ELSEVIER

Computer Networks 31 (1999) 1879–1889

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

Analysis of a general limited scheduling mechanism for a distributed communication system [1]

Lain-Chyr Hwang [a], Chung-Ju Chang [b,*]

[a] *Network Planning Department, Telecommunication Labs., Ministry of Transportations and Communications, Hsinchu, Taiwan*
[b] *Department of Communication Engineering and Center for Telecommunications Research, National Chiao Tung University, Hsinchu 300, Taiwan*

Abstract

This paper studies an alternative *scheduling mechanism* of *general limited* service discipline for a distributed communication system, where the distributed system has a modular architecture and the module has tasks in the queue. We successfully analyze the system by way of imbedded Markov chains and use a recursive method to derive the mean waiting time of task. The arrival processes for modules are assumed to be Poisson processes; the service time of a task in queues and the walking time between queues are assumed to be generally distributed. We also presents an optimal-pattern design of the general limited scheduling mechanism for the distributed communication system via a genetic algorithm. A distributed communication system, if designed with the general limited scheduling mechanism and its optimal pattern, can be flexible to meet the system requirements as much as possible. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Scheduling mechanism; Limited; Imbedded Markov chain; Distributed communication system

1. Introduction

A communication system has many resources of various kinds; a switching system, for example, has modules of subscriber line, trunk line, service trunk, switching network, and central processor [10]. And a communication system is characterized by real-time, high quality and reliability, and flexibility for implementing additional functions. To achieve these characteristics, one of the cost-effective manners is to have a distributed modular architecture for the communication system.

Usually, there is a kernel (an operating system) in a distributed communication system that deals with the scheduling problem of tasks coming from different queues (modules). Tasks within queues are arranged as processes and are administered by the operating system via process queues. The operating system has to perform the task (process) functions under real-time conditions. Therefore, there were scheduling schemes of interrupt-driven and priorities adopted [8,13].

* Corresponding author. Tel.: +886-3-5731923; fax: +886-3-5710116; E-mail: cjchang@cc.nctu.edu.tw.
1 This work is supported by National Science Council, Taiwan, under contract number NSC-83-0404-E-009-002.

PII: S0169-7552(98)00288-8

This paper presents an alternative scheduling mechanism of general limited service discipline [1,4] for a distributed communication system. In the general service order sequence, the repetition of multiple polls of a process queue and the sequence length are equivalent to the priority class and the duration of interrupt timer, respectively. On the other hand, in the limited service discipline, a limited number of tasks in a process queue is executed as the processor attends the queue; the limited service discipline is regarded as the most realistic one in all the service disciplines [2,3,5,6,8,9,14,15].

We describe the system model in Section 2. We successfully obtain the mean waiting time of a task via an imbedded Markov chain approach and use a new recursive method to find parameters in formula for the mean waiting time in Section 3. Section 4 illustrates a numerical example and presents an optimal-pattern design of the general limited scheduling mechanism for a distributed communication system by using a genetic algorithm (GA) [7]. Finally, some concluding remarks are given in Section 5.

2. System model

Consider a distributed communication system which has R process queues. The scheduling mechanism for the process queues is assumed to be with general service order sequence, where a high priority queue is polled more often, depending on the system requirements. For example, a distributed communication system has four process queues, say queue {1, 2, 3, 4}. If the queue 1 is with high priority and is designed to be polled three times in a whole cycle, the service order sequence could be {1, 2, 1, 3, 1, 4}. The turn in the sequence is called stage; the number of turns in the sequence is denoted by P. On the other hand, the service discipline of each stage is assumed to be limited, either gated-limited (G-limited) or exhaustive-limited (E-limited). The limitation number for stage i is assumed to be k_i such that there are at most k_i tasks served in stage i for an attendance (arrival) of the processor at stage i. The service disciplines and limitation numbers can be different even for stages corresponding to the same queue.

Let r and i denote the indexes of a process queue and a stage, respectively, and let r_i stand for the underlying process queue of stage i. The arrival process for queue r is an independent Poisson process with rate λ_r; the service time S_r of a task at queue r follows an independent and general distribution with the first moment (mean) s_r and the second moment $s_r^{(2)}$, $1 \le r \le R$. The traffic intensity for queue r is $\rho_r = \lambda_r s_r$ and the total traffic intensity is $\rho = \sum_{r=1}^{R} \rho_r$. The walking time U_i, defined as the time from the processor's departure from stage i to the processor's arrival at stage $i+1$, follows an independent and general distribution with the first moment u_i and the second moment $u_i^{(2)}$, $1 \le i \le P$, and the total mean walking time is $u = \sum_{i=1}^{P} u_i$. Here, the index corresponding to stage is with modulo-P arithmetic which equals to P if the remainder is zero.

3. Analysis

3.1. Relations among observation points

Some notations for stage i, $1 \le i \le P$, at four observation points are first defined in the following:

$a_i(t)$ = the number of processor's arrivals at stage i during $(0,t)$;
$b_i(t)$ = the number of service beginnings of i-tasks during $(0,t)$;
$d_i(t)$ = the number of processor's departures from stage i during $(0,t)$;
$e_i(t)$ = the number of service endings of i-tasks during $(0,t)$;
$a_i(n,t)$ = the number of processor's arrivals at stage i when there are n tasks in queue r_i during $(0,t)$;
$b_i(n,t)$ = the number of service beginnings of i-tasks when there are n tasks in queue r_i during $(0,t)$;
$d_i(n,t)$ = the number of processor's departures from stage i when there are n tasks in queue r_i during $(0,t)$;
$e_i(n,t)$ = the number of service endings of i-tasks when there are n tasks in queue r_i during $(0,t)$.

Time 0 in above definitions is an arbitrarily selected initial time. The i-task is the task receiving service at stage i.

