Chapter 10
Queueing Networks

10.1 Introduction of Queueing Networks

Up to now, we have overviewed the main methods for the analysis of individual queueing systems. But the analysis of large telecommunication systems or computer systems executing complex interrelated tasks (e.g., transaction processing systems, Web server farms) requires the application of systems models that contain several servers (potentially of different kinds) where customers are traveling among these servers for consecutive services.

Queueing network models are commonly used for the analysis of these kinds of systems. A queueing network is a graph with directed arcs whose nodes represent the kinds of queueing systems that we have studied till now. The arcs of the graph describe the potential transitions of customers among these queueing systems.

It is a commonly applied modeling assumption in queueing networks that the transition of a customer from one node to the next is memoryless and independent of the network state, i.e., it is independent of the past history of the network, the current number of customers at the network nodes, and the status of the servers. After being served at a network node a customer chooses the next node according to the weight (probability) associated with the outgoing arcs of the given node.

There are two main classes of queueing networks: open and closed queueing networks. In closed queueing networks, a fixed number of customers circulate in the network, and there is no arrival/departure from/to the environment. In open queueing networks customers arrive from the environment, obtain a finite number of services at the network nodes (nodes are potentially visited more than once), and leave the network eventually.

Queueing networks are classified also based on the structure of the directed arcs. Queueing networks without a loop (series of directed arcs forming a loop) are referred to as acyclic or feedforward queueing networks, and those with a loop are referred to as cyclic or feedback queueing networks. Acyclic networks are meaningful only in the case of open queueing networks. The nodes of acyclic

L. Lakatos et al., *Introduction to Queueing Systems with Telecommunication Applications*,
DOI 10.1007/978-1-4614-5317-8_10, © Springer Science+Business Media, LLC 2013

networks can be numbered such that arcs are always directed from a node with a lower index to a node with a higher index or to the environment. Henceforth we assume that the nodes of acyclic networks are numbered in this way.

10.2 Burke's Theorem

It is possible to analyze a class of open acyclic queueing networks based on the following theorem.

Theorem 10.1 ([17]). *The customer departure process of a stable $M/M/m$ queue is a Poisson process with the same rate as the arrival process of the queue.*

Proof. The number of customers in an $M/M/m$ queue is a *reversible* Markov chain (Sect. 3.3.6). The time reverse of the process is stochastically identical (according to all finite-dimensional joint probabilities) with the original process. In this way the departure instances of the original process (which are the arrival instants of the reverse process) are stochastically identical with the arrival instants of the original process (which are the departure instants of the reverse process) which is a Poisson process. □

An important consequence of the theorem is that in equilibrium the time till the next departure is exponentially distributed, i.e., memoryless.

Let $D^*(s)$ be the Laplace transform of the time till the next departure, $A^*(s)$ the Laplace transform of the interarrival time distribution, $B^*(s)$ the Laplace transform of the service time distribution, and p the probability that in equilibrium the queue will be idle; then

$$D^*(s) = p\ B^*(s) + (1-p)\ A^*(s)\ B^*(s).$$

Using that $B^*(s) = \frac{\mu}{s+\mu}$, $A^*(s) = \frac{\lambda}{s+\lambda}$, $p = \frac{\lambda}{\mu}$, we have

$$D^*(s) = \frac{\lambda}{\mu}\ \frac{\mu}{s+\mu} + \frac{\mu-\lambda}{\mu}\ \frac{\lambda}{s+\lambda}\ \frac{\mu}{s+\mu},$$

and after some algebra

$$D^*(s) = \frac{\mu}{s+\mu}\ \frac{s\lambda + \lambda^2 + \mu\lambda - \lambda^2}{\mu(s+\lambda)} = \frac{\lambda}{s+\lambda}.$$

This expression indicates that we often have exponentially distributed interarrival, interdeparture times in Markovian queueing networks.

10.3 Tandem Network of Two Queues

The simplest queueing network is the open tandem network (Fig. 10.1) composed of two $M/M/1$ queues in which customers arriving from the environment get in queue 1 and after being served in queue 1 get in queue 2, from where, after being served, they depart to the environment. Let the arrival rate from the environment to queue 1 be λ and the service rate at queue 1 and 2 be μ_1 and μ_2, respectively.

