# Delay Analysis of Clock-Driven Message Transfer in Distributed Processing Systems

by Phuoc Tran-Gia * and Herbert Jans **

In modern computer and communication systems the system control is distributed among a number of individual devices or processors operating in modes of function or load sharing. Communication between distributed control units via an interconnection network is often organized in the form of message interchanging according to a message transfer mechanism.

In this paper, a performance analysis is given for the commonly used clock-driven messaging protocol, for which a two-level queueing system is developed and investigated. The analysis is done using a two-dimensional imbedded Markov chain, for which the calculation is done in conjunction with a dimension reducing choice of regeneration points.

Numerical results for dimensioning purposes are given for message delay characteristics under different traffic conditions, clock intervals, and buffer sizes. The results can be used to optimize the messaging delay for the class of clock-driven message transfer mechanisms. Finally, the distribution function of the queueing delay is derived for the first-in first-out message transfer discipline.

**Wartezeitanalyse von taktgesteuerten Kommunikationsmechanismen in Systemen mit verteilter Steuerung**

In modernen Rechner- und Kommunikationssystemen ist die Systemsteuerung oft auf eine Anzahl von Steuerungseinheiten bzw. Prozessoren verteilt, die nach Funktions- und Lastteilungsprinzipien arbeiten. Die Kommunikation zwischen den verteilten Steuerungseinheiten erfolgt häufig über ein Kommunikationssystem mittels eines Meldungsaustauschprotokolles.

In diesem Beitrag wird die Leistungsfähigkeit des in Telekommunikationssystemen oft verwendeten taktgesteuerten Meldungsaustauschverfahrens untersucht, wobei ein zweistufiges Warteschlangenmodell entwickelt und analysiert wird. Die Analyse geht von einer eingebetteten Markoff-Kette mit einer zweidimensionalen Zustandsbeschreibung aus, welche für die Berechnung jedoch durch geeignete Wahl der Regenerationszeitpunkte um eine Dimension reduziert werden kann.

Für Dimensionierungszwecke werden numerisch gewonnene Ergebnisse der Meldungswartezeiten bei verschiedenen Verkehrsintensitäten, Taktdauern und Pufferspeicherkapazitäten diskutiert. Im letzten Kapitel des Beitrages wird die Wartezeitverteilungsfunktion der Meldungen für die Abfertigungsstrategie FIFO (first-in, first-out) abgeleitet.

## 1. Introduction

In distributed processing systems, especially in communication applications, the traffic intensities of real-time messages, which have to be handled and interchanged by several processors, are very high. These messages are generated by users or peripheral devices or they are caused for purposes of interprocessor communications.

Fig. 1 illustrates a basic control structure of a multi-processor system where communications are performed by message interchanging via an interconnection network. Messages are preprocessed by an access (I/O) controller (process level 1) and stored in an intermediate message buffer as valid messages in a logical sense for further processing in the device control unit. Due to the function of the considered device the messages will be transferred to the device control unit (process level 2) according to a messaging protocol. The transfer protocol influences very strongly the message delay characteristics and the throughput performance of the entire system.

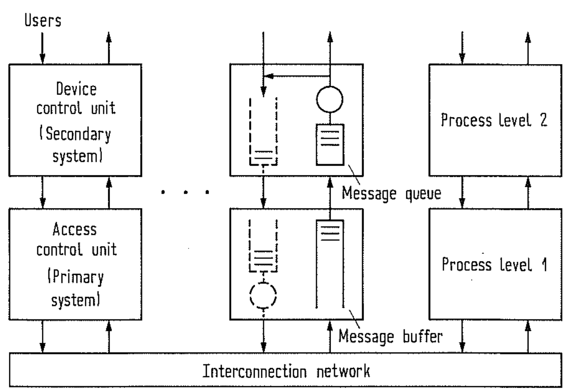


Fig. 1. Messaging in distributed processing environments.

* Dr.-Ing. P. Tran-Gia, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität, Seidenstrasse 36, D-7000 Stuttgart 1.

** Dr.-Ing. H. Jans, Siemens AG, Öffentliche Vermittlungstechnik, Boschetsriederstrasse 133, D-8000 München 70.

The most commonly used messaging protocol is the clock-driven scheme, whereby messages are transmitted from process level 1 to process level 2 in a batch-wise manner at a scheduled time, initialized by a real-time clock. The messaging protocol includes all activities to transfer messages like initialisation of the transfer task, transmission control, acknowledgement etc.

There are a number of studies [1]−[4] which investigate the performance of message sampling and transfer schemes by means of basic queueing models with batch arrivals. Some of them [2]−[4] deal with models having infinite waiting capacity. Finite capacity aspects are discussed in [1], in conjunction with several messaging schemes with batch arrivals and overhead. Messages of an arriving batch are considered to be lost when the batch is larger in size than the actual number of free waiting places. In [5] dimensioning aspects are considered for models with clocked batch input where priorities for different classes of customers are taken into account. A delay optimizing messaging scheme is presented and investigated in [6], which is modelled by means of a two-level queueing system with an arrival-driven scheduling mechanism.