In a process queue, the point of task's service beginning is the point of processor's arrival or the point of task's service ending; similarly, the point of task's service ending is the point of task's service beginning or the point of processor's departure. Thus, during $(0,t)$, the number of the service beginnings of i-tasks when there are n tasks in queue r_i, $b_i(n,t)$, plus the number of processor's departures when there are n tasks in queue r_i, $d_i(n,t)$, equals the number of service endings of i-tasks when there are n tasks in queue r_i, $e_i(n,t)$, plus the number of processor's arrivals when there are n tasks in queue r_i, $a_i(n,t)$. That is

$$b_i(n,t) + d_i(n,t) = e_i(n,t) + a_i(n,t). \tag{1}$$

Dividing Eq. (1) by the number of processor's arrivals at stage i during the time interval $(0,t)$, $a_i(t)$, and taking the limit as t goes to infinity, Eq. (1) can be expressed as

$$b_i(n) \cdot \beta_i + d_i(n) = e_i(n) \cdot \beta_i + a_i(n), \tag{2}$$

where

$$x_i(n) \equiv \lim_{t\to\infty} \frac{x_i(n,t)}{x_i(t)}, \tag{3}$$

x could be b, d, e, or a, and

$$\begin{cases} \beta_i \equiv \lim\limits_{t\to\infty} \dfrac{b_i(t)}{a_i(t)} = \lim\limits_{t\to\infty} \dfrac{e_i(t)}{a_i(t)}, \\ \lim\limits_{t\to\infty} \dfrac{d_i(t)}{a_i(t)} = 1. \end{cases} \tag{4}$$

Noticeably, $b_i(n)$, $d_i(n)$, $e_i(n)$, and $a_i(n)$ are the probabilities that there are n tasks in queue r_i when an i-task begins service, when the processor departs from stage i, when an i-task ends service, and when the processor arrives at stage i, respectively; and β_i is the mean number of i-tasks served in a whole cycle. Let $\hat{b}_i(z)$, $\hat{d}_i(z)$, $\hat{e}_i(z)$, and $\hat{a}_i(z)$, denote the probability generating functions (pgfs) of the random variables with probability mass functions (pmfs) of $b_i(n)$, $d_i(n)$, $e_i(n)$, and $a_i(n)$, respectively. Then Eq. (2) can be expressed as

$$\beta_i \cdot \hat{b}_i(z) + \hat{d}_i(z) = \beta_i \cdot \hat{e}_i(z) + \hat{a}_i(z). \tag{5}$$

Note that Eq. (5) is valid for all service disciplines ([4], Eq. (14)) and $\hat{a}_i(1)$, $\hat{b}_i(1)$, $\hat{d}_i(1)$, and $\hat{e}_i(1)$ are all equal to 1.

3.2. Mean waiting times

Intuitively, the mean waiting time of tasks at queue r, υ_r, $1 \le r \le R$, in a distributed communication system can be obtained by

$$\upsilon_r = \frac{1}{\lambda_r c} \sum_{\{i|r_i=r\}} \beta_i \cdot w_i, \tag{6}$$

where w_i is the mean waiting time of i-task and c is the mean whole cycle time. The mean whole cycle time is given by

$$c = \sum_{\{i|r_i=r\}} c_i = \frac{u}{1-\rho}, \tag{7}$$

where c_i is the mean pseudocycle time for stage i which is defined as the mean time from the processor's last arrival at queue r_i to the processor's present arrival at stage i. It can be seen from Eq. (6) that the mean waiting times for queues can be obtained from the mean waiting times for stages. Hence, the mean waiting times for stages are here derived. Some notations are defined for the derivation.

Y_i: the system time of an i-task; the system time of an i-task is defined as the time an i-task spending in the system;
$Y_i^*(s)$ = the Laplace–Stieltjes transform (LST) of the cumulative distribution function (CDF) of Y_i;
$\hat{Y}_i(z)$ = the pgf of the number of tasks arriving at queue r_i during Y_i;
$S_r^*(s)$ = LST of CDF of S_r;
$\hat{S}_r(z)$ = the pgf of the number of tasks arriving at queue r during S_r;
$W_i^*(s)$ = LST of CDF of the waiting time for i-task.

The number of tasks in queue r_i when an i-task ends service is equal to the number of tasks in queue r_i when this i-task begins service, plus the number of tasks arriving at queue r_i during the service time of this i-task, and minus the just served task. Thus $\hat{b}_i(z)$ can be expressed as

$$\hat{b}_i(z) = \frac{\hat{e}_i(z)\cdot z}{\hat{S}_{r_i}(z)}. \tag{8}$$

Substitute Eq. (8) into Eq. (5) and obtain $\hat{e}_i(z)$ by

$$\hat{e}_i(z) = \frac{\hat{S}_{r_i}(z)\left[\hat{a}_i(z) - \hat{d}_i(z)\right]}{\beta_i\left[z - \hat{S}_{r_i}(z)\right]}. \tag{9}$$

On the other hand, the number of tasks arriving at queue r_i during the system time of an i-task is equal to the number of tasks at queue r_i when this task departs from stage i. This yields

$$\hat{Y}_i(z) = \hat{e}_i(z). \tag{10}$$

Since the arrival process is Poisson process, pgf $\hat{Y}_i(z)$ in Eq. (10) and pgf $\hat{S}_{r_i}(s)$ in Eq. (9) can be transformed into LSTs of $Y_i^*(s)$ and $S_{r_i}^*(s)$ by replacing z with $1 - s/\lambda_{r_i}$, and $Y_i^*(s)$ can be obtained by

$$Y_i^*(s) = \frac{S_{r_i}^*(s)\left[\hat{a}_i\left(1 - \frac{s}{\lambda_{r_i}}\right) - \hat{d}_i\left(1 - \frac{s}{\lambda_{r_i}}\right)\right]}{\beta_i\left[1 - \frac{s}{\lambda_{r_i}} - S_{r_i}^*(s)\right]}.$$