From Burke's theorem we have that the arrival intensity to both queues is λ, and in this way the condition of stability is

$$\frac{\lambda}{\mu_1} < 1 \qquad \frac{\lambda}{\mu_2} < 1$$

that is

$$\lambda < \min(\mu_1, \mu_2).$$

Let us consider a Markov chain describing the number of customers in both queues. We identify the states of this Markov chain by a vector of the number of customers in the first queue and the second queue. That is, state $\{i, j\}$ refers to the state where there are i customers in the first and j customers in the second queue. The transition rates of this Markov chain are as follows:

$$\begin{aligned} &\{i, j\} \to \{i+1, j\} &&: \lambda, \\ &\{i, j\} \to \{i-1, j+1\} &&: \mu_1 \text{ when } i \geq 1, \\ &\{i, j\} \to \{i, j-1\} &&: \mu_2 \text{ when } j \geq 1. \end{aligned}$$

We denote the stationary probability of state $\{i, j\}$ by $p_{i,j}$. The balance equations of the Markov chains are

$$\begin{cases} \lambda p_{0,0} & = \mu_2 p_{0,1}, \\ (\lambda + \mu_2) p_{0,j} & = \mu_1 p_{1,j-1} + \mu_2 p_{0,j+1} & \text{when } j \geq 1, \\ (\lambda + \mu_1) p_{i,0} & = \lambda p_{i-1,0} + \mu_2 p_{i,1} & \text{when } i \geq 1, \\ (\lambda + \mu_1 + \mu_2) p_{i,j} & = \lambda p_{i-1,j} + \mu_1 p_{i+1,j-1} + \mu_2 p_{i,j+1} & \text{when } i, j \geq 1. \end{cases}$$

According to Burke's theorem, in equilibrium the arrival process of queue 2 is a Poisson process with rate λ. Using this fact the stationary state probabilities are

$$p_{i,j} = p_i^{(1)} p_j^{(2)} = \left(1 - \frac{\lambda}{\mu_1}\right) \left(\frac{\lambda}{\mu_1}\right)^i \left(1 - \frac{\lambda}{\mu_2}\right) \left(\frac{\lambda}{\mu_2}\right)^j ,$$

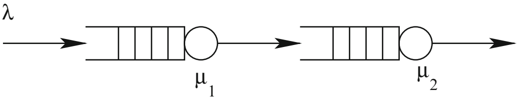


Fig. 10.1 Tandem network of two nodes

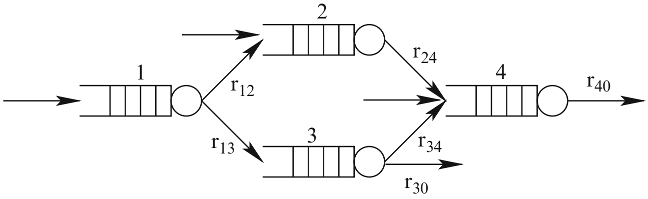


Fig. 10.2 Acyclic queueing network

where $p_i^{(1)}$ and $p_j^{(2)}$ are the stationary distributions of the corresponding $M/M/1$ queues.

Stationary solutions of this kind are referred to as *product-form solution* because the joint distribution is the product of two marginal distributions. It is important to note that despite the product-form stationary distribution the number of customers in the two queues is not independent. There is a very strong correlation between those processes, namely, a departure from the first queue results in an arrival at the second queue.

Based on the stationary distribution we can easily determine the important performance indices. For example, the mean number of customers in the system, the mean time spent in the network, and the mean waiting time spent in the network are

$$\mathbf{E}(N) = \sum_i \sum_j (i+j) p_{i,j} = \sum_i i p_i^{(1)} + \sum_j j p_j^{(2)} = \frac{\frac{\lambda}{\mu_1}}{1-\frac{\lambda}{\mu_1}} + \frac{\frac{\lambda}{\mu_2}}{1-\frac{\lambda}{\mu_2}},$$

$$\mathbf{E}(T) = \frac{\mathbf{E}(N)}{\lambda} = \frac{\frac{1}{\mu_1}}{1-\frac{\lambda}{\mu_1}} + \frac{\frac{1}{\mu_2}}{1-\frac{\lambda}{\mu_2}} = \frac{1}{\mu_1-\lambda} + \frac{1}{\mu_2-\lambda},$$

$$\mathbf{E}(W) = \mathbf{E}(T) - \frac{1}{\mu_1} - \frac{1}{\mu_2},$$

where we used Little’s law to obtain the last two quantities.