In order to investigate the messaging performance of the system depicted in Fig. 1, we restrict the consideration to the message traffic offered from the interconnection network and the delay characteristics of messages before being served by the device control unit. A two-level queueing system is developed and analysed in this paper, where a clock-driven messaging scheme is considered. The results obtained can be used to optimize the message transfer delays for a wide range of messaging mechanisms.

In principle an approximate analysis of this model by decomposition is possible using two separate one-level queues. This method implies an independence assumption for the two queues; it is only exact for infinite capacity queues, where no backward blocking effect can occur. Due to the correlation between the two queues, which is caused by the clock-driven nature of the messaging scheme, the accuracy of the decomposition approximation is expected to be reasonable for only a small range of system parameters.

This paper presents an exact analysis method, which requires a coupled two-level queueing system to a two-dimensional process description. By observing the two-dimensional process at particular points, the analysis can be simplified according to a one-dimensional state description, for which computational efforts can be remarkably reduced.

## 2. The Two-Level Queueing Model

The queueing model considered in this paper has the structure shown in Fig. 2. Message arrivals constitute a Poisson process. The Poisson assumption is based on the observation that the incoming message stream is the superposition of offered traffic from a large number of different devices and processors connected to the observed control unit. Taking into account the different types of events and corresponding tasks and programs they may activate, the service time $T_H$ of messages is assumed to be negative exponentially distributed.

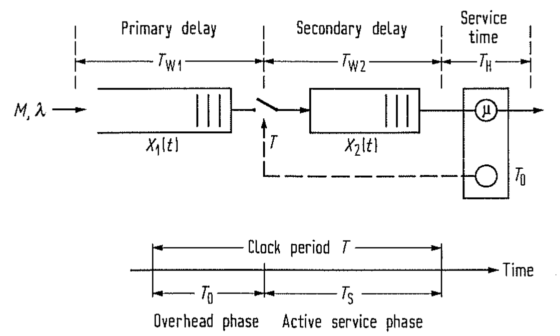


Fig. 2. The two-level queueing system.

Every message transfer activity which is controlled by the processor is usually performed by the same I/O task and has approximately the same run time during which the processor is not available for message processing. Thus, the whole clock period $T$ consists of two parts: the overhead phase $T_0$ and the active service phase $T_S$ (c.f. Fig. 2). It should be noted here that only during the active service phase the server is available for message processing but not necessarily busy. In this paper the clock period is chosen to be constant; this is often the case in systems where the I/O phases are activated by a real-time clock. Another reason for this choice of $T$ can be found in [1]. It is shown there that a well dimensioned clocked scheme is relatively robust with respect to the message traffic intensity. The primary queue is considered to be infinite whereas the secondary queue (process level 2) is limited to the finite capacity $S$. At a clock instant, when the actual batch (i.e. the number of waiting messages in the primary queue prior to the clock transfer instant) is larger in size than the number of free waiting places in the secondary queue (process level 2), all free positions will be filled in a first-in first-out (FIFO) order and the remaining messages must wait for retrial until the next clock transfer instant.

Thus, each batch consists of two parts, the fresh part and the reattempt part. All arrivals during the clock period form the fresh part; the reattempt part contains messages which have been rejected at the previous clock instant.

The total sojourn time of a message in the system is composed of three components:

- the primary delay $T_{W1}$, i.e. the waiting time in the primary queue (process level 1),
- the secondary delay $T_{W2}$, i.e. the waiting time in the secondary queue, including overhead periods (process level 2),
- the service time $T_H$.

## 3. State Analysis of the Two-Level Queueing System

In this section performance measures of the above described two-level queueing system are presented. Subsection 3.1 discusses queue stability conditions and subsections 3.2 and 3.3 deal with the Markov chain state probabilities and the arbitrary-time state probabilities, respectively. Subsequently, system characteristics will be derived in subsection 3.4 and some results will be presented in subsection 3.5.

