

# A continuum of time-sharing scheduling algorithms\*

by LEONARD KLEINROCK

*University of California*  
Los Angeles, California

## INTRODUCTION

The study of time-sharing scheduling algorithms has now reached a certain maturity. One need merely look at a recent survey by McKinney<sup>1</sup> in which he traces the field from the first published paper in 1964<sup>2</sup> to a succession of many papers during these past six years. Research which is currently taking place within the field is of the nature whereby many of the important theoretical questions will be sufficiently well answered in the very near future so as to question the justification for continuing extensive research much longer without first studying the overall system behavior.

Among the scheduling algorithms which have been studied in the past are included the round robin (RR), the feedback model with  $N$  levels ( $FB_N$ ), and variations of these.<sup>1</sup> The models introduced for these scheduling algorithms gave the designer some freedom in adjusting system performance as a function of service time but did not range over a continuum of system behaviors. In this paper we proceed in that direction by defining a model which allows one to range from the first come first served algorithm all the way through to a round robin scheduling algorithm. We also find a variety of other models within a given family which have yet to be analyzed.

Thus the model analyzed in this paper provides to the designer a degree of freedom whereby he may adjust the relative behavior for jobs as a function of service time; in the past such a parameter was not available. Moreover, the method for providing this adjustment is rather straightforward to implement and is very easily changed by altering a constant within the scheduler.

\* This work was supported by the Advanced Research Projects Agency of the Department of Defense (DAHC15-69-C-0285).

## A GENERALIZED MODEL

In an earlier paper<sup>3</sup> we analyzed a priority queueing system in which an entering customer from a particular priority group was assigned a zero value for priority but then began to increase in priority linearly with time at a rate indicative of his priority group. Such a model may be used for describing a large class of time-sharing scheduling algorithms. Consider Figure 1. This figure defines the class of scheduling algorithms which we shall consider. The principle behind this class of algorithms is that when a customer is in the system waiting for service then his priority (a numerical function) increases from zero (upon his entry) at a rate  $\alpha$ ; similarly, when he is in service (typically with other customers sharing the service facility simultaneously with him as in a processor shared system<sup>4</sup>) his priority changes at a rate  $\beta$ . All customers possess the same parameters  $\alpha$  and  $\beta$ . Figure 1 shows the case where both  $\alpha$  and  $\beta$  are positive although, as we shall see below, this need not be the case in general. The history of a customer's priority value then would typically be as shown in Figure 1 where he enters the system at time  $t_0$  with a 0 value of priority and begins to gain priority at a rate  $\alpha$ . At time  $t_1$  he joins those in service after having reached a value of priority equal to  $\alpha(t_1 - t_0)$ . When he joins those in service he shares on an equal basis the capacity of the service facility and then continues to gain priority at a different rate,  $\beta$ . It may be that a customer is removed from service before his requirement is filled (as may occur when one of the slopes is negative); in this case, his priority then grows at a rate of  $\alpha$  again, etc. At all times, the server serves *all* those with the highest value of priority. Thus we can define a slope for priority while a customer is queueing and another slope for priority while a cus-

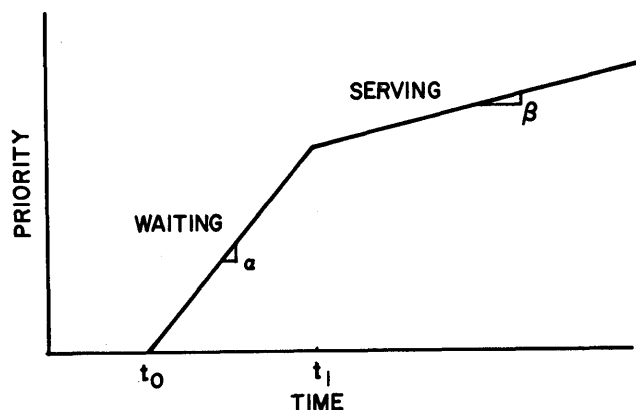


Figure 1—Behavior of the time-varying priority

customer is being served as

$$\text{queueing slope} = \alpha \quad (1)$$

$$\text{serving slope} = \beta. \quad (2)$$

A variety of different kinds of scheduling algorithms follow from this model depending upon the relative values of  $\alpha$  and  $\beta$ . For example, when both  $\alpha$  and  $\beta$  are positive and when  $\beta \geq \alpha$  then it is clear that customers in the queue can never catch up to the customer in service since he is escaping from the queueing customers at least as fast as they are catching up to him; only when the customer in service departs from service after his completion will another customer be taken into service. This new customer to be taken into the service facility is that one which has the highest value of priority. Thus we see that for the range