10.4 Acyclic Queueing Networks

Acyclic queueing networks (Fig. 10.2) are queueing networks in which the outgoing arcs of the nodes are directed toward nodes with a higher index or to the environment. Consequently, in such queueing networks a customer visits each node at most once.

Based on Burke’s theorem and the results on the superposition and filtering of independent Poisson processes [Property (h) of Poisson processes in Sect. 2.7.3], we can apply the same approach as the one applied for the analysis of the tandem queueing network. That is, we can (explicitly) compute the arrival rate to each node of the network, and we can assume that the arrival process at the given node is a Poisson process with that arrival rate. Based on this assumption, the product-form solution remains valid, that is,

$$p_{k_1,k_2,\cdots,k_N} = \prod_{i=1}^{N} p_{k_i}^{(i)},$$

where $p_{k_i}^{(i)}$ is the stationary probability of the k_i state of an M/M/1 queue with a Poisson arrival process with the parameter λ_i and exponentially distributed service time with the parameter μ_i, which is

$$p_{k_i}^{(i)} = \left(1 - \frac{\lambda_i}{\mu_i}\right)\left(\frac{\lambda_i}{\mu_i}\right)^{k_i}.$$

10.5 Open, Jackson-Type Queueing Networks

In the previous subsections we discussed acyclic queueing networks and, based on Burke's theorem, we assumed that the arrival processes of the queues were independent Poisson processes. Based on this assumption we obtained product-form solutions. From now on we consider cyclic queueing networks and consequently we can no longer apply Burke's theorem due to the dependencies on the arrival processes of customers at a queue.

The main results of this kind of queueing networks were published by Jackson [44] in 1963. Since then, these kinds of networks have often been referred to as Jackson-type networks (Fig. 10.3). Jackson considered the following queueing network model:

- The network is composed of N nodes.
- There are m_i servers at node *i*.

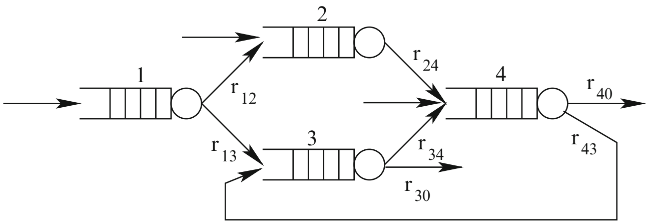


Fig. 10.3 Jackson-type queueing network

- The service time distribution at node i is exponentially distributed with the parameter μ_i.
- From the environment customers arrive at node i according to a Poisson process at rate γ_i.
- A customer getting served at node i goes to node j with probability $r_{i,j}$ $(i, j = 1, 2, \cdots, N)$, and the probability that the customer departs from the network is

$$r_{i,0} = 1 - \sum_{k=1}^{N} r_{i,k} \qquad i, j = 1, 2, \cdots, N.$$

Stability Condition of Jackson-Type Queueing Networks

The following *traffic equations* define the traffic rate at the nodes of the network:

$$\lambda_i = \gamma_i + \sum_{j=1}^{N} \lambda_j \; r_{j,i} \quad i = 1, 2, \cdots, N. \tag{10.1}$$

The left-hand side of the equation represents the aggregate traffic intensity arriving at node i. Due to the stability of the network nodes, the arriving traffic intensity is identical with the departing traffic intensity from node i. The right-hand side of the equation gives the traffic components arriving at node i. γ_i is the traffic component arriving from the environment, and $\lambda_j \; r_{j,i}$ is the traffic component that departs from node j and goes to node i.

Introducing the row vector $\lambda = \{\lambda_i\}$ and $\gamma = \{\gamma_i\}$ and matrix $\boldsymbol{R} = \{r_{ij}\}$ the traffic equation can be written in the following vector form:

$$\lambda = \gamma + \lambda \boldsymbol{R},$$

whence

$$\lambda = \gamma(\boldsymbol{I} - \boldsymbol{R})^{-1}$$

if $(\boldsymbol{I} - \boldsymbol{R})$ is nonsingular.

The elements of the matrix $(\boldsymbol{I} - \boldsymbol{R})^{-1}$ have a well-defined physical interpretation according to the following theorem. Let L_{ij} denote the number of visits to node j (before departing to the environment) by a customer arriving at node i:

Theorem 10.2.

$$\left[(\boldsymbol{I} - \boldsymbol{R})^{-1}\right]_{i,j} = \mathbf{E}\left(L_{i,j}\right),$$

where the left-hand side denotes the i, j *element of the matrix* $(\boldsymbol{I} - \boldsymbol{R})^{-1}$.

