Chapter 2
Queueing Network Models - Analytical Issues

Department of Statistics, University of Calicut, 2016
CHAPTER 2

QUEUING NETWORK MODELS - ANALYTICAL ISSUES

2.1 Introduction

This chapter provides a literature review on queueing networks and addresses some analytical issues of queueing networks. In the first section Jackson networks and some other types of networks are discussed. Product form networks and related issues are described. The concept of blocking and methods used to analyze various types of blocking are described. The concept of feedback flows and related literature in queueing networks are discussed. Capacity restriction in queueing networks are described. Specific methods- decomposition technique and expansion method are discussed. The concept of simulation and Matlab simulation extension SimEvents are introduced.
2.2 Jackson Queueing Networks

Jackson (1957,1963) made most important contributions to the development of queueing network models. Jackson model is considered to be the most researched and widely applied network model in different areas of research. Jackson find a product-form steady-state solution for in open and closed models.

Jackson queueing network is an arbitrary network of $M|M|c|\infty$ queueing nodes where customers arrive in Poisson stream from outside and transferred probabilistically from one node to another node until their eventual departure from the system.

Features of Jackson open network with K nodes:

- There is only one class of customers in the network.

- Customers arrive from outside at each queue $i$ according to a Poisson process with rate $\gamma_i \geq 0$.

- The service times are independent and exponentially distributed with mean $\mu_i$.

- A customer served at queue $i$ is routed to queue $j$ with a probability $p_{ij}$ or leave the network with probability

$$1 - \sum_{j=1}^{K} p_{ij}.$$

- By applying the flow balance conditions, the arrival rates of customers to each
individual queue is

\[ \lambda_j = \gamma_j + \sum_{i=1}^{K} p_{ij} \quad \text{for} \quad j = 1, 2, \ldots K. \]

According to Jackson’s theorem, the joint probability distribution for this type of network with \( M|M|c|\infty \) queues has a product form solution.

\[ P(n_1, n_2, \ldots n_K) = \prod_{j=1}^{K} P_j(n_j), \]

Note that the whole network is stable if \( \rho_j = \lambda_j/\mu_j < c_j \quad \text{for} \quad j = 1, 2, \ldots K, \) where \( c_j \) is the number of servers at queue \( j. \)

Important implications of Jackson network:

- Once the average flow rates are determined, the queues may be considered separately in isolation even when there is feedback present in the network.

- The states of individual queues behave as if they are independent of each other and hence the joint probability of the system is the product of the state probabilities of the individual queues.

- The flows entering the individual queues behave as if they are Poisson even though it is not Poisson due to the feedback paths.
Burke’s theorem

Burke (1956) suggested that if the input rate to the station is Poisson; service time at the station is exponential; and there is no restriction on exiting the station then output distribution from a station is identical to the input distribution. The internal arrival pattern in the Jackson model is actually related to this findings by Burke (1956). Burke’s theorem assures that the internal arrival rate follows a Poisson distribution as long as no feedback flows are allowed. Burke’s theorem enables us to do queue by queue-by-queue decomposition and analyze each queue separately when multiple server queues are connected together in a feed forward fashion without any feedback path.

Burke’s theorem provides a more general results for a departure process of a $M|M|c|\infty$ queue. This theorem states that the steady-state output of a stable $M|M|c|\infty$ with input parameter $\alpha$ and service-time parameter $\mu$ for each of the $c$ servers is in fact a Poisson process at the same rate $\alpha$. So the output does not depend on the other processes in the system.

Many authors expanded and modified the Jackson properties. Melamed (1979) studied Burke’s finding in an open Jackson system. He showed that departure rates from internal stations to outside the system are mutually independent if arriving rates to all internal stations follow the Poisson distribution and the sum of all departure rates from the network must also be Poisson. Disney (1981) examined the internal arrival rate distribution with feedback flow as a generalization of Jackson network
model. He showed that when a system has any kind of feedback flow, the internal flows in the system do not follow the Poisson distribution.

**Network performance measures**

After obtaining the actual flow rate to each queue in the network, Jackson’s theorem and Little’s theorem are used to obtain the state distribution of each queue in isolation and overall state distribution as the product of the individual distributions. Using these flows and the state distributions, the following performance measures are calculated.