$$0 < \alpha \leq \beta \quad (3)$$

we have a pure *first come first served* (FCFS) scheduling algorithm. This is indicated in Figure 2 where we show the entire structure of the general model.

Now consider the case in which

$$0 \leq \beta \leq \alpha. \quad (4)$$

This is the case depicted in Figure 1. Here we see that the group of customers being served (which act among themselves in a processor-shared round robin (RR) fashion) is attempting to escape from the group of customers in the queue; their attempt is futile, however, and it is clear from this range of parameters that the queueing customers will eventually each catch up with the group being served. Thus the group being served is selfishly attempting to maintain the service capacity for themselves alone and for this reason we refer to this system as the *selfish round robin* (SRR).

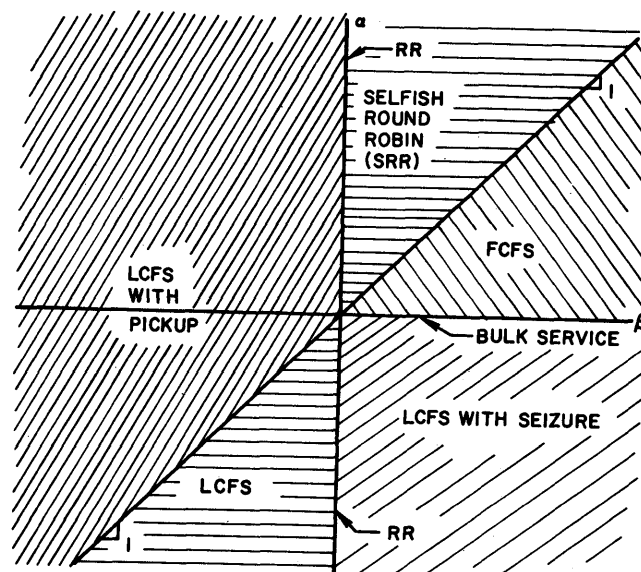


Figure 2—The structure of the general model

What happens in this case is that entering customers spend a period of time in the queue and after catching up with the serving group proceed to be served in a round robin fashion. The duration of the time they spend in the queue depends upon the relative parameters  $\alpha$  and  $\beta$  as we shall see below. It is clear however that for  $\beta = 0$  we have the case that customers in service gain no priority at all. Thus any newly entering customer would have a value of priority exactly equal to that of the group in service and so will immediately pass into the service group. Since all serving customers share equally, we see that the limiting case,  $\beta = 0$ , is a processor-sharing round robin (RR) scheduling algorithm! It happens that SRR yields to analysis very nicely (whereas some of the other systems mentioned below are as yet unsolved) and the results of this analysis are given in the next section.

Another interesting range to consider is that for which

$$\alpha \leq \beta < 0. \quad (5)$$

Here we have the situation in which queueing customers lose priority faster than serving customers do; in both cases however, priority decreases with time and so any newly entering customer will clearly have the highest priority and will take over the complete service facility for themselves. This most recent customer will continue to occupy the service facility until either he leaves due to a service completion or some new customer enters the system and ejects him. Clearly what we have here is a classical *last come first served* (LCFS) scheduling algorithm as is indicated in Figure 2.

Now consider the range

$$\alpha < 0 < \beta. \quad (6)$$

In this case a waiting customer loses priority whereas a customer in service gains priority. When an arriving customer finds a customer in service who has a negative value for priority then this new customer preempts the old customer and begins service while at the same time his priority proceeds to increase at a rate  $\beta$ ; from here on no other customer can catch him and this customer will be served until completion. Upon his completion, service will then revert back to that customer with the largest value of priority. Since customers lose priority with queuing time, then all customers in the system when our lucky customer departed must have negative priority. One of these will be chosen and will begin to gain priority; if now he is lucky enough to achieve a positive priority during his service time, then he will seize the service facility and maintain possession until his completion. Thus we call this range *LCFS with seizure* (see Figure 2).

