Feedback Queues with Preemption-Distance Priorities

M. Paterok O. Fischer *

University of Erlangen-Nuremberg
Institute for Mathematical Machines and Data Processing VII
Chair for Computer Architecture and Performance Evaluation

* Loewe Opta GmbH
Kronach, Federal Republic of Germany

This paper is dedicated to Prof. Ulrich Herzog on the occasion of his 50th birthday.

Abstract

The method of moments is used to derive exact analytical solutions for an open priority queueing system with preemption-distance priorities and feedback. Customers enter from outside in a Poisson stream. They can feed back for several times, changing priorities and service demands in an arbitrary manner. During feedback they can fork and branch according to user-defined probabilities. The service demands of the different classes are pairwise independent and can be arbitrarily distributed. A customer who has been interrupted resumes his service from the point where he was interrupted (preemptive resume). A system of linear equations is to be solved to obtain the mean sojourn times of each customer class in the system.

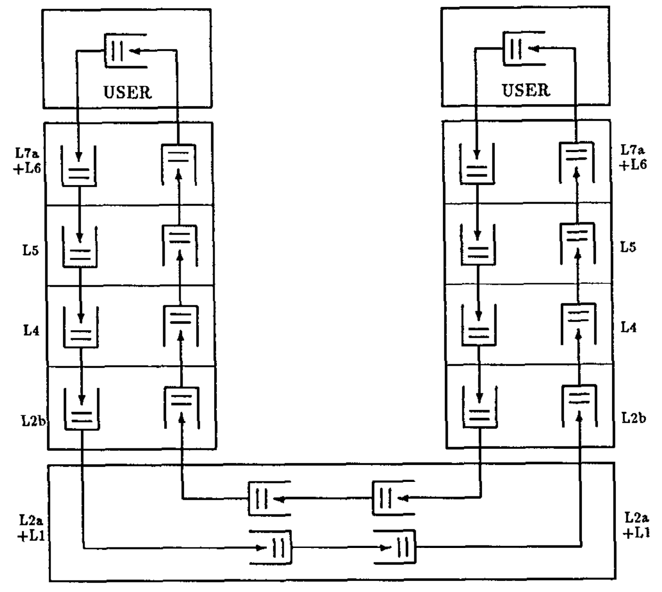
1 Introduction

Real computer systems generally use priority strategies for the scheduling of their central processor. As a result, in the framework of queueing sytems there has been interest in mathematical solutions for systems with priorities. In 1954, Cobham offered formulae for the mean waiting times for queues of the type M/G/1-PRIO[NP] [Cobham 54]. In order to obtain his results, he developed a technique that has become typical for the analysis of such systems: the method of moments. A typical job is traced as it proceeds through the system and the components of its path are individually analyzed. Using the expected values of these components, a system of linear equations can be derived and solved to obtain the first moments of the characteristical performance measures. The only assumptions about the model are that it is in steady state, and that interarrival times are exponentially distributed. This is caused by the well-known result, that customers arriving in a Poisson stream see time averages upon arrival [Wolff 82]. Herzog's equations for arbitrarily chosen preemption-distance priorities ([Herzog 75], [Herzog 76]) give results for open priority systems without feedback. In such systems, only an arriving customer whose priority level is at least by the preemption-distance higher than the priority of the customer that is actually served will cause a preemption. Almost any combination of static priority strategies can be modelled by a proper choice for the priority levels and the preemption-distance (see [Herzog 76]). The method of moments is briefly discussed in [Kleinr 76].

However, in real computer systems processes often run through several phases during their runtime on the processor. Those phases often show significant differences in service demands and in the priorities assigned to the different levels of execution. Such feedback systems with priorities have not been adequately studied.

In the context of our recent research feedback queues are used to model the behavior of layered communication systems. Figure 1a shows Herzog's logical flow model [Herzog 86] for the layers of the ISO Reference Model for open systems ([Halsa 88]).

Packets pass the different layers of the communication software. Each layer marks a different phase of service that a packet receives and is denoted according to the OSI Model. Each queue has its own server (In Figure 1a, the servers are not depicted). However, in real systems all layers within one of the large boxes are executed on one single processor. Thus, all servers within one box are collected to a single server serving all queues of the boxes. For each of the two large boxes of Figure 1a we obtain the model shown in Figure 1b [Herzog 86]. The corresponding layer of the logical



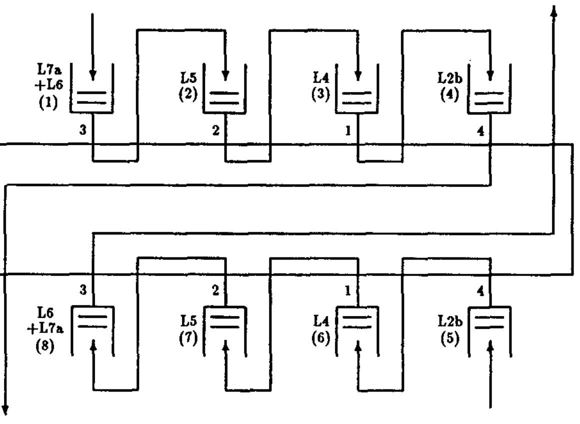


Figure 1: a. Logical Flow Model b. Model Including Hardware Restrictions

flow model is denoted for each queue. Priorities are assigned to the different queues (numbers below and above the queues) in order to schedule them on the processor. The numbers in paranthesis are the class numbers.

