (12). Further steps are analogous to the case of BP1.

### 2.3 Blocking Probabilities

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

#### a) 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.

It is obvious that

$$P_B(k) = \sum_{j=S+2-k}^{\infty} x(j), \quad k = 0, 1, \dots, S+1. \quad (13)$$

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

$$\begin{aligned} P_B &= \sum_{k=0}^{S+1} u(k) \sum_{j=S+2-k}^{\infty} x(j) \\ &= \sum_{k=S+2}^{\infty} (u(k) \star x(k)) \end{aligned} \quad (14)$$

b) *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.

1) *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  $X=k$  observed upon arrival.

The probability for the test data-unit to be in a batch of size  $i$  is  $i \cdot x(i)/EX$ . For a batch of size  $i$ , blocking will occur for  $i+k > S+1$ , where a fraction of  $(k+i-(S+1))/i$  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

$$\begin{aligned}
 P_{DU}(k) &= \sum_{i=S-k+2}^{\infty} \frac{k+i-S-1}{i} \cdot \frac{i \cdot x(i)}{EX} \\
 &= \frac{1}{EX} \sum_{i=S-k+2}^{\infty} (k+i-S-1) \cdot x(i), \quad k = 0, \dots, S+1.
 \end{aligned} \tag{15}$$

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

$$\begin{aligned}
P_{DU} &= \sum_{k=0}^{S+1} u(k)P_{DU}(k) \\
&= \frac{1}{EX} \sum_{k=0}^{S+1} u(k) \sum_{i=S+2-k}^{\infty} (k+i-S-1) \cdot x(i) .
\end{aligned} \tag{16}$$

## 2) 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

$$\begin{aligned}
P_{DU}(k) &= \sum_{i=S-k+2}^{\infty} 1 \cdot \frac{i \cdot x(i)}{EX} \\
&= \frac{1}{EX} \sum_{i=S-k+2}^{\infty} i \cdot x(i), \quad k = 0, \dots, S+1.
\end{aligned} \tag{17}$$

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

$$P_{DU} = \frac{1}{EX} \sum_{k=0}^{S+1} u(k) \sum_{i=S+2-k}^{\infty} i \cdot x(i) . \tag{18}$$

## 3. 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 here the inter-arrival 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

$$r(k) = \binom{y+k-1}{k} p^y (1-p)^k, \quad 0 \leq p < 1, \quad y \text{ real}, \quad (19)$$

where

$$p = \frac{1}{ER \cdot c_R}, \quad y = \frac{ER}{ER \cdot c_R - 1}, \quad ER \cdot c_R > 1.$$

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

$$\rho = \frac{EX}{EA}. \quad (20)$$

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 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 = 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 inter-arrival 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 inter-arrival 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.

#### 4. 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. Our intent is to show how this model can be used to solve these two cases, by selecting the inter-arrival and batch-size distribution appropriately.

The first example is from the paper by Morris [14], which models a packet-switch node consisting of  $N$  independent binary sources feeding a single-server queue or multiplexer (like in Fig. 1a). Each source is in either an "off" state during which it is not transmitting packets, or in an "on" state, during which it transmits packets (data-units) at a constant rate of one packet per unit time. Since more than one source can be in the "on" state, multiple packets can arrive at the queue per service time. Therefore, the arrival process can be modeled as a batch arrival with constant inter-arrival time. Using the same assumptions and notations as in [14], each of the  $N$  sources is represented by a two-state discrete-time Markov chain, with transition probabilities  $t_{01}$  (transition "off" to "on" state) and  $t_{10}$  (transition from "on" to "off"), the batch-size distribution is given by a binomial distribution



$$x(k) = \binom{N}{k} Q^k (1-Q)^{N-k}, \quad (21)$$

where

$$Q = \frac{t_{01}}{t_{01} + t_{10}}.$$

In this view, the behavior of the packets generated by the  $N$  independent sources is represented by eq. (21). The remaining solution is to solve the system behavior based on the method presented in this paper. This way all the results in [14] can easily be obtained. In the Morris' model, the system state is presented by a bivariate discrete-time Markov chain for which the system of state equations has been solved for the queue with finite waiting places. It should be noted that for this application, only the blocking probability of data-units under policy 1 is appropriate.

The second example is taken from the paper by Karol et al. [12], in which they model a crossbar  $N \times N$  space division switch with output FIFOs (like the one shown in Fig. 1b). 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 with binomial distribution exactly as in eq. (21), with  $Q = p/N$ . Using our model, one can easily obtain the probability of packet loss for this system. It should be noted that the authors in [12] model the buffer state as a discrete-time Markov chain, and propose a recursive algorithm which numerically provides the steady-state probabilities. Because of the specific assumptions made for this problem (i.e., constant inter-arrival time, fixed-length packets, and a single-packet departure during an inter-arrival time), it is possible to solve the

steady-state probabilities numerically by a recursive algorithm directly from the balance equations. However, if any of these assumptions is relaxed, it is not in general possible to solve for the state probabilities recursively other than by either matrix inversion or by numerical inversion of z-transform. Both of these methods are computationally very complex. The method proposed in our paper can appropriately model this switch under variable length packets and a general inter-arrival time distribution. Once the arrival process has been correctly modeled, the system performance can be analyzed in a straightforward manner.