In the special case

$$\alpha = 0 < \beta \quad (7)$$

we have the situation in which a newly emptied service facility will find a collection of customers who have been waiting for service and who have been kept at a zero value priority. Since all of these have equal priority they will all be taken into service simultaneously and then will begin to gain priority at a rate  $\beta > 0$ . Any customers arriving thereafter must now queue in bulk fashion since they cannot catch up with the current group in service. Only when that group finishes service completely will the newly waiting group be taken into service. We refer to this case as *bulk service*.

The last case to consider is in the range

$$\beta < 0, \quad \beta < \alpha. \quad (8)$$

In this case a customer being served always loses priority whereas a queuing customer loses priority at a slower rate or may in fact gain priority. Consequently, serving customers will tend to "run into" queuing customers and pick them up into the service facility at which point the entire group continues to decrease in priority at rate  $\beta$ . We refer to this region as *LCFS with pickup* (see Figure 2).

Thus Figure 2 summarizes the range of scheduling algorithms which this two-parameter priority function can provide for us. We have described a number of regions of interest for this class of algorithms. The FCFS, LCFS, and RR systems, of course, are well known and solved. The three regions given by Equations 4, 6, and 8 are as yet unsolved. As mentioned

before, the SRR system yields very nicely to analysis and that analysis is given in this paper. This system has the interesting property that we may vary its parameters and pass smoothly from the FCFS system through the SRR class to the familiar RR system. The others (LCFS with seizure and LCFS with pickup) are as yet unsolved and appear to be more difficult to solve than the SRR. Of course other generalizations to this scheme are possible, but these too are yet to be studied. Among these generalizations, for example, is the case where each customer need not have the same  $\alpha$  and  $\beta$ ; also one might consider the case where growth (or decay) of priority is a non-linear function of time. Of all these cases we repeat again that the SRR has been the simplest to study and its analysis follows in the next section.

#### THE SELFISH ROUND ROBIN (SRR) SCHEDULING ALGORITHM

We consider the system for which customers in service gain priority at a rate less than or equal to the rate at which they gained priority while queuing (see Equation (4)); in both cases the rate of gain is positive. We assume that the arrival process is Poisson at an average rate of  $\lambda$  customers per second

$$P[\text{inter-arrival time} \leq t] = 1 - e^{-\lambda t} \quad t \geq 0 \quad (9)$$

and that the service times are exponentially distributed

$$P[\text{service time} \leq x] = 1 - e^{-\mu x} \quad x \geq 0 \quad (10)$$

Thus the two additional parameters of our system are

$$\text{average arrival rate} = \lambda \quad (11)$$

$$\text{average service time} = 1/\mu \quad (12)$$

As usual, we define the utilization factor

$$\rho \equiv \lambda/\mu \quad (13)$$

For the range of  $\alpha, \beta$  under consideration it is clear that once a customer enters the service facility he will not leave until his service is complete. Consequently, we may consider the system as broken into two parts: first, a collection of queued customers; and second, a collection of customers in service. Figure 3 depicts this situation where we define\*

$$T_w = E[\text{time spent in queue box}] \quad (14)$$

$$T_s = E[\text{time spent in service box}] \quad (15)$$

$$N_w = E[\text{number in queue box}] \quad (16)$$

$$N_s = E[\text{number in service box}] \quad (17)$$

\* The notation  $E[x]$  reads as "the expectation of  $x$ ."