Initial analyses of these systems were done by Daigle and Houstes [Daigle 81], Gay and Seaman [Gay 75], Schrage [Schrag 67], Altaminaro and Fontana [Alta 88] and Simon [Simon 84]. Altaminaro and Fontana [Alta 88] treat feedback queues with priorities and single and multiple servers. In their approach the results for classes with high priorities are obtained without explicit consideration of classes with lower priorities. Thus, we do not expect this algorithm to yield good results. Simon examines priority queues with feedback, mixed strategies and Poisson arrivals [Simon 84]. Nevertheless, Simon's method is important and the basic method fits very well to the analysis of open systems with feedback. However, some features of his modell are unfavourable for practical performance modelling: a repeat strategy on preemption is used, the classes are statically divided into preemptive and non-preemptive classes, and branching can only be introduced by adding artificial complexity to the model. In this paper Simon's technique is used as mathematical basis of the analysis. Also Simon's notation is adopted. In the paper presented, the disadvantages described above have been overcome.

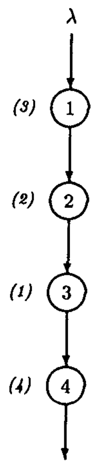
In the next section, the model is briefly described. In Section 3, measures are introduced which are needed for the description of the characteristic performance measures. Section 4 gives a short overview of the presuppositions and basic ideas of the algorithm. In Section 5 the expected system times are derived as a function of the expected system states at arrival times. These system states are expressed in Section 6. In Section 7, a system of linear equations is derived to compute the expected system time of each class. In Section 8, a numerical example is presented.

2 The Model

In this section the feedback model is described more precisely. The following input measures and assumptions are introduced.

1. The model consists of a single server and an infinite waiting room (i.e. queue).
2. There are c open chains of customers. Each chain i has $N(i)$ classes ($i = 1, \dots, c$), which are tree-structured. The root of each tree is given by an external class, whose customer arrive from outside. On their way through the system customers of an external class can only enter classes within the corresponding tree. Therefore, only customers whose class upon arrival is the same may reach the same classes by feedback. In order to simplify the notation, the classes are single-indexed by $1, \dots, N$ with $N = \sum_{j=1}^{c} N(j)$. The function $g(j)$ $(j = 1, \dots, N)$ denotes the class of the external arrivals of the chain containing class j, i.e., the root of its tree. The classes are partially ordered by the precedence relation '$\prec$'. $i \prec j$ means that a class i customer can become a customer of class j via feedbacks. The relation '$\preceq$' is the reflexive extension of '$\prec$'. Only classes of the same chain can stand in these relations. EXT denotes the set of external customer classes. The other classes are called internal classes.

Figure 2a shows another representation of the simple example given by the upper chain in Figure 1b. For simplicity of notation the classes are represented by circles and the feedbacks by arrows. The numbers inside the circles are the class numbers, the numbers in paranthesis are the priorities assigned to the customers of the various classes.



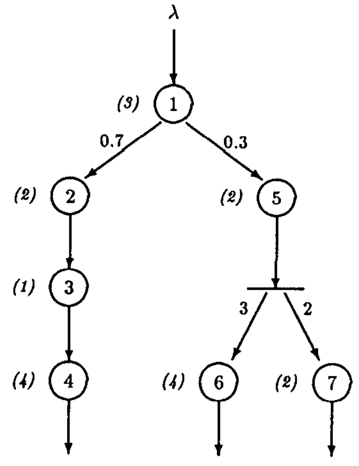


Figure 2: a. Simple Example b. Example Including Branch and Fork

3. A customer of class j ($j = 1, \dots, N$) has priority $f(j)$. A higher value of $f(j)$ implies a higher priority level.
4. Customers of an external customer class j arrive as a Poisson stream with rate λ_j.
5. The service demand of a class j customer is described by the arbitrarily chosen probability distribution function G_j and its density function g_j.
6. Upon feedback a customer may split into several customers of different classes *(fork)*. Furthermore, a customer can decide to continue as a customer of either one or another class according to given probabilities *(branch)*. Upon branching a customer cannot fork (This case can be modeled by introducing artificial classes). The value $p(j)$ stands for the probability that a customer of the predecessor-class of class j ($pre(j)$) branches to class j. $num(j)$ denotes the number of class j customers that grow out of the predecessor customer. M_j is the mean number of class j customers that one predecessor becomes and is given by $M(j) = num(j) * p(j)$, where either $num(j)$ or $p(j)$ has the value 1.
   Taking Figure 2b as an example, we obtain: $M(2) = 0.7,\ M(3) = 1.0,\ M(6) = 3.0,\ M(7) = 2.0$.
   The throughput of an internal class j is given by
   $$\lambda_j = M(j) * \lambda_{pre(j)} = \prod_{k \preceq j} M(k) * \lambda_{g(j)} \tag{1}$$
   where the function $pre(j)$ denotes the predecessor class of class j.
   In Figure 2a: $\lambda_j = \lambda$ for all classes j; in Figure 2b: $\lambda_2 = 0.7\,\lambda_1 = 0.7\,\lambda,\ \lambda_3 = \lambda_2 = 0.7\,\lambda,\ \lambda_6 = 3\,\lambda_5 = 3 * 0.3\,\lambda$.
   $A(j)$ is needed to describe branching. It is set to zero if a class j customer enters the queue as a result of branching. Otherwise it stands for $num(j)$.
   $$A(j) = \begin{cases} num(j) & \text{if } p(j) = 1 \quad \text{(no branching)} \\ 0 & \text{otherwise} \quad \text{(branching)} \end{cases} \tag{2}$$
   In Figure 2a: $A(j) = num(j) = 1$ for all classes j. In Figure 2b: $A(2) = 0,\ A(3) = 1,\ A(5) = 0,\ A(7) = 2$.
   Figure 3 illustrates the dependencies between the measures presented by an example for the heterogenous fork (left) and for the branch (right) arrising in Figure 2b.
7. The customers are scheduled according to preemption-distance priorities. Only an arriving customer whose priority level is at least by the global preemption-distance PD higher than that of the customer in service will cause an interrupt. After completion of a service, the customer with the highest priority is scheduled. The scheduling process takes place after the feedback. Customers of the same priority level are served in FCFS order. A preempted customer resumes his service from the point of interruption. The scheduling process, the feedbacks, and the interrupts are assumed to take negligible time.
   In Figure 2a, $PD = 2$ means, that a class 2 customer cannot be preempted, but a class 3 customer can.
8. The arrival and service processes are pairwise independent of each other.

