Contribution to the 8th International Symposium on Computer Performance Modelling, Measurement and Evaluation, Nov.1981, Amsterdam

QUEUEING ANALYSIS OF SCHEDULED COMMUNICATIONS PHASES IN DISTRIBUTED PROCESSING SYSTEMS

D.R. Manfield*

Node Traffic Studies Dept., Bell-Northern Research
P.O. Box 3511, Station C, Ottawa, Canada

P. Tran-Gia

Dept. of Communications, University of Siegen
Postfach 210209, 5900 Siegen 21, West-Germany

In a distributed real-time processing system, the actions of the various processors are co-ordinated by the passing of messages. Events generated in one processor destined to be handled in another are formed into messages which must wait until the next scheduled I/O or communication phase, at which time they are transferred to the event queue of the destination processor. The performance analysis of blocking and delays for the inter-processor message traffic is analysed by means of a queueing model with batch input, finite storage and overhead. Results are presented chiefly for the commonly occurring case of a clocked communications schedule, although much of the analysis is more generally applicable.

1. INTRODUCTION

In many modern real-time processing systems it is often the case that all the intelligence is not concentrated in one large central processor. Rather there is a tendency, for reasons of reliability and flexibility, towards the development of distributed systems composed of a number of individual processors operating in modes of functional or load sharing or a combination of these. In a stored program controlled (SPC) telephone exchange, for example, one possibility is that there is a number of small peripheral processors (pre-processors) concerned with the routine and time-consuming tasks of event detection and signalling, while the higher level call control is performed by a central processor. Or, instead of such a hierarchical system, an exchange may be fully distributed, composed of processors of equal ranking but with different functions, for example processors associated with blocks of incoming and outgoing trunks (or subscriber circuits), and individual processors concerned with translation or common channel signalling.

* This work was done while Dr. Manfield was at the University of Siegen, Dept. of Communications, under a fellowship from the Alexander von Humboldt Foundation.

For the overall real-time performance of distributed systems, which in an SPC exchange could mean the call set-up time, the response time of the interprocessor communication is a critical factor. To meet real-time constraints it is common to schedule I/O phases at clocked instants by means of high priority interrupts. The events generated in one processor which need to be handled in another are formed into messages which are placed into an output buffer to await the interrupt beginning the next I/O phase. When this occurs the messages are transferred to the input buffer of the appropriate destination processor, under the control of a simple transmission protocol, where they wait for processing.

In order to investigate some aspects of the performance of a distributed processing system, a queueing model is built up for the traffic analysis of the inter-processor communication. We take into account the overhead incurred by I/O phases, the blocking caused by finite processor input buffers, and the feature of batch job arrivals caused by the scheduling of I/O phases. There are a number of works [1-6] which deal with related queueing problems, mostly queues with clocked, batch arrivals in the context of switching system control structures. [1,2,4] consider queueing models with infinite waiting space. In [6] finite systems are treated but with single arrivals and clocked, batch service. In [5] is an approximate analysis by a phase method of a finite system with clocked, batch arrivals and overhead. Overhead is also treated in [6] but in the context of a priority queueing model.

2. MODELLING

In this section the queueing model for event sampling, interprocessor event-message transfer and processing is developed. Although we now proceed to do this with explanation based on one particular example, it should be clear that the model is useful for a wide range of different applications in distributed or multi-level control processing systems.

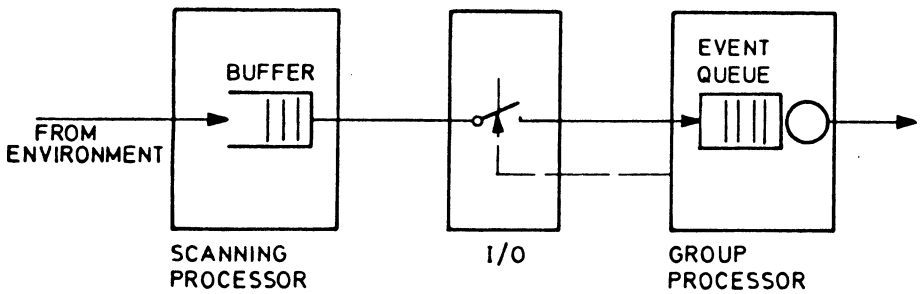

Fig.1 : Example for an event sampling mechanism

Consider the system depicted in Fig. 1, a typical scenario for peripheral processing in an SPC switching system. At the hardware scanning level, physical events must pass a persistency check to be registered as valid telephonic events. These events are the input for state transition actions at the call processing level. They will be transferred to the event queue in the group processor according to some I/O scheduling scheme as shown in Fig. 2. The I/O phases

represent overhead to the group processor, since it can not do any event processing during these times.

The details of the traffic model are as follows. The process of registered event arrivals is assumed to be Poisson, considering that these events arise from a large number of different sources, either subscribers or devices. The times between the initiation of scheduled transfers of groups of events are assumed to be random variables with a common distribution function. The lengths of successive I/O (i. e., overhead) periods are also generally assumed to be random variables with a common distribution. Although much of the following analysis will be for the general case, our chief consideration will be for the case of a clocked communication schedule, with overhead phases of fixed length.

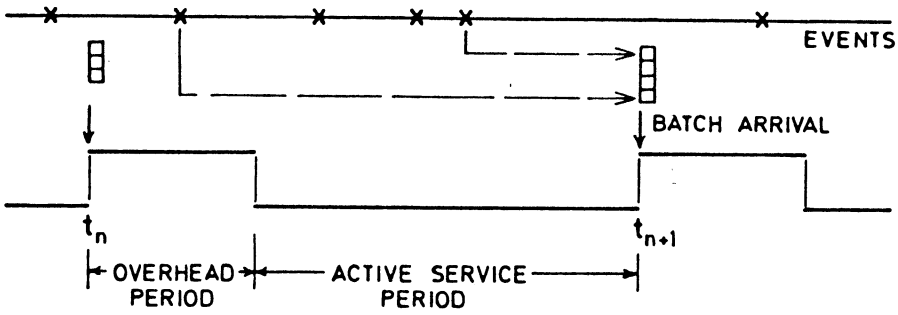

Fig.2 : Event collection and processing

The size of the group of events transmitted to the destination processor during an I/O phase will clearly depend on the length of the sampling period and on the intensity of the underlying process of event arrivals. Although the event messages in the primary buffer will be transferred individually during an I/O phase, this is exactly equivalent from the point of view of modelling to an instantaneous transfer of the whole group batch-wise at the scheduled I/O instant. Every event transfer activity which is controlled by the processor is usually performed by the same I/O task (the *overhead phase*) and has approximately the same run-time during which the processor is not available for event processing. The intervals between overhead periods are called *active service periods*, during which the server is available but not necessarily busy.

The capacity of the event queue is considered to be finite, and when an arriving batch is too large for the number of free queue positions, the free positions are filled and the remaining events are assumed lost. While this is not very realistic in real systems (e.g., the transmission protocol would simply stop the transfer of messages when the event queue was full), it is a useful starting assumption and also allows the dimensioning of the event queue size.

Taking into account the mixture of types of arriving events, each type requiring a different program with different resident times in the processor, the service times for events are assumed to be exponentially distributed [4,5].