The following symbols will be used in the analysis:

| | |
|---|---|
| $\lambda$ | parameter of Poisson process of message arrivals, |
| $\mu$ | service rate ($\mu = 1/\mathrm{E}\,[T_H]$), |
| $\varrho = \lambda/\mu$ | offered traffic intensity, |
| $S$ | capacity of the secondary queue, |
| $N = S+1$ | secondary system size (process level 2), |
| $T_{W1}$ | random variable (r.v.) for the primary delay (pretransfer delay), |
| $T_{W2}$ | r.v. for the message waiting time in the secondary queue, |
| $T_D$ | r.v. for the message total delay, |
| $T_F$ | r.v. for the sojourn time (flow time) of messages. |

Furthermore, the following notations will be used

$$a_k(t) = \Pr\{k \text{ messages arrive in a time interval of length } t\} = g(k, \lambda t), \quad (1\,a)$$

$$d_k(t) = \Pr\{k \text{ service completions during a time interval of length } t, \text{ server busy}\} = g(k, \mu t) \quad (1\,b)$$

where $g(k, u)$ is the Poisson distribution given by

$$g(k, u) = \frac{(u)^k}{k!}\,\mathrm{e}^{-u}, \qquad k = 0, 1, \ldots, u \geqq 0. \quad (1\,c)$$

### *3.1. Queue stability conditions*

In accordance with the considered transfer overhead $T_0$ and the finite capacity $S$ of the secondary queue, the system is only stable under certain conditions which will be derived below. For given values of the overhead period $T_0$, the primary queue capacity $S$, and the offered traffic intensity $\varrho$ we will calculate a lower and an upper limit for the clock period $T$, for which the system is stable, i.e. the steady state conditions of the whole queueing system exist. The lower limit $T_{\min}$ is found using the fact that, on average, the active service interval $T_S$ in which the server is available must be long enough to serve all arriving messages:

$$\lambda T < \mu T_S = \mu (T - T_0)$$

or

$$T_{\min} = T_0/(1-\varrho). \quad (2\,a)$$

On the other hand, if the clock period $T$ becomes too long, the mean batch sizes to be transferred are also very large while the finite queue is likely to be filled completely at each transfer instant and tends to be empty before the next transfer period. Thus, the upper limit $T_{\max}$ can be defined as

$$\lambda T < N$$

or

$$T_{\max} = N/\lambda. \quad (2\,b)$$

The whole queueing system is stable for

$$T_{\min} < T < T_{\max}. \quad (2\,c)$$

The queue stability condition (2c) is illustrated in Fig. 3. For $T_0 = 0.5$ and $S = 10$ the system is stable in the hatched area. The dashed lines show the upper clock period limit for other values of $S$.

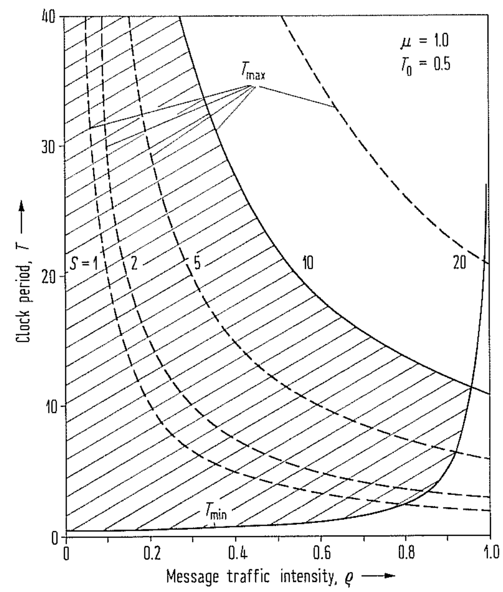


Fig. 3. Queue stability conditions.

### *3.2. Markov chain state probabilities*

Define $X_1(t,n)$ and $X_2(t,n)$ to be the random variables for the number of messages in the primary and the secondary system at time $t$ after the $n$-th clock transfer period $(0 < t < T)$, respectively, and the system state probability

$$P(i, j, t, n) = \Pr\{X_1(t,n) = i,\, X_2(t,n) = j\}. \quad (3)$$

Assume further a stable system under steady-state conditions, i.e.

$$p(i, j, t) = P(i, j, t, n+1) = P(i, j, t, n), \quad (4)$$
$$i = 0, 1, \ldots,\; j = 0, 1, \ldots, N, \quad 0 < t < T,$$

taking into account the periodical behaviour of the state process.

The two-dimensional quasi-stationary state process is illustrated in Fig. 4. It can be clearly seen that it is here convenient to choose the time epochs just after the clock transfer instants ($t = 0^+, T^+, 2T^+, \ldots$) as regeneration points of the imbedded Markov chain, at which the two queues can be considered as connected and a one-dimensional state description is possible. Thus, the Markov chain state probabilities

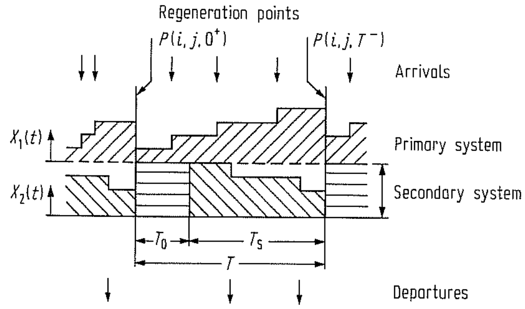


Fig. 4. The two-dimensional state process.