Consequently, $W_i^*(s)$ is gotten by

$$W_i^*(s) = \frac{Y_i^*(s)}{S_{r_i}^*(s)} = \frac{\left[\hat{a}_i\left(1 - \frac{s}{\lambda_{r_i}}\right) - \hat{d}_i\left(1 - \frac{s}{\lambda_{r_i}}\right)\right]}{\beta_i\left[1 - \frac{s}{\lambda_{r_i}} - S_{r_i}^*(s)\right]}, \tag{11}$$

and from L'Hospital law, w_i is derived by

$$w_i = \frac{\lambda_{r_i}}{2\beta_i(1-\rho_{r_i})}\cdot\left[\frac{\left(\hat{a}_i'(1) - \hat{d}_i'(1)\right)s_{r_i}^{(2)}}{1-\rho_{r_i}} + \frac{\hat{a}_i''(1) - \hat{d}_i''(1)}{\lambda_{r_i}^2}\right]. \tag{12}$$

Then the mean waiting time of tasks at queue r, v_r, can be obtained by Eq. (6) accordingly. Note that Eq. (12) is indifferent to service disciplines of the distributed communication system, but moments in the mean waiting time, $\hat{a}_{i'}(1)$, $\hat{a}i''(1)$, $\hat{d}i'(1)$, and $\hat{d}i''(1)$, may vary for different service disciplines.

3.3. Moments in the mean waiting time

To find the moments of $\hat{a}_i'(1)$, $\hat{a}_i''(1)$, $\hat{d}_i'(1)$, and $\hat{d}_i''(1)$ and β_i in Eq. (12), all stages have to be observed. The following notations are further defined:

$\hat{\boldsymbol{S}}_r(Z)$ = the pgf of the numbers of tasks arriving at all independent queues during S_r, where $Z = (z_1, z_2, \ldots, z_R)$;
$\hat{\boldsymbol{a}}_i(Z)$ = the pgf of the numbers of tasks in all independent queues when the processor arrives at stage i; i.e., $\hat{\boldsymbol{a}}_i(Z) = \sum_A \boldsymbol{P}_i(A) Z^A$, where $\hat{\boldsymbol{P}}_i(A)$ is the probability of $A \equiv (a_1, a_2, \ldots, a_R)$ when the processor arrives at stage i and $Z^A \equiv z_1^{a_1} z_2^{a_2} \cdots z_R^{a_R}$;
$\hat{\boldsymbol{d}}_i(Z)$ = the pgf of the numbers of tasks in all independent queues when the processor departs from stage i;
$\hat{\boldsymbol{U}}_i(Z)$ = the pgf of the numbers of tasks arriving at all independent queues during U_i;
ν_i = the number of tasks served at stage i in a cycle;
η_r^i = the number of tasks arriving at queue r during serving stage i.

The number of tasks in queue r when the processor arrives at stage $i+1$ is equal to the number of tasks in queue r when the processor departs form stage i plus the number of tasks arriving at queue r during U_i, $1 \le r \le R$, $1 \le i \le P$. The relation can be expressed as

$$\hat{\boldsymbol{a}}_{i+1}(Z) = \hat{\boldsymbol{d}}_i(Z) \cdot \hat{\boldsymbol{U}}_i(Z). \tag{13}$$

It is also known that the distribution of ν_i be different for the two limited service disciplines. If there are a_{r_i} customers in station r_i when the server arrives at stage i, ν_i is $\min(a_{r_i}, k_i)$ for G-limited service discipline but is $\min(a_{r_i} + \eta_{r_i}^i, k_i)$ for E-limited service discipline. On the other hand, if there are a_r customers in station r when the server arrives at stage i, the number of customers in station r when the server departs from stage i is equal to $a_r + \eta_r^i$ for $r \ne r_i$ but is equal to $a_r + \eta_r^i - \nu_i$ for $r = r_i$, for both disciplines, $1 \le r \le R$. Hence, the relationship between $\hat{\boldsymbol{a}}_i(Z)$ and $\hat{\boldsymbol{d}}_i(Z)$ should not be the same for these two limited service disciplines.

3.3.1. Gated-limited service discipline

If stage i adopts the G-limited service discipline and there are a_{r_i} customers in station r_i when the server arrives at stage i, the processor, at the attendance, will serve $\min(a_{r_i}, k_i)$ tasks and leave $\max(0, a_{r_i} - k_i)$ tasks to the service of its next attendance. The pgf of the numbers of tasks arriving at all independent queues during serving stage i is $\hat{\boldsymbol{S}}_{r_i}^{\max(a_{r_i}, k_i)}(Z)$. Thus $\hat{\boldsymbol{d}}_i(Z)$ is given by

$$\begin{aligned} \hat{\boldsymbol{d}}_i(Z) &= \sum_{a_1=0}^{\infty} \cdots \sum_{a_R=0}^{\infty} \boldsymbol{P}_i(A)\, z_1^{a_1} \cdots \hat{\boldsymbol{S}}_{r_i}^{\min(a_{r_i}, k_i)}(Z)\, z_{r_i}^{\max(0, a_{r_i} - k_i)} \cdots z_R^{a_R} \\ &\equiv \mathscr{G}_i[\hat{\boldsymbol{a}}_i(Z)], \end{aligned} \tag{14}$$

where $\mathscr{G}_i$ is defined as an operator on $\hat{\boldsymbol{a}}_i(Z)$ that transforms $z_{r_i}^{a_{r_i}}$ of $\hat{\boldsymbol{a}}_i(Z)$ to $\hat{\boldsymbol{S}}_{r_i}^{a_{r_i}}(Z)$ if $a_{r_i} < k_i$ but transforms $z_{r_i}^{a_{r_i}}$ to $\hat{\boldsymbol{S}}_{r_i}^{k_i}(Z) z_{r_i}^{a_{r_i} - k_i}$ if $a_{r_i} \ge k_i$. Therefore, $\hat{\boldsymbol{a}}_{i+1}(Z)$ in Eq. (13) can be expressed as

$$\hat{\boldsymbol{a}}_{i+1}(Z) = \mathscr{G}_i[\hat{\boldsymbol{a}}_i(Z)] \cdot \hat{\boldsymbol{U}}_i(Z). \tag{15}$$