Gathering together all this information, the basic component of the model is seen to be that of a $GI^{[x]}/M/1$-s queue with overhead, and s waiting places. The waiting time for customers (events) in this queueing system is made up of two components. Firstly an event must wait in the primary buffer from the time it is generated until the next scheduled I/O phase. This is called the pre-queueing delay. Secondly an event suffers a queueing delay beginning at the time it is transferred.

The sampling period consists of two parts, the overhead period and the active service period. When these two are both generally distributed it is possible an interrupt for the next I/O phase to occur before the current I/O phase has ended. To overcome this problem we will asume that by a method of moment subtraction that the overhead and active service periods have independent distributions (after the nature of an alternating renewal process [10]) whose convolution yields the required scheduling distribution, or at least its moments. It is important to note that in the case of major interest, namely the clocked schedule with constant overhead, there is no approximation involved in this assumption.

3. ANALYSIS

The basic component of the model for inter-processor communication is the $GI^{[x]}/M/1$-s queue with overhead. In subsections 3.1 to 3.5 performance measures for this queueing system are developed, and in subsection 3.6 the prequeueing delay is investigated. The following terminology will be used:

λ	parameter of Poisson process of customer arrivals
μ	service rate
$\rho = \lambda/\mu$	offered traffic intensity
s	number of waiting places in queue
$N = s+1$	system size
$F_1(t)$	distribution function (d.f.) of overhead period
$F_2(t)$	d.f. of active service period (period when server is available)
g_n	Pr{batch of n customers offered to queue at scheduled interrupt}

$$MBS = \sum_{n=0}^{\infty} n g_n \quad (\textit{m}\text{ean } \textit{b}\text{atch } \textit{s}\text{ize}).$$

3.1 The Imbedded Markov Chain

The first step in the analysis of the queueing part of the model is by means of the well-known technique of the imbedded Markov chain (MC) [7], at time instants just prior to the arrival points. Let t_n be the time of the n^{th} batch arrival and let η_n be the number of customers in the system at time t_n^- (i.e., just prior to the arrival point), the system being assumed to be in steady state. The intervals $(t_{n+1}-t_n)$ are then assumed to be independent and identically distributed. The transition probabilities of the imbedded MC are defined from

$$p_{jk} = \Pr\left\{\eta_{n+1} = k \,\middle|\, \eta_n = j\right\}$$

The state probabilities of the MC are

$$P_k = \Pr\{\eta_n = k\}$$

which are calculated from the following relations

$$P_k = \sum_{j=0}^{N} p_{jk} P_j \ , \qquad \sum_{k=0}^{N} P_k = 1 \qquad \ldots(3.1)$$

By conditioning on the size of an arriving batch, the transition probabilities are

$$p_{j0} = \sum_{i=0}^{N-j} g_i \left(1 - \sum_{r=0}^{i+j-1} d_r\right) + \left(1 - \sum_{r=0}^{N-1} d_r\right) \sum_{i=N-j+1}^{\infty} g_i \qquad (3.2a)$$

$$p_{jk} = \sum_{i=0}^{N-j} g_i \, d_{i+j-k} + d_{N-k} \sum_{i=N-j+1}^{\infty} g_i \ ; (0<k\leq N) \qquad (3.2b)$$

where we have used

$$d_m = \begin{cases} \int_0^{\infty} e^{-\mu x} \frac{(\mu x)^m}{m!} dF_2(x) & ; \ m \geq 0 \\ 0 & ; \ m < 0 \end{cases} \qquad \ldots(3.3)$$

This is just the probability of m departures during an active service period, and in the case of the clocked schedule with active service period of fixed length T, we have simply

$$d_m = e^{-\mu T} \frac{(\mu T)^m}{m!} \quad ; \ m \geq 0 \qquad \ldots(3.4)$$

Using eqns. (3.2) and (3.3), eqn. (3.1) is solved for the state probabilities of the imbedded MC by means of the numerical method of Gauss-Seidel iteration with over-relaxation.

3.2 Blocking Probabilities

A performance measure of primary importance is the blocking probability of the system. When a batch of customers arrives, and its length exceeds the number of free waiting places, then some of the customers are lost. The probability B_c of an arbitrary test customer is lost is determined as follows. The probability that this customer arrives in a batch of size i is just ig_i/MBS [8], each position in the batch being equally probable. By conditioning on the state found by the batch containing the test customer,

$$B_c = \frac{1}{\text{MBS}} \sum_{k=0}^{N} P_k \sum_{i=N-k+1}^{\infty} (i-N+k) \, g_i \qquad \ldots(3.5)$$

A batch of size zero is stipulated never to be blocked even when the system is full. Of secondary importance, it is possible to evaluate the probability B_b that an arbitrary batch is either partly or fully blocked. From straightforward argument,

$$B_b = \sum_{k=0}^{N} P_k \sum_{i=N-k+1}^{\infty} g_i \qquad \ldots(3.6)$$

At this point it is opportune to point out that the effective arrival rate of customers into the system, defined as ρ', is

$$\rho' = \rho\ (1-B_c) \qquad \ldots(3.7)$$

3.3 Arbitrary-Time State Probabilities

In order to deduce the mean waiting time in the queue for an arbitrary customer, it is easiest to use Little's law [9], for which it is necessary to know the queue length at an arbitrary point in time, as distinct from the queue length at the points of the imbedded MC. An outside observer chooses an arbitrary time origin. He will observe the system either during an overhead phase or during an active service phase with probabilities π_1 and π_2 respectively. Denoting the lengths of the overhead and active service phases by X_1 and X_2, from the theory of alternating renewal processes

$$\pi_1 = E(X_1)/(E(X_1)+E(X_2))$$

$$\pi_2 = 1-\pi_1 \qquad \ldots(3.8)$$

The two observation possibilities are labelled respectively case I and case II. From the arbitrary time origin the observer looks back to the last point of the imbedded process and notes the state changes which have occured since then. In case I, the state must be the same as immediately after the last arrival, since no servicing occurs during the overhead phase. In case II, it is necessary however to reckon the numberof departures which occurred since the end of the previous overhead phase. The time since the end of the previous overhead phase is a random variable which is just the backwards recurrence time for the active service period and hence has the probability density function (p.d.f.) $(1-F_2(t))/E(X_2)$ [10].

Define the arbitrary time state probabilities

$$P_k^* = \Pr\ \{\text{ outside observer sees } k \text{ customers in system}\}$$

By consideration of the two observation cases I and II, the $\{P_k^*\}$ are found from the following relations:

$$P_o^* = \pi_1 g_o P_o + \pi_2 \sum_{j=0}^{N} \left[\sum_{i=0}^{N-j} g_i \sum_{r=i+j}^{\infty} d_r^* + \sum_{r=N}^{\infty} d_r^* \sum_{i=N-j+1}^{\infty} g_i \right] P_j \qquad (3.9a)$$

$$P_k^* = \pi_1 \sum_{j=0}^{k} g_{k-j} P_j + \pi_2 \sum_{j=0}^{N} \left[\sum_{i=0}^{N-j} d_{i+j-k}^* g_i + d_{N-k}^* \sum_{i=N-j+1}^{\infty} g_i \right] P_j \qquad (0 < k < N) \qquad \ldots(3.9b)$$

$$P_N^* = \pi_1 \sum_{j=0}^{N} \sum_{i=N-j}^{\infty} g_i P_j + \pi_2 \sum_{j=0}^{N} \left[d_o^* g_{N-j} + d_o^* \sum_{i=N-j+1}^{\infty} g_i \right] P_j \qquad \ldots(3.9c)$$

where

$$d_m^* = \begin{cases} \dfrac{1}{E(X_2)} \displaystyle\int_0^{\infty} e^{-\mu x} \frac{(\mu x)^m}{m!} (1 - F_2(x))\, dx & ; \; m \geq 0 \\ 0 & ; \; m < 0 \end{cases} \qquad \ldots(3.10)$$

From this formulation it is possible to see directly that for the special case of Markov inter-arrival periods and zero overhead that $d_m = d_m^*$ and $P_k = P_k^*$.