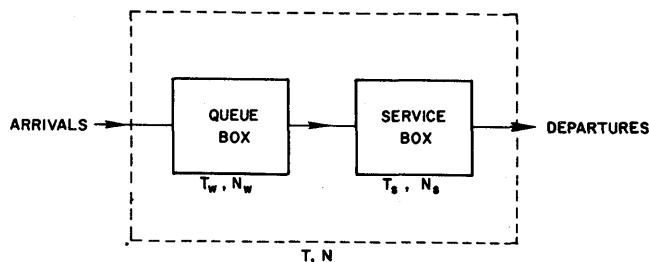


Figure 3—Decomposition of the SRR system

We further define

$$T = T_w + T_s = E[\text{time in system}] \quad (18)$$

$$N = N_w + N_s = E[\text{number in system}] \quad (19)$$

Due to the memoryless property of the exponential service time distribution, it is clear that the average number in system and average time in system are independent of the order of service of customers; this follows both from intuition and from the conservation law given in Reference 5. Thus we have immediately

$$T = \frac{1/\mu}{1 - \rho} \quad (20)$$

$$N = \frac{\rho}{1 - \rho} \quad (21)$$

For our purposes we are interested in solving for the average response time for a customer requiring  $t$  seconds of service; that is for a customer requiring  $t$  seconds of complete attention by the server or  $2t$  seconds of service from the server when he is shared between two customers, etc. Recall that more than one customer may simultaneously be sharing the attention of the service facility and this is just another class of processor-sharing systems.<sup>4</sup> Thus our goal is to solve for

$$T(t) = E[\text{response time for customer requiring } t \text{ seconds of service}] \quad (22)$$

where by response time we mean total time spent in the system. The average of this conditional response time without regard to service time requirement is given by Equation 20. Due to our decomposition we can write immediately

$$T(t) = T_w(t) + T_s(t) \quad (23)$$

where  $T_w(t)$  is the expected time spent in the queue box for customers requiring  $t$  seconds of service and  $T_s(t)$  is the expected time spent in the service box for customers requiring  $t$  seconds of service. Since the

system is unaware of the customer's service time until he departs from the system, it is clear that the time he spends in the queue box must be independent of this service time and therefore

$$T_w(t) = T_w \quad (24)$$

Let us now solve for  $T_s(t)$ . We make this calculation by following a customer, whom we shall refer to as the "tagged" customer, through the system given that this customer requires  $t$  seconds of service. His time in the queue box will be given by Equation 24. We now assume that this tagged customer has just entered the service box and we wish to calculate the expected time he spends there. This calculation may be made by appealing to an earlier result. In Reference 4, we studied the case of the processor-shared round robin system (both with and without priorities). Theorem 4 of that paper gives the expected response time conditioned on service time and we may use that result here since the system we are considering, the service box, appears like a round robin system. However, the arrival rate of customers to the service box conditioned on the presence of a tagged customer in that box is no longer  $\lambda$ , but rather some new average arrival rate  $\lambda'$ . In order to calculate  $\lambda'$  we refer the reader to Figure 4. In this figure we show that two successive customers arrive at times  $t_1$  and  $t_2$  where the average time between these arrivals is clearly  $1/\lambda$ . The service group moves away from the new arrivals at a rate  $\beta$  and the new arrivals chase the service group at a rate  $\alpha$ ; as shown in Figure 4, these two adjacent arrivals catch up with the service group where the time between their arrival to the service box is given by  $1/\lambda'$ . Recall that the calculation we are making is conditioned on the fact that our tagged customer remains in the service box during the interval of interest; therefore the service box is guaranteed not to empty over the period of our

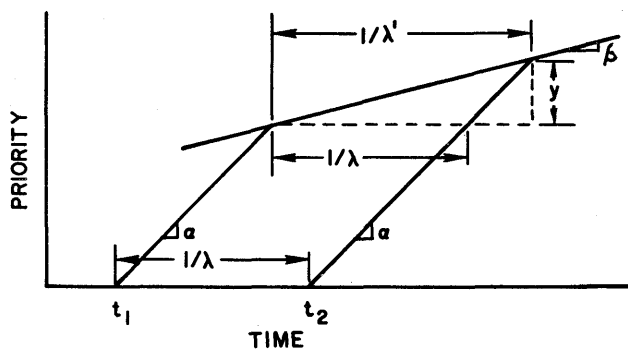


Figure 4—Calculation of the conditional arrival rate to the service box

calculations.  $\lambda'$  is easily calculated by recognizing that the vertical offset  $y$  may be written in the following two ways