3.3.2. Exhaustive-limited service discipline

If stage i adopts the E-limited service discipline, the processor, after completing service of an i-task, will continue to serve the next i-task on condition that the number of served i-task is less than k_i and there are still i-tasks in queue r_i, otherwise it will stop serving and leave stage i. The pgf of the numbers of tasks arriving at all independent queues during an i-task service is $\hat{\boldsymbol{S}}_{r_i}(Z)$, and denote such an i-task service process that results in an increment of tasks in all queues by an operator $\mathscr{J}_i$. When the number of tasks served at stage i, ν_i, is less than k_i, the E-limited service discipline can still be considered as k_i repetitions of i-task service if the later

$k_i - \nu_i$ tasks are regarded as pseudo-ones. The k_i repetition of i-task service processes is denoted by $\mathscr{J}_i^{k_i}[\cdot]$ and is expressed as

$$\mathscr{J}_i^{k_i}[\cdot] = \mathscr{J}\Bigg[\mathscr{J}_i\Bigg[\cdots\Bigg[\mathscr{J}_i\Bigg[\underbrace{\mathscr{J}_i[\cdot]}_{k_i}\Bigg]\Bigg]\cdots\Bigg]\Bigg]$$

Therefore, $\hat{\boldsymbol{d}}_i(Z)$ is given by

$$\hat{\boldsymbol{d}}_i(Z) = \sum_{a_1=0}^{\infty}\cdots\sum_{a_R=0}^{\infty}\boldsymbol{P}_i(A)\, z_1^{a_1}\cdots\mathscr{J}_i^{k_i}\left[z_{r_i}^{a_{r_i}}\right]\cdots z_R^{a_R} \equiv \mathscr{E}_i\left[\hat{\boldsymbol{a}}_i(Z)\right], \tag{16}$$

where the operator $\mathscr{J}_i$ changes $z_{r_i}^{a_{r_i}}$ to $\hat{\boldsymbol{S}}_{r_i}(Z) z_{r_i}^{a_{r_i}-1}$ if $a_{r_i} \geq 1$ but does not change if $a_{r_i} = 0$, and $\mathscr{E}_i$ is an operator on $\hat{\boldsymbol{a}}_i(Z)$ that transforms $z_{r_i}^{a_{r_i}}$ in $\hat{\boldsymbol{a}}_i(Z)$ to $\mathscr{J}_i^{k_i}[z_{r_i}^{a_{r_i}}]$, i.e., transforms $z_{r_i}^{a_{r_i}}$ to $\mathscr{J}_i^{k_i - a_{r_i}}[\hat{\boldsymbol{S}}_{r_i}^{a_{r_i}}(Z)]$ if $a_{r_i} < k_i$ but to $\hat{\boldsymbol{S}}_{r_i}^{k_i}(Z) z_{r_i}^{a_{r_i}-k_i}$ if $a_{r_i} \geq k_i$. Consequently, $\hat{\boldsymbol{a}}_{i+1}(Z)$ in Eq. (13) can be expressed as

$$\hat{\boldsymbol{a}}_{i+1}(Z) = \mathscr{E}_i\left[\hat{\boldsymbol{a}}_i(Z)\right]\cdot\hat{\boldsymbol{U}}_i(Z). \tag{17}$$

From Eq. (15) or Eq. (17), it is found that $\hat{\boldsymbol{a}}_i(Z)$ can be obtained recursively by

$$\hat{\boldsymbol{a}}_i(Z) = \hat{\boldsymbol{a}}_{i+P}(Z) = S_{i+P-1}\Big[\cdots\Big[S_{i+1}\Big[S_i\big[\hat{\boldsymbol{a}}_i(Z)\big]\cdot\hat{\boldsymbol{U}}_i(Z)\Big]\cdot\hat{\boldsymbol{U}}_{i+1}(Z)\Big]\cdots\Big]\cdot\hat{\boldsymbol{U}}_{i+P-1}(Z), \tag{18}$$

where $\mathscr{S}_i$ is $\mathscr{G}_i$ or $\mathscr{E}_i$, depending on the service discipline of stage i. If $\hat{\boldsymbol{a}}_i(Z)$ is given, $\hat{\boldsymbol{d}}_i(Z)$ can be gotten from Eq. (14) for G-limited or from Eq. (16) for E-limited. And $\hat{a}_i(z)$ and $\hat{d}_i(z)$ for all i can be found from $\hat{\boldsymbol{a}}_i(Z_{r_i})$ and $\hat{\boldsymbol{d}}_i(Z_{r_i})$ for all i, respectively, where Z_{r_i} is obtained by replacing all elements of Z by 1 except z_{r_i} by z. Finally, $\hat{a}_i'(1)$, $\hat{a}_i''(1)$, $\hat{d}_i'(1)$, and $\hat{d}_i''(1)$ can be obtained for all i. On the other hand, β_i can be gotten by taking limit of Eq. (9) as z goes to 1 and by L'Hospital rule; β_i is given by

$$\beta_i = \frac{\hat{a}_i'(1) - \hat{d}_i'(1)}{1 - \rho_{r_i}}. \tag{19}$$

4. Optimal pattern design and numerical examples

4.1. Optimal pattern design

The paper further explores an optimal-pattern design of the general limited scheduling mechanism for the distributed communication system by a genetic algorithm (GA). GA is a random method that combines the survival of the fittest with the innovative flair of a human search [7]. GAs evolve generation by generation as in biological evolution. Assume that there is a population of m candidates denoted by $\{x_1^n, x_2^n, \ldots, x_m^n\}$ in generation n, where each candidate is expressed by a k-length binary string of genes. The candidates of the nth generation $\{x_1^n, x_2^n, \ldots, x_m^n\}$ generate the candidates of the $(n+1)$st generation $\{x_1^n+1, x_2^{n+1}, \ldots, x_m^{n+1}\}$ by crossover and mutation. An objective function, denoted by f, is defined to find the fitness of candidates. The fitness of the candidates in the present generation will influence the production of candidates for the next generation.