By evaluating the integral in (3.10) it can be seen how to determine the $\{d_m^*\}$ from the $\{d_m\}$. Namely

$$d_m^* = \frac{1}{\mu E(X_2)} \sum_{r=m+1}^{\infty} d_r$$

and furthermore for use in (3.9) it is possible to show that

$$\sum_{m=n}^{\infty} d_m^* = \frac{1}{\mu E(X_2)} \left[\mu E(X_2) - \sum_{r=0}^{n} r\, d_r - n \sum_{r=n+1}^{\infty} d_r \right] \qquad (n > 0)$$

3.4 Mean Waiting Time in Queue

The use of Little's law [9] in queues with overhead involves a little care since one must keep in mind that during an overhead period it is possible for a customer to be waiting in the server itself. Thus the mean waiting time for non-blocked customers W_q is calculated indirectly from the mean sojourn time in the system W as follows :

$$W_q = W - \frac{1}{\mu}$$

where $W = \dfrac{L}{\rho' \mu}$ and $L = \displaystyle\sum_{k=0}^{N} k\, P_k^*$ $\qquad \ldots(3.11)$

3.5 Waiting Time Distribution Function

The distribution function of waiting time in a queue with overhead is in general not easy to determine, even for a system with Markov server. We proceed in two steps. First we calculate the complementary distribution function (c.d.f.) for the component of the waiting time generated by the workload of all customers ahead of test customer in the queue (that is not including waiting time from overhead) assuming FIFO queue discipline. Conditioned on the system state and the size of the batch containing the test customer, this will be an Erlang distribution. In the second step we include the waiting time induced by the overhead periods to obtain the full waiting time c.d.f. for the case of the clocked schedule with constant overhead.

The first component of waiting, that caused by the service times of customers ahead of the test customer, with c.d.f. $\bar{W}_q(t)$, can be derived by consideration of the principles in [8]. We find, conditioning on customer acceptance,

$$\bar{W}_q(t) = \frac{\rho}{\rho' MBS}\left[P_o\left\{\sum_{i=2}^{N} g_i \sum_{j=0}^{i-2} \bar{H}^{*(j+1)}(t) + \sum_{i=N+1}^{\infty} g_i \sum_{j=0}^{N-2} \bar{H}^{*(j+1)}(t)\right\}\right.$$
$$\left. + \sum_{k=1}^{N-1} P_k\left\{\sum_{i=1}^{N-k} g_i \sum_{j=0}^{i-1} \bar{H}^{*(j+k)}(t) + \sum_{i=N-k+1}^{\infty} g_i \sum_{j=0}^{N-k-1} \bar{H}^{*(j+k)}(t)\right\}\right] \qquad \ldots(3.12)$$

where

$$\bar{H}^{*(j)}(t) = \sum_{r=0}^{j-1} \frac{(\mu t)^r}{r!} e^{-\mu t} \quad ,$$

that is the c.d.f. of the j-fold convolution of the service time distribution function.

Now to include the effect of overhead on the waiting time it is necessary to take into account the number of overhead periods which occur before the test customer reaches service. We now restrict ourselves to the case of clocked schedule with constant overhead (formulae for the full waiting time c.d.f. in the general case have been derived but do not seem to be useful for explicit calculation and hence have not been included here). Let the test customer enter the system at time t=0 and have waiting time c.d.f. $\bar{W}_q^*(t)$, i.e., for the time until he first reaches the server. Let the clock period be T_c and the overhead time T_o.

For a particular time $t = t'$ we need to determine the probability that the test customer has not yet entered service. We condition on the number of overhead periods which have occurred before time t'. This number is uniquely determined by t', owing to the deterministic nature of the overhead and active service periods, and we denote this (unique) number by n(t') which may be integer or real depending on t'. Hence the c.d.f. we require is just

$$\bar{W}_q^*(t) = \bar{W}_q \left[t - n(t)T_o\right] \qquad \ldots(3.13)$$

To determine n(t) define $m(t) = \lfloor t/T_c \rfloor^-$ to be the largest integer less than t/T_c. Then

$$n(t) = \begin{cases} m(t) + 1 & ; \quad t - m(t)T_c \geq T_o \\ m(t) + \dfrac{t - m(t)T_c}{T_o} & ; \quad t - m(t)T_c < T_o \end{cases}$$

3.6 Pre-Queueing Delay

Again in this subsection we restrict ourselves to the case of clocked systems with constant overhead, although there seems to be no reason why the general case cannot also be so treated (albeit with more complicated algebra). The mean pre-queueing delay is the expectation of the time the test customer has to wait before he enters the queueing system at a clocked batch arrival point, as described in Section 2. For non-blocking systems this is simply shown to be one half of the clock period, but it is a little more difficult for systems with blocking. The problem is depicted in Fig. 3.

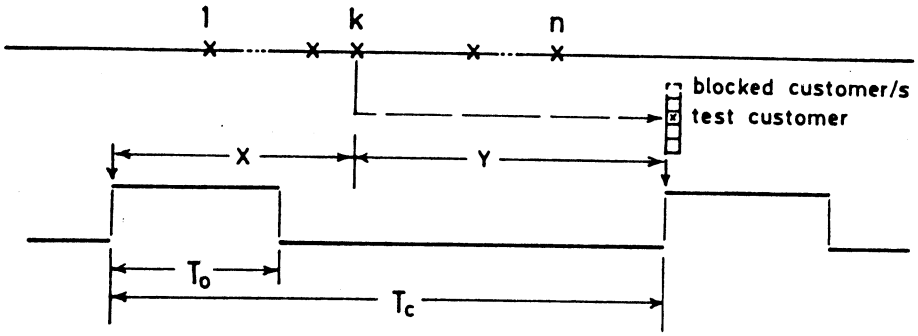

Fig.3 : The Pre-Queueing Delay

Assume that in the clock period n customers arrive and that the test customer occupies the k^{th} position. Let the time from the previous clock instant to the k^{th} arrival be denoted by the random variable X and the subsequent time to the next clock instant by the random variable Y (the pre-queueing delay). Hence

$$E[Y] = T_c - E[X] \qquad \ldots(3.14)$$

Let $E[Y|k,n]$ denote the conditional expectation of pre-queueing delay for the k^{th} customer given precisely n arrive in the clock period. For simplicity of notation denote the clock period by T instead of T_c. Then, from a renewal argument [10] considering the position in time of the k^{th} customer,

$$E[Y|k,n] = T - \frac{1}{g_n} \int_0^T x \frac{\rho(\rho x)^{k-1} e^{-\rho x}}{(k-1)!} \cdot \frac{e^{-\rho(T-x)} [\rho(T-x)]^{n-k}}{(n-k)!} dx$$

