A feedback queueing model for an interactive computer system

by GISAKU NAKAMURA
Musashino Electrical Communication Laboratory, NTT
Musashinoshi, Tokyo, Japan

INTRODUCTION

Recently considerable effort has been directed to the development of computer systems that are able to serve a large number of users in an interactive manner. The model of interactive computer system is described by stating what a single user does during an elementary operation at his console, the "interaction." Roughly stated, an interaction consists of the user requesting and then receiving service from the computer system. The events usually forming an interaction are: the user's thinking, typing at his remote console, waiting for a response from the computer system, and finally watching output. These interactions are repeated until the user finds the desired output. The number of interactions depends on the contents of a job which is processed by the computer system and on the goodness of program which is processed by the user in each interaction. Since this number fluctuates stochastically, it may be considered as a random variable.

The turnaround time is defined as the time interval between the generation of the first request and the reception of the final service from the computer system. Thus, the complete service for a job is made during the turnaround time. For users the turnaround time is one of the most important characteristics of the computer system. There are some characteristics of the interactive computer system which are of operational importance. These characteristics are:

(a) The service time, which is the duration of time required to complete the service for a request in an interaction;
(b) The response time, which is the time interval between the generation of a request in an interaction and the reception of service from the computer system in the same interaction;
(c) The think time, which is the time interval between the reception of service in an interaction and the generation of a request in the next interaction; and
(d) The interaction time, which is the time interval during which an interaction is completed, that is, the sum of the response time and the think time.

In this paper, a simple mathematical model of the operation is proposed for an interactive computer system, and some analyses are made for the turnaround time, the response time, the interaction time, and the number of jobs in the computer system.

MOTIVATION OF THE ANALYSIS

There are various analytical models for time sharing computer systems, which are extensively surveyed by J. M. McKinney. Most of the models have been constructed for the purpose of estimating the response characteristics in each interaction. Some typical models are the round-robin model, the multi-level foreground background model, and the external priority model. The analyses of the models have well explained the servicing behavior for requests in each interaction, and the probability distribution of response time and related characteristics have usually been obtained in postulated time sharing environments. These models, however, are not suitable to explain the servicing behavior for jobs through the complete turnaround time, namely, the models are too complex to be used in estimating the characteristics in connection with the entire sequence of interactions between users and the computer.

Unfortunately, little work has been carried out in analyzing the overall servicing behavior of interactive computer systems from a mathematical point of view, at least to the author's knowledge. The motivation for lack of the analysis is directed toward constructing an
ANALYTICAL MODEL

Assumptions

To construct an analytical model for the interactive computer system, we will make some assumptions on the arrival to the system, on the stochastic behavior of the service times and the think times, and on the number of interactions during the turnaround time. First, it is natural to assume that the sequence of job arrivals constitutes a Poisson process with a constant arrival rate, because jobs arrive at the computer system independently each other. In fact, this assumption has been empirically justified in various situations, and has been adopted in almost all queueing models of time sharing systems. Second, the service times should depend on a scheduling algorithm by which requests in each interaction are processed. But, in a very rough estimation, it may be considered that they are mutually independent and identically distributed. We assume the exponential distribution of service times in the usual way. This is also demonstrated by practical data observed by A. L. Scherr with compatible time sharing systems. Similar considerations can be made on the think times. It is evident that a user's think time in an interaction is independent from his think time in another interaction as well as from another user's think time. Scherr's observation also supports this assumption as an approximation. Here, it is noted that the exponential distribution assumptions of the service times and the think times make the model tractable. Finally, the number of interactions between a user and the computer fluctuates stochastically and may be considered as a random variable. We assume that the probability of having another interaction of a job is independent of the number of interactions preceding it.