$$y = \left(\frac{1}{\lambda'}\right)\beta$$

$$y = \left(\frac{1}{\lambda'} - \frac{1}{\lambda}\right)\alpha$$

and so we may solve for  $\lambda'$  as follows

$$\lambda' = \lambda \left(1 - \frac{\beta}{\alpha}\right) \quad (25)$$

(recall that for the SRR system  $\beta \leq \alpha$ ). For convenience we now define

$$\rho' = \lambda'/\mu \quad (26)$$

We may now apply Theorem 4 of Reference 4 and obtain the quantity we are seeking, namely,

$$T_s(t) = \frac{t}{1 - \rho'} \quad (27)$$

The only difference between Equation 27 and the referenced theorem is that here we use  $\rho'$  instead of  $\rho$  since in all cases we must use the appropriate utilization factor for the system under consideration. That theorem also gives us immediately that

$$N_s = \frac{\rho'}{1 - \rho'} \quad (28)$$

This last equation could be derived from Equation 27 and the application of Little's result<sup>6</sup> which states that

$$\lambda' T_s = N_s \quad (29)$$

and where

$$T_s \equiv \int_0^{\infty} T_s(t) \mu e^{-\mu t} dt$$

$$T_s = \frac{1/\mu}{1 - \rho'} \quad (30)$$

We may now substitute Equation 27 into Equation 23 to give

$$T(t) = T_w + \frac{t}{1 - \rho'} \quad (31)$$

In order to evaluate  $T_w$  we form the average with respect to  $t$  over both sides of Equation 31 to obtain

$$\int_0^{\infty} T(t) \mu e^{-\mu t} dt = T_w + \int_0^{\infty} \frac{t}{1 - \rho'} \mu e^{-\mu t} dt$$

and so

$$T = T_w + \frac{1/\mu}{1 - \rho'} \quad (32)$$

Using Equation 20 we have the result

$$T_w = \frac{1/\mu}{1 - \rho} - \frac{1/\mu}{1 - \rho'} \quad (33)$$

Upon substituting Equation 33 into Equation 31 we obtain our final result as

$$T(t) = \frac{1/\mu}{1 - \rho} + \frac{t - 1/\mu}{1 - \rho'} \quad (34)$$

Another convenient form in which to express this result is to consider the average time wasted in this SRR system where wasted time is any extra time a customer spends in the system due to the fact that he is sharing the system with other customers. Thus, by definition, we have

$$W(t) = T(t) - t \quad (35)$$

and this results in

$$W(t) = \frac{\rho/\mu}{1 - \rho} + \frac{(t - 1/\mu)\rho'}{1 - \rho'} \quad (36)$$

In both Equations 34 and 36 we observe for the case of a customer whose service time is equal to the average service time ( $1/\mu$ ) that his average response time and average wasted time are the same that he would encounter for any SRR system; thus his performance is the same that he would receive, for example, in a FCFS system. We had observed that correspondence between the RR system and the FCFS system in the past; here we show that it holds for the entire class of SRR systems. In Figure 5 below we plot the performance of the class of SRR systems by showing the dependence of the wasted time for a customer whose service time is  $t$  seconds as a function of his service time. We show this for the case  $\rho = 3/4$  and  $\mu = 1$ . The truly significant part regarding the behavior of the SRR system is that the dependence of the conditional response time upon the service time is linear. Once observed, this result is intuitively pleasing if we refer back to Figure 3. Clearly, the time spent in the queue box is some constant independent of service time. However, the time spent in the service box is time spent in a round robin system since all customers in that box share equally the capability of the server; we know that the response time for the round robin system is directly proportional to service time required (in fact, as shown in Reference 8, this statement is true even for arbitrary service time). Thus the total time spent in the SRR system must be equal to some con-