By integration by parts we arrive at

$$E[Y|k,n] = T - \frac{k(\rho T)^{n+1} e^{-\rho T}}{\rho g_n (n+1)!} \qquad \ldots(3.15)$$

Removing the conditioning on k and n, but conditioning on customer acceptance, and noting MBS = ρT, the mean pre-queueing delay (conditioned on customer acceptance) is given by:

$$E[Y] = \frac{\rho}{\rho'} \sum_{j=0}^{N-1} P_j \left[\sum_{n=1}^{N-j} \frac{g_n}{\rho T} \sum_{k=1}^{n} E[Y|k,n] + \sum_{n=N-j+1}^{\infty} \frac{g_n}{\rho T} \sum_{k=1}^{N-j} E[Y|k,n] \right]$$

After a little algebraic manipulation we have in full :

$$E[Y] = \frac{\rho}{\rho'} \sum_{j=0}^{N-1} P_j \left[\frac{1}{\rho} \sum_{n=1}^{N-j} n g_n - \frac{e^{-\rho T}}{2} \sum_{n=1}^{N-j} \frac{\rho^{n-1} T^n}{(n-1)!} \right.$$

$$\left. + \frac{(N-j)}{\rho} \sum_{n=N-j+1}^{\infty} g_n - \frac{e^{-\rho T}(N-j)(N-j+1)}{2} \sum_{n=N-j+1}^{\infty} \frac{\rho^{n-1} T^n}{(n+1)!} \right]$$

$\ldots(3.16)$

When the blocking is zero (3.16) reduces to T/2 as required.

4. RESULTS AND DISCUSSION

In this section numerical results are presented from the analysis of the clocked queueing system with overhead, and the effects of queue capacity, offered traffic level, and choice of clock period are discussed. In all the results, time is assumed to be normalised by the customer mean service time. Furthermore the (constant) length of the overhead period is chosen to be one time unit, that is the I/O routine needs one average customer service time.

4.1 System Performance

Fig. 4 shows the customer blocking probability as a function of the number of waiting places s, for different values of traffic intensity and length of clock period. These curves allow for a suitable dimensioning of the processor event queue. The crossover of the curves for fixed offered traffic and a suitable value of s indicate the existence of an optimum choice of clock period, a subject which is discussed in the next subsection. The blocking probabilities for customers and batches, and the mean queueing time are depicted in Figs. 5 and 6 respectively, as a function of offered

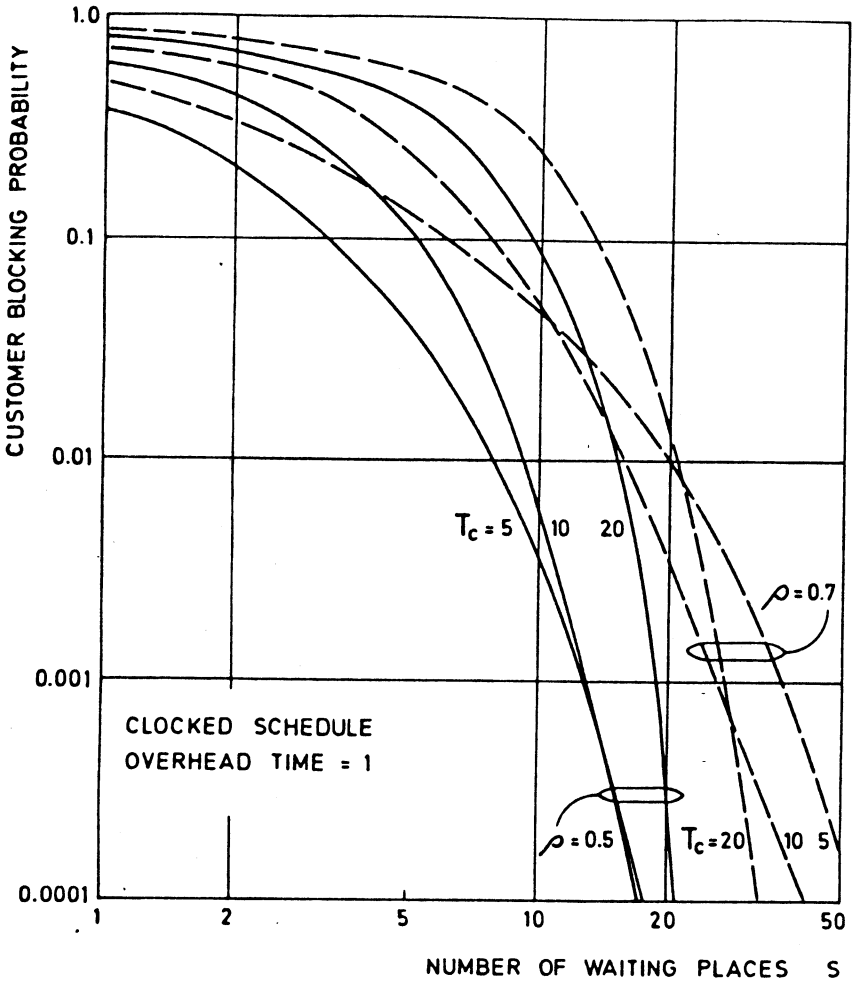

Fig.4 :

Customer Blocking vs Queue Size (Parameters: Offered Traffic and Clock Period)

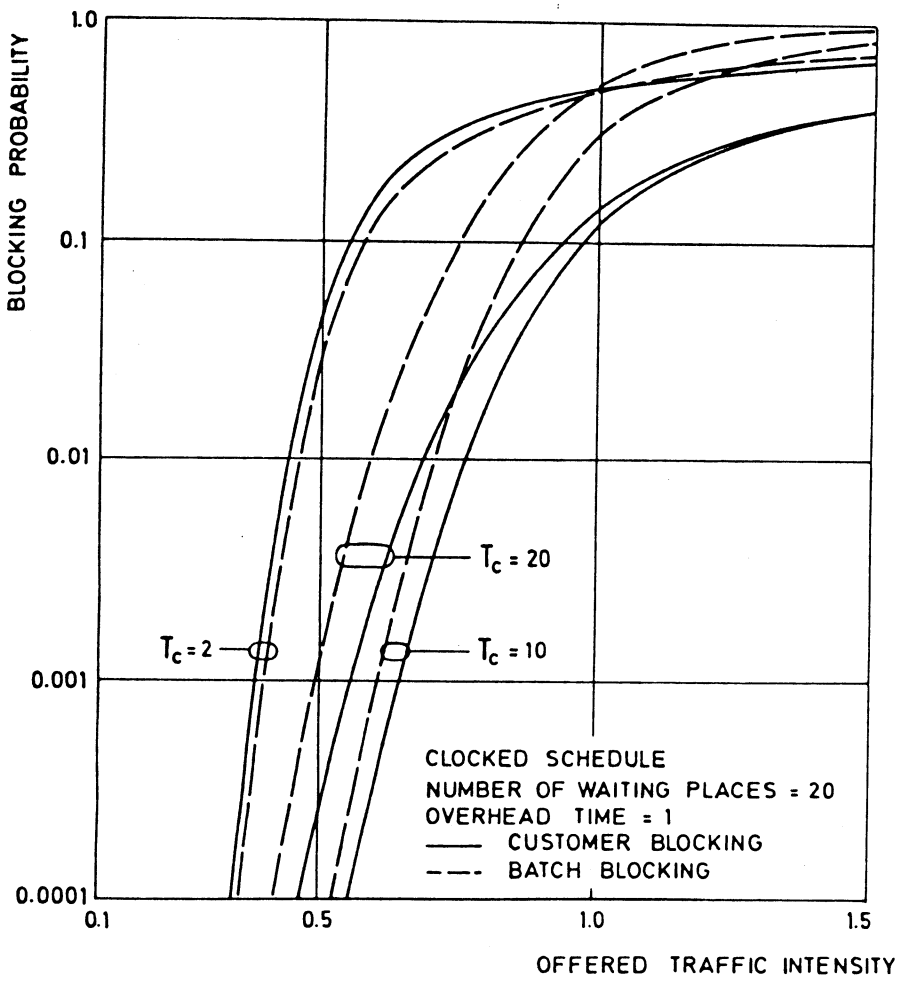

Fig.5 :