- Total throughput of the network \((\gamma)\): This follows from the flow conservation principle, as the total number of customers entering the system must be equal to the total number of customers leaving the system, if it is stable.

  \[
  \gamma = \sum_{j=1}^{K} \gamma_j.
  \]

- The visit count to node \(j\), \((V_j)\): This is a measure of average number of times a customer visits a node/queue \(j\) once he enters the network.

  \[
  V_j = \frac{\lambda_j}{\gamma}.
  \]

- Average number of customers at node \(j\), \((N_j)\): Using the definition of the
state probability for each of the queues, $j = 1, 2, \cdots K$.

$$N_j = \sum_{k=0}^{\infty} kp_j(k)$$

- Average number of jobs in the network ($N$):- This is the sum of the average number of customers in each queue in the network.

$$N = \sum_{j=1}^{K} N_j.$$

- Mean sojourn time ($W$):- The average total time spent in the system by an arriving customer before he leaves the network. This can be calculated by either as the sum of the total times that the customer will spend in each of the $K$ queues in the network or by applying Little’s theorem.

$$W = \frac{N}{\gamma}.$$

**Closed Jackson’s networks**

In closed queueing networks, customers can neither enter nor leave the network, except this assumption all the other assumptions are as same as Jackson’s networks. These networks are also known as Gordon-Newell networks (see Gordon and Newell (1967a)). The system is started with a certain number of customers and these customers continually circulate in the network with a probabilistic routing
between the queues.

Consider a closed queueing network with $K$ nodes and assume that there are $M$ customers of the same class circulating in the network. Service times for a particular customer at a queue is assumed to be exponentially distributed and each node $j$ is FCFS queue. A customer departing from queue $i$ chooses queue $j$ next with probability $p_{ij}$ and the sum of the transition probabilities from queue $i$ to any other queue $j$, in the network including queue $i$ must add up to unity.

Let $\lambda_j$ be the total average arrival rate to queue $j$, the application of flow balance conditions would lead to the following $K$ equations

$$
\lambda_j = \sum_{i=1}^{K} \lambda_i p_{ij} \quad for \quad j = 1, 2, \cdots K.
$$

These equations are not independent and the solution is determined uniquely up to a constant factor. Assume that $\alpha(M)$ is an unknown scalar quantity and let $\hat{\lambda}_j, \ j = 1, 2, \cdots K$ be a particular solution. Then the true average arrival rates $\lambda_j(M), \ j = 1, 2, \cdots K$ are

$$
\lambda_j(M) = \alpha(M)\hat{\lambda}_j, \quad j = 1, 2, \cdots K.
$$

where $\alpha(M)$ and $\lambda_j(M)$ are functions of the population size $M$ though $\hat{\lambda}_j, j = 1, 2, \cdots K$ are independent of $M$.

Let $\mu_j(m)$, be the state dependent service rate at queue $j$ when queue $j$ is in
state \( m \). Using any particular solution of \( \hat{\lambda}_j \) we can then define the relative utilization of queue \( j \) with respect to the relative utilization of the reference queue as

\[
u_j(m) = \frac{\hat{\lambda}_j}{\mu_j(m)}, \quad j = 1, 2, \cdots K, \quad m = 1, 2, \cdots M.
\]

Let

\[
\hat{P}_j(n_j) = \begin{cases} 
1 & n_j = 0 \\
u_j(1)u_j(2)\cdots u_j(n_j) & n_j \geq 1
\end{cases}
\]

and the corresponding normalization constant \( G(M) \) for a population \( M \) is given by

\[
G(M) = \sum_{n_1+n_2+\cdots+n_K = M} \hat{P}_1(n_1)\hat{P}_2(n_2)\cdots\hat{P}_K(n_K).
\]

Then according to Jackson’s theorem for closed networks, for all states \( n_1, n_2, \cdots, n_K \) such that \( n_1+n_2+\cdots+n_K = M \), the probability of the state vector have the following product form solution expression.

\[
P(n_1, n_2, \cdots, n_K) = \frac{1}{G(M)} \prod_{i=1}^{K} \hat{P}_i(n_i).
\]

**Kelly’s networks**

This is another popular queueing network with different classes of customers. This type of networks has Poisson arrival process and fixed route in the network.
Customers served at each queue with an exponential service time distribution with infinite capacity.