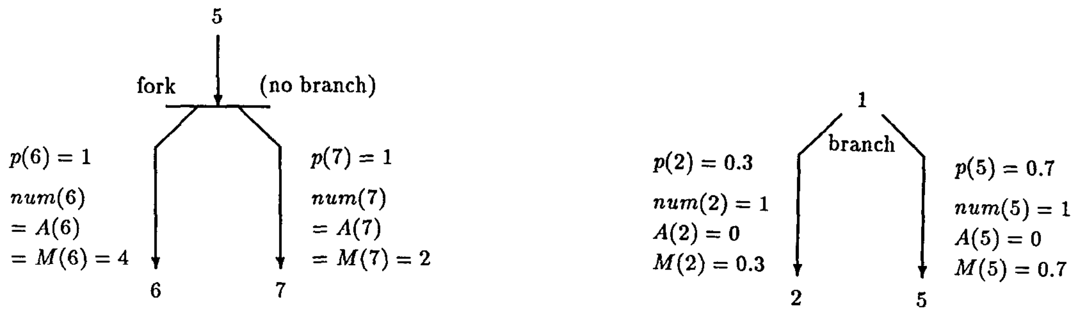


Figure 3: Measures for Fork and Branch

9. The system is stable.

3 Some Important Measures

In this section, some measures that are needed for the analysis are presented. They have been derived in [Pate 89], and partly in [Herzog 75] and [Simon 84]. For application purposes they can also be expressed in terms of the Branching Erlang Distribution [Pate 89].

Since a preempted customer resumes his service, the mean service time t_j of a class j customer is the first moment of G_j. The utilization ρ_j of the server by class j customers is given by $\rho_j = \lambda_j * t_j$.

The total remaining service time $\hat{t}_j$ of a class j customer at arrival of another customer is well-known from Pollaczek's and Khintchine's result [Kleinr 75] for M/G/1-FCFS.

$\tilde{t}_j$ is the expected time the server keeps on serving a customer of class j after the arrival of a non-interrupting customer. This represents the mean time until the service is either completed or preempted by an interrupting customer. $\tilde{t}_j$ is needed to describe the following situation. Consider an arriving customer having a priority that is higher than that of the customer being served. But due to the preemption-distance the arriving customer is not allowed to interrupt. The delay that the served customer causes the arriving customer is limited by either the completion of service or by a preemption due to another arriving customer. A preemption implies that the previously arrived customer is served before the preempted customer can resume service. $\tilde{t}_j$ is given by

$$\tilde{t}_j = \int_0^\infty t \frac{d\tilde{G}_j(t)}{dt}\, dt \qquad \text{where} \qquad \tilde{G}_j = \frac{\int_0^r (1 - G_j(t))e^{-\alpha_j t} dt}{\int_0^\infty (1 - G_j(t))e^{-\alpha_j t} dt} \qquad \text{and} \qquad \alpha_j = \sum_{\substack{f(i) \geq f(j)+PD \\ i \in EXT}} \lambda_i \tag{3}$$

α_j is the rate at which a class j customer is preempted.