Blocking Probability vs Offered Traffic (Parameter: Clock Period)

traffic intensity. With regard to Fig. 6, the mean waiting time asymptotically approaches T_o as the traffic goes to zero, since even customers entering an empty system have to wait at least one overhead time. Fig. 7 gives the c.d.f. of waiting time in the queue until a customer first reaches the server, for different values of offered traffic and clock period. The effect of the overhead times can be recognised easily by the "step" characteristic of the curves. These results may be useful for dimensioning the queue to a percentile delay for some fixed offered traffic level. The mean pre-queueing delay as a function of offered traffic is shown in Fig. 8. For a certain value of clock period, the pre-queueing delay increases with offered traffic because the latest-arriving customers tend to be blocked. However in a useful working range, say ρ less than 0.7, blocking is seen to have little effect and the mean pre-queueing delay can be approximated by one half the clock period.

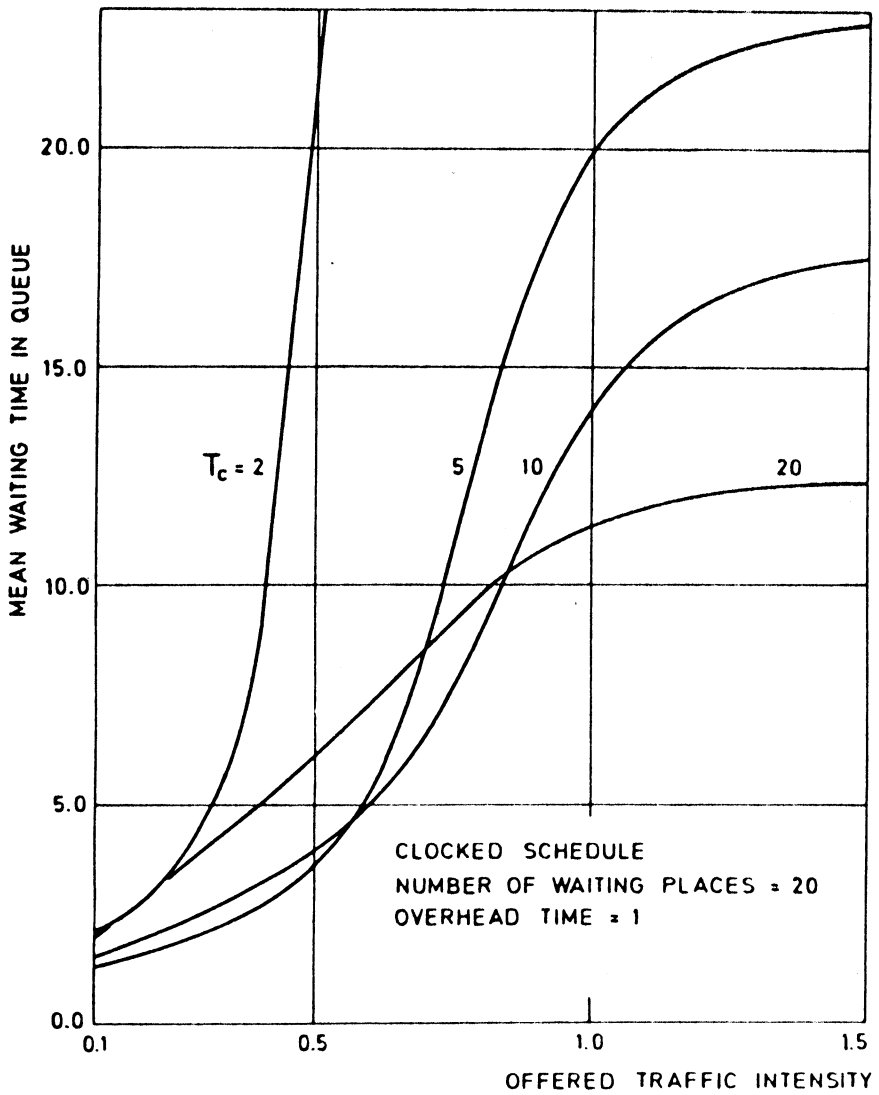

Fig.6 : Mean Waiting Time vs Offered Traffic. Parameter : Clock Period

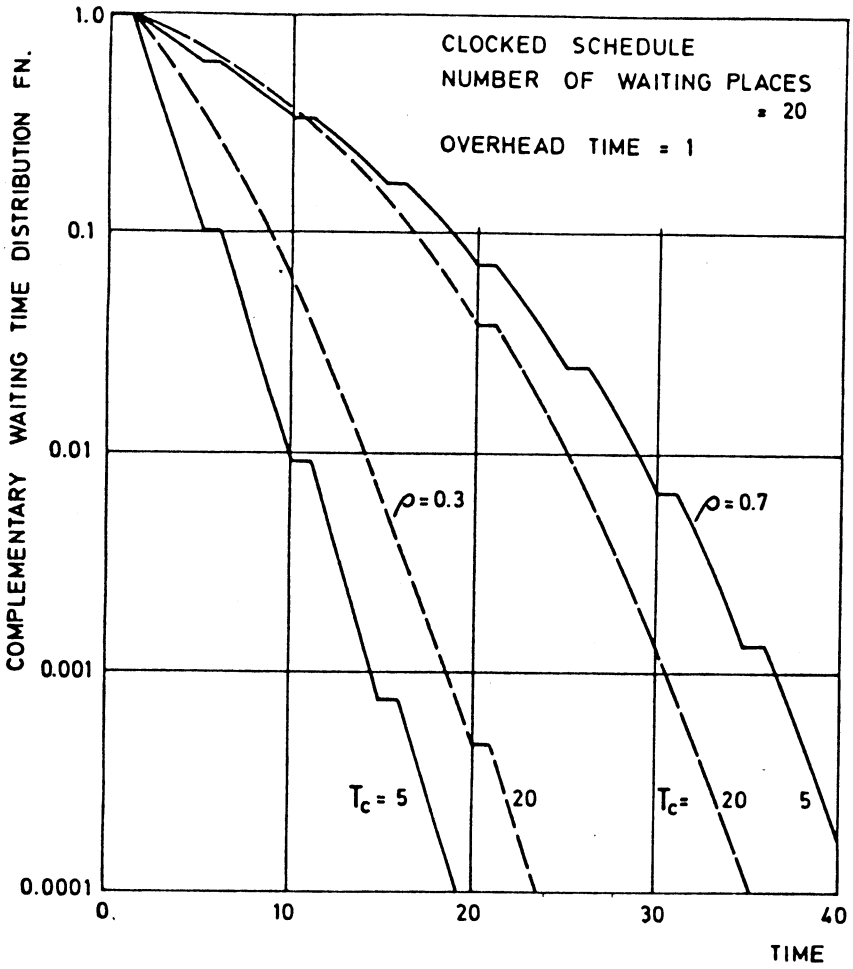

Fig.7 :