Proof. The number of visits to node j satisfies the following equation:

$$\mathbf{E}\left(L_{i,j}\right) = \delta_{i,j} + \sum_{k=1}^{N} r_{i,k} \mathbf{E}\left(L_{k,j}\right),$$

where $\delta_{i,j}$ is the Kronecker delta, that is, $\delta_{i,j} = 1$ if $i = j$, 0 otherwise. Introducing matrix $\boldsymbol{L}$ whose i, j element is $\mathbf{E}\left(L_{i,j}\right)$ we can rewrite the preceding equation in matrix form:

$$\boldsymbol{L} = \boldsymbol{I} + \boldsymbol{R}\boldsymbol{L},$$

from which the theorem comes. □

The theorem gives a condition for the nonsingularity of the matrix $(\boldsymbol{I} - \boldsymbol{R})$. $(\boldsymbol{I} - \boldsymbol{R})$ is nonsingular if all customers leave the queueing network after a finite number of visits to the nodes of the network.

A queueing network is said to be stable if all queues are stable, which holds when

$$\lambda_i < m_i \mu_i, \qquad i = 1, 2, \cdots, N.$$

Stationary Distribution of Jackson-Type Queueing Networks

According to the properties of Jackson-type queueing networks, the number of customers at the nodes of the network is a continuous-time Markov chain. Let k_i denote the number of customers at node i, and let us introduce the following notations:

$$\begin{aligned}
\mathbf{N} &= (k_1, \cdots, k_i, \cdots, k_j, \cdots, k_N),\\
\mathbf{N}_{i,0} &= (k_1, \cdots, k_i + 1, \cdots, k_j, \cdots, k_N),\\
\mathbf{N}_{0,j} &= (k_1, \cdots, k_i, \cdots, k_j - 1, \cdots, k_N),\\
\mathbf{N}_{i,j} &= (k_1, \cdots, k_i + 1, \cdots, k_j - 1, \cdots, k_N),
\end{aligned}$$

where in the last two cases $k_j \geq 1$. Using these notations we can describe the possible transitions of Markov chains representing the number of customers at the network nodes.

- $\mathbf{N}_{0,j} \to \mathbf{N}$: a new customer arrives at node j from the environment, increasing the number of customers at node j from $k_j - 1$ to k_j. This happens at rate γ_j.
- $\mathbf{N}_{i,0} \to \mathbf{N}$: a customer departs to the environment from node j, decreasing the number of customers at node j from $k_j + 1$ to k_j. This happens at rate $r_{i,0}\alpha_i(k_i + 1)\mu_i$.
- $\mathbf{N}_{i,j} \to \mathbf{N}$: a customer gets served at node i and goes to node j. This transition decreases the number of customers at node i from $k_i + 1$ to i_j and increases the number of customers at node j from $k_j - 1$ to k_j. This happens at rate $r_{i,j}\alpha_i(k_i + 1)\mu_i$.

In the preceding expressions $\alpha_i(k_i) = \min\{k_i, m_i\}$ defines the coefficient of the service rate of node i when there are k_i customers at the node. When there are more customers at the node than servers, then all servers are working and the service rate is $m_i \mu_i$; when there are fewer customers than servers, then there are idle servers and the service rate is $k_i \mu_i$.

Theorem 10.3. *A Markov chain characterized by the previously defined state transitions has a product-form stationary distribution, that is,*

$$p_{\mathbf{N}} = p_{k_1,\cdots,k_N} = p^{(1)}_{k_1} p^{(2)}_{k_2} \cdots p^{(N)}_{k_N}, \tag{10.2}$$

where $p^{(i)}_{k_i}$ is the stationary distribution of an M/M/m_i queue with a Poisson arrival process at rate λ_i and exponentially distributed service time with the parameter μ_i. The stationary probabilities of such queues are given as a function of $p^{(i)}_0$:

$$p^{(i)}_{k_i} = \begin{cases} p^{(i)}_0 \left(\dfrac{\lambda_i}{\mu_i}\right)^{k_i} \dfrac{1}{k_i!} & 0 \le k_i \le m_i, \\[2ex] p^{(i)}_0 \left(\dfrac{\lambda_i}{\mu_i}\right)^{k_i} \dfrac{1}{m_i!}\, m_i^{m_i-k_i}, & k_i \ge m_i \end{cases} \tag{10.3}$$

and $p^{(i)}_0$ can be obtained from the normalizing equation $\sum_{k_i=0}^{\infty} p^{(i)}_{k_i} = 1$.