## 5. 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 statistical multiplexer, and a modeling of a space-division packet switch with output FIFOs.

## REFERENCES

1. Anick, C., D. Mitra and M. M. Sondhi, "Stochastic Theory of a Data Handling System with Multiple Sources," *BSTJ*, Vol.61, No. 8, pp. 1871-1894, October 1982.
2. Bagchi, T. P. and J. G. C. Templeton, "Finite Waiting Bulk Queueing Systems," *J. Eng. Math.*, Vol. 7, pp. 313-317, 1973.

3. Bhat, U. N., "Imbedded Markov Chain Analysis of Single Server Bulk Queues," J. Aust. Math. Soc., Vol. 4, pp. 244-263, 1964.
4. Burke, P. J. "Delays in Single-Server Queues with Batch Input," Operation Research, Vol. 23, pp. 830-833, 1975.
5. Chu, W. W., "Buffer Behavior for Batch Poisson Arrivals and Multiple Synchronous Constant Outputs," IEEE Trans. Commun., Vol. COM-19, pp. 30-534, June 1971.
6. Chu, W. W. and L. C. Liang, "Buffer Behavior for Mixed Input Traffic and Single Constant Outputs Rate," IEEE Trans. Commun., Vol. COM-20, pp. 230-235, April 1972.
7. Chu, W. W. and A. G. Konheim, "On the Analysis and Modeling of a Class of Computer Communication Systems," IEEE Trans. Commun., Vol. COM-20, No. 3, pp. 645-660, June 1972.
8. Eng, K. Y., M. G. Hluchyj and Y. S. Yeh, "A Knockout Switch for Variable-Length Packets," Proc. ICC, Seattle, Washington, pp. 22.6.1-22.6.6, 1987.
9. Gopinath, B. and A. J. Morrison, "Discrete-time Single Server Queues with Correlated Inputs," BSTJ, Vol. 56, No. 9, pp. 1743-1768, November 1977.
10. Halfin, S., "The Blocking of Data in Buffers with Variable Input and Output Rate," in Performance of Computer-Communication Systems, edited by H. Rudin and W. Bux, North-Holland, Amsterdam, 1984, pp. 307-319.
11. Kabak, I. W., "Blocking and Delays in  $M^{[X]}/M/c$  Bulk Arrival Queueing Systems," Management Science, Vol. 17, No. 1, pp.112-115, 1970.
12. Karol, M. J., M. G. Hluchyj and S. P. Morgan, "Input vs. Output Queueing on a Space-Division Packet Switch," Proc. Globecom, Houston, Texas, pp-19.4.1-19.4.7, 1986.
13. Manfield, D. R. and P. Tran-Gia, "Analysis of a Finite Storage System with Batch Input Arising out of Message Packetization," IEEE Trans. Commun., Vol. COM-30, No. 3, pp. 456-462, March 1982.
14. Morris, R. J. T, "An Algorithmic Technique for a Class of Queueing Models with Packet Switching Applications," Proc. ICC, Denver, Colorado, pp. 41.2.1-41.2.1, 1981.
15. Schwaertzel, H. G., "Serving Strategies of Batch Arrivals in Common Control Switching Systems," Proc. 7th Int. Telecomm. Conference, Stockholm, Sweden, 1973.
16. Tran-Gia, P., "Discrete Time Analysis for the Interdeparture Distribution of a GI/G/1 Queue," in Teletraffic Analysis and Computer Performance Evaluation, edited by O. J. Boxma, J. W. Cohen and H. C. Tijms, North Holland, Amsterdam, 1986, pp. 341-357.

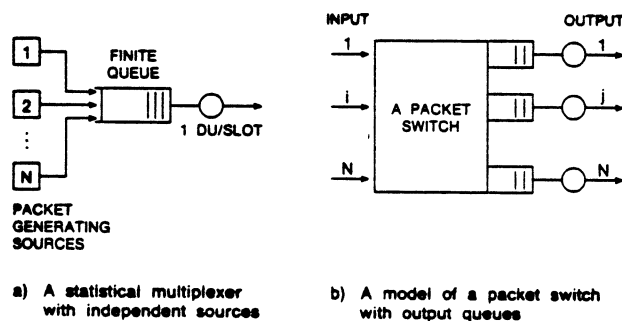


Fig. 1. Modeling examples.