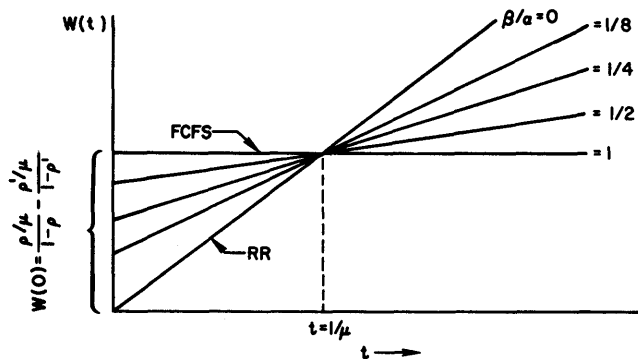


Figure 5—Performance of the SRR system

stant plus a second term proportional to service time as in fact our result in Equation 34 indicates. Again we emphasize the fact that customers whose service time requirements are greater than the average service time requirement are discriminated against in the SRR system as compared to a FCFS system; conversely, customers whose service time requirement is less than the average are treated preferentially in the SRR system and compared to the FCFS system. The degree of this preferential treatment is controlled by the parameters  $\alpha$  and  $\beta$  giving the performance shown in Figure 5.

## CONCLUSION

In this paper we have defined a continuum of scheduling algorithms for time-shared systems by the introduction of two new parameters,  $\alpha$  and  $\beta$ . The class so defined is rather broad and its range is shown in Figure 2. We have presented the analysis for the range of parameters that is given in Equation 4 and refer to this new system as the selfish round robin (SRR) scheduling algorithm. Equation 34 gives our result for the average response time conditioned on the required service time and we observed that this result took the especially simple form of a constant plus a term linearly dependent upon the service time. Moreover, we observe that the parameters  $\alpha$  and  $\beta$  appear in the solution only as the ratio  $\beta/\alpha$ . This last is not overly surprising since a similar observation was made in the paper<sup>3</sup> which was our point of departure for the model described herein; namely, there too the slope parameters

appeared only as ratios. Thus in effect we have introduced one additional parameter, the ratio  $\beta/\alpha$ , and it is through the use of this parameter that the designer of a time-sharing scheduling algorithm is provided a degree of freedom for adjusting the extent of discrimination based upon service time requirements which he wishes to introduce into his algorithm; the implementation of this degree of freedom is especially simple. The range of the algorithm is from the case where there is zero discrimination based on service time, namely the FCFS system, to a case where there is a strong degree of discrimination, namely the RR system.

The mathematical simplicity of the SRR algorithm is especially appealing. Nevertheless, the unsolved systems referred to in this paper should be analyzed since they provide behavior distinct from the SRR. In any event, this continuum of algorithms is simply implemented in terms of the linear parameters  $\alpha$  and  $\beta$ , and the scheduling algorithm can easily choose the desired behavior by adjusting  $\alpha$  and  $\beta$  appropriately.

## REFERENCES

- 1 J M MCKINNEY  
*A survey of analytical time-sharing models*  
Computing Surveys Vol 1 No 2 pp 105-116 June 1969
- 2 L KLEINROCK  
*Analysis of a time-shared processor*  
Naval Research Logistics Quarterly Vol 11 No 1 pp 59-73  
March 1964
- 3 L KLEINROCK  
*A delay dependent queue discipline*  
Naval Research Logistics Quarterly Vol 11 No 4 pp 329-341  
December 1964
- 4 L KLEINROCK  
*Time-shared systems: A theoretical treatment*  
JACM Vol 14 No 2 pp 242-261 April 1967
- 5 L KLEINROCK  
*A conservation law for a wide class of queueing disciplines*  
Naval Research Logistics Quarterly Vol 12 No 2 pp 181-192  
June 1965
- 6 J D C LITTLE  
*A proof of the queueing formula  $L = \lambda W$*   
Operations Research Vol 9 pp 383-387 1961
- 7 L KLEINROCK  
*Distribution of attained service in time-shared systems*  
J of Computers and Systems Science Vol 3 pp 287-298  
October 1967
- 8 M SAKATA S NOGUCHI J OIZUMI  
*Analysis of a processor shared queueing model for time-sharing systems*  
Proc of the Second Hawaii International Conference on  
Systems Science pp 625-628  
University of Hawaii Honolulu Hawaii January 22-24 1969