$$p(i,j,0^+), \quad i=0,1,\ldots, \quad j=0,1,\ldots,N$$

can be simply denoted by

$$p(i,j,0^+) = \pi(k) \quad \text{with} \quad k=i+j. \tag{5}$$

In order to calculate the transition probabilities of the Markov chain, we observe two consecutive transfer epochs $n$ and $n+1$. The main steps of the transition probability calculation are:

i) Calculate $P(i,j,T^-,n)$ out of $P(i,j,0^+,n)$

$$\begin{aligned} P(i,j,T^-,n) &= \sum_{y=j}^{N}\sum_{x=0}^{i} P(x,y,0^+,n)\, a_{i-x}(T)\cdot \\ &\quad \cdot d_{y-j}(T_S), \qquad j>0 \\ P(i,0,T^-,n) &= \sum_{y=0}^{N}\sum_{x=0}^{i} P(x,y,0^+,n)\cdot \\ &\quad \cdot a_{i-x}(T)\sum_{z=y}^{\infty} d_z(T_S). \end{aligned} \tag{6a}$$

ii) Calculate $P(i,j,0^+,n+1)$ out of $P(i,j,T^-,n)$

$$\begin{aligned} P(0,j,0^+,n+1) &= \sum_{y=0}^{j} P(j-y,y,T^-,n), \\ &\qquad j=0,1,\ldots,N-1, \\ P(i,N,0^+,n+1) &= \sum_{y=0}^{N} P(N+i-y,y,T^-,n), \\ &\qquad i=0,1,\ldots. \end{aligned} \tag{6b}$$

Taking eqs. (6a) and (6b) together, the transition probabilities, which give a relationship between state probabilities of two consecutive regeneration points of the Markov chain, are calculated. Using eqs. (4), (5) and after simple algebraic manipulations, the system of difference equations, which implicitly contains the transition probabilities, is derived:

$$\begin{aligned} \pi(k) &= \sum_{i=0}^{N-1}\pi(i)\left[a_k(T)\sum_{j=i}^{\infty} d_j(T_S) + \right. \\ &\quad \left. + \sum_{j=1}^{\min(i,k)} d_{i-j}(T_S)\, a_{k-j}(T)\right] + \\ &\quad + \sum_{i=N}^{N+k}\pi(i)\left[a_{k+N-i}(T)\sum_{j=N}^{\infty} d_j(T_S) + \right. \\ &\quad \left. + \sum_{j=1}^{\min(N+k-i,N)} d_{N-j}(T_S)\, a_{k+N-i-j}(T)\right] \end{aligned} \tag{7}$$

with $\sum_{j=a}^{b}(.) = 0$ for $b<a$.

Using eq. (7) the state probabilities of the imbedded Markov chain can be obtained by means of the numerical method of Gauss-Seidel iteration with overrelaxation, whereby a proper adaptive truncation of the state space has been used.

### *3.3. Arbitrary-time state probabilities*

In order to calculate the mean waiting time in the primary queue $E[T_{W1}]$ and in the secondary queue $E[T_{W2}]$, it is convenient to use Little's law [9], for which it is necessary to know the mean queue lengths at an arbitrary time instant, i.e. the system state probabilities seen by an outside observer. The arbitrary-time state probabilities can be, e.g., calculated by using results of semi-Markov processes [6]. In this subsection, a more simpler approach will be presented, which bases on the fact of constant intervals between successive regeneration points of the imbedded Markov chain.

We consider the two-dimensional state space characterized by the two r.v. $X_1(t,n)$ and $X_2(t,n)$, $0<t<T$. Between two transfer instants, $X_1(t,n)$ follows a pure birth process and $X_2(t,n)$ a pure death process (with exception of the overhead period). Under stationary conditions, taking into account the periodical property of the state process, we only have to observe the system during one clock period.

Assuming a stable system under steady state conditions, i.e.

$$X_i(t,n) = X_i(t,n+1) = x_i(t), \quad 0<t<T, \quad i=1,2, \tag{8}$$

and denoting

$$\begin{aligned} p_1(k,t) &= \Pr\{x_1(t)=k\}, \quad k=0,1,\ldots, \quad 0<t<T, \\ p_2(k,t) &= \Pr\{x_2(t)=k\}, \quad k=0,1,\ldots,N, \quad 0<t<T, \end{aligned} \tag{9}$$

we obtain from the Markov chain state probabilities the arbitrary-time state probabilities of the primary and the secondary system (secondary queue and server):

i) Primary system

$$\begin{aligned} p_1(0,0^+) &= \sum_{i=0}^{N}\pi(i), \\ p_1(k,0^+) &= \pi(k+N), \qquad k=1,2,\ldots \end{aligned} \tag{10a}$$

and, due to the evolution of the pure birth process between two transfer epochs

$$p_1(k,t) = \sum_{i=0}^{k} p_1(i,0^+)\, a_{k-i}(t). \tag{10b}$$

ii) Secondary system

$$\begin{aligned} p_2(k,0^+) &= \pi(k), \qquad k=0,1,\ldots,N-1 \\ p_2(N,0^+) &= \sum_{i=N}^{\infty}\pi(i) \end{aligned} \tag{11a}$$

and, due to the evolution of the pure death process during an active service period ($T_0 < t < T$)

