1.2 A Simplified Treatment of a Simple Queue Model

If we are not averse to making some crude simplifying assumptions, then analysis of simple queueing models may be easily done. (These assumptions will be justified later.) For example, consider the queue of Figure 1 with only one server and an infinite number of waiting positions. Let the arrival process of jobs be such that arrivals come at rate λ. Assume that when $\Delta t \to 0$, P{one arrival in time interval Δt}, the probability of one arrival in a time interval Δt, is $\lambda \Delta t$. Similarly, P{no arrivals in time interval Δt] is $1-\lambda \Delta t$ and P{more than one arrival in time interval Δt} will be negligibly small and may be considered to be zero. For the service process of the given server, we assume that the average service rate is μ (mean service time is $1/\mu$). Assume that when the server is working and when $\Delta t \to 0$, P{one departure from the system in time interval Δt] is $\mu \Delta t$, the P{no departures in time interval Δt] is $1-\mu \Delta t$ and P{more than one departure in time interval Δt] is zero.

For this queue, we need to describe the *system state* in some fashion. The system state at any time instant may be taken as the number in the system at that instant. Note that this will include both the number waiting in the queue and the customer currently in service, if any. Let $p_N(t)$ = P{system in state N at time t}. By ignoring terms with $(\Delta t)^2$ and higher order terms, the probability of the system state at time $t+\Delta t$ may then be found as

$$p_0(t+\Delta t) = p_0(t)[1-\lambda \Delta t] + p_1(t)\mu \Delta t \qquad N=0 \qquad (1.1)$$

$$p_N(t+\Delta t) = p_N(t)[1-\lambda \Delta t - \mu \Delta t] + p_{N-1}(t)\lambda \Delta t + p_{N+1}(t)\mu \Delta t$$

$$N>0 \qquad (1.2)$$

subject to the normalisation condition that $\sum_{\forall i} p_i(t) = 1$ for all $t \geq 0$.

Taking the limits as $\Delta t \to 0$, and subject to the same normalisation, we get

$$\frac{dp_0(t)}{dt} = -\lambda p_0(t) + \mu p_1(t) \qquad N=0 \qquad (1.3)$$

$$\frac{dp_N(t)}{dt} = -(\lambda + \mu)p_N(t) + \lambda p_{N-1}(t) + \mu p_{N+1}(t) \qquad N>0 \qquad (1.4)$$

These differential equations (along with the normalisation condition) may be used to get both the *transient* and the *equilibrium* solutions. For the transient solutions, the queue is assumed to start at time *t=0* with some initial state *K*, i.e. $p_i(0)=d_{iK}$ and the differential equations are solved to obtain the state probabilities $p_i(t)$ *i=0, 1, 2…..*μ as a function of the time *t*. For the equilibrium solutions, the conditions invoked are

$$\frac{dp_i(t)}{dt} = 0 \text{ and } p_i(t) = p_i \text{ for } i=0, 1, 2\ldots..\mu$$

For this, defining $r = \lambda/\mu$ *erlangs*, with r *<1* for stability, we get

$$\begin{aligned} p_1 &= r\,p_0 \\ p_{N+1} &= (1+r)p_N - r\,p_{N-1} = r\,p_N = r^{N+1}p_0 \qquad N \geq 1 \end{aligned} \qquad (1.5)$$

Solving Eq. (1.5) with the normalisation condition $\sum p_i = 1$, we get the system state probabilities to be

$$p_i = r^i(1-r) \qquad i = 0, 1, \ldots..,\mu \qquad (1.6)$$

It should be noted that the summation in the normalisation condition would only have a finite value when $r<1$. This condition is therefore required for the queue to be stable. Note that once we know the equilibrium state probabilities, we can use these to compute various mean performance parameters of this *simple queue*, Some examples of these are given next.

(a) Mean Number in System, *N*

$$N = \sum_{i=0}^{\infty} ip_i = \sum_{i=0}^{\infty} ir^i(1-r) = \frac{r}{1-r} \qquad (1.7)$$

(b) Mean Number Waiting in Queue (prior to service), N_q

$$N_q = \sum_{i=1}^{\infty} (i-1)p_i = \frac{r}{1-r} - (1-p_0) = \frac{r}{1-r} - r = \frac{r^2}{1-r} \qquad (1.8)$$

(c) Mean Time Spent in System, *W*

To obtain this, we need to assume *first come first served* (FCFS) service even though the result obtained does actually hold for any other service discipline where "*the server is never idle while the system is non-empty*". Note that this equality only holds for the mean. The second moment and higher moments, as well as the actual distribution will be different for different service disciplines.

Assuming FCFS discipline, consider a job, which arrives to the system when it is in state *k*, i.e. there are *k* users already in the system of which one is currently being served. We need to make one more assumption before we can proceed further with this analysis. We will assume that the mean remaining service time needed to finish serving the customer currently being served, is still *1/m*. This is indeed justified if the service time distribution is *exponential* (with mean *1/m*), which is effectively what is being assumed here. (This is because of the memory-less property of the exponential distribution.) With these assumptions, we get

$$W = \sum_{k=0}^{\infty} \frac{(k+1)}{m} p_k = \frac{1}{m(1-r)} \tag{1.9}$$

(d) Mean Time Spent waiting in Queue, prior to service, W_q

It is easy to see that W_q will always be one mean service time less than the value of W obtained earlier from Eq. (1.9). Hence, we get

$$W_q = W - \frac{1}{m} = \frac{r}{m(1-r)} \tag{1.10}$$

Note that we can also obtain W_q directly if we know the probabilities of the system states. Using the same arguments and assumptions as in (c) above (for deriving W), we can see that a customer arrival which finds the system in state k will encounter a mean queueing delay of k/m. This may also be used to obtain W_q directly as shown below.

$$W_q = \sum_{k=0}^{\infty} \frac{k}{m} p_k = \frac{r}{m(1-r)} \tag{1.11}$$

Note that this mean result will also be independent of the service discipline actually followed.

(e) Server Utilisation

This will be a parameter of interest to the service provider arranging for the server serving the jobs in the queue. Obviously, the service provider would like the server to be utilised as much as possible, even though that might lead to large queueing delays and a large number of jobs waiting for service in the queue.

The server is busy except when the system is empty, i.e. the system state is zero. Therefore, we can see that the utilisation of the server under equilibrium conditions will be $1-p_0 = r$. This quantity is indicative of how hard the server is actually working.

(f) Probability that an arriving customer has to wait for service

In a way, this parameter is indicative of the quality of service being provided by the queue. Obviously, customers will not be happy with the service provided if they have to wait for service every time they arrive at the queue. From the customers' point of view, a good system will be one where they immediately start getting served on arrival.

It is obvious that a customer will get served immediately on arrival, i.e. will not have to wait for service, if it sees the queue to be empty on arrival. The probability of this is simply p_0, the probability that the system is empty and will be given by $1-r$. Note that as the traffic r $(0<r<1)$ increases, the server utilisation improves at the cost of lower customer satisfaction with the quality of service being provided by the queue.

The *simple queue* described above is actually what is referred to as the $M/M/1/\mu$ queue, i.e. a queue with Poisson arrivals, exponentially distributed service times, single server and infinite waiting positions. (The notation used for representing the queue is called *Kendall's Notation* and will be described later.)

In general, we find that the analysis of a single server queue is attractive, because -

(a) It is generally more tractable than the analysis of a multi-server queue

(b) For a queue with C servers, bounding results may be obtained by considering a system with C parallel, single server queues where an arriving customer can join any of the queues randomly. The latter system will always be less efficient than the actual C server queue.