In this paper, we create $\{\hat{x}_1^{n+1}, \hat{x}_2^{n+1}, \ldots, \hat{x}_m^{n+1}\}$ by choosing with replacement from $\{x_1^n, x_2^n, \ldots, x_m^n\}$ with probabilities $\{f(z_1^n)/\Delta, f(x_2^n)/\Delta, \ldots, f(x_m^n)/\Delta\}$, where $f(x_i^n)$ is the fitness of x_i^n and $\Delta = \sum_{j=1}^m f(x_j^n)$. Note that the elements of $\{\hat{x}_1^{n+1}, \hat{x}_2^{n+1}, \ldots, \hat{x}_m^{n+1}\}$ are replicas of candidates in $\{x_1^n, x_2^n, \ldots, x_m^n\}$ and some elements in $\{\hat{x}_1^{n+1}, \hat{x}_2^{n+1}, \ldots, \hat{x}_m^{n+1}\}$ can be the same. The greater the fitness of x_i^n is, the greater the number of replicas of x_i^n in $\{\hat{x}_1^{n+1}, \hat{x}_2^{n+1}, \ldots, \hat{x}_m^{n+1}\}$ will be. Next, operate on the elements of $\{\hat{x}_1^{n+1}, \hat{x}_2^{n+1}, \ldots, \hat{x}_m^{n+1}\}$ by *crossover* and *mutation* to generate $\{x_1^{n+1}, x_2^{n+1}, \ldots, x_m^{n+1}\}$. In the crossover operation, the elements of $\{\hat{x}_1^{n+1}, \hat{x}_2^{n+1}, \ldots, \hat{x}_m^{n+1}\}$ are mated randomly and each mated pair of candidates exchanges their genes of the string from 1 to ζ, where ζ is an integer selected uniformly from $1, 2, \ldots, k-1$. For example, if $\hat{x}_i^{n+1} = (01010011)$ and $\hat{x}_j^{n+1} = (10101001)$ are mated together and ζ is selected to be 3, then x_i^{n+1} will be (10110011) and x_j^{n+1} will be (01001001). In mutation, one or more of the genes of a candidate are changed with a small probability; this is in fact a random walk through the string space in order to yield a potentially useful candidate. The evolution in the GA will be terminated when an acceptable approximation is found, the number of searched candidates has reached a predetermined number, or some other reasonable criterion is satisfied [9]. During the evolution, we find the optimal candidate x_{op}, which is defined to have the maximum value of fitness $f(x_{op})$. The framework of the numerical procedures to obtain the mean waiting time of task and the optimal pattern of general limited scheduling mechanism is described below.

Step 1: [Initialization]
- $n = 0$
- Heuristically choose $\{x_1^n, x_2^n, \dots, x_m^n\}$
- x_{op} (Optimal candidate) $= x_1^0$
- Without loss of generality, we first find $\hat{\boldsymbol{a}}_1(Z)$
- Give an initial $\hat{\boldsymbol{a}}_1(Z) \equiv \sum_A \boldsymbol{P}_1(A) Z^A$

Step 2: [Find $\hat{\boldsymbol{a}}_1(Z)$ by an Iterative Algorithm]
- $\hat{a}_1^*(Z)(\equiv \sum_A \boldsymbol{P}_1^*(A) Z^A) = \hat{\boldsymbol{a}}_1(Z)$, i.e., $\boldsymbol{P}_1^*(A) = \boldsymbol{P}_1(A)$ for all A
- Find $\hat{\boldsymbol{a}}_{1+P}(Z) = \mathscr{S}_P[\cdots[\mathscr{S}_2[\mathscr{S}_1[\hat{\boldsymbol{a}}_1(Z)] \cdot \hat{\boldsymbol{U}}_1(Z)] \cdot \hat{\boldsymbol{U}}_2(Z) \cdots] \cdot \hat{\boldsymbol{U}}_P(Z)]$
- Find a weighting factor ω according to the convergence situation by Seelen's Algorithm [10]
- $\hat{\boldsymbol{a}}_1(Z) = \hat{\boldsymbol{a}}_{1+P}(Z) + \omega(\hat{\boldsymbol{a}}_{1+P}(Z) - \hat{\boldsymbol{a}}_1^*(Z))$
- IF $|(\boldsymbol{P}_1(A) - \boldsymbol{P}_1^*(A))/\boldsymbol{P}_1(A)| > 10^{-3}$ for any N

    GO TO Step 2

  END IF

Step 3: [Obtain the pgfs]
- DO $i = 1, P$

    $\hat{\boldsymbol{d}}_i(Z) = \mathscr{G}_i[\hat{\boldsymbol{a}}_i(Z)]$ for G-limited service discipline (from Eq. (14)), or

    $\hat{\boldsymbol{d}}_i(Z) = \mathscr{E}_i[\hat{\boldsymbol{a}}_i(Z)]$ for E-limited service discipline (from Eq. (16))

    $\hat{a}_i(z) = \hat{\boldsymbol{a}}_i(Z_{r_i})$

    $\hat{d}_i(z) = \hat{\boldsymbol{d}}_i(Z_{r_i})$

    Get $\hat{a}_i'(1)$, $\hat{a}_i''(1)$, $\hat{d}_i'(1)$, and $\hat{d}_i''(1)$

    Get $\beta_i = (\hat{a}_i'(1) - \hat{d}_i'(1))/(1 - \rho_{r_i})$

    IF i is not equal to P

$$\hat{\boldsymbol{a}}_{i+1}(Z) = \hat{\boldsymbol{d}}_i(Z) \cdot \hat{\boldsymbol{U}}_i(Z) (\text{ from Eq. (13)})$$