Description of the model

Suppose that jobs arrive at the computer system in accordance with a Poisson process with density \( \lambda \). Denote by \( \tau_n \) \((n=1, 2, 3, \ldots)\) the arrival epoch of the \( n \)th job. Then, the interarrival times \( \tau_{n+1} - \tau_n \) \((n=1, 2, 3, \ldots)\) are identically distributed, mutually independent random variables with distribution function

\[
A(t) = \begin{cases} 
1 - e^{-\lambda t} & \text{if } t \geq 0, \\
0 & \text{if } t < 0. 
\end{cases}
\]

The jobs are served by a single processor in order of arrival. The processor is idle if and only if there is no job in the computer system. The service times are supposed to be identically distributed, mutually independent random variables with exponential distribution function

\[
H(t) = \begin{cases} 
1 - e^{-\mu t} & \text{if } t \geq 0, \\
0 & \text{if } t < 0. 
\end{cases}
\]

After being served, each job either returns the computer system with probability \( \gamma \), requesting further service, or goes away permanently with probability \( 1 - \gamma \). The event that a job returns is independent of any other event involved and, in particular, independent of the number of its previous returns. In the case that a job returns the computer system, some delay (the think time) is required before the job joins the queue again. The think times are supposed to be identically distributed, mutually independent random variables with exponential distribution function

\[
G(t) = \begin{cases} 
1 - e^{-\psi t} & \text{if } t \geq 0, \\
0 & \text{if } t < 0. 
\end{cases}
\]

It is supposed that the newly arrived jobs and the returned jobs are equally treated in the computer system. Thus, all jobs are served in order of arrival. The distribution function of service times for the returned jobs is supposed to be equal to that for the newly arrived jobs.

The model described is a kind of single-server queueing model with feedback. A single-server queueing model with feedback has been investigated by L. Takács. In his model, the Poisson arrival is assumed and the feedback rate \( \gamma \) is defined in the same way. The service times are supposed to be identically distributed, mutually independent random variables with general distribution function. From this point of view,
Takács' model is more general than ours. However, immediate returns are required when jobs join the queue again, while some delay is involved in our model.

The model is well described by introducing a virtual thinking system in which infinitely many servers are furnished. Let the whole system consist of the computer system and the thinking system, as is shown in Figure 1. Jobs that arrive at the whole system enter the computer system and join the queue. The jobs are served by a single processor in order of arrival. After being served, each job either immediately enters the thinking system with probability \( \gamma' \) or departs from the whole system with probability \( 1 - \gamma' \).

Since infinitely many servers are furnished in the thinking system, there is no queue in it. Therefore, the service in the thinking system is immediately commenced when a job enters the thinking system. The distribution function of service times in the thinking system is given by \( G(t) \).

After being served, each job immediately enters the computer system and joins the queue. The cycles from computer system to thinking system are repeated until the job departs from the whole system. Then, the probability with which a job has \( n \) returns is given by

\[
    r_n = \gamma^n (1 - \gamma), \quad n = 0, 1, 2, \ldots
\]  

**ANALYSIS**

**The mean number of jobs in the system**

Assume that the whole system is in statistical equilibrium. Let \( \xi \) be the random variable representing the number of jobs in the computer system, and let \( \eta \) be the random variable representing the number of jobs in the thinking system. Denote by \( p(i, j) \) the probability with which \( \xi = i \) and \( \eta = j \). Then, it is obtained that

\[
    \lambda p(0, 0) = \mu(1 - \gamma) p(1, 0), \quad (2)
\]

\[
    (\lambda + \mu + j\nu) p(0, j) = \mu(1 - \gamma) p(1, j) + \mu \gamma p(1, j - 1), \quad j > 0 \quad (3)
\]

\[
    (\lambda + \mu + j\nu) p(i, 0) = \lambda p(i - 1, 0) + \mu(1 - \gamma) p(i + 1, 0) + \mu \gamma p(i - 1, 1), \quad i > 0 \quad (4)
\]

\[
    (\lambda + \mu + j\nu) p(i, j) = \lambda p(i - 1, j) + \mu(1 - \gamma) p(i + 1, j) + \mu \gamma p(i + 1, j - 1) + (j + 1) \nu p(i - 1, j + 1), \quad i > 0, j > 0 \quad (5)
\]

The probability generating function of \( \xi \) and \( \eta \) is defined by

\[
    P(x, y) = E[x^\xi y^\eta] = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} p(i, j) x^i y^j,
\]

where \( E \) represents the mathematical expectation.