**The BCMP queueing networks**

The BCMP queueing network is a multi class queueing network discussed in Baskett et al. (1975). In this type of networks different class of jobs, different queuing discipline and generally distributed service times are included. Routing through the network depend on the customer type and class of a customer can change its class while passing through the network. Usually this network involve four types of systems:

- **Type a**: system with multiple servers, the service times are exponentially distributed and different customer classes must be identical and the queue discipline is first-come-first-served;

- **Type b**: system with one server, different customer classes have different general service time distribution with a rational Laplace transform, the queue discipline is Processor Sharing;

- **Type c**: system with an infinite number of servers and the mean service time for job classes can be different, the service times of the customers of different classes must have a rational Laplace transform;
• Type d: system with one server, different customer classes have different general service time distribution with a rational Laplace transform, the service discipline is last-come-first-serve with preemptive.

**Fork-Join networks**

An N-dimensional fork-join queue is a queueing system operated by N parallel servers with synchronized arrival and departure streams (see Baccelli et al. (1989)). Each server is attended by a buffer of infinite capacity and individually operates according to the FCFS discipline. Customers arrive into the system in batches of size not larger than N. The customers arriving to a fork-join queue splits (at the fork point) into N independent tasks that are simultaneously assigned to N servers. Each customer requires a server. At each server tasks can belong to the different jobs. When a job completes execution, it will wait at the join queue until all its sibling tasks are done. A join queue merge several tasks into a single job. There are mainly two special cases of fork join systems available in literature. One is fission-fusion system in which all tasks are considered identical and a customer can leave the system, as soon as any N customers are finished. These customers do not necessarily have to belong to the same job. Other case is split-merge system with no processor queues (a new job is served when all the N tasks are finished) are special cases of basic fork-join systems. These type of systems have non-product form solutions but many techniques are available for solving such systems (see Bolch et al. (1998)).


2.3 Product Form Networks

Queueing networks having special structures and their solutions can be obtained without generating their underlying state space and are generally known as product-form networks or separable networks. According to Balsamo (2001), product form queueing networks have a simple closed form expression of the stationary state distribution that allow to define efficient algorithms to evaluate average performance measures. We discuss product form queueing networks and their properties.

A queueing network is said to have product form solution if it can be represented by the form:

$$P(n_1, n_2, \cdots n_K) = \prod_{i=1}^{K} P_i(n_i),$$

where $P(n_1, n_2, \cdots n_K)$ is the the joint distribution of states in the $K$ queues in the network and $P_i(n_i)$ is a function of individual queues $n_i$, $i = 1, 2, \cdots K$.

The idea of product form networks was first introduced by Jackson (1963). He introduced product form queueing network models for open exponential networks with several assumptions. Product form networks for closed exponential networks are presented by Gordon and Newell (1967a). He also introduce several assumptions on the model characteristics and provide a simple closed form expression of the stationary state distribution and some average performance indices. This class of network models was extended by several authors to include various useful characteristics to represent more complex system. The main such characteristics are different types of customers of the networks, queueing discipline, state-dependent service rate,
state-dependent routing between the service centers and some constraints on the population of subnetworks.

The main popular result concerning product form queueing networks was BCMP theorem given in Baskett et al. (1975). This paper defines the BCMP class of queueing networks with product form solution for open, closed or mixed queueing models with multiple classes of customers and various service disciplines and service time distributions. It states that the stationary state distribution is expressed as the product of the distributions of the single queues with appropriate parameters and a normalization constant. Arrival theorem for product form queueing networks is the another remarkable property. This theorem states that a customer arriving at a queue of an open network sees the stationary state distribution of the network, and a customer arriving at a queue of a closed network sees the stationary state distribution with one less customer of the network.

The arrival theorem for product form queueing networks with infinite capacity was studied by Lavenberg and Reiser (1980) and Sevcik and Mitrani (1981). So open or closed queueing networks with FCFS queues with exponential service times, LCFS preemptive resume queues (if the LCFS queue discipline is used, then the customer currently being served is interrupted and preempted if there is a customer in the queue with a higher priority) with Coxian distributed service times, process sharing (PS) and infinite server queues with Coxian distributed service times have a product form solution. Complex queueing networks with finite capacities, subnetwork with population constraints and blocking have product form solution in some special
cases. Actually the product-form solutions have been derived only under special constraints, for different blocking types. Generally the queueing networks with blocking do not have a product form solution. So some approximate analytical techniques and simulation are used for solving the non product form networks. Several authors have studied these type of networks such as Gordon and Newell (1967b), Lam (1977), Towsley (1980), Akyildiz (1987), van Dijk (1993), Nelson (1993), Balsamo and de Nitto (1994), Balsamo and Cló (1998).