$0 < t \leqq T_0$

$$p_2(k,t) = p_2(k,0^+), \qquad k = 0, 1, \ldots, N$$

$T_0 < t < T$

$$p_2(0,t) = \sum_{i=0}^{N} p_2(i,0^+) \sum_{j=i}^{\infty} d_j(t-T_0),$$
$$p_2(k,t) = \sum_{i=k}^{N} p_2(i,0^+)\, d_{i-k}(t-T_0), \qquad k = 1, 2, \ldots, N. \tag{11 b}$$

*3.4. System characteristics*

Using the results of the state analysis described above, system characteristics can be obtained. From eqs. (10 a, b) and (11 a, b) the mean number of messages in the primary system $E[X_1]$ and in the secondary system $E[X_2]$ can be derived:

$$E[X_1] = \frac{1}{T} \int_0^T \sum_{k=1}^{\infty} k\, p_1(k,t)\, dt = \sum_{k=1}^{\infty} k\, p_1(k,0^+) + \frac{\lambda T}{2} = E[x_1(0^+)] + \frac{\lambda T}{2}, \tag{12}$$

$$E[X_2] = \frac{1}{T} \int_0^T \sum_{k=1}^{N} k\, p_2(k,t)\, dt = \frac{T_0}{T} \sum_{i=1}^{N} i\, p_2(i,0^+) + \frac{1}{\mu T} \sum_{i=1}^{N} p_2(i,0^+) \sum_{k=1}^{i} k \sum_{j=i-k+1}^{\infty} d_j(T_S). \tag{13}$$

Using Little's law, the mean waiting time in the primary queue $E[T_{W1}]$ and the mean waiting time in the secondary queue $E[T_{W2}]$ can be given as

$$E[T_{W1}] = E[X_1]/\lambda, \tag{14}$$
$$E[T_{W2}] = E[X_2]/\lambda - 1/\mu. \tag{15}$$

The mean sojourn time of messages in the system is

$$E[T_F] = (E[X_1] + E[X_2])/\lambda \tag{16}$$

and the mean total delay of messages in the system is given by

$$E[T_D] = E[T_{W1}] + E[T_{W2}] = E[T_F] - 1/\mu. \tag{17}$$

*3.5. Results and discussions*

In this section numerical results are presented for the delay characteristics of messages in the system. All values are normalised with respect to $E[T_H] = 1/\mu$. Fig. 5 depicts the different mean values of delays as a function of the clock period $T$. By the chosen parameters ($\varrho = 0.7$, $S = 10$, $T_0 = 0.5$) the system is stable for $1.666 < T < 15.714$ (c.f. eq. (2)). It is clearly shown that a minimum of the total delay for messages exists as expected. Fig. 6 gives the mean total delay of messages for different message traffic intensities. The cross-over effect of the curves indicates an optimum choice of clock period as will be discussed in Fig. 7 in more detail.

In Fig. 7 the total delay is plotted as a function of clock period $T$. An optimum choice for $T$ can be defined for a given level of offered traffic intensity. The sensitivity of these optimum values has to be taken into account for dimensioning purposes. The best choice $T = 3$ for $\varrho = 0.6$ would lead for example to an instable system for $\varrho > 0.833$ during an overload situation. The mean total delay is shown in Fig. 8 as a function of the secondary queue capacity. For a fixed value of the clock period $T$, the mean total delay of messages cannot be further reduced by increasing the secondary queue length $S$ beyond a typical minimum value.

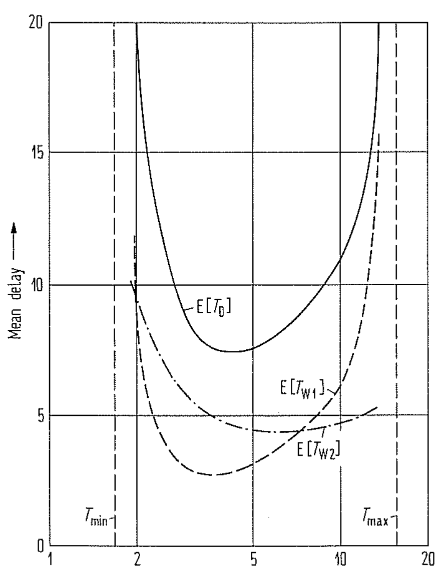

Fig. 5. Delays versus clock period; $S = 10$, $\varrho = 0.7$, $T_0 = 0.5$.