Complementary Waiting Time Distribution Function

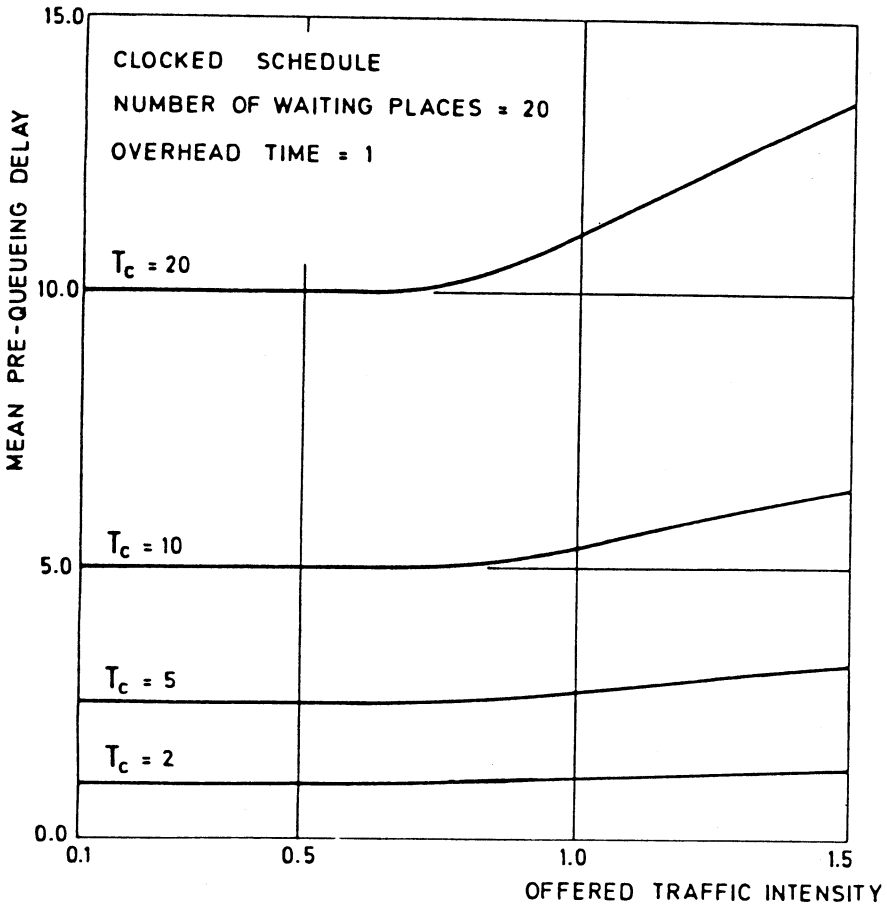

Fig.8 :

Mean Pre-Queueing Delay vs Offered Traffic (Parameter :Clock Period)

4.2 Effect of Clock Period

In Fig. 9, blocking probabilities for customers and batches are plotted as a function of clock period, and it is quite clear an optimum choice of T_c exists for a given level of offered traffic. Moreover this optimum value varies very little with the level of offered traffic. Related to the clock period are two opposing effects which give rise to the optimum. For small T_c, the percentage of time taken up with overhead is large, and so customers spend longer in the queue and the blocking is therefore high. If T_c is large, the mean batch size at a clock instant is also large, and blocking is again high.

The mean waiting time in the queue, depicted in Fig. 10 exhibits a similar characteristic with respect to clock period. As T_c grows very large, the batch sizes are also very large and the system will tend to be filled at each clock instant, and tend to be emptied before the next clock instant. An arbitrary customer therefore has to wait, on the average, for the service of s/2 customers (those ahead of him in the batch), plus one overhead period. This explains the asymptote at high traffic for the mean waiting time.