Multiplying (2), (3), (4), and (5) by \( 1, \nu x', \nu x^2 \), and \( \nu x^3 \) respectively and summing them, it is derived that

\[
    \nu (y - x) P_x(x, y) + [\lambda (1 - x)] P(x, y) + \mu [1 - (1 - \gamma)/x - \gamma y/x] P(x, y)
\]

\[
    = \mu [1 - (1 - \gamma)/x - \gamma y/x] P(0, y), \quad (6)
\]

where we put

\[
    P_x(x, y) = dP(x, y)/dx.
\]

Let \( L_3 \) be the mean number of jobs in the thinking system. Then,

\[
    L_3 = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} p(i, j) = P_x(1, 1). \quad (7)
\]

\( L_3 \) is easily found in the following way. Putting \( x = 1 \) in (6), it is obtained that

\[
    \nu (y - 1) P_x(1, y) + \mu \gamma (1 - y) P(1, y)
\]

\[
    = \mu \gamma (1 - y) P(0, y). \quad (8)
\]

Then, by differentiating both sides of (8) with respect to \( y \) and then putting \( y = 1 \), it is obtained that

\[
    \nu P_x(1, 1) = \mu \gamma \{P(1, 1) - P(0, 1)\}. \quad (9)
\]

Substituting (7) and \( P(1, 1) = 1 \) in (9), it is obtained that

\[
    L_3 = \nu \gamma [1 - P(0, 1)]. \quad (10)
\]

Here, \( P(0, 1) \) is found as follows. Put \( y = x \) in (6). Then,

\[
    (1 - x) [\mu (1 - \gamma) - \lambda x] P(x, x)
\]

\[
    = (1 - x) \mu (1 - \gamma) P(0, x). \quad (11)
\]

By differentiating both sides of (11) with respect to \( x \) and then putting \( x = 1 \), it is obtained that

\[
    P(0, 1) = 1 - \lambda / (1 - \gamma) \mu. \quad (12)
\]

Hence, by substituting (12) in (10), it is obtained that

\[
    L_3 = \nu \gamma (1 - 1/\gamma). \quad (13)
\]

Next, let \( L_2 \) be the mean number of jobs in the computer system. Then,

\[
    L_2 = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} i \left( \sum_{j=0}^{\infty} p(i, j) \right) = P_x(1, 1) \quad (14)
\]

where we put

\[
    P_x(1, 1) = P_x(x, y) \big|_{x=1, y=1} = dP(x, y)/dx \big|_{x=1, y=1}.
\]

Since

\[
    dP(x, x)/dx = P_x(x, x) + P_y(x, x),
\]

it is derived that

\[
    dP(x, x)/dx \big|_{x=1} = L_1 + L_2. \quad (15)
\]
Differentiating both sides of (11) two times with respect to \( x \) and then putting \( x = 1 \), it is obtained that
\[
\lambda - \{ \mu (1 - \gamma) - \lambda \} (L_1 + L_2) = -\mu (1 - \gamma) P_y(0, 1). \tag{16}
\]
Here, it is noted that
\[
dP(0, x)/dx = P_y(0, x).
\]
Thus, (16) yields
\[
L_1 + L_2 = \rho/(1 - \rho) + P_y(0, 1)/(1 - \rho), \tag{17}
\]
where we put
\[
\rho = \lambda/(1 - \gamma) \mu.
\]
\( \rho \) is the utilization factor of the computer system, because
\[
\lambda (1 + \gamma + \gamma^2 + \cdots) = \lambda/(1 - \gamma)
\]
is the arrival rate and \( 1/\mu \) is the mean service time.\(^4\)
The second term in the right hand side of (17) is the mean number of jobs in the thinking system, under the condition that there is no job in the computer system. Then, if the state in the thinking system is stochastically independent of that in the computer system, \( L_2 \) may be given by
\[
L_2 = P_y(1, 1) = P_y(0, 1)/(1 - \rho).
\]
Under the assumption of this stochastic independence, \( L_1 \) may be given by
\[
L_1 = \rho/(1 - \rho). \tag{18}
\]
In the following section we will prove the stochastic independence between the states in the computer system and in the thinking system.

