PREEMPTIVE PRIORITY QUEUES

Consider an M/M/1 queuing system in which there are two classes of customers--high and low priority--who arrive under independent Poisson processes with parameters of, respectively, λ_1 and λ_2. We assume that:

- No low-priority customer enters service when any high-priority customers are present.

- If a low-priority customer is in service, his service will be interrupted at once if a high-priority customer arrives, and will not be resumed until the system is again clear of high-priority customers.

- Service times for different customers are independent, and all follow an exponential distribution with parameter μ.

We seek here the average queue length and mean time in system for high-priority (hereafter Type 1) customers, and the corresponding quantities for low-priority (Type 2) customers.

Easy Start

The analysis for L_1 and W_1 (L and W for the Type-1's) is straightforward. Type-2 customers are nonexistent as far as Type-1 service is concerned, so the Type-1's enjoy an M/M/1 queuing system with arrival λ_1 and service rate μ. We can therefore invoke previous results and write:

$$W_1 = 1/(\mu - \lambda_1) \qquad\qquad L_1 = \lambda_1/(\mu - \lambda_1)$$

Light Turbulence

Obviously, matters are more complicated for the Type-2 analysis. We proceed first to calculate $E_2(w\Box k_1,k_2)$, the conditional mean time in the system for a Type-2 customer who arrives to find k_1 Type-1's and k_2 Type-2's already there.

First of all, we note that all Type 1's present will have to be served before any of the Type 2's get any attention. Moreover, any other Type 1's who arrive while the present Type-1's are being served--and any Type 1's who arrive while this second group is getting served, etc.--will be handled before the server turns to the first Type-2. Thus, even if there are only five Type-1's now, there might be 25 Type-1 services before the first Type-2 service begins. On average, how long will it take to clear the system of the k_1 Type-1's and their high-priority "descendants"?

Things are easier if we make the observation that the mean time it takes the server to get rid of the Type-1's is independent of the order in which she serves them. Thus, it does not change the problem mathematically if we assume that the server first handles the longest waiting of the k_1 Type-1's and his "descendants," then turns to the second-longest waiting and proceeds similarly, and continues in this way until she gets to the k_i^{th} and her descendants. The average time to handle the first group (longest waiting plus his descendants) is simply the average busy period for an M/M/1 queue with parameters λ_1 and μ. (Do you see why? Ask us if not.) As we have seen earlier, this busy period B_1 follows:

$$B_1 = 1/(\mu - \lambda_1)$$

Similarly, the time to deal with each of the other k groups of Type-1's is on average B_1. Thus, the total mean time to get rid of the Type-1 customers and turn to the low-priority group is $k_1 B_1$.

The Type-2 Ordeal

A Type-2 customer's "service" can be humiliating and protracted. He starts service and either (i) finishes up before any Type-1's arrive, or(ii) is instead preempted by a Type-1 arrival. Once such preemption occurs, the server will turn to this Type-1 and her descendants and, on average, will not resume service to the Type-2 until a full Type-1 busy period (B_1) has elapsed. And, once service resumes, the Type-2 is essentially back at the beginning: under an exponential service time distribution, he gets no “partial credit” for the time he already spent in service.

To turn the preceding description into mathematics, we need an important general result about exponential processes:

Suppose that the time to the next event in process A is exponential with parameter r_A, and that in process B it is exponential with parameter r_B. Then:

- the time until the next event in either process follows an exponential distribution with parameter $r_A + r_B$.

- The probability the next event comes from process A is $r_A/(r_A + r_B.)$, and from process B is $r_B/(r_A + r_B.)$.

Now, let’s return to the start of a Type-2 service. Two exponential processes will be competing to stop this service: completion of the service, and the arrival of a high-priority customer. From the previous paragraph, we see that the average time until the stoppage will be $1/(\lambda_1 + \mu)$, while the probability that it represents a service completion is $\mu/(\lambda_1 + \mu)$. If service is interrupted rather than completed (probability $\lambda_1/(\lambda_1 + \mu)$), then it will be an average of B_1 time units until the server returns to the Type-2 customer, at which point the customer is effectively back at the beginning. Putting all this stuff together via the conditional expectation rule, we find the following equation for the mean Type-2 service time $E_2(s)$:

$$E_2(s) = [\mu/(\lambda_1 + \mu)][1/(\lambda_1 + \mu)] + [\lambda_1/(\lambda_1 + \mu)][1/(\lambda_1 + \mu) + B_1 + E_2(s)]$$

(The first term on the right covers the case when the Type-2 service is not interrupted; the second covers the case when it is.)

Note that we have one linear equation in one unknown above, so we can solve immediately for $E_2(s)$.

The time it will take the server to clear the system of all k_2 of the Type-2's already present is simply k_2 $E_2(s)$; it will take her an additional $E_2(s)$ to serve the new arrival. It follows that:

$$E_2(w \mid k_1,k_2) = k_1\ B_1 + (k_2 + 1)\ E_2(s)$$

<u>Our Service Ends</u>

We're almost finished. We need recall that k_1 and k_2 are themselves random variables, and thus that, to work out the overall mean time-in-system for a Type-2 customer, we need use the conditional expectation rule once more:

$$W_2 = B_1E(k_1) + E_2(s)E(k_2+1)$$

Now, $E(k_1)$ is just L_1 and $E(k_2+1)$ is just $L_2 + 1$. Thus, we can rewrite this last equation as:

$$W_2 = B_1L_1 + E_2(s)(L_2+1)$$

With a Little help from our friends, we know too that $L_2 = \lambda_2W_2$. Given that we know L_1 and the other constants in the above expression for W_2, it combines with Little's law to give us two linear equations for the two unknowns L_2 and W_2. You can work out the algebra.