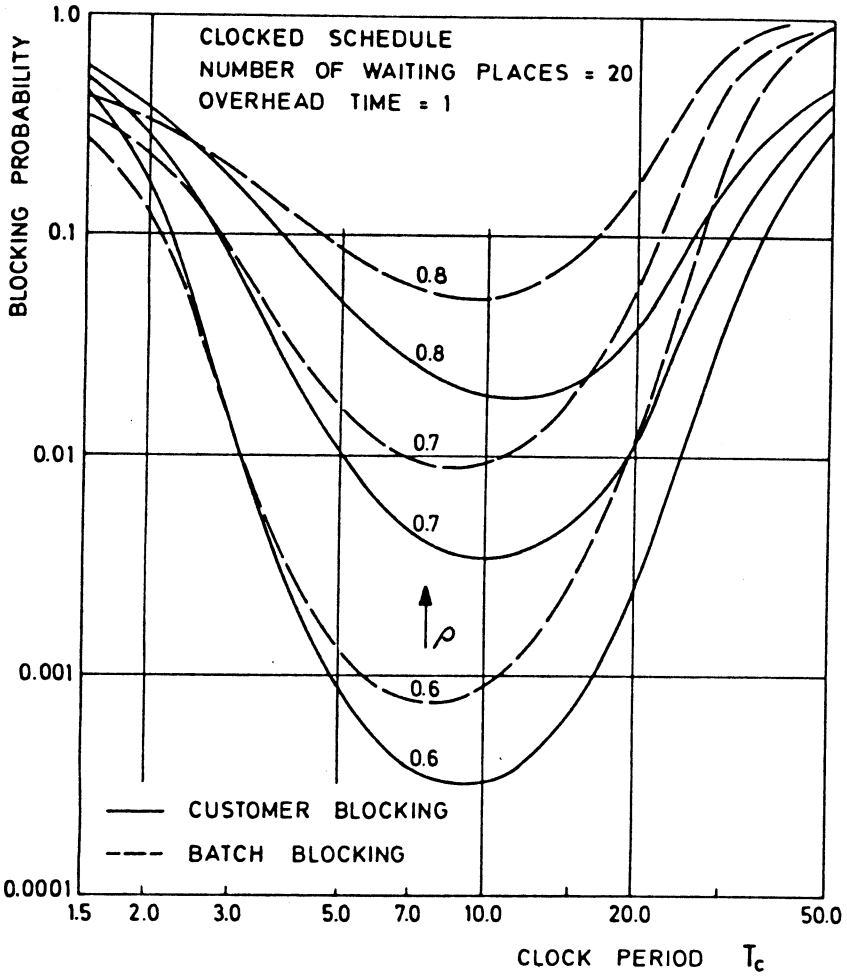

Fig.9 : Blocking vs Clock Period.
Parameter : Offered Traffic

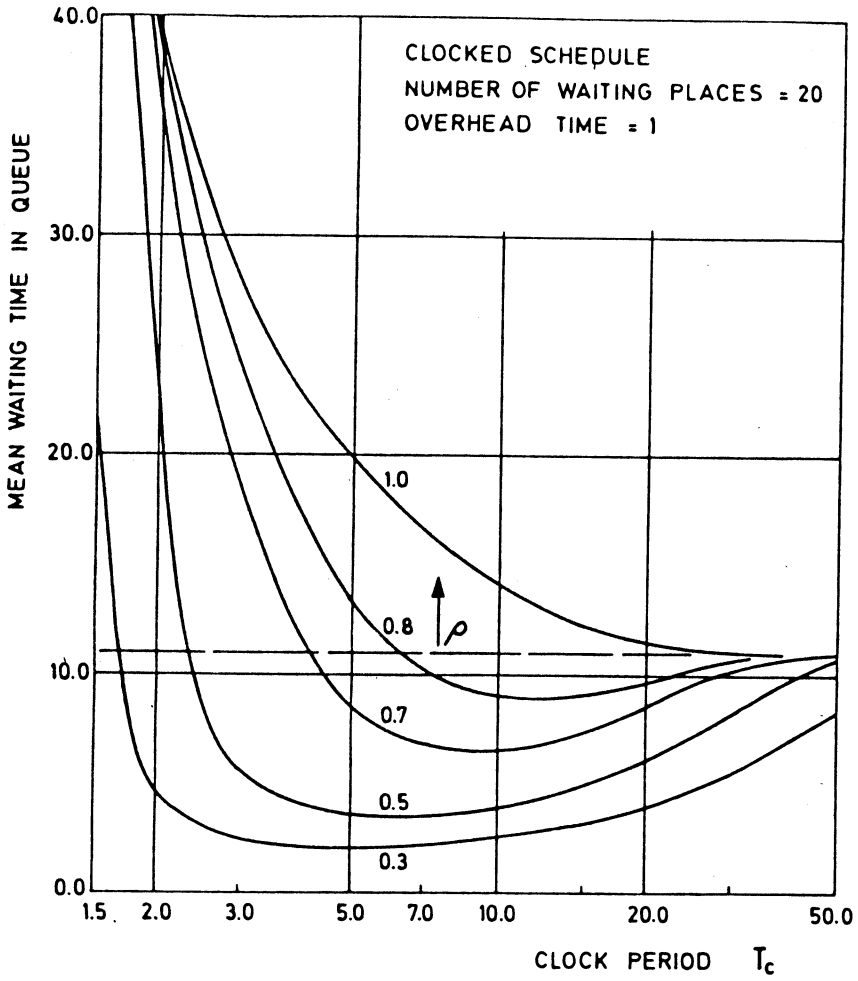

Fig. 10

Mean Waiting Time vs Clock Period (Parameter : Offered Traffic)

5. ALTERNATIVE I/O SCHEDULING SCHEME

5.1 Description

The previous section is devoted to results for the case of a clocked communication schedule. An alternative scheme easily implemented in software is an "event collection" scheme as follows. Two processors performing different functions generate events which need to be handled in the other. For simplicity only one direction of communication is considered. When the primary processor has collected a fixed number, n say, of event messages which need to be sent to the secondary processor, then it generates an interrupt to begin an I/O phase. For example this is very efficient at low traffic levels when we set $n = 1$, and clearly in this case the pre-queueing delay is zero. During the I/O phase both processors are unavailable for event processing and so no new event messages for transmission can be generated in these intervals. During the active periods it is assumed that events to be transmitted are generated according to a Poisson process with intensity

$$a = \rho\left[T_o + E[X_2]\right] / E[X_2] \qquad \ldots(5.1)$$

where the overhead periods are again assumed to be constant and of length T_o. In this way the active period has an Erlang distribution of order n, and the analytical tools of section 3 (except for the calculation of waiting time c.d.f. and pre-queueing delay) may be applied.

5.2 Pre-queueing Delay

To calculate the expected pre-queueing delay we assumed the same notation as in subsection 3.6. For a system without blocking it is not difficult to see that

$$E[Y] = \sum_{k=1}^{n} \frac{n-k}{na} = \frac{n-1}{2a} \qquad \ldots(5.2)$$

When blocking is considered, we condition on the state of the system just prior to the next I/O phase, and condition on customer acceptance to obtain

$$E[Y] = \frac{\rho}{\rho'} \left(\sum_{j=0}^{N-n} P_j \sum_{k=1}^{n} \frac{n-k}{na} + \sum_{j=N-n+1}^{N-1} P_j \sum_{k=1}^{N-j} \frac{n-k}{na} \right)$$

which quickly reduces to

$$E[Y] = \frac{\rho}{\rho' a} \left(\frac{n-1}{2} \sum_{j=0}^{N-n} P_j + \sum_{j=N-n+1}^{N-1} P_j (N-j) \left(1 - \frac{N-j+1}{2n}\right) \right) \qquad \ldots(5.3)$$

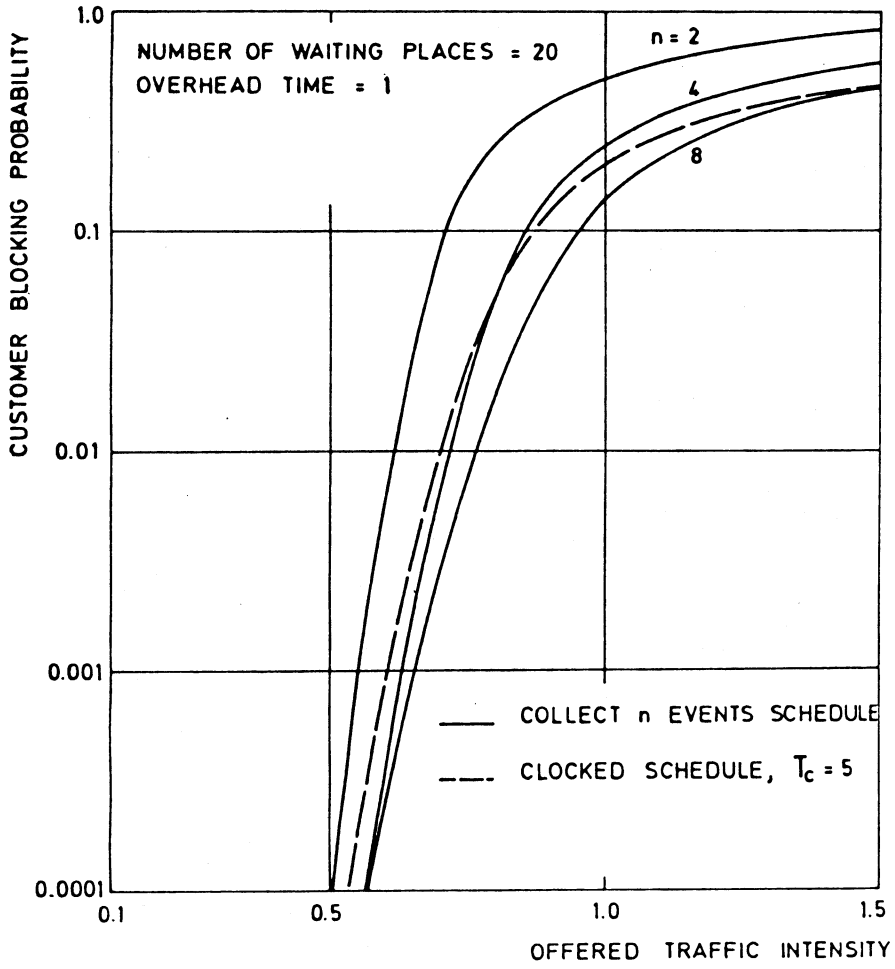

Fig. 11 :

Customer Blocking vs Offered Traffic; Comparison of Scheduling Schemes

In the case of n = 1 it is clear from the development that the second term of (5.3) will not exist.