**Method of finding \( P(x, y) \)**

Assume that the states in the computer system and in the thinking system are stochastically independent. Then, the probability generating function \( P(x, y) \) is factorized as
\[
P(x, y) = P(x, 1)P(1, y). \tag{19}
\]
We will show that \( P(x, y) \) given by (19) becomes the solution of (6) if \( P(x, 1) \) and \( P(1, y) \) are suitably chosen.

Denote by \( M/M/1 \) the single-server queueing system with a Poisson arrival and exponential service times.\(^5\) It is known that the mean number of jobs in the system \( M/M/1 \) is given by \( \rho/(1 - \rho) \), where \( \rho \) is the utilization factor of the system. This formula coincides with the first term in the right hand side of (17). Thus, it is suspected that the computer system forms the system \( M/M/1 \) from the queueing point of view. Since the probability generating function of the number of jobs in the system \( M/M/1 \) is given by \( (1 - \rho)/(1 - \rho x) \), we put
\[
P(x, 1) = (1 - \rho)/(1 - \rho x). \tag{20}
\]
Next, denote by \( M/M/\infty \) the infinitely many-server queueing system with a Poisson arrival and exponential service times.\(^5\) It is also known that the mean number of jobs in the system \( M/M/\infty \) is given by \( \lambda'/\nu \), where \( \lambda' \) is the arrival rate and \( 1/\nu \) is the mean service time. Now, under the assumption of the stochastic independence, the second term in the right hand side of (17) is written as
\[
\lambda'/(1 - \gamma) = \lambda(\gamma + \gamma^2 + \gamma^3 + \cdots)
\]
is the arrival rate of the thinking system. Thus, it is suspected that the thinking system forms the system \( M/M/\infty \) from the queueing point of view. Since the probability generating function of the number of jobs in the system \( M/M/\infty \) is given by \( \exp\{-(\lambda'/\nu)(1 - x)\} \), we put
\[
P(1, y) = e^{-\beta(1-y)} \tag{21}
\]
where
\[
\beta = \lambda\gamma/((1 - \gamma)\nu).
\]
Substituting (20) and (21) in (19), it is obtained that
\[
P(x, y) = P(x, 1)P(1, y) = (1 - \rho)e^{-\beta(1-y)} (1 - \rho x) \tag{22}
\]
To show that \( P(x, y) \) given by (22) is the solution of (6), put
\[
A_1 = v(y-x)P_y(x, y),
\]
\[
A_2 = \mu[1 - (1 - \gamma)/(x - \gamma y/x)]P(x, y),
\]
\[
A_3 = \mu[1 - (1 - \gamma)/(x - \gamma y/x)]P(0, y).
\]
Then, it is sufficient to prove that
\[
A_1 + A_2 - A_3 = 0.
\]
Since \( A_1 \) and \( A_3 \) are calculated as
\[
A_1 = v\beta(y-x)P(x, y),
\]
\[
A_3 = \mu[1 - (1 - \gamma)/(x - \gamma y/x)](1 - \rho x)P(x, y),
\]
it is derived that
\[
(A_1 + A_2 - A_3)/P(x, y) = \lambda\gamma(y-x)/(1 - \gamma) + \lambda(1-x) + \lambda[x - (1 - \gamma) - \gamma y]/(1 - \gamma) = 0.
\]
Thus, (22) becomes the solution of (6) and our conjecture that the states in the computer system and in the thinking system are stochastically independent is justified.

In the above analysis, it is shown that the computer system and the thinking system form the systems $M/M/1$ and $M/M/\infty$ respectively. This fact can be intuitively explained in the following way. Assume that the computer system forms the system $M/M/1$ from the queueing point of view. It is known that the output process of the system $M/M/1$ is a Poisson process. 6 It is also known that the probabilistic selection of jobs from a Poisson process results in a Poisson process. 7 Hence, the input process of the thinking system becomes a Poisson process and then the thinking system forms the system $M/M/\infty$ from the queueing point of view. Since the output process of the system $M/M/\infty$ is a Poisson process, and the aggregation of several independent Poisson processes results in a Poisson process, 7 the input process of the computer system becomes a Poisson process. Thus, the computer system forms the system $M/M/1$ from the queueing point of view, which is our first assumption. Therefore, no contradiction is derived, and our assumption is justified. This intuitive argument will be used later.