The expected non-preempted final part of the service of class j is denoted by t_j^{fin}. It stands for the mean length of the last (uninterrupted) phase of service received by a class j customer. It describes the mean length of the period during which a non-preempting customer cannot overtake an actually served customer with lower priority.

$$t_j^{fin} = \int_0^\infty \frac{1}{\alpha_j}(1 - e^{-\alpha_j t}) g_j(t) dt \tag{4}$$

$\tilde{z}_j$ is the probability that the remaining service time (with mean $\hat{t}_j$) of a class j customer is not preempted.

$$\tilde{z}_j = \frac{\int_0^\infty t\, e^{-\alpha_j t} g_j(t) dt}{\int_0^\infty t\, e^{-\alpha_j t} g_j(t) dt + \int_0^\infty t\, \alpha_j e^{-\alpha_j t}(1 - G_j(t)) dt} \tag{5}$$

4 Basic Concepts of the Algorithm

The algorithm presented is based on four suppositions [Simon 84]:

1. Little's theorem establishes a simple relationship between queue sizes and waiting times [Jewell 67].
2. The waiting time of a customer depends linearly on the number of customers that the customer sees upon arrival in the system.
3. Customers entering the system from external in a Poisson stream see time averages upon arrival [Wolff 82]. Thus, by Little's theorem they see steady state.
4. For customers entering the system via feedback the system state at arrival time is a linear function of the steady state and of the system state upon the arrival of the customer's predecessor (which became the customer via feedback).

Therefore the expected system time W_j for class j customers can be described in a simplified (and informal) way by

$$W_j \;=\; \tilde{w}_j\,(a_{j-1}(\,\ldots\, a_{g(j)+1}(a_{g(j)}(\Lambda\underline{W}))\;\ldots)),\underline{W})$$

where $\tilde{w}_j$ denotes the linear function mentioned in Supposition 2 and a_i is the linear function that determines the system state seen by an arriving class i customer. $\Lambda\underline{W}$ stands for the steady state, which is defined to be the mean number of customers of each class in the system. The expression inside the brackets represents the system state upon arrival of a class j customer. It is determined by continously determining the system states as a function of the predecessor state, which is steady state for the external ancestor of the class j customer. Since all relationships are linear, a system of linear equations can be set up to solve for the expected system times. Many more interesting performance measures can be easily derived from the expected system times.

In the following Section 5 the expected system times are derived as a function ('$\tilde{w}_j$') of the system states at arrival times. Section 6 describes the relationship ('a_j') between the system states at arrival times. In Section 7 the system of linear equations is introduced.

For the sake of clarity and brevity of the description only the basic equations are presented. Other equations which can be derived straightforward and the complete system of linear equations are only outlined. The whole set of equations is given in [Pate 89] where an extended version containing heterogenous bulk arrivals is presented.

5 The Expected System Times

The expected system time of a class is the mean time a customer stays in the class when entering it on its way through the system. Applying the method of moments, equations for the expected system times dependent on the expected system states upon (external and internal) arrivals can be derived.

First of all, the expected total delay that a class i customer causes a class j customer has to be introduced. It must be recognized that a class i customer may feed back and enter successor classes which have a higher priority than class j. If the classes i and j have the same priority, $D_j(i)$ applies to situations where the class j customer entered the system before the class i customer. $D_j(i)$ is given by

$$D_j(i) = \begin{cases} 0 & \text{if } f(i) \le f(j) \\ t_i + \sum\limits_{l \in DEC(i)} M(l) D_j(l) & \text{if } f(i) > f(j) \end{cases} \tag{6}$$

where DEC is the set of all successor classes of class i.

With respect to our example in Figure 2a we obtain: $D_2(4) = t_4$, $D_3(1) = t_1 + t_2$, $D_1(2) = 0$, $D_1(1) = 0$. In Figure 2b: $D_3(5) = t_5 + 3t_6 + 2t_7$, $D_2(5) = 0$, $D_7(1) = t_1$, $D_3(1) = t_1 + 0.7t_2 + 0.3(t_5 + 3t_6 + 2t_7)$.

The expected total delay $SD_j(i)$ of a class j customer caused by all successors of a class i customer is given by

$$SD_j(i) = \sum_{l \in DEC(i)} M(l)\, D_j(l) \tag{7}$$

In Figure 2b we obtain: $SD_3(1) = 0.3\,D_3(5) + 0.7t_2$, $SD_2(5) = 3t_6$, $SD_7(1) = 0$.

t_j^r stands for the expected residual service time of a class j customer. This is the expected time from the service begin of a class j customer until its end including all preemptions and overtakings of customers with higher priority level.

$$t_j^r \;=\; \left(t_j \;-\; t_j^{fin} \sum_{\substack{f(j)<f(k)<f(j)+PD \\ k \in EXT}} \lambda_k\, D_j(k) \right) * \left(1 - \sum_{\substack{f(k)>f(j) \\ k \in EXT}} \lambda_k\, D_j(k) \right)^{-1} \tag{8}$$

Now the expected system times can be derived for internal and for external customer classes.

external customers:

It is important to notice that Poisson arrivals see steady state [Wolff 82]. So the probability that a class i customer holds the server at Poisson arrival times is given by ρ_i. According to Little's Law [Jewell 67] the mean queue population of class j is $\lambda_j W_j - \rho_j \quad (j = 1, \ldots, N)$. The expected system time W_j for an external class j customer is given by

$$\begin{aligned} W_j \;=\; & \sum_{f(i)<f(j)<f(i)+PD} \rho_i\left(\tilde{t}_i + \tilde{z}_i SD_j(i)\right) \\ & + \sum_{f(i)\geq f(j)} \left\{(\lambda_i W_i - \rho_i)(t_i + SD_j(i)) \;-\; \lambda_i (t_i^r - t_i)(t_i + SD_j(i)) \;+\; \lambda_i t_i^r (\hat{t}_i + SD_j(i))\right\} \\ & + \sum_{\substack{f(i)\geq f(j)+PD \\ i\in EXT}} \lambda_i W_j D_j(i) \;+\; \sum_{\substack{f(j)<f(i)<f(j)+PD \\ i\in EXT}} \lambda_i (W_j - t_j^{fin}) D_j(i) \;+\; t_j \end{aligned} \tag{9}$$

where $SD_j(i)$ is given by equation (7). The five terms of W_j are characterized as follows:

