6.3.1 Fork/Join Node without Synchronising Queues in Open or Closed Networks of Infinite Capacity Queues

In this case, the service time encountered by a job entering the fork/join node will be the *maximum of the service times* for the sub-jobs at the k sibling queues. Assuming that the service provided at the sibling queues are independent of each other, the probability density function of this service time (i.e. of the fork/join node) may be found from the probability density function/cumulative distribution function of the service times of the k sibling queues. For doing this, we can use the result that if X and Y are independent random variables and Z is a random variable defined as $Z=max(X,Y)$, then the cumulative distribution function $F_Z(z)$ and probability density function $f_Z(z)$ of Z are respectively given by

$$F_Z(z) = F_X(z)F_Y(z) \tag{6.39}$$

$$f_Z(z) = f_X(z)F_Y(z) + f_Y(z)F_X(z) \tag{6.40}$$

where $F_X(x)$, $F_Y(y)$, $f_X(x)$ and $f_Y(y)$ are the cumulative distribution function and probability density function of X and Y, respectively.

Note that the probability density function and cumulative distribution function of Z may be found using Eqs. (6.39) and (6.40), if the probability density function and cumulative distribution function of X and Y are known. This result may also be easily extended for the case of k random variables corresponding to the service times of the k sibling queues. One way to do this will be to take the maximum of any two random variables and then take the maximum of this result with the next random variable and continue this until all the variables have been considered. Given the service time distributions of the individual (single server) sibling queues, the overall service time distribution of a job entering the fork/join node can then always be found. We can then also use this service time distribution to find the mean and SQV of the job's service time in a fork/join node of this type.

Even though the above calculation of the service time distribution can be done for any given sub-job service time distribution at the sibling queues, the results are greatly simplified if we assume the sub-jobs to be independent, exponentially distributed random variables. In this case, let $1/m_i$ be the mean of the (exponentially distributed) service time of a sub-job at the i^{th} sibling queue, $i=1,......,k$. Let X be the random variable denoting the overall service time of a job at the fork/join node without synchronising queues. Using the earlier approach and Eq. (6.39), we can then write the cumulative distribution function $F_X(x)$ of the overall service time at this fork/join node as

$$F_X(x) = \prod_{i=1}^{k} \left(1 - \exp(-m_i x)\right) \tag{6.41}$$

Open Network

In this case, we consider the situations where there are fork/join nodes of this type in an open network of GI/G/m queues or if such nodes are being considered in isolation. We can then find the cumulative distribution function and probability density function of the overall service times at the fork/join nodes; we can use Eq. (6.41) directly if the sub-jobs have independent, exponentially distributed service times. We can then use these distributions to find the mean and SQV of the resultant overall service time random variable at each of the fork/join nodes. Using these two service parameters, the approach of Section 6.2 may be directly applied to obtain the required solution.

Closed Network

In this case, we consider the situation where there are fork/join nodes of this type in a closed network. For analysing such a network, we would like to apply the MVA or the Convolution Algorithms. This however requires that the service times at all the queues should be exponentially distributed in nature. To approximately satisfy this condition, we fit an exponential distribution to the resultant distribution of the overall service time at each of the fork/join nodes. Note that the resultant distributions are the ones obtained as the distribution of the maximum of the k random service times at each of the k sibling queues of a fork/join node. This may be done by simply matching the first moments, as an exponential distribution is completely characterised by its mean. However, simulations show that a somewhat better way is to minimise the mean square error between the two distributions to get the best exponential fit. For this, let $F_X(x)$ be the resultant cumulative distribution function of the service time X at the fork/join node. (Note that if the sibling queues have exponentially distributed service times then this will be given by Eq. (6.41).) Let $F_{estimated}(x)$ be the cumulative distribution function (to be found) of the exponentially distributed minimum mean square error fit to $F_X(x)$. This may then be found as

$$F_{estimated}(x) = 1 - e^{-\hat{U}x} \qquad \text{exponential fit to } F_X(x) \tag{6.42}$$

$$e(x,\hat{U}) = \int_0^{\infty} [F_X(x) - F_{estimated}(x)]^2\, dx \qquad \text{mean square error} \tag{6.43}$$

$$\hat{U}_{\min} = \min_{\hat{U}} [e(x,\hat{U})] \qquad \text{minimise the mean square error} \tag{6.44}$$

This leads to

$$F_{estimated}(x) = 1 - e^{-\hat{U}_{\min} x} \tag{6.45}$$

as the desired distribution. Once these resultant approximate exponential service time distributions are found for each of the fork/join nodes without synchronisation queues, we can use standard MVA or Convolution Algorithms to solve the overall queueing network in the same manner as described in Sections. 5.6 and 5.7.