    END IF

  END DO

Step 4: [Mean Waiting Times]
- DO $i = 1, P$

$$w_i = \frac{\lambda_{r_i}}{2\beta_i(1-\rho_{r_i})}\left[\frac{\left(\hat{a}_i'(1) - \hat{d}_i'(1)\right)s_{r_i}^{(2)}}{1-\rho_{r_i}} + \frac{\hat{a}_i''(1) - \hat{d}_i''(1)}{\lambda_{r_i}^2}\right]$$

  END DO

- DO $r = 1, R$

$$v_r = \frac{1}{2(1-\rho_{r_i})c}\sum_{\{i|r_i = r\}}\left[\frac{\left(\hat{a}_i'(1) - \hat{d}_i'(1)\right)s_{r_i}^{(2)}}{1-\rho_{r_i}} + \frac{\hat{a}_i''(1) - \hat{d}_i''(1)}{\lambda_{r_i}^2}\right]$$

  END DO

Step 5: [Evaluate fitness]

- DO $i=1,m$
  Find $f(x_i^n)$
  IF $f(x_i^n) > f(x_{op})$
  $x_{op} = x_i^n$
  END IF
  END DO

Step 6: [Check the termination criterion]

- IF the termination criterion is satisfied
  GO TO **Step 8**
  END IF

Step 7: [Produce the next generation]

- Create $\{\hat{x}_1^{n+1}, \hat{x}_2^{n+1}, \ldots, \hat{x}_m^{n+1}\}$ from $\{x_1^n, x_2^n, \ldots, x_m^n\}$ according to the weights of fitness $f(x_i^n)$, $i=1, 2, \ldots, m$
- Generate $\{x_1^{n+1}, x_2^{n+1}, \ldots, x_m^{n+1}\}$ by crossover and mutation
- $n = n+1$
- GO TO **Step 5**

Step 8: [End]

- Print the optimal candidate x_{op} (optimal pattern of the general limited service discipline) and its fitness $f(x_{op})$

The Seelen's algorithm in Step 2 is used to speed up the convergence of the iteration [12]. The reader can refer to [12] for details.

The paper heuristically defines a fitness function for a given candidate x_i^n of the nth generation in the system, denoted by $f(x_i^n)$, as

$$f(x_i^n) = \left[\sum_{r=1}^{R} \frac{\lambda_r}{\lambda} \cdot |v_r - v_r^*| \right]^{-1},$$

where v_r is the mean waiting time of task in queue r with respect to the candidate x_i^n and v_r^* is the required mean waiting time of task in queue r. As the equation implies, the fitness function f is used to find an optimal pattern such that the mean waiting time of queue r in the distributed communication system satisfies the system requirement and specification as much as possible. The explicit parameters of the fitness function $f(x_i^n)$ are the mean waiting times for a candidate x_i^n (the implicit parameters) that represents a pattern of the general limited service discipline. Note that the service order and the service discipline of each stage are coded into a binary string of genes. The paper utilizes the GA ucsd 1.4 [11] developed at the University of California, San Diego, and adopts a predetermined number of searched candidates as the termination criterion.

4.2. Numerical example and discussion

This subsection first investigates the influence of the limitation number of limited service discipline on the mean waiting time. An M/G/1 queue with E-limited service discipline of limitation numbers 1, 2, 3, 4, 5, 10, 50, and is observed. It is found that the mean waiting times of tasks as the limitation numbers equal to 1, 2, 3, 4, and 5 are more distinct than those as the limitation numbers are greater than 5 for a wide range of traffic intensity. G-limited service discipline also has the same phenomenon. Therefore, the following example only considers limitation numbers of 1, 2, 3, 4, 5, and ∞.

The example of the optimal pattern design for a distributed communication system assumes that there are four process queues ($R=4$) in the system. The service time distribution S_r is exponentially distributed and the mean service time s_r is equal to 1, $1 \le r \le R$; the walking time U_i is deterministic and u_i is equal to 0.1, $1 \le i \le P$. The arrival rates are 4:1:1:1 for queues 1–4, respectively. Furthermore, queue 1 is regarded as more

Table 1
The optimal patterns of service order sequence and service discipline

Traffic intensity	Optimal pattern	Cost (f^{-1})
0.2	Sequence C, $E_1E_1E_1E_1E_1E_1$	2.207
0.3	Sequence C, $E_1E_1E_1E_1E_1E_1$	1.966
0.4	Sequence C, $E_1E_1E_1E_1E_1E_1$	1.638
0.5	Sequence C, $E_\infty E_1E_1E_1E_1E_1$	1.235
0.6	Sequence C, $E_\infty E_1E_\infty E_1E_\infty E_1$	0.642
0.7	Sequence C, $E_\infty E_2E_\infty E_2E_\infty E_2$	0.267
0.8	Sequence C, $E_\infty E_\infty E_\infty E_\infty E_\infty E_\infty$	1.968

delay-sensitive than queues 2–4 that the required mean waiting times are assumed to be $v_1^* = 1$ and $v_r^* = 5$ (or $v_1^*/v_r^* = \frac{1}{5}$), $2 \le r \le R$. Three possible types of service order sequences that poll the queue 1 from one to three times are considered in the design of optimal pattern. The three sequences are: Sequence A with $P = 4$: {1, 2, 3, 4}; Sequence B with $P = 5$: {1, 2, 1, 3, 4}; and Sequence C with $P = 6$: {1, 2, 1, 3, 1, 4}. The paper uses G_k (E_k) to denote the G-limited (E-limited) service discipline with limitation number k and refer the aggregation of G_k or E_k for every stage as the design pattern of the distributed communication software system. For example, $E_3G_2E_2E_2G_1$ means E-limited, G-limited, E-limited, E-limited and G-limited for stages 1, 2, 3, 4, and 5 with limitation numbers 3, 2, 2, 2, and 1, respectively.