2.4 Blocking

Consider a network of queues with some or all have limited capacities. That is the total number of customers receiving or waiting service are limited. So the new arrival finds the queue is full it cannot enter the queue. In this case the arriving customer is "blocked" and cannot move to the destination queue as required. This can be handled depending on the blocking mechanism is being adopted. A blocking mechanism is considered as the description of customer’s behaviour. According to the behaviour of blocking different blocking mechanisms are considered in the literature are given below:

Blocking After Service (BAS)

Consider a queueing network when a customer upon completion of its service at a source node attempts to enter a destination node that is full, he is forced to wait at the source node blocking the server there, until a space become available
at the destination node. Server of the source node is forced to stop processing the customers that might be waiting on its queue until the blocked customer can join its destination queue. Therefore, in this blocking mechanism, the server at the source node becomes unblocked only when a departure occurs from the destination node or the number of customers drops below its maximum capacity. It is possible that the destination queue of customers receiving service in two or more different queues is the same (arbitrary topology). If these customers complete their service and they find that the destination queue full, more than one queue is blocked by the same queue. In this case First-Blocked-First-Unblocked rule is used to handle the situation, that is the customer that was blocked first will be the first one joins the blocking queue when a departure occurs.

Deadlock conditions are possible in this type of blocking such as all the stations in any directed cycle are full at the same time and a blocked customer is wants to go to the next station in the cycle. This situation may be handled by including some kinds of deadlock strategy into the model. A usual assumption made for removing the deadlock immediately is that, when a deadlock occurs it is immediately resolved by simultaneously moving all the blocked customers to their respective destinations. The another way is to restrict the system to situations where deadlocks are impossible.

BAS blocking mechanism has been used mainly to model systems such as production systems and disk I/O configurations. This type of blocking mechanism has also known as Type-I, classical, manufacturing, production and transfer blocking in the literature.
Blocking Before Service (BBS)

In this type of blocking mechanism, a customer declares its destination node, say $j$, before it starts receiving service at the source node $i$. The destination of the customer does not change until it completes its service and departs from the queue. If upon entering service at node $i$, a customer finds that destination node $j$ is full, then the service at the source node $i$ does not start and server $i$ is blocked. The other possibility is that at the time a customer enters for service, there was a space available at node the destination node $j$ but becomes full prior to service completion at node $i$. When the destination node becomes full, the server is blocked. The service at node $i$ is unblocked as soon as a departure occurs from node $j$. It is generally assumed that a new service starts when a node becomes unblocked and the amount of service received has lost until it is blocked.

According to the server space utilization of blocking time BBS blocking is further divided into two categories Blocking Before Service Server Occupied (BBS-SO) and Blocking Before Service Server Not Occupied (BBS-SNO). In BBS-SO the service space is used and in BBS-SNO the service space cannot be used to hold a customer during the time the server is blocked. In the case of queuing network with arbitrary topologies BBS-SNO is not defined and is applicable only in network topologies in which a node with a finite capacity has only one upstream node with a finite capacity.

BBS blocking mechanism has been used to model systems such as production, telecommunication, and computer systems. It has also referred to as Type-II,
immediate and service blocking in the literature.

**Repetitive Service (RS)**

A customer completes its service at source node $i$ and attempts to enter destination node $j$. If node $j$ at that time is full, the customer at node $i$ immediately receives a new and independent service according to its service discipline. This type of blocking is called Repetitive service blocking. This can be further divided into two types namely repetitive service with fixed destination or repetitive service with random destination. In the first case, the customer attempts to enter the same destination node after each repeated service completion until it departs from the queue. In the latter case, each time the customer completes the service, a new destination node is chosen independent of the previous choice.

This RS blocking mechanism is used to model flexible manufacturing systems and telecommunications systems. It has also referred to as Type-III, rejection, retransmission and repeat blocking in the literature.

**Rejection Blocking**

If the customer is forced to leave the system as soon as its destination node is full, this type of blocking is called rejection blocking. This type of blocking models loss systems and can be used in open networks of finite capacity queues. Deadlock does not arise for this type of blocking as a blocked customer is always made to leave the system altogether.