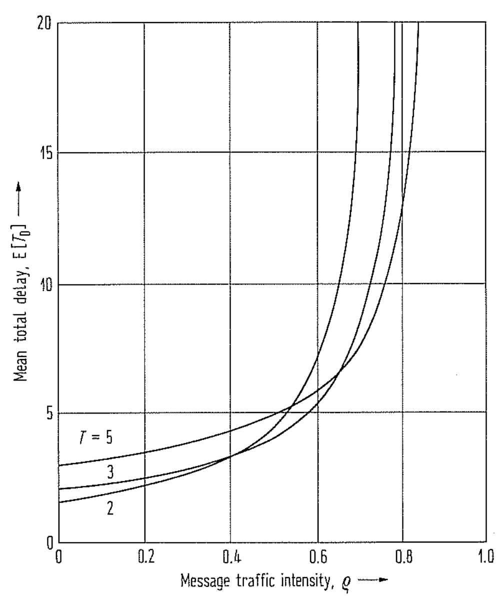

Fig. 6. Total delay versus offered traffic; $S = 10$, $T_0 = 0.5$.

The delay characteristics discussed here can be used for dimensioning purposes where the clock period $T$ and the capacity $S$ of the secondary queue have to be chosen for a given traffic range.

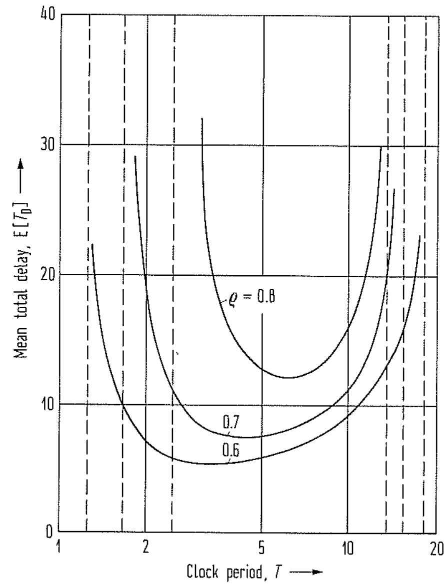


Fig. 7. Total delay versus clock period; $S = 10$, $T_0 = 0.5$.

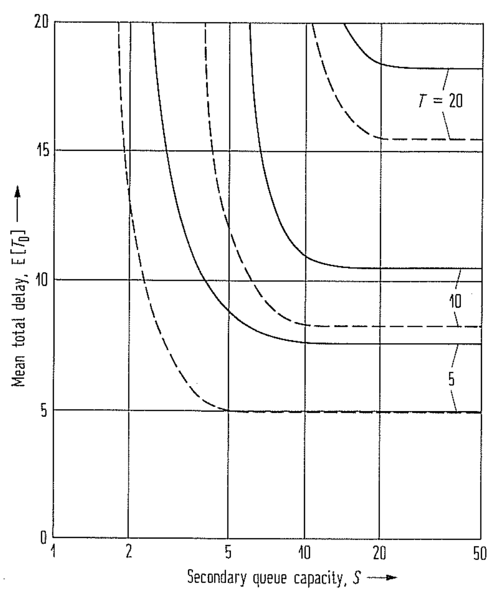


Fig. 8. Total delay versus secondary queue capacity; $T_0 = 0.5$, – – – $\varrho = 0.5$, —— $\varrho = 0.7$.

## 4. Delay Analysis

Due to the clock-driven transfer discipline between the access control unit and the device control unit, the delay characteristics are expected to be strongly influenced by the I/O access delay. Hence, in this section, attention will be devoted to the pretransfer delay. The distribution function of the pretransfer delay will be derived in the following, whereby the FIFO (first-in first-out) transfer discipline is considered.

### *4.1. Pretransfer delay distribution function*

In order to determine the distribution function

$$F_{W1}(t) = \Pr\{T_{W1} \leqq t\} \tag{18}$$

of the pretransfer delay $T_{W1}$, the fate of a test message (t.m.) is observed. As illustrated in Fig. 9 the pretransfer delay consists of two components:

$t_B$ delay from test message arrival epoch until the next clock instant,

$i \cdot T$ a number of $i$, $i = 0, 1, 2, \ldots$, clock periods until the test message is transferred.