**The mean turnaround time**

The turnaround time is defined as the time interval between the generation of the first request and the reception of the final service from the computer system. In other words, the turnaround time is the time interval during which a job stays in the whole system. The mean turnaround time is one of the most important characteristics for users.

Denote by $T(n)$, ($n=0, 1, 2, \ldots$), the mean turnaround time for jobs with $n$ returns. Since the probability with which a job has $n$ returns is given by (1), the arrival rate for jobs with $n$ returns is written as

$$\lambda(n) = \lambda r_n = \lambda \gamma^n (1-\gamma), \quad n=0, 1, 2, \ldots \quad (23)$$

Let $L_1(n)$ be the mean number of jobs with $n$ returns in the computer system, and let $L_2(n)$ be that in the thinking system. Then, by applying Little’s theorem to the whole system, 6 it is obtained that

$$T(n) = \frac{L_1(n) + L_2(n)}{\lambda(n)}, \quad n=0, 1, 2, \ldots \quad (24)$$

We will find $L_1(n)$ and $L_2(n)$.

Let $L_1(n, k)$ be the mean number of jobs with $n$ returns in the computer system, under the condition that those jobs already have $k (k \leq n)$ returns. And let $L_2(n, k)$ be that in the thinking system under the same condition. Then, it is evident that

$$L_1(n) = \sum_{k=0}^{n} L_1(n, k), \quad (25)$$

$$L_2(n) = \sum_{k=0}^{n-1} L_2(n, k). \quad (26)$$

In statistical equilibrium, the input rate in any system is equal to the output rate in the same system. Now, define

$$u_1(n, k) = \frac{L_1(n, k)}{L_1}, \quad u_2(n, k) = \frac{L_2(n, k)}{L_2}.$$ 

Then, by equating the input rate of jobs with $n$ returns in the computer system and the output rate, it is obtained that

$$\lambda \gamma^n (1-\gamma) = \frac{\lambda}{(1-\gamma)} u_1(n, n), \quad (27)$$

$$\lambda \gamma^n (1-\gamma) u_2(n, k-1) = \frac{\lambda}{(1-\gamma)} u_1(n, k), \quad n \geq k \geq 1. \quad (28)$$

Here, it is noted that the input rates in the computer system and in the thinking system are $\lambda/(1-\gamma)$ and $\lambda/(1-\gamma)$ respectively, as is shown in Figure 2. From (27) and (28), it is derived that

$$u_1(n, n) = \gamma^n (1-\gamma)^2, \quad (29)$$

$$\gamma u_2(n, k-1) = u_1(n, k), \quad n \geq k \geq 1. \quad (30)$$

Similarly, by equating the input rate of jobs with $n$ returns in the thinking system and the output rate, it is obtained that

$$\lambda/(1-\gamma) u_1(n, k) = \lambda/(1-\gamma) u_2(n, k), \quad n \geq k \geq 0. \quad (31)$$

From (31), it is derived that

$$u_1(n, k) = \gamma u_2(n, k), \quad n \geq k \geq 0. \quad (32)$$

Using (29), (30), and (32), $u_1(n, k)$ and $u_2(n, k)$ are

![Figure 2—The aspect of flows in the system](image-url)
given by
\[ u_1(n, k) = \gamma^n (1 - \gamma)^k, \quad n \geq k \geq 0, \]
\[ u_2(n, k) = \gamma^{n-1} (1 - \gamma)^k, \quad n > k \geq 0. \]

Then, by the definitions of \( u_1(n, k) \) and \( u_2(n, k) \), it is obtained that
\[ L_1(n, k) = \gamma^n (1 - \gamma)^2 L_1, \quad (33) \]
\[ L_2(n, k) = \gamma^{n-1} (1 - \gamma)^2 L_2. \quad (34) \]