This model has been used to model computer systems and packet-switching net-
works. It is also referred to as stop and delay blocking. This also referred to as stop blocking and delay blocking in the literature. For more details on various aspects of blocking one may refer to Balsamo et al. (2001) and Bose (2002).

2.5 Feedback

In a network, a customer finishing service at one queue may return to the same queue later for another round of service is considered as a feedback network or cyclic network. In the case of healthcare systems, feedback represents readmission of patients to the same unit or department they have already visited. So there may be two types of patient feedbacks possible namely immediate and delayed (after a certain period of time).

The concept of feedback was first introduced by Finch (1959) through his paper cyclic queues with feedback. He considered two types of feedback in a network of m servers in series. In the first type, he has assumed that once a customer completes service at a server, it is possible to feedback to an earlier server. Finch called such type of feedback as terminal feedback. In the second type, he referred that a customer completes service at a server it is possible to return to the same server with a certain probability. This type of feedback is called single service feedback. In each case new customers are assumed to arrive randomly to the first server, and service time was assumed according to the negative exponential distribution. A constraint is also imposed on the upper limit on the number of customers in the system. He also made
the assumption that the probability of feeding back does not depend on the state of
the system at the time of return and of the customer who just completed service. So
at service completion, there was a non-zero probability of leaving the system. Finch
obtained the joint probability distribution of the number of customers at each service
stage under equilibrium conditions.

Takacs (1963) considered a system of single server queue with feedback. He
assumed Poisson arrivals and a general service time distribution with service times
were assumed to be mutually independent and identically distributed. After service
completion, a customer can immediately return with a certain probability to the
server for another round of service, or the customer may depart from the system.
He also assumed that the probability of feeding back was independent of any other
event. Takacs derived the stationary distribution of queue size, and the station-
ary distribution of total time spent by a customer in the system. State dependent
Feedback flows was first considered by Davignon and Disney (1976). Foley and Dis-
ney (1983) described delayed feedback considering a network with two nodes. The
authors derived the properties of the time-dependent queue length process.

2.6 Capacity Restriction (Blocking with Feedback)

Queueing network with different blocking mechanisms are extensively studied in lit-
erature see Balsamo et al. (2001) and Perros (1994). According to Disney (1981),
the poisson arrival theoretically incorporate feedback. Next reason is the that FCFS
queue discipline could be violated while trying to resolve the deadlocks due to feedback flows. There are many studies in open queueing network models with arbitrary configurations such as Altiok and Perros (1987), Jun and Perros (1989), Lee et al. (1998), Takahashi et al. (1980), but only Perros (1981a,1981b), Jun and Perros (1989) and Lee et al. (1998) considered queueing networks with feedback flows.

Perros (1981a,1981b) studied exponential open queueing network model with blocking and feedback by independent first-level servers in tandem with a second-level server. The flow of customers are blocked through a first-level server occurs each time the server completes a service. The server remains blocked until its blocking unit completes its service at the second-level server. An approximate expression of the probability distribution of the number of blocked first-level servers conditioned upon a service completion of a first-level server is obtained. This expression is derived assuming a processor-sharing type of service.

Jun and Perros (1989) are considered blocking and feedback flows by assuming simultaneous exchange of blocked customers and so the deadlocks are resolved instantaneously. Arrival distribution were assumed to be Poisson and the service time was followed by a two-phase Coxian distribution. The algorithm says that, in order for a two-phase Coxian distribution to reflect all the possible deadlocks and delays due to blocking, a very complicated phase-type distribution should be constructed first. Then, using a three-moment approximation, the phase-type distribution is simplified to the two-phase Coxian distribution. Although accurate, the algorithm is restricted to networks consisting of nodes with no more than two directly-linked
upstream servers.

Lee et al. (1998) extended the previous work of Lee and Pollock (1990) to account for feedback flows, and also the extended the symmetrical approach provided by Dallery and Frien’s (1993) to open queueing networks with arbitrary configurations. Authors assumed that deadlocks are resolved instantaneously by simultaneously transferring all the blocked customers. Using the decomposition technique, the authors decomposed the network into a set of subsystems, and assumed that service times and inter-arrival times are characterized by a generalized exponential distribution. Each subsystem was expressed by one or many upstream servers and one downstream server, separated by a finite buffer. The authors considered the service rate as the unknown parameter. The proposed algorithm provides accurate results with less execution time, even when deadlocks exits, as long as they are not too frequent.