Proof. Based on the possible state transitions of a Markov chain, the balance equation of state **N** is as follows:

$$p_{\mathbf{N}} \left(\sum_{i=1}^{N} \gamma_i + \sum_{i=1}^{N} \alpha_i(k_i)\, \mu_i \right) = \sum_{i=1}^{N} p_{\mathbf{N}_{i,0}} \alpha_i(k_i+1)\, \mu_i\, r_{i,0}$$
$$+ \sum_{j=1}^{N} p_{\mathbf{N}_{0,j}}\, \gamma_j\, \mathcal{I}_{\{k_j>0\}} + \sum_{i=1}^{N} \sum_{j=1}^{N} p_{\mathbf{N}_{i,j}}\, \alpha_i(k_i+1)\, \mu_i\, r_{i,j}\,, \tag{10.4}$$

where $\mathcal{I}_{\{k_j>0\}}$ is the indicator of $k_j > 0$, i.e., $\mathcal{I}_{\{k_j>0\}} = 1$ if $k_j > 0$ and $\mathcal{I}_{\{k_j>0\}} = 0$ otherwise.

The left-hand side of the equation is the rate at which the process departs from state **N** in equilibrium. It contains the state transitions due to a new customer arrival from the environment and due to a service completion. The right-hand side of the equation is the rate at which the process moves to state **N** in equilibrium. This can happen due to a service of a queue from which the customer leaves the network, due to the arrival of a new customer from the environment, or due to a service completion at node i from where the customer moves to node j.

If $\gamma_i > 0$ and $\mu_i > 0$, then the Markov chain is irreducible, the solution of the stationary equation is unique, and it is enough to show that the product-form solution (10.2) satisfies the balance Eq. (10.4). First we substitute the product-form solution into the right-hand side of the balance equation and use the fact that from Eq. (10.3) we have $p^{(i)}_{k_i+1} = p^{(i)}_{k_i} \frac{\lambda_i}{\mu_i \alpha_i(k_i+1)}$ and $p^{(i)}_{k_i-1} = p^{(i)}_{k_i} \frac{\mu_i \alpha_i(k_i)}{\lambda_i}$. We obtain that

$$
\begin{aligned}
&\sum_{i=1}^{N} p_{k_1}^{(1)} \cdots p_{k_i+1}^{(i)} \cdots p_{k_N}^{(N)}\, \alpha_i(k_i+1)\, \mu_i\, r_{i,0} \\
&\quad + \sum_{j=1}^{N} p_{k_1}^{(1)} \cdots p_{k_j-1}^{(j)} \cdots p_{k_N}^{(N)}\, \gamma_j\, I_{k_j>0} \\
&\quad + \sum_{i=1}^{N}\sum_{j=1}^{N} p_{k_1}^{(1)} \cdots p_{k_i+1}^{(i)} \cdots p_{k_j-1}^{(j)} \cdots p_{k_N}^{(N)}\, \alpha_i(k_i+1)\, \mu_i\, r_{i,j} \\
&= p_{k_1}^{(1)} \cdots p_{k_N}^{(N)} \left(\sum_{i=1}^{N} \lambda_i\, r_{i,0} + \sum_{j=1}^{N} \frac{\mu_j \alpha_j(k_j)}{\lambda_j}\, \gamma_j + \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\mu_j \alpha_j(k_j)}{\lambda_j}\, \lambda_i\, r_{i,j} \right) \\
&= p_{k_1}^{(1)} \cdots p_{k_N}^{(N)} \left(\sum_{i=1}^{N} \lambda_i\, r_{i,0} + \sum_{j=1}^{N} \frac{\mu_j \alpha_j(k_j)}{\lambda_j}\, \gamma_j + \sum_{j=1}^{N} \frac{\mu_j \alpha_j(k_j)}{\lambda_j} \underbrace{\sum_{i=1}^{N} \lambda_i\, r_{i,j}}_{\lambda_j - \gamma_j} \right) \\
&= p_{k_1}^{(1)} \cdots p_{k_N}^{(N)} \left(\sum_{i=1}^{N} \lambda_i\, r_{i,0} + \sum_{j=1}^{N} \mu_j \alpha_j(k_j) \right) \\
&= p_{k_1}^{(1)} \cdots p_{k_N}^{(N)} \left(\sum_{i=1}^{N} \gamma_i + \sum_{j=1}^{N} \mu_j \alpha_j(k_j) \right) . \qquad (10.5)
\end{aligned}
$$