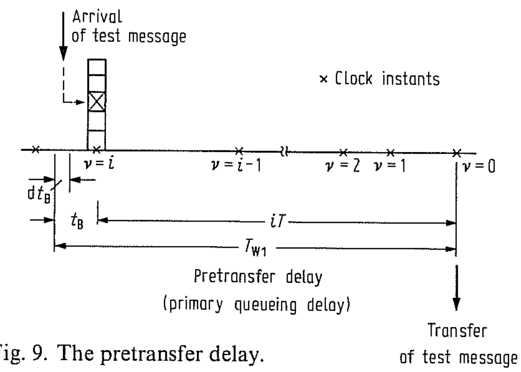


Fig. 9. The pretransfer delay.

Based on the stationary state distribution of the imbedded Markov chain and according to the FIFO transfer discipline the pretransfer delay depends only on the service process of those messages in the system in front of the t.m. For the derivation below we observe a t.m. which

- arrives $t_B$ before the transfer instant and takes the $k$-th position in the fresh batch (batch of new messages arriving during the clock interval),
- be in the $n$-th position in the entire system after the next transfer instant.

The probability density $b(k, t_B)$ that the t.m. arrives $t_B$ before the next transfer epoch and takes position $k$ in the fresh batch is

$$b(k, t_B) = a_{k-1}(T - t_B)\,\lambda, \quad k = 1, 2, \ldots. \tag{19}$$

The probability density $r(n, t_B)$ that the observed t.m. takes the $n$-th position in the whole system just after the next transfer instant is given by

$$r(n, t_B) = \sum_{k=1}^{n} b(k, t_B)\, q_{n-k}(0^+) \tag{20}$$

where

$q_{n-k}(0^+) = \Pr\{n-k$ messages are in the whole system at the next transfer instant$\}$,

$$q_{n-k}(0^+) = \begin{cases} \sum\limits_{j=0}^{N} \pi(j) \sum\limits_{r=j}^{N} d_r^{(1)}(T_S), & n = k \\ \sum\limits_{j=n-k}^{N+n-k} \pi(j)\, d_{j-n+k}^{(1)}(T_S), & n > k \end{cases} \tag{21}$$

and

$$d_j^{(1)}(T_S) = \begin{cases} d_j(T_S), & j = 0, 1, \ldots, N-1, \\ \sum_{k=N}^{\infty} d_k(T_S), & j = N, \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

By summing over all positions of $n$ eq. (20) results to the probability density $r(t_B) = \lambda$, as expected, according to the Poisson arrival process.

If the t.m. takes a position $i, i = 1, \ldots, N$, it will be delayed only for $t = t_B$ with the probability density

$$w(0, t_B) = \sum_{n=1}^{N} r(n, t_B). \quad (23a)$$

Because of the finiteness of the secondary queue, the maximum number of services during a clock period is limited by $N$. To be delayed $i$ clock periods the t.m. must take a position between $n = N+1$ and $n = (i+1)N$. Therefore we obtain

$$w(i, t_B) = \sum_{n=N+1}^{(i+1)N} r(n, t_B) \sum_{m=1}^{n-N} d_{n-m-N}^{(i-1)}(T_S) \cdot \sum_{j=m}^{N} d_j^{(1)}(T_S), \quad i = 1, 2, \ldots \quad (23b)$$

where $d_i^{(r)}(T_S)$ is the $r$-fold convolution of $d_i^{(1)}(T_S)$ with itself, and can be thought of as the probability for $i$ services during $r$ transfer periods, based on the fact that all waiting places in the secondary queue are occupied after each transfer epoch. It should be noted that

$$d_i^{(0)}(T_S) = \begin{cases} 1, & i = 0, \\ 0, & \text{otherwise.} \end{cases}$$

After integration of eqs. (23a, b) from 0 to $t_B$ we arrive at the probability

$$\Pr\{iT < T_{W1} \leqq iT + t_B\} = \int_0^{t_B} w(i, u)\,du, \quad (24a)$$

and using the definition of a distribution function

$$F_{W1}(iT + t_B) = \Pr\{T_{W1} \leqq iT + t_B\} = \Pr\{iT < T_{W1} \leqq iT + t_B\} + \Pr\{T_{W1} \leqq iT\}. \quad (24b)$$

By solving recursively eq. (23b) we obtain the pretransfer delay distribution function of messages

$$F_{W1}(t) = \sum_{k=0}^{i-1} I(k, T) + I(i, t_B), \quad i = 0, 1, \ldots \quad (25a)$$

where

$i = [t/T]^{-}$: largest integer less than $t/T$,

$t_B = t - iT, \quad \sum_a^b (.) = 0 \quad \text{for} \quad b < a$

and the function $I(i, t)$ is defined by

$$I(i, t) = \int_0^t w(i, u)\,du = \begin{cases} \sum_{n=1}^{N} \sum_{k=1}^{n} [a_{\geqq k}(T) - a_{\geqq k}(T-t)]\, q_{n-k}(0^+), & i = 0 \\ \sum_{n=N+1}^{(i+1)N} \sum_{k=1}^{n} [a_{\geqq k}(T) - a_{\geqq k}(T-t)]\, q_{n-k}(0^+) \cdot \sum_{m=1}^{n-N} d_{n-m-N}^{(i-1)}(T_S) \sum_{j=m}^{N} d_j^{(1)}(T_S), & i > 0 \end{cases} \quad (25b)$$