2.7 Decomposition Technique

The most commonly used approximation technique for analyzing queueing networks with blocking is the decomposition technique. In this approach the whole network is decomposed into a set of subsystems. The decomposition technique consists of mainly three steps: (i) characterization of subsystems, (ii) derivation of a set of equations to identify the unknown parameters in each subsystem, and (iii) building an algorithm for solving the sets of equations and determining the unknown parameters of each
subsystem.

According to Dallery and Frein (1993) this technique provides an approximation to the original system. Generally, a system of \((K+1)\) servers is decomposed into \(K\) subsystems. There are two ways to represent each subsystem. To represent the subsystem with a finite queue and a server being fed by an external arrival process is one way. Another way is to represent the subsystem by upstream and downstream servers separated by a finite capacity buffer. The upstream server is assumed to be never starved. In the second way for characterizing subsystems, the service time distribution of each server is represented by an exponential distribution or a phase-type distribution.

Takahashi et al. (1980) suggested an approximation method for the analysis of open restricted queueing networks. Assuming Poisson arrivals and exponential service times, the authors proposed pseudo-arrival rates and effective service rates for the approximation technique. Where the pseudo arrival rate refers to the rate of actual arrivals for non-blocked time intervals only. That is the pseudo arrival rate at node \(i\) is the ratio of the rate of actual arrivals to node \(i\) from other nodes over all time space, including blocked intervals to the probability of no blocking at node \(i\). Effective service time is a combination of treatment time and blocked time. The proposed approximation technique was applied to different networks, and the results were compared with simulation and exact calculations. Finally authors showed that the suggested technique gives a good approximation to system performance measures such as the blocking probability.
An algorithm for approximately analyzing open exponential queueing networks with blocking has been proposed by Altiok and Perros (1987). The authors did not consider deadlocks in the queueing network. The suggested algorithm decomposes the network into several $M/PH/1/K$ queues (where service time follows phase-type ($PH$) distribution) and each of these queues are then analyzed independently. The approximation algorithm gives results in the form of the marginal probability distribution of number of units in each queue. The proposed algorithm was applied in a three-node and four-node queueing networks and compared with exact numerical data. Comparison showed that results based on the algorithms had an acceptable level of error.

Then Perros and Snyder (1989) suggested a computationally more efficient version of the algorithm. Here an open queueing network with transfer blocking and a feed-forward configuration is considered. Arrivals were assumed to be Poisson and service times were assumed to be exponentially distributed. Single class of customers was considered. Customers were served according to first come first serve queue discipline. Authors considered the case of external arrivals occur to one particular queue only. They used a two-phase Coxian distribution with simple structure, which can be easily applied to large networks. When compared with the algorithm proposed by Altiok and Perros (1987), this algorithm was found to have comparable accuracy.

Lee and Pollock (1990) studied similar queueing network as in Perros and Snyder (1989). But the external arrivals here could occur at any server. The authors restricted the analysis to acyclic networks. Using the information from its nearest
queue, the proposed algorithm analyzes each queue separately. So this algorithm provides marginal steady-state occupancy probability for each queue. Clearance time and effective inter-arrival rate were considered in the algorithm. In the algorithm, it is assumed that the arrivals be Poisson, clearance time exponentially distributed and also of no arrivals to blocked queues. The authors claimed that the proposed algorithm can be used to solve large networks with general topology and yields accurate results whether with finite or infinite buffer external arrivals and high or low service rates.

2.8 Expansion method

We know that the majority of the real life queueing networks are with finite capacity queues. The expansion method is a technique used for open queueing networks where each node has finite waiting room space and also arrival to each node is Poisson and service rate is exponentially distributed. As similar to a product form solution approach, this method transforms the queueing network into an equivalent Jackson network, which can be decomposed so that each node can be solved independently of each other. This method is considered an effective and robust approximation technique to measure the performance of open finite queueing networks. This technique was proposed by Kerbache and Smith (1987,1988) and further extended to general queueing networks. Kerbache and Smith (2000) used this method in their paper multi-objective routing within large scale facilities using open finite queueing
networks. Then Jain and Smith (1994), Smith (2003), Andriansyah et al. (2010), Dallery and Gershwin (1992), van Woensel and Vandaele (2006,2007) recently Cruz and van Woensel (2014) and many authors used this technique.