In the third step of the derivation we used the traffic equation of queue j, Eq. (10.1), and in the fourth step we utilized that the intensity of customer arrivals from the environment $\sum_{i=1}^{N} \gamma_i$ is identical to the intensity of customer departures to the environment, $\sum_{i=1}^{N} \lambda_i\, r_{i,0}$, in equilibrium.

The obtained expression is the left-hand side of the balance equation assuming a product-form solution of the stationary distribution. □

There might be loops in a Jackson-type queueing network of which the arrival processes of the nodes are not independent Poisson processes and to which Burke's theorem is not applicable. Consequently, in this case we obtain a product-form solution despite the queues' dependent input processes. The reverse reasoning cannot be applied. The product-form solution has no implications for the dependencies of the arrival processes of the queues.

Traffic Theorem for Open Queueing Networks

Jackson-type queueing networks possess a traffic property similar to the PASTA (Poisson arrival sees time average) property of queueing systems with a Poisson arrival process.

Theorem 10.4. *The distribution of the number of customers in the queues at the arrival instants of node j is identical to the stationary distribution of the number of customers in the queues.*

Proof. We define an extended queueing network that contains one additional single-server node, node 0, with respect to the original queueing network. The traffic matrix is also similar to the original one. It is modified only such that customers going to node j are driven to node 0 and from node 0 to node j. The rest of the traffic matrix is unchanged. The extended queueing network is also of a Jackson type, and consequently its stationary distribution is product form: $p_{\mathbf{N}'} = p_{k_0}^{(0)} p_{k_1}^{(1)} p_{k_2}^{(2)} \cdots p_{k_N}^{(N)}$.

The service rate of node 0 is μ_0. As $\mu_0 \to \infty$, the behavior of the extended queueing network becomes identical to that of the original and the arrival instants of node j are the instants when there is one customer in node 0. In this way the distribution of the customers at an arrival instants of node j is

$$\begin{aligned}\mathbf{P}(K_1 = k_1, \cdots, K_N = k_N | K_0 = 1) &= \frac{\mathbf{P}(K_0 = 1, K_1 = k_1, \cdots, K_N = k_N)}{\mathbf{P}(K_0 = 1)} \\ &= p_{\mathbf{N}}.\end{aligned}$$

□

This theorem is important for computing the delays in a queueing system.

10.6 Closed, Gordon–Newell-Type Queueing Networks

The analysis of the closed queueing network counterpart of Jackson-type queueing networks was first published by Gordon and Newell in 1967 [40]. Since that time, this kind of queueing network has often carried their name. The node behavior of Gordon–Newell-type queueing networks is identical to that of Jackson-type networks. At node i there are m_i servers with exponentially distributed service time with parameters μ_i and an infinite buffer.

In contrast to the Jackson-type networks, there is no arrival from or departure to the environment in closed queueing networks. Thus, the number of customers in the network is constant, denoted by K. If k_i denotes the number of customers at node i, then in each state of the network we have

$$\sum_{i=1}^{N} k_i = K.$$

As with the Jackson-type network, the number of customers at the nodes of the network form a Markov chain. In a closed queueing network the only possible state transition in this Markov chain is the $\mathbf{N}_{i,j} \to \mathbf{N}$ transition, that is, a customer gets served at node i and moves to node j; the transition rate of this state transition is

$\alpha_i(k_i+1)\mu_i r_{i,j}$. This state transition decreases the number of customers at node i from k_i+1 to k_i and increases the number of customers at node j from k_j-1 to k_j.

The aggregate arrival rate of the nodes are characterized by the traffic equation

$$\lambda_i = \sum_{j=1}^{N} \lambda_j\ r_{j,i} \quad i = 1, 2, \cdots, N. \tag{10.6}$$

Equation (10.6) indicates that customers arriving at node i are those customers that departed from node j and were directed to node i with probability r_{ij}. In a closed queueing network, $\sum_{j=1}^{N} r_{ij} = 1$ since there is no departure to the environment. The solution of the traffic equation of closed queueing networks is not unique. Multiplying an arbitrary solution by a constant gives another solution of the traffic equation.