1. The total expected delay due to an actually served customer with lower priority.
2. Waiting for customers that have at least the same priority as class j and that are already in the system when the class j customer arrives. Upon arrival a class j customer sees $\lambda_i W_i$ customers of class i in the system, of which $\lambda_i t_i^r$ have already received service. Therefore, the class j customer has to wait for
$$\sum_{f(i)\geq f(j)} \left\{\lambda_i (W_i - t_i^r)(t_i + SD_j(i)) \;+\; \lambda_i t_i^r (\hat{t}_i + SD_j(i))\right\} \tag{10}$$
In order to keep the expression $(\lambda_i W_i - \rho_i)$ in the formula, equation (10) is transformed.
3. The total service time of customers that can overtake the class j customer during its hole system time.
4. The total service time of customers which overtake the class j customer without preempting it.
5. The mean service time of the class j customer.

An example taken from Figure 2a will illustrate equation (9). For the external class 1 we obtain:

$$\begin{aligned} W_1 = \; & \rho_2(\tilde{t}_2 + \tilde{z}_2 SD_1(2)) + (\lambda_1 W_1 - \rho_1)(t_1 + SD_1(1)) - \lambda_1(t_1^r - t_1)(t_1 + SD_1(1)) + \lambda_1 t_1^r(\hat{t}_1 + SD_1(1)) \\ & + (\lambda_4 W_4 - \rho_4)(t_4 + SD_1(4)) + \lambda_4(t_4^r - t_4)(t_4 + SD_1(4)) + \lambda_4 t_4^r(\hat{t}_4 + SD_1(4)) + t_1 \end{aligned}$$

A system of linear equations is formed by this and the following equations. It is solved to obtain the expected system times $\underline{W}$ in steady state. Therefore, the equations are rewritten as sums of two terms. The first one, $T_j(i)$, is the part depending on the expected system times in steady state. It forms the homogeneous part of the system of linear equations. r_j represents the inhomogeneous part. $T_j(i)$ and r_j can be derived straightforward from equation (9) and are therefore omitted in this paper (see [Pate 89]). W_j can be rewritten as

$$W_j \;=\; T_j\,(\Lambda \underline{W} - \rho) \;+\; r_j \tag{11}$$

where $(\Lambda \underline{W} - \rho)$ stands for the vector representing the steady state of the queues.

internal customers:

Since customers of an internal class do not see steady state upon arrival, the expected system time of a customer of an internal customer class j is a function of the system state $V = (v_1, \ldots, v_N)$ at the arrival time of a class j customer. An internal customer enters the system as a result of a feedback of his predecessor class. Therefore the server can be treated as empty. We obtain W_j as a sum of six parts.

$$\begin{aligned} & W_j(v_1, \ldots, v_N) = \\ & \sum_{f(i)\geq f(j)} \left\{ v_i\,(t_i + SD_j(i)) \right. \\ & \qquad \left. - \left((\lambda_i t_i^r - \rho_i \tilde{z}_i)\; 1_{f(g(j))<f(i)+PD} \;+\; \lambda_i t_i^r \; 1_{f(g(j))\geq f(i)+PD}\right) \prod_{k=g(j)}^{j-1} 1_{f(i)<f(k)}\left(t_i - \hat{t}_i\right)\right\} \end{aligned}$$

$$+ \sum_{\substack{f(i)\geq f(j)+PD \\ i\in EXT}} \lambda_i W_j(v_1,\ldots,v_N) D_j(i) + \sum_{\substack{f(j)<f(i)<f(j)+PD \\ i\in EXT}} \lambda_i \left(W_j(v_1,\ldots,v_N) - t_j^{fin}\right) D_j(i)$$

$$+ \sum_{\substack{pre(i)=pre(j) \\ f(i)>f(j)}} A(i) D_j(i) + \frac{1}{2} \sum_{\substack{pre(i)=pre(j) \\ f(i)=f(j) \\ A(i)>0}} \left(A(i) - 1_{i=j}\right)\left(t_i + SD_j(i)\right) + t_j \qquad (12)$$

where $pre(j)$ denotes the predecessor class of class j. The function denoted by '1' is the indication function.

Similar to the external case, an internal class j customer has to wait for

1. Customers that are already in the system upon arrival. Again there may be customers having already received service when the class j customer enters the system. The subtracted term represents the already received service that benefits the class j customer. The sum describes the number of preempted customers seen by the external ancestor of the class j customer. The cases where the ancestor preempts upon arrival or not have to be distincted. The product determines whether a preempted customer seen by the ancestor is still in the system when the class j customer enters.
2. Entering customers which can preempt the class j customer.
3. Entering customers which can overtake, but which cannot preempt the class j customer.
4. Customers with higher priority that forked together with him.
5. Customers of another class with the same priority that forked together with him. On the average the class j customer has to wait for half of them. If some of these customers are of class j, the customer itself has to be subtracted, since it is not waiting for itself.
6. The end of his own service.