There are total 1947253 cases in an enumerative search for the optimal pattern of this example. GAUCSD suggests only 490 cases to be searched; the efficiency is about 99.97%. The optimal patterns and costs are shown in Table 1, where the cost is defined as $f^{-1}(x_i^n)$. It is found that all the service order sequences are sequence C and the limitation number for all queues, especially queue 1, tends to E_∞ as traffic intensity becomes larger. Choosing C that the queue 1 needs multiple polls is because queue 1 is highly loaded and its requirement

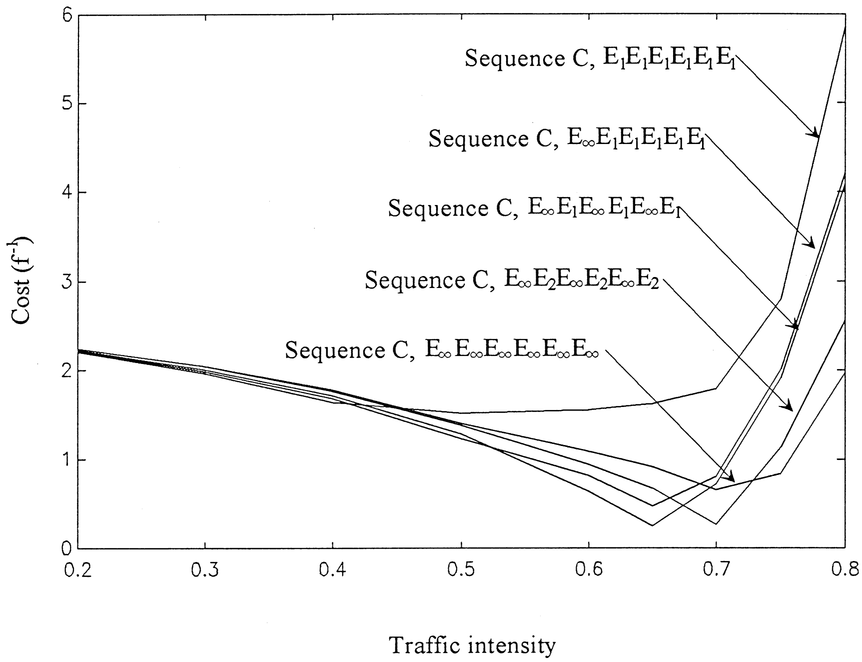


Fig. 1. The cost for various optimal patterns in the four-queue example system with arrival rates = 4:1:1:1, $v_1^* = 1$, $v_r^* = 5$, $r = 2, 3, 4$, mean service time = 1, and mean walking time = 0.1.

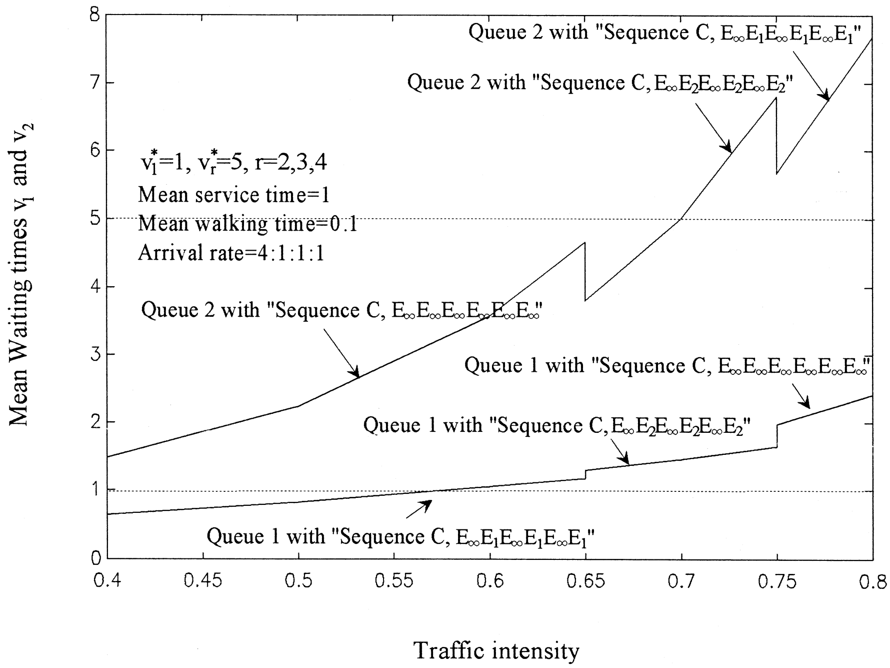


Fig. 2. The mean waiting times of the three suggested optimal patterns versus traffic intensity.

is stricter. Adopting exhaustive (E_∞) service discipline when the traffic intensity is high is because the exhaustive service discipline processes a hogging property which will result in a smaller waiting time [15].

The optimal pattern is altered if the traffic intensities are changed. Fig. 1 plots the costs of the optimal patterns shown in Table 1 for all traffic intensities. It is found that there is no much difference at low traffic intensities for all these optimal patterns. Therefore, it is suggested that the pattern of ''Sequence C and $E_\infty E_1 E_\infty E_1 E_\infty E_1$'' be adopted around the traffic intensities below 0.65, the pattern of ''Sequence C and $E_\infty E_2 E_\infty E_2 E_\infty E_2$'' be adopted around the traffic intensities between 0.65 and 0.75, and the pattern of ''Sequence C and $E_\infty E_\infty E_\infty E_\infty E_\infty E_\infty$'' be adopted around the traffic intensities above 0.75 for the example software system.