with

$$a_{\geqq k}(t) = \sum_{i=k}^{\infty} a_i(t).$$

### 4.2. Numerical results

The complementary pretransfer delay distribution function calculated by means of eqs. (25a, b) is illustrated in Fig. 10. For the given set of parameters ($\varrho = 0.5$, $S = 10$, $T_0 = 0.5$) the influence of the clock period length on the distribution function characteristic can clearly be seen. The two building parts of the pretransfer delay can be recognized in the diagram, in which an optimal choice of the intertransfer intervals with respect to the coefficient of variation can be estimated.

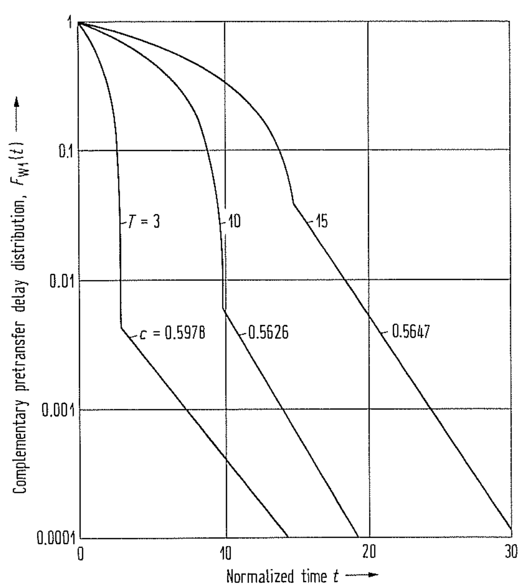


Fig. 10. Complementary pretransfer delay distribution function (FIFO transfer discipline, $c$: coefficient of variation of the pretransfer delay); $S = 10$, $\varrho = 0.5$, $T_0 = 0.5$, $\mu = 1$.

## 5. Conclusion

In this paper, a two-level queueing system has been developed and investigated, which models a clock-driven messaging scheme between the device control unit and the communication control unit of distributed controlled devices in real-time processing systems. The analysis is done using a time effective computing technique, which involves a one-dimensional description and computation of the two-dimensional Markov chain of the state process.

From the steady state distribution, system characteristics like message delays and mean system occupancies are derived and discussed. Finally, the distribution function of message delay according to the FIFO transfer discipline is investigated.

The model can be applied to performance investigations of a wide range of distributed computing systems, in which high rate of real-time messages have to be interchanged between decentralized controlled processors and the critical device response time has to be optimized. Numerical results are provided for dimensioning purposes, whereby critical aspects for the system performance like queue stability conditions and delay characteristics of messages, especially in high load situations, are taken into account.

*Acknowledgements*

The authors would like to express their thanks to Prof. P. J. Kühn for fruitful discussions during the course of this work. The programming efforts of M. Weixler are greatly appreciated.

(Received February 6th, 1985.)

**References**

[1] Manfield, D. R. and Tran-Gia, P., Queueing analysis of scheduled communications phases in distributed processing systems. Proc. Performance '81, Amsterdam 1981, pp. 233–250.

[2] Langenbach-Belz, M., Two-stage queueing system with sampled parallel input queues. Proc. 7th Internat. Teletraffic Congress, Stockholm 1973.

[3] Chu, W. W., Buffer behaviour for batch Poisson arrivals and single constant output. Transact. IEEE COM-**18** [1970], 613–618.

[4] Schwaertzel, H. G., Serving strategies of batch arrivals in common control switching systems. Proc. 7th Internat. Teletraffic Congress, Stockholm 1973.

[5] Jans, H., Traffic analysis of switching system control structures with clocked I/O and priorities. 37th Rep. on Studies in Congestion Theory, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1983.

[6] Tran-Gia, P. and Manfield, D. R., Performance analysis of communication delay optimization in distributed processing environments. Proc. 2nd Internat. Symp. on Perf. of Comp. Comm. Systems, Zürich 1983.

[7] Tran-Gia, P. and Jans, H., Delay analysis of clocked event transfer protocol in distributed processing systems. Proc. 6th Internat. Conf. on Comp. Comm., London 1982.

[8] Burke, P. J., Delays in single server queues with batch input. Operations Res. **23** [1975], 830–833.

[9] Little, J. D. C., A proof of the queueing formula $L = \lambda W$. Operations Res. **9** [1961], 383–387.

[10] Gross, D. and Harris, C. M., Fundamentals of queueing theory. J. Wiley & Sons, New York 1974.