Theorem 10.5. *The stationary distribution of the number of customers in a Gordon–Newell-type queueing network has product form. That is,*

$$p_{\mathbf{N}} = p_{k_1,\cdots,k_N} = \frac{1}{G}\prod_{i=1}^{N} h_{k_i}^{(i)}, \tag{10.7}$$

where λ_i is an arbitrary nonzero solution of the traffic equation,

$$h_{k_i}^{(i)} = \begin{cases} \left(\dfrac{\lambda_i}{\mu_i}\right)^{k_i} \dfrac{1}{k_i!} & 0 \le k_i \le m_i, \\[2ex] \left(\dfrac{\lambda_i}{\mu_i}\right)^{k_i} \dfrac{1}{m_i!}\ m_i^{m_i-k_i} & k_i \ge m_i, \end{cases} \tag{10.8}$$

and $G = \sum_{\mathbf{N}} \prod_{i=1}^{N} h_{k_i}^{(i)}$.

Proof. The proof follows the same pattern as that for the Jackson-type network. The balance equation for $\mathbf{N}$ is

$$p_{\mathbf{N}}\left(\sum_{i=1}^{N} \alpha_i(k_i)\ \mu_i\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} p_{\mathbf{N}_{i,j}}\ \alpha_i(k_i+1)\ \mu_i\ r_{i,j}, \tag{10.9}$$

where the left-hand side of the equation is the rate at which state $\mathbf{N}$ is left and the right-hand side is the rate at which state $\mathbf{N}$ is entered in equilibrium. Due to the irreducibility of a Markov chain, we assume a unique solution of the balance equations (together with the normalizing equation, $\sum_{\mathbf{N}\in\mathcal{S}} p_{\mathbf{N}} = 1$), and we only show that the product form satisfies the balance equation.

Substituting the product form into the right-hand side of the balance equation gives

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N} p_{k_1}^{(1)}\cdots p_{k_i+1}^{(i)}\cdots p_{k_j-1}^{(j)}\cdots p_{k_N}^{(N)}\,\alpha_i(k_i+1)\,\mu_i\,r_{i,j}\\
&= p_{k_1}^{(1)}\cdots p_{k_N}^{(N)}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\mu_j\alpha_j(k_j)}{\lambda_j}\,\lambda_i\,r_{i,j}\right)\\
&= p_{k_1}^{(1)}\cdots p_{k_N}^{(N)}\left(\sum_{j=1}^{N}\frac{\mu_j\alpha_j(k_j)}{\lambda_j}\underbrace{\sum_{i=1}^{N}\lambda_i\,r_{i,j}}_{\lambda_j}\right)\\
&= p_{k_1}^{(1)}\cdots p_{k_N}^{(N)}\left(\sum_{j=1}^{N}\mu_j\alpha_j(k_j)\right)\,, \qquad (10.10)
\end{aligned}$$

which is identical to the left-hand side of the balance equation when the product-form solution is assumed. The normalizing constant, G, ensures that the normalizing equation is satisfied. □

The main difficulties of the analysis of closed queueing networks are that the solution of the traffic equation is not unique and that the normalizing constant cannot be computed in a node-based manner only for the whole network. The computation of G requires the evaluation of all system states, which gets very high even for reasonably small networks. When there are N nodes and K customers in a network, the number of system states is $\binom{N+K-1}{K}$ (e.g., for $N = 10, K = 25$ there are 52,451,256 states).

The commonly applied solution of the first problem is to add an additional equation to the set of traffic equations, $\lambda_1 = 1$, which makes its solution unique.

The second problem, the computation of the normalizing constant, G, is a real research challenge. Many proposals exist for computing the normalizing constant efficiently. Here we summarize the convolution algorithm [18] and the mean value analysis (MVA) algorithm [79].

Convolution Algorithm

The convolution algorithm was first published by Buzen [18]. In the original paper the nodes have a single server, but it is easy to extend the algorithm to Gordon–Newell-type queueing networks where the node i has m_i ($m_i \geq 1$) servers and an infinite buffer. We present the more general version of the algorithm.