Furthermore, Fig. 2 plots the mean waiting times of queue 1 and queue 2 with pattern of ''Sequence C and $E_\infty E_1 E_\infty E_1 E_\infty E_1$'' at traffic intensities below 0.65, pattern of ''Sequence C and $E_\infty E_2 E_\infty E_2 E_\infty E_2$'' at traffic intensities between 0.65 and 0.75, and pattern of ''Sequence C and $E_\infty E_\infty E_\infty E_\infty E_\infty E_\infty$'' at traffic intensities above 0.75, where the dotted lines denotes the delay requirements $v_1^* = 1$ and $v_2^* = 5$ for queue 1 and queue r, $2 \le r \le R$ respectively. Note that the queues 2–4 should have similar mean waiting time. It is found that the ratio of the mean waiting time of queue 1 and queue 2, (v_1/v_2), is close to the required ratio $(v_1^*/v_2^* = \frac{1}{5})$ and the mean waiting times of queue 1 and queue 2 are also kept as near to the requirement as possible. A distributed communication software system with scheduling mechanism of general service order and limited service discipline seems flexible to meet the system requirements.

5. Concluding remarks

This paper studies a distributed communication system with an alternative scheduling mechanism of general service order sequence and limited service discipline. The analytical approach is by way of imbedded Markov chains and utilizes some operators to facilitate the analysis. Design of the optimal pattern of the general limited scheduling mechansim for a distributed communication system via a genetic algorithm is also presented. The results show that the distributed communication system with a scheduling mechanism of general limited service discipline can be flexibly programmed to meet the system requirement and specification and can be improved if an optimal pattern for the general limited scheduling mechanism is adopted.

References

[1] J.E. Baker, I. Rubin, Polling with a general-service order table, IEEE Trans. Comm. 35 (3) (1987) 283–288.

[2] C.J. Chang, L.C. Hwang, An asymmetrical polling system with general service order sequence and gated-limited service discipline, IEEE Int. Symp. Inf. Theory, Hundary, June 1991, p. 158.

[3] E. Drakopoulos, M.J. Merges, Performance analysis of client-server storage system, IEEE Trans. Comp. 41 (11) (1992) 1442–1452.

[4] M. Eisenberg, Queues with period service and changeover time, Oper. Res. 20 (1972) 440–451.

[5] D. Everitt, A note on the pseudoconservation laws for cyclic service systems with limited service discipline, IEEE Trans. Comm. 37 (7) (1989) 781–783.

[6] S.W. Fuhrmann, Y.T. Wang, Analysis of cyclic service systems with limited service: bound and approximation, Perf. Eval. 9 (1988) 35–54.

[7] D.E. Goldbeg, Genetic Algorithm in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1986.

[8] M.G. Härbour, M.H. Klein, J.P. Lehoczky, Timing analysis for fixed-priority scheduling of hard real-time systems, IEEE Trans. Soft. Eng. 20 (1) (1994) 13–28.

[9] L.C. Hwang, C.J. Chang, Optimal design of a finite-buffer polling network with mixed service discipline and general service order sequence, IEE Proceeding – Communications 142 (1) (1995) 1–6.

[10] B.E. Keiser, E. Strange, Digital Telephony and Network Integration, Van Nostrand Reinhold, New York, 1995.

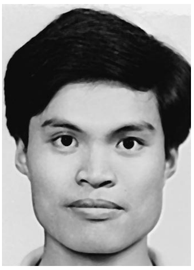
[11] N.N. Schraudolph, J.J. Grefenstette, A user's guide to GAucsa 1.4, Tech. Rep. CS92-249, CSE DEP., UC San Diego, La Jolla, CA 92093-0114.

[12] L.P. Seelen, An algorithm for Ph/Ph/*c* queue, European J. Oper. Res. 23 (1986) 118–127.

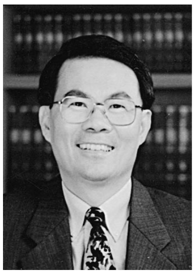
[13] M. Schwartz, Telecommunication Networks: Protocols, Modeling and Analysis, Addison-Wesley, Reading, MA, 1987.

[14] C.B. Silio Jr., H.M. Ghafir, M.R. Parikh, An approximate method for the performance analysis of PLAYTHROUGH rings, IEEE Trans. Comp. 41 (9) (1992) 1137–1155.

[15] H. Takagi, Queueing analysis of polling models: An update, in: H. Takagi (Ed.), Stochastic Analysis of Computer and Communication Systems, North-Holland, Amsterdam, 1990, pp. 267–318.



Lain-Chyr Hwang was born in Taiwan on 26 December 1965. He received his B.S. degree in Electrical Engineering from the National Sun Yat-Sen University, Kaohsiung, Taiwan in 1988, his M.S. degree in Communication Engineering (1990) and his Ph.D. in Electronic Engineering (1994) from National Chiao-Tung University, Hsinchu, Taiwan. From 1995 to 1996 the was with Telecommunications Laboratories, Chunghwa Telecom Co., Ltd, Taiwan as an associate researcher. There, he was involved in network planning of the intelligence network and personal communication services. In August 1996 he became associate professor in the Department of Management Information Systems of the I-Shou University, Taiwan. His research interests include performance evaluation and computer communication networks.



Chung-Ju Chang received the B.E. and the M.E. degrees in Electronics Engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 1972, and 1976, respectively, and the Ph.D. degree in Electrical Engineering from National Taiwan University, Taiwan, in 1985. From 1976–1988, he was with Telecommunication Laboratories, Directorate General of Telecommunications, Ministry of Communications, Taiwan, as a Design Engineer, Supervisor, Project Manager, and then Division Director. There, he was involved in designing digital switching system, ISDN user-network interface, and ISDN service and technology trials. In the meantime, he also acted as a Science and Technical Advisor for Minister of the Ministry of Communications from 1987–1989. In August 1988, he joined the faculty of the Department of Communication Engineering, College of Electrical Engineering and Computer Science, National Chiao Tung University, where he is currently a Professor. He is also acting as an Advisor for the Ministry of Education to promote the education of communication science and technologies for the universities of the whole country, and is serving as the Chairman of IEEE Vehicular Technology Society, Taiwan Chapter. He was Director of the Institute of Communication Engineering from August 1993 to July 1995. His research interests include performance evaluation, PCS (personal communications services), and broadband networks.