Here, it is noted that \( L_1(n, k) \) and \( L_2(n, k) \) do not depend on \( k \). By substituting (33), (34) in (25), (26) respectively, it is obtained that
\[ L_1(n) = (n+1) \gamma^n (1 - \gamma)^2 L_1, \quad (35) \]
\[ L_2(n) = n \gamma^{n-1} (1 - \gamma)^2 L_2. \quad (36) \]

Then, by using (23), (24), (35), and (36), the mean turnaround time \( T(n) \) for jobs with \( n \) returns is given by
\[ T(n) = \frac{(n+1) L_1}{\lambda (1 - \gamma)} + \frac{n L_2}{\lambda \gamma / (1 - \gamma)}. \quad (37) \]

Now, we will consider the meaning of (37). Let \( R \) be the mean response time of the interactive computer system, and let \( K \) be the mean think time. It is shown that the input rate in the computer system is \( \lambda / (1 - \gamma) \), and the mean number of jobs in that system is denoted by \( L_1 \). Then, by applying Little’s theorem to the computer system, it is obtained that
\[ R = \frac{L_1}{\lambda / (1 - \gamma)}. \quad (38) \]

Similarly, the input rate in the thinking system is \( \lambda \gamma / (1 - \gamma) \) and the mean number of jobs in that system is \( L_2 \), then, by applying Little’s theorem to the thinking system,
\[ K = \frac{L_2}{\lambda \gamma / (1 - \gamma)} = 1 / \nu. \quad (39) \]

Hence, (37) is written as
\[ T(n) = R + n(R + K). \quad (40) \]

Here, \((R + K)\) is the mean interaction time for the interactive computer system. Thus, the mean turnaround time for jobs with \( n \) returns is the sum of the mean response time and \( n \) times the mean interaction time. \( R, K, \) and \( T(n) \) are shown in Figures 3, 4, and 5 respectively. In the case that a job has no return, of course, the mean turnaround time coincides with the mean response time.

Finally, the mean turnaround time for any job, regardless of the number of its returns, is given by
\[ T = \sum_{n=0}^{\infty} r_n T(n) = R + (1 - \gamma) (R + K) \sum_{n=1}^{\infty} n \gamma^n \]
\[ = R + \frac{1}{\nu} (R + K), \quad (41) \]
which coincides with \((L_1 + L_2) / \lambda\).

**Extension to the interactive computer system with multiple processors**

It is easy to extend the previous analysis to the interactive computer system with multiple processors. Suppose that there are \( s \) processors in the computer system. Then, the computer system forms the system
\[ K = \frac{L_2}{\lambda \gamma / (1 - \gamma)} = 1 / \nu. \quad (39) \]

Hence, (37) is written as
\[ T(n) = R + n(R + K). \quad (40) \]

Here, \((R + K)\) is the mean interaction time for the interactive computer system. Thus, the mean turnaround time for jobs with \( n \) returns is the sum of the mean response time and \( n \) times the mean interaction time. \( R, K, \) and \( T(n) \) are shown in Figures 3, 4, and 5 respectively. In the case that a job has no return, of course, the mean turnaround time coincides with the mean response time.

Finally, the mean turnaround time for any job, regardless of the number of its returns, is given by
\[ T = \sum_{n=0}^{\infty} r_n T(n) = R + (1 - \gamma) (R + K) \sum_{n=1}^{\infty} n \gamma^n \]
\[ = R + \frac{1}{\nu} (R + K), \quad (41) \]
which coincides with \((L_1 + L_2) / \lambda\).
M/M/s from the queueing point of view, where M/M/s means the s servers queueing system with a Poisson arrival and exponential service times. It is known that the output process of the system M/M/s is a Poisson process. Then, the intuitive argument given for the analysis of the computer system with a single processor is entirely applied to the analysis of the computer system with s processors. The probability generating function of the number of jobs in the system M/M/s is given by

\[ p_0 \left( \sum_{k=0}^{s} (\rho x)^k / k! \right) \sum_{k=0}^{\infty} (\rho x)^k / s! e^{x-a} \]  

where \( p_0 \) is the probability with which there is no job in the system M/M/s. \( p_0 \) is calculated by

\[ p_0 = 1 / \left( \sum_{k=0}^{s} (\rho x)^k / k! + \rho^s / (s-1)! (s-\rho) \right) \]  