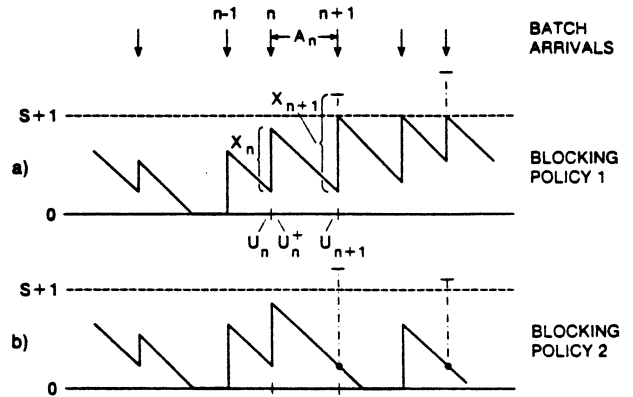


Fig. 2. Snapshot of the state process.

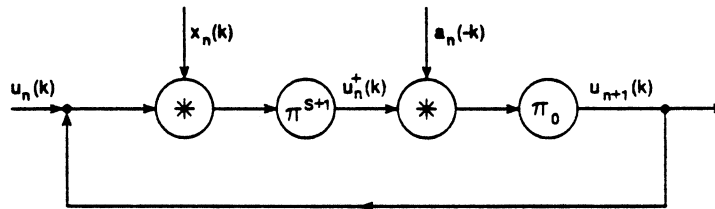


Fig. 3. Computational diagram of state probabilities. Blocking policy 1.

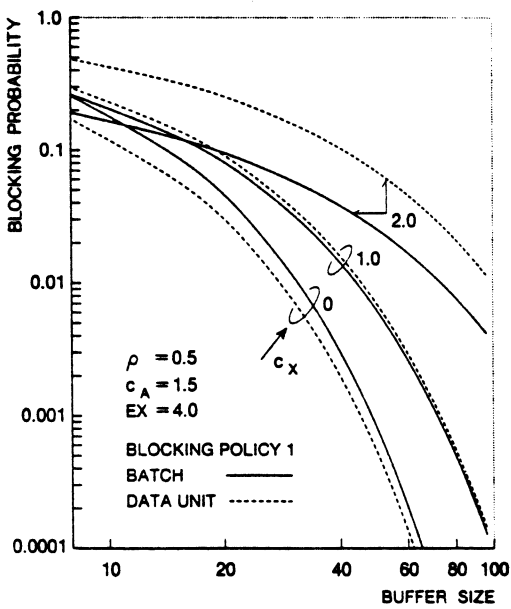


Fig. 4. Blocking probabilities vs buffer size: impact of batch statistics on the first blocking policy (BP1).

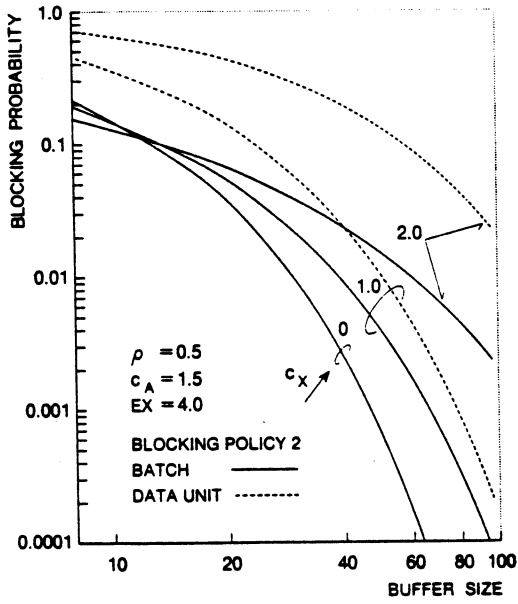


Fig. 5. Blocking probabilities vs buffer size: impact of batch statistics on the second blocking policy (BP2).