5.3 Results and Comparison

In Fig. 11 the blocking curves for the queue with 20 waiting places are drawn, for various values of n in the "collect n events" scheduling scheme. For comparison the blocking for the clocked schedule with $T_c = 5$ is also drawn. From Fig. 9 it is seen that $T_c = 5$ is not the optimum with respect to blocking, and at approximately $T_c = 10$ the clocked scheme works as well as the best event collection scheme.

In Fig. 12 the expected total delay (pre-queueing delay plus mean waiting time in queue) is also depicted for a range of n for the event collection scheme and compared to the clocked scheme for $T_c=5$. Generally speaking it may be said that from the point of view of waiting time one can always do better with the event collection scheme by an optimal choice of n. On the other hand, the total delay and the optimal value of n are very sensitive to changes in traffic load, and it may not be easy to quickly compute the optimal n in real-time. The clocked scheme in contrast is very robust with respect to changes in traffic load.

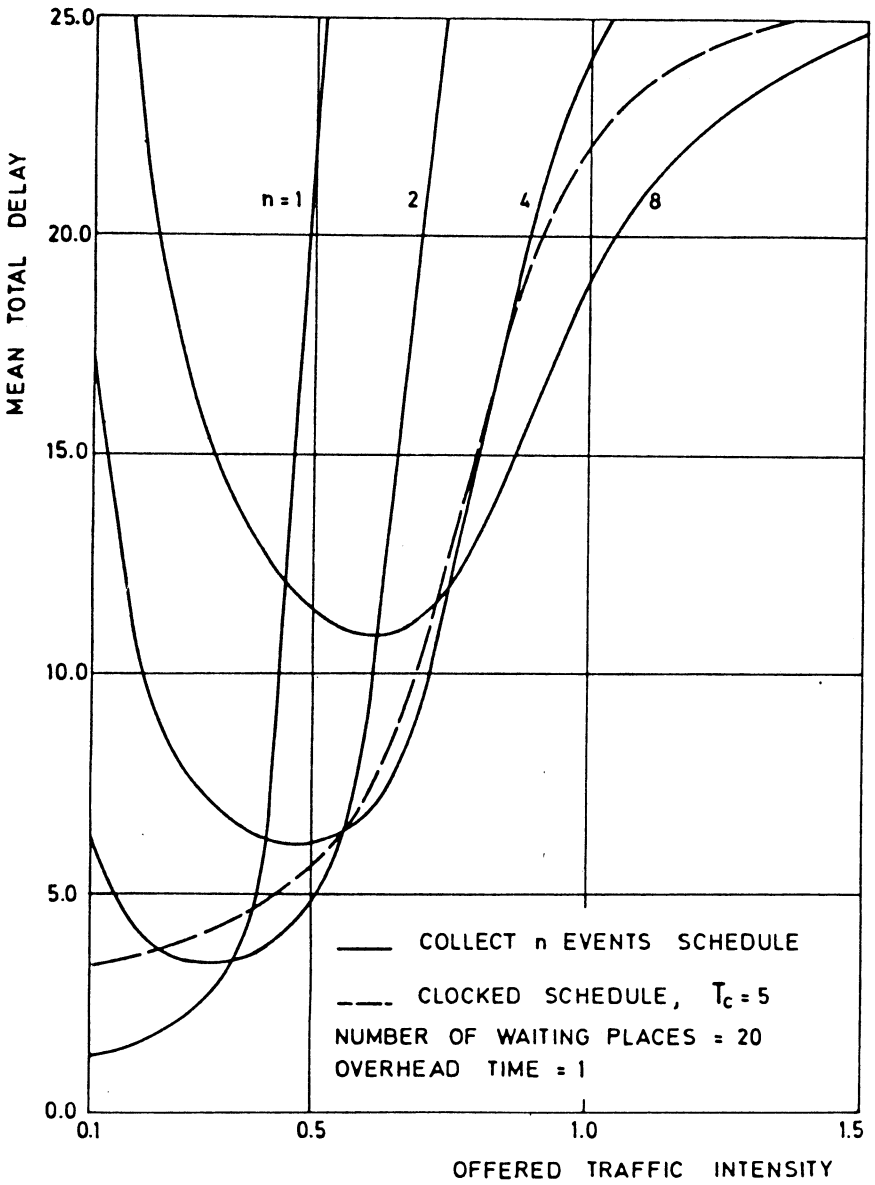

Fig. 12 :

Mean Total Delay vs Offered Traffic; Comparison of Scheduling Schemes

6. CONCLUSION

The queueing model and accompanying analysis which have been developed in this paper give a range of tools useful in the performance evaluation for inter-processor message traffic in a distributed system. The model is applicable to a range of systems, not only to stored program controlled switching systems, but also to more general distributed computing systems, for example in the traffic analysis of bus systems. The results show how blocking and delay criteria can be taken into account in the traffic engineering of inter-processor communication.

For the scheme of a clocked communications schedule, it is seen that the clock period is a critical parameter for system performance, with an optimum value that is largely determined by the system overhead rather than the level of inter-processor message traffic. In comparison to the alternative communication scheme, the "collect n events" schedule considered in Section 5, the clocked scheme is seen to be relatively robust with respect to traffic level, when the clock period is well chosen.

ACKNOWLEDGEMENT

The authors would like to thank Professor P. J. Kuehn and also H. Jans for helpful discussions during the course of this work.

REFERENCES

[1] Langenbach-Belz M., "Sampled Queueing Systems". Proc. Symp. on Computer Communications Networks and Teletraffic, Brooklyn, 1972.

[2] Langenbach-Belz M., "Two-Stage Queueing System with Sampled Parallel Input Queues". Proc. 7 th I.T.C., Stockholm, 1973.

[3] Schwaertzel H.G., "Serving Strategies of Batch Arrivals in Common Control Switching Systems".Proc. 7 th I.T.C., Stockholm,1973.

[4] Weisschuh H. and Wizgall M., "Investigations on the Traffic Behaviour of the Common Control in SPC Switching Systems".Proc. 8 th I.T.C., Melbourne, 1976.

[5] Weisschuh H., "Development of the Control Software for a Stored Program Controlled PCM Switching System". 24 th Report on Studies in Congestion Theory, Univ. of Stuttgart, 1977.

[6] Kuehn P.J., "Analysis of Switching System Control Structures by Decomposition". Proc. 9 th I.T.C., Spain, 1979.

[7] Gross D. and Harris C.M., "Fundamentals of Queueing Theory". Wiley, 1974.

[8] Burke P.J., "Delays in Single Server Queues with Batch Input". B.S.T.J.,23, 830-833(1975).

[9] Little J.D.C., "A Proof of the Queueing Formula $L = \lambda W$". Operations Research, 9, 383-387(1961).

[10] Cox D.R., "Renewal Theory". Methuen and Co., London, 1962.

[11] Manfield D.R. and Tran-Gia P., "Analysis of a Finite Storage System with Batch Input Arising out of Message Packetisation". Univ. of Siegen 1980, to be published in IEEE Transactions on Communications.