(42)

Then, if we use

\[ P(x, 1) = p_0 \left( \sum_{k=0}^{s} (\rho x)^k / k! \right) + \sum_{k=0}^{\infty} (\rho x)^k / s! e^{x-a} \]  

(43)

instead of (20), the probability generating function \( P(x, y) \) is factorized as

\[ P(x, y) = P(x, 1) P(1, y) \]

where \( P(1, y) \) is given by (21). In this case, the mean number of jobs in the computer system is given by

\[ L_1 = \frac{\rho^{s+1}}{(s-1)! \sum_{k=0}^{s} (\rho^s / k!) \{(s-k)^2-k\}} \]  

(44)

Then, the previous results (37)-(41) still hold for the interactive computer system with s processors.

**DISCUSSION**

As an analytical model for the interactive computer system, the paper has proposed a feedback queueing model in which some delay is required before jobs join the queue again. From the analysis of the model, the mean turnaround time is related to the mean response time and the mean think time in a very simple way. Although the validity of this simple relation largely depends on the exponential distribution assumptions for the service times and the think times, it is considered that the result obtained is a good approximation of actual behavior. In fact, Equation (40) which gives the relation can be intuitively justified and can be empirically recognized.

The exponential distribution assumption for the service times is frequently adopted in various queueing models. This is due to the tractability of models as well as to the reasonability of the assumption. Sometimes we may be interested in queueing models with non-exponential distribution assumption. These models usually become hardly tractable in a theoretical way, when they are slightly complicated. However, it is well known that the adoption of the exponential distribution assumption causes results to be on the safe side.

**SUMMARY**

The paper has proposed a simple mathematical model of the operation for an interactive computer system with a single processor, and has presented some characteristics of the system, such as the mean turnaround time, the mean interaction time, etc. From the queueing point of view, the proposed model is a kind of single-server queueing model with feedback. But, unlike the usual queueing models with feedback, the proposed model requires some delay when a job returns the queueing system. This delay represents the user's think time in the interactive computer system.

The model is well described by introducing a virtual thinking system in which infinitely many servers are furnished. Thus, the whole system consists of the computer system and the thinking system. Here, the user's thinking is represented by the service in the thinking system. The analysis is first made for the
mean number of jobs in the thinking system and for that in the computer system, under the assumptions of a Poisson arrival and exponential service times. Then, the probability generating function of the number of jobs is derived by solving a differential equation. It is shown that the states in the computer system and in the thinking system are stochastically independent, and the computer system and the thinking system form the systems M/M/1, M/M/∞ respectively.

ACKNOWLEDGMENTS

The author would like to thank Dr. N. Ikeno, Mr. K. Naemura, and Mr. Y. Yoshida for numerous discussions and suggestions which aided in the preparation of this paper.

REFERENCES

1 J M MC KINNEY
A survey of analytical time-sharing models
Computing Surveys Vol 1 No 2 1969

2 A L SCHERR
An analysis of time-shared computer systems
Research Monograph No 36 MIT Cambridge Massachusetts

3 L TAKÁCS
A single-server queue with feedback
Bell System Technical Journal Vol 42 No 2 1963

4 T L SAATY
Elements of queuing theory

5 D G KENDALL
Stochastic processes occurring in the theory of queues and their analysis by the method of the embedded Markov chain
Annals of Mathematical Statistics Vol 24 No 3 1953

6 P J BURKE
The output of a queuing system
Operations Research Vol 4 No 6 1956

7 R W CONWAY W L MAXWELL L W MILLER
Theory of scheduling Chap 8
Addison-Wesley Reading Massachusetts Palo Alto London Don Mills Ontario 1967

8 J D C LITTLE
A proof for the queuing formula: \( L = \lambda W \)
Operations Research Vol 9 No 3 1961