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

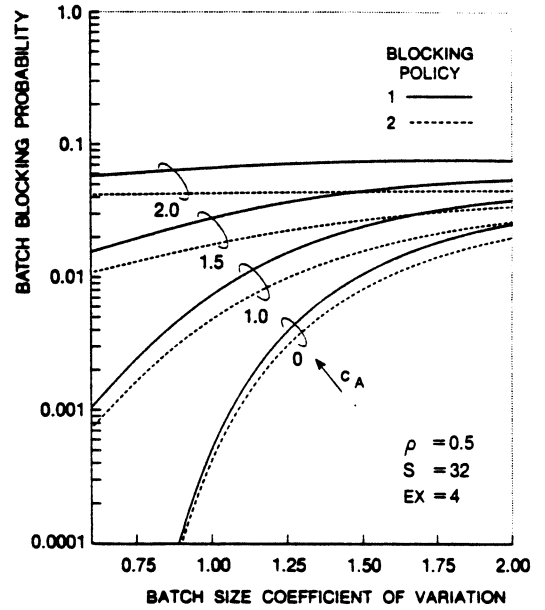
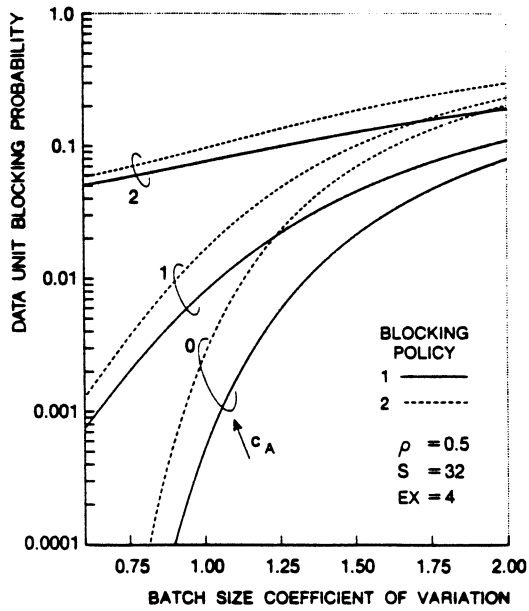


Fig. 7. Data-unit blocking vs batch variation: comparison of blocking policies.

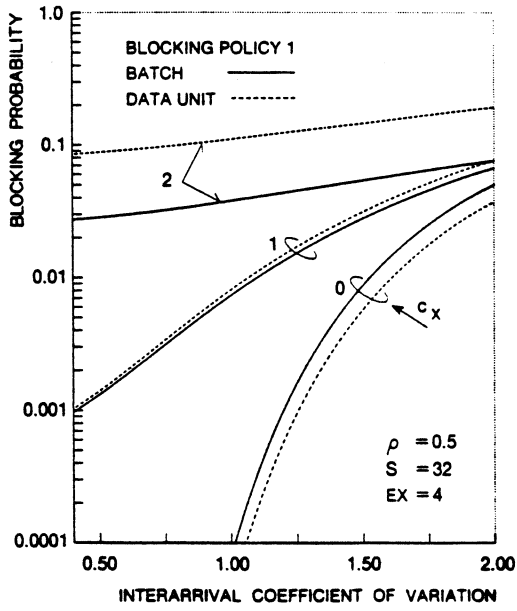


Fig. 9. Blocking probabilities vs inter-arrival variation: comparison of blocking policies.

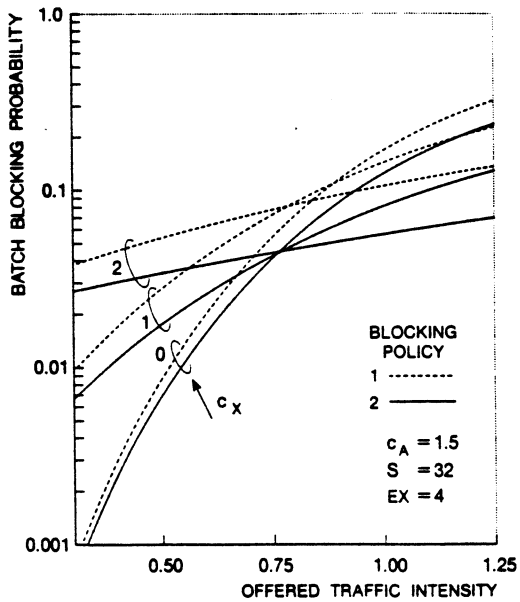


Fig. 8. Blocking probabilities vs inter-arrival variation: comparison of blocking policies.

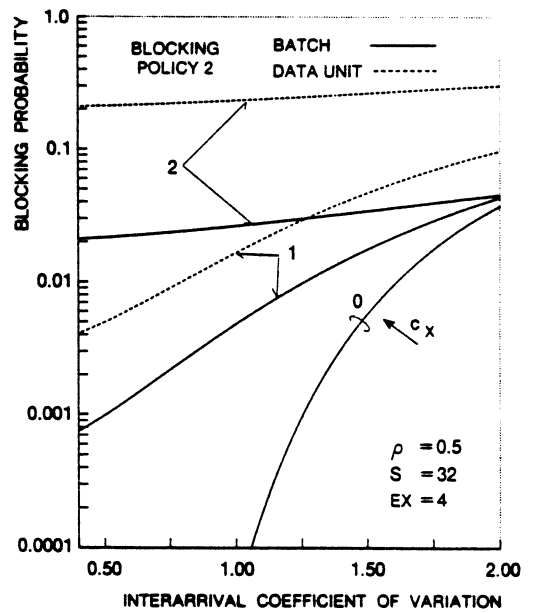


Fig. 10. Batch blocking vs offered traffic intensity: impact of batch variation.