In Figure 2a: $W_2 = v_1(t_1 + SD_2(1) + v_2(t_2 + SD_2(2)) + v_4(t_4 + SD_2(4)) + \lambda(W_2(v_1,\ldots,v_4) - t_2^{fin})D_2(1) + t_2$.

Similar to the external case (equation(11)) and using $V = (v_1,\ldots,v_N)$, this may be rewritten as $W_j(V) = T_j V + r_j$. Again the term T_j, which depends on the system state and the term r_j representing the inhomogenous part are omitted. To complete the system of linear equations the expected system states V at arrival times have to be expressed.

6 The Expected System States at Customer Feedback Times

In the previous section the system time for each class was derived as a function of the system state at arrival times. The expected system states at external Poisson arrival times are steady state. The issue addressed here is the expected system state at customer feedback times and its derivation. The state an internal customer sees when he enters the queue is the state at the feedback time of his predecessor, where forks have to be considered additionally. The system state at feedback time depends linearly on this state and on steady state.

First a function is needed which expresses the number of customers of a certain class that have grown out of a class j customer at the moment when a customer with priority n completes service. This function is given by

$$U_i^n(j) = \begin{cases} \prod_{j\prec k\preceq i} M(k) & \text{if } \begin{cases} g(i) = g(j) \\ f(j) \geq n \\ f(i) \leq n \\ f(k) > n \text{ for } j \prec k \prec i \end{cases} \\ 1 & \text{if } i = j \wedge f(i) < n \\ 0 & \text{otherwise} \end{cases} \qquad (13)$$

$$U_i^n(0) = \begin{cases} \prod_{k\preceq i} M(k) & \text{if } \begin{cases} f(i) \leq n \\ f(k) > n \text{ for } k \prec i \end{cases} \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

where the precedence relations '$\prec$' and $\preceq$ were previously introduced.

Consider a customer with priority n entering the queue. Upon arrival he sees a customer of class j. If $U_i^n(j) > 0$, the class j customer may become a class i customer when the arriving customer with priotity n completes service. More precisely, the arriving customer will see, on the average, $U_i^n(j)$ customers of class i upon departure for every class j customer he saw upon arrival. The class j customers are assumed to be already in the system when the customer with priority n arrives.

$U_i^n(0)$ stands for the class i customers which enter from outside as customers of class $g(i)$ and become a class i customer during the system time of the customer with priority n.
For example in Figure 2a: $U_3^1(2) = 1$, $U_4^2(3) = 0$, $U_2^2(2) = 0$, $U_3^2(2) = 1, U_1^4(0) = U_1^3(0) = 1$, $U_2^1(0) = 0$, $U_3^1(0) = 1$.
In Figure 2b: $U_2^1(1) = 0$, $U_3^1(1) = 0.7$, $U_6^3(5) = 0$, $U_5^3(5) = 1$, $U_6^2(5) = 0$, $U_7^2(5) = 2, U_2^5(0) = 1, U_3^1(0) = 0.7, U_7^2(0) = 0$.

A customer of the internal class j enters the system at state $X = (x_1, \ldots, x_N)$. Consider the expected system state $(y_1, \ldots, y_N)$ when he leaves the system by feedback or departure. Five cases have to be distinguished.

1. $i \in EXT \wedge f(i) \geq f(j) + PD$: $y_i = 0$
2. $i \in EXT \wedge f(j) < f(i) < f(j) + PD$: $y_i = \lambda_i t_j^{fin}$
3. $i \in EXT \wedge f(i) \leq f(j)$: $y_i = \sum_{k \preceq i} U_i^{f(j)}(k)\, x_k + \lambda_i W_j(X)$
4. $i \notin EXT \wedge f(i) > f(j)$: $y_i = 0$
5. $i \notin EXT \wedge f(i) \leq f(j)$: In this case y_i is given by:

$$\begin{aligned} y_i \;=\; & \sum_{k \preceq i} U_i^{f(j)}(k)\, x_k \\ & + \lambda_{g(i)} W_j(X)\, U_i^{f(j)}(0)\, 1_{f(g(i)) \geq f(j)+PD} + \lambda_{g(i)} (W_j(X) - t_j^{fin})\, U_i^{f(j)}(0)\, 1_{f(g(i)) < f(j)+PD} \\ & + \sum_{\substack{pre(k)=pre(j) \\ f(k)>f(j)}} A(k)\, U_i^{f(j)}(k) + \frac{1}{2} \sum_{\substack{pre(k)=pre(j) \\ f(k)=f(j) \\ A(k)>0}} (A(k) - 1_{k=j})\,(U_i^{f(j)}(k) + 1_{k=i}) + \sum_{\substack{pre(i)=pre(j) \\ f(i)<f(j)}} A(i) \end{aligned} \tag{15}$$

The variable y_i represents the expected number of class i customers that are in the system when a customer of an internal class j feeds back. In case 5 three types of customers may be of class i upon departure:
1. Customers that are already in the system when the class j customer enters.
2. Preempting and non-preempting customers that enter during system time (second line).
3. Customers that arrive together with the class j customer due to forking (third line).