The method is a combination of two approximation techniques, namely, the repeated trials and the node-by-node decomposition. In order to evaluate the performance measures of a queueing network, the method first divides the network into single nodes with revised service and arrival parameters. Blocked customers are routed into an artificial node namely holding node and are repeatedly sent to this node until they are served. The addition of this holding node expands the network and transforms the network into an equivalent Jackson network in which each node can be solved independently (see Cruz and van Woensel (2014)).

In this method each finite node could be treated as one of the two phases of operation: Phase (A): Not Saturated - where node with finite capacity is empty or it has at most \((K - 1)\) customers in the system including those in service by assuming finite capacity of queue is \(K\) and Phase (B): Saturated - where no more customers can join the queue as queue is full. There are basically three stages in this method, namely,

- Stage 1: Network reconfiguration.
- Stage 2: Parameter estimation.
- Stage 3: Feedback elimination.
Notations and steps are described as follows:

Notations used

Following are the notations used in this paper:

\( h \) = The holding node established in the expansion method,

\( \alpha \) = External Poisson arrival rate to the network,

\( \lambda_j \) = Poisson arrival rate to node \( j \),

\( \tilde{\lambda}_j \) = Effective arrival rate to node \( j \),

\( \mu_j \) = Exponential mean service rate to node \( j \),

\( \tilde{\mu}_j \) = Effective service rate at node \( j \) due to blocking,

\( p_K \) = Blocking probability of finite queue of size \( K \),

\( p'_K \) = Feedback blocking probability,

\( L_j \) = Average number of customers in node \( j \),

\( W_j \) = Average time a customer spends at node \( j \),

\( X_j \) = Rate at which customers leave the node \( j \),

\( U \) = Probability that the server is busy.

Stage 1 : Network reconfiguration

Add an artificial node ‘\( h \)’ in the network for each finite capacity node. If node two is in phase A, then customers proceed to its queue with probability \( (1 - p_K) \). If node is in phase B, then customers are blocked with probability \( p_K \) and will have to incur delay before it can join the queue at node two. Note that we are not loosing the customers here, instead customers wait for queue of node two to be
empty to accommodate them. They wait in artificial node. After incurring a delay at the artificial node, customer proceeds to the capacitated node from which it was previously rejected. To represent this process, the added node has a feedback arc to account for these attempts. If queue is still full, it incurs another delay. This holding node is modeled as an $M|M|\infty$ queue. The infinity number of servers simply implies an overflow customer is serviced a delay time without queueing.

**Stage 2 : Parameter estimation**

In this stage we determine $p_K$, $p'_K$, and $\mu_h$ as follows:

First determine $p_K$ using the known analytical result of $M|M|1|K$ queue by

$$p_K = P(K \text{ customers in queue or service}) = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}}.$$ 

where

$$\rho = \frac{\text{Arrival rate}}{\text{Service rate}}.$$ 

After a service completion at the holding node, with probability $p'_K$ a customer is
forced to return to the holding node for another immediate service time (delay). $p'_K$ cannot be calculated exactly, we use an approximate formula,

$$p'_K = \left[ \frac{\mu_j + \mu_h}{\mu_h} - \frac{\lambda (r_2^K - r_1^K) - (r_2^{K-1} - r_1^{K-1})}{\mu_h (r_2^{K+1} - r_1^{K+1}) - (r_2^K - r_1^K)} \right]^{-1}$$

where $r_1$ and $r_2$ are roots to

$$\lambda - (\lambda + \mu_h + \mu_j) x + \mu_h x^2 = 0$$

while

$$\lambda = \lambda_j - \lambda_h (p'_K)$$

and $\lambda_j$ and $\lambda_h$ are the actual arrival rates to the finite and artificial nodes respectively.

$$\mu_h = \frac{2\mu_j}{1 + \rho^2 \mu_j^2}$$

where $\rho^2$ is the service time variance.