Equation (15) is illustrated by an example taken from Figure 2a:
For $j = 2$: $y_1 = \lambda_1 t_2^{fin}$, $y_2 = U_2^2(1)\, x_1 + U_2^2(2)\, x_2 + \lambda_1 (W_2(\underline{X}) - t_2^{fin})\, U_2^2(0)$, $y_3 = U_3^2(1)\, x_1 + U_3^2(2)\, x_2 + U_3^2(3)\, x_3 + \lambda_1 (W_2(\underline{X}) - t_2^{fin})\, U_3^2(0)$, $y_4 = 0$. Since class 3 is the successor class of class 2, an arriving class 3 customer sees system state (y_1, y_2, y_3, y_4) upon arrival (on the average).

For external classes, the destiny of the customer served at arrival time has to be considered additionally. This leads to

$$\begin{aligned} R_j'(i) \;=\; & \sum_{k \preceq i} \rho_k \Big(1_{\substack{f(j) \geq f(k)+PD \\ k=i}} + (1 - \tilde{z}_k)\, 1_{\substack{f(k)<f(j)<f(k)+PD \\ \wedge k=i}} \\ & + \tilde{z}_k\, 1_{f(k)<f(j)<f(k)+PD}\, M(k+1)\, (U_i^{f(j)}(k+1)\, 1_{f(k+1)>f(j)} + 1_{f(k+1) \leq f(j) \wedge k+1=i}) \\ & + U_i^{f(j)}(k)\, 1_{f(j) \leq f(k)} \Big) \end{aligned} \tag{16}$$

where $k+1$ denotes the successor class of class k on the way to class i.

The transformation function of the system state X can be devided into the matrices A_j, which form the part that depends on steady state. The vector R_j represents the remaining parts. A_j and R_j can be derived from equation (15) and are omitted in this context. R_j' must be added for external customers. We obtain
for internal classes j: $Y = A_j X + R_j$,
for external classes j: $Y = A_j(\Lambda W - \rho) + R_j + R_j'$

7 The System of Linear Equations

Each customer has an ancestor that arrived as an external Poisson arrival seeing steady state. Via feedback the ancestor suffers several transformations and finally enters the system as a class j customer. These linear transformations are represented by the measures A, R_j, and R_j' which have been defined in the previous section. As outlined in Section 4, the expected system state at feedback time of a customer of class $pre(j)$ can be derived by continously transforming steady state by multiplication of the matrices $A_{g(j)}, \ldots, A_{pre(j)}$ under consideration of R_j and R_j'. The expected system time of a class j customer can be expressed by the linear function $W_j = T_j X_j + r_j$, where X_j is a linear function of the steady state. Thus, we can write for all classes j $(j = 1, \ldots, N)$:

$$\begin{aligned} W_j \;=\; & T_j \left(\prod_{k=g(j)}^{pre(j)} A_k \right) \Lambda W \;-\; T_j \left(\prod_{k=g(j)}^{pre(j)} A_k \right) \rho \;+ \\ & T_j \left(\prod_{k=g(j)+1}^{pre(j)} A_k \right) R_{g(j)}' \;+\; T_j \sum_{s \prec j} \left(\prod_{k=s+1}^{pre(j)} A_k \right) R_s \;+\; r_j \end{aligned} \tag{17}$$

Thereby a system of linear equations is formed, where the expected system times are expressed as linear functions of the steady state. This system of linear equations can be solved by using standard techniques. The size of the matrix is $N \times N$, which is tractable. Other interesting performance measures can be derived from the expected system times.

8 Numerical Example

Even for the analysis of a simple example, complex formulae have to be computed (in [Simon 84] a very simple model is presented). Therefore, in this section results are presented to demonstrate the practical significance of the model.

The feedback model of a communication controller shown in Figure 1b is the basic system of the example. Four schedules are analyzed and compared, where different priorities are assigned to the classes.

- In Schedule A the priorities are assigned per layer. The lower the layer, the higher the priority.
- In Schedule B the priorities are assigned in opposite order to Schedule A. The higher the layer, the higher the priority.
- In Schedule C a customer in service holds the server until it leaves the system at the end of his chain.
- In Schedule D the priorities are assigned opposite to Schedule C.

The following table shows the service time distributions and priorities of each class. The preemption-distance is set to 2. In Figure 8 the expected time a customer needs to proceed through the upper chain of the model (sending part) is depicted over the system load. This time is sum of the expected system times for the classes 1 to 4.

Class	Service Time		Priority			
	mean	distr.	A	B	C	D
1	5.0	M	1	4	1	4
2	2.5	E_2	2	3	2	3
3	1.5	D	3	2	3	2
4	2.5	D	4	1	4	1
5	2.5	D	4	1	1	4
6	1.5	D	3	2	2	3
7	2.5	E_2	2	3	3	2
8	5.0	M	1	4	4	1

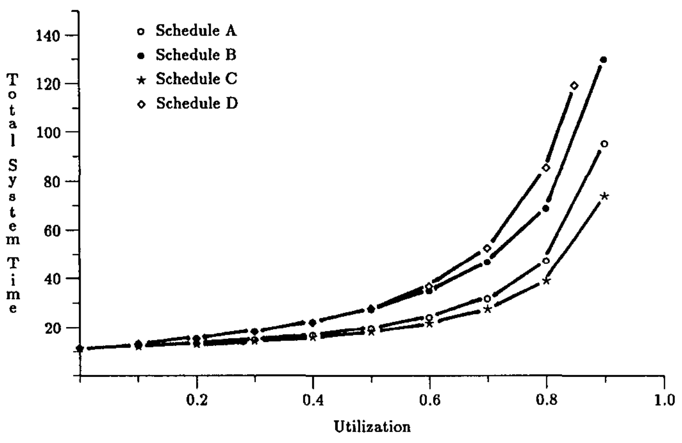


Figure 4: Total System Times for the Sender Part

Schedule C marks the best of the four schedules with respect to the overall system times, while Schedule D is the worst. Since Schedule A corresponds to Schedule C for the sending side, it exhibits better performance than Schedule B for this side. For the receiving side they perform in opposite order.

The technique presented is implemented on a PCS CADMUS 9940 computer with floating point coprocessor and was tested extensively. The execution time for all 40 analyzed models of this example is about 40 secs. More complex models will be presented in further reports (see also [Pate 89]).

9 Conclusion

Models of systems where different processes execute on a single processor often contain feedback queues with priorities. These systems have usually been analyzed only by simulation. In this paper analytical solutions for an open queueing system of this type are presented. The model includes features, that are important for modelling real computer systems, such as branching, forking, and a priority strategy. Preemption-distance priorities allow the flexible representation even of sophisticated scheduling strategies.

The technique presented is implemented and is now available as a tool for performance analysis. Its suitability for applied performance evaluation is outlined by the evaluation of a simple model of a communication controller. The computational requirements are very small compared to other techniques like simulation. Thus, the tool may be used to analyze large scale models of that type as well as for the analysis of submodels in hierarchical techniques.

Anknowledgements:

We are thankful to numerous persons for their helpful hints and comments. Special thanks are to Prof. U. Herzog for initiating this research, and to A. Sieber and L. Dowdy for reading the drafts with patience and competence.

References

[Alta 88] V.Altaminaro, B.Fontana: Models to Evaluate Response Times in Single-Processor Systems and their Apllication to a Multiprocessor System, Proc. 12. International Teletraffic Congress, Torino, Italy, June 88, 4.3A5.1-10

[Cobham 54] A.Cobham: Priority Assignments in Waiting Line Problems, Operations Ressearch 2, 70-76, 1954

[Daigle 81] J.Daigle, C.E.Houstis: Analysis of a Task Oriented Multipriority Queueing System, IEEE Transactions on Comm., Vol. 29, No. 11, 1669-1677, 1981

[Gay 75] T.W.Gay, P.H.Seaman: Composite Priority Queues, IBM Journal of Research and Developement, Vol.19 , 78-81, 1975

[Halsa 88] F.Halsall: Data Communications, Computer Networks and OSI, 2nd ed., Addison-Wesley, 1988.

[Herzog 73] U.Herzog: Verkehrsfluß in Datennetzen, habilitation thesis, University of Stuttgart, 1973 (in German)

[Herzog 75] U.Herzog: Optimal Scheduling Strategies for Real-Time-Computers, IBM Journal of Research and Developement, Vol. 19, 494-504, 1975

[Herzog 76] U.Herzog: Priority Models for Communication Processors Including System Overhead, International Telegraffic Congress 8, Melbourne, 623/1-623/7, 1976

[Herzog 86] U.Herzog: Flexible Networks of Tightly and Loosely Coupled Processors, Proc. Teletraffic Analysis and Computer Performance Evaluation, O.J. Boxma et al. (ed.), North-Holland, 439-446, 1986.

[Jewell 67] W.S.Jewell: A Simple Proof of $Q = \lambda W$, Oper.Res. 15, 1109-1116, 1967

[Kleinr 75] L.Kleinrock: Queueing Systems Volume I: Theory, John Wiley & Sons, New York, 1975

[Kleinr 76] L.Kleinrock: Queueing Systems Volume II: Computer Applications, John Wiley & Sons, New York, 1976

[Pate 88] M.Paterok, O.Fischer: Feedback Queues with Preemption-Distance Priorities, Technical Report, Institute for Mathematical Machines and Data Processing VII, University of Erlangen-Nuremberg, 1988

[Pate 89] M.Paterok, O.Fischer: Feedback Queues with Preemption-Distance Priorities, Bulk Arrivals, Forking and Branching, Technical Report, Institute for Mathematical Machines and Data Processing VII, University of Erlangen-Nuremberg, 1989

[Schrag 67] L.E.Schrage: The Queue M/G/1 with Feedback to Lower Priority Queues, Management Science, Vol.13 No.7, 466-474, 1967

[Simon 84] B.Simon: Priority Queues with Feedback, JACM, Vol.31 No.1, 134-149, 1984

[Wolff 82] R.W.Wolff: Poisson Arrivals see Time Averages, Operations Research, Vol. 20, 4, 223-231, 1982

Mail to the authors can currently be sent to Martin Paterok, Universität Erlangen-Nürnberg, Institut für mathematische Maschinen und Datenverarbeitung VII, Lehrstuhl für Rechnerarchitektur und -Verkehrstheorie, Martensstr. 3, D-8520 Erlangen, Federal Republic of Germany