**Stage 3 : Feedback elimination**

Feedback creates strong dependencies in the arrival processes. So we have to recompute the service time at the artificial node and get rid of the feedback arc. The new service rate is

$$\mu'_h = (1 - p'_K)\mu_h$$
The mean service time at node $i$ preceding a finite node is

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + p_K \mu_h^{-1}.$$

In the same manner, we can establish similar equations with regard to each finite queue node. Finally we have simultaneous equations in variables $p_K, \tilde{\mu}_K, \mu_h^{-1}$. Solving these equations, all performance measures of the network can be obtained. The approximate blocking probability derived from exact formula for the $M|M|1|K$ case is

$$p_K = P(K \text{ customers in queue or service}) = \frac{\left(1 - \frac{\tilde{\lambda}_i}{\mu_h}\right) \left(\frac{\tilde{\lambda}_i}{\mu_h}\right)^K}{1 - \left(\frac{\tilde{\lambda}_i}{\mu_h}\right)^{K+1}}.$$

### 2.9 Simulation

Simulations often give a convenient tool to study complex systems, which cannot be accurately modelled for exact or approximate mathematical analysis. A system can be simulated very closely to the reality. Compared to analytical methods sensitivity analysis of the various system performance parameters is easier to study using simulations. Simulation shows the relationship between the real system and its model. So the basic idea behind developing a simulation model is to study the system and understand its behaviour. Some times, simulations are carried out to see the logically inconsistent behaviour in the original system. Simulations of queueing networks may be used to find the different kinds of deadlock conditions that may arise in the
real system and suggest possible ways of handling them. Bose (2002) has effectively used simulations to understand the behaviour of several complex queueing network systems. A change in the system can happen only at discrete time instants and the simulator may effectively use this to simulate events and handle them only at discrete time instants and in this case it is known as discrete event simulation.

The simulation technique consists of basically three steps: (a) construct a model, (b) describe the model, and (c) execute the model. In the first step the structure of the model is to be designed. The second step describes the characteristics of the inherent activities in the system, such as arrival rates, service times and the associated distributions. In the last step, for executing the simulation, we have to specify the number or length or running time.

To study the behaviour of any attribute of any entity in the system one may do it by monitoring those during the simulation run. But a good simulation model typically takes a long time to construct and may also take a long time to run on a computer. The major disadvantages of taking a simulation rather than an analytical approach are complexity of simulator construction and computational expenditure.

2.10 SimEvents

Simulink is a graphical extension to MATLAB for modelling and simulation of systems. Using Simulink, systems are drawn on screen as block diagrams. SimEvents gives a discrete-event simulation engine and component library for Simulink. We can
model event-driven communication between components to analyze and optimize end-to-end latencies, throughput, packet loss, and other performance characteristics. ‘SimEvents’ provides libraries of predefined blocks, such as queues, servers, and switches to represent systems and customize routing, processing delays, prioritization and other operations.

Using SimEvents we can design distributed control systems, hardware architectures, and sensor and communication networks for aerospace, automotive, and electronics applications. We can also simulate event-driven processes, such as the execution of a mission plan or the stages of a manufacturing process, to determine resource requirements and identify bottlenecks. The following are key features of SimEvents.

(a) Discrete-event simulation engine for multi domain modelling of complex systems in Simulink.

(b) Predefined block libraries, including queues, servers, generators, routing, and entity combiner/splitter blocks.

(c) Entities with custom data attributes for flexible representation of packets, tasks and parts.

(d) Built-in statistics aggregation for obtaining delay, throughput, average queue length, and other metrics.

(e) Library blocks for defining domain-specific constructs, such as communication channels, messaging protocols, and conveyor belts.
(f) Model animation for visualizing model operation and debugging discrete-event simulation in Simulink models.

In Simulink modelling, we construct a discrete-event system by adding a variety of blocks, such as generators, queues, and servers, from the SimEvents block library. These blocks are used for producing and processing entities. Packets within a communication network, planes on a runway, or trains within a signalling system are the examples of entities. Asynchronous events corresponding to motion and changes in entity attributes through updating states of the underlying system. The states are queue length or service time for an entity in a server (for more details see Mathlab documentation, SimEvents, Getting Started Guide (2015)).

2.11 Summary and Conclusions

This chapter reviewed Jackson queueing networks, product form networks and related references. The concept of blocking and feedback flows and related and literature reviews are discussed. The decomposition techniques and expansion method are described. The concept of simulation and the Matlab simulation extension SimEvents are presented. Following chapters we will make use of these concepts and methods to deduce the performance measures the system.

We know that the solution of complex queueing networks are often analytically intractable. So approximation techniques and simulation techniques are necessary. In this thesis we are dealing with both restricted and unrestricted queueing networks.
In two chapters we are dealing with complex networks which do not satisfy product form solution. Then for the analysis of such networks simulation and approximation methods are used. So these concepts are inevitable in this thesis.