Assuming that there are n nodes and k customers in the network, let the assumed normalizing constant be

$$g(k,n) = \sum_{(k_1,\ldots,k_n),\sum_j k_j=k} \prod_{i=1}^{n} h_{k_i}^{(i)},$$

and $g(0,n) = 1$. When $g(k,n)$ is known, we obtain the normalizing constant of the network with N nodes and K customers as $G = \sum_{\mathbf{N}} \prod_{i=1}^{N} h_{k_i}^{(i)} = g(K,N)$.

The following formula allows one to determine $g(k,n)$ in a recursive manner:

$$g(k,n) = \begin{cases} h_k^{(1)} & \text{ha } n = 1, \\ \sum_{j=0}^{k} h_j^{(n)} g(k-j,n-1) & \text{ha } n > 1. \end{cases} \tag{10.11}$$

In the case of one node ($n = 1$) and $k \geq 1$ customers, the recursive formula gives $h_k^{(1)}$, and in the case of more than one nodes we have

$$\begin{aligned} g(k,n) &= \sum_{(k_1,\ldots,k_n),\sum_j k_j=k} \prod_{i=1}^{n} h_{k_i}^{(i)} \\ &= \sum_{(k_1,\ldots,k_n),\sum_j k_j=k,k_n=0} h_0^{(n)} \prod_{i=1}^{n-1} h_{k_i}^{(i)} + \ldots \\ &\quad + \sum_{(k_1,\ldots,k_n),\sum_j k_j=k,k_n=k} h_k^{(n)} \prod_{i=1}^{n-1} h_{k_i}^{(i)} \\ &= h_0^{(n)} g(k,n-1) + \ldots + h_k^{(n)} g(0,n-1). \end{aligned}$$

This expression relates the normalizing constant of a network with n nodes to the normalizing constant of a network with $n-1$ nodes.

The convolution algorithm starts from $n = 1$, $k = 1,\ldots,K$, and increases n to N step by step according to Eq. (10.11). The computational complexity of this algorithm is proportional to N and K^2 [denoted by $O(NK^2)$], and its memory complexity is proportional to K [denoted by $O(K)$].

Another benefit of the convolution algorithm is that some interesting performance parameters are closely related to the $g(k,n)$ parameters. For example, the probability that there are ℓ customers in queue k is

$$\mathbf{P}(k_\ell = k) = \sum_{(k_1,\ldots,k_n),\sum_j k_j=K,k_\ell=k} \frac{1}{G} \prod_{i=1}^{n} h_{k_i}^{(i)} = h_k^{(\ell)} \frac{g(K-k,N-1)}{g(K,N)},$$

and from this the utilization of node ℓ is

$$U_\ell = 1 - \mathbf{P}(k_\ell = 0) = 1 - h_0^{(\ell)} \frac{g(K, N-1)}{g(K, N)}.$$

Traffic Theorem for Closed Queueing Networks

The MVA algorithm is based on the traffic theorem for closed queueing networks, so we present the theorem first.

Theorem 10.6. *In a closed Gordon–Newell-type queueing network containing K customers, the distribution of the number of customers upon a customer's arrival at node j is identical to the stationary distribution of the same network with $K - 1$ customers.*

Proof. The proof is practically identical to that provided for open queueing networks. We extend the network with a single-server node 0 and redirect all customers going to node j to node 0 and from node 0 all customers go to node j. The rest of the network is left unchanged. The extended network is of a Gordon–Newell type as well; thus it has a product-form stationary distribution, $p_{k_0,k_1,\ldots,k_N,\sum_{i=0}^N k_i=K} = \frac{1}{G'} \prod_{i=0}^N h_{k_i}^{(i)}$.

The service rate of node 0 is μ_0. As $\mu_0 \to \infty$, the behavior of the extended network and that of the original networks are identical, and the arrival instances of node j are the instances when the number of customers in node 0 is 1. Thus,

$$\mathbf{P}\left(K_1 = k_1, \cdots, K_N = k_N, \sum_{i=0}^{N} k_i = K | K_0 = 1\right)$$

$$= \frac{\mathbf{P}\left(K_0 = 1, K_1 = k_1, \cdots, K_N = k_N, \sum_{i=0}^N k_i = K\right)}{\mathbf{P}(K_0 = 1)}$$

$$= \mathbf{P}\left(K_1 = k_1, \cdots, K_N = k_N, \sum\nolimits_{i=1}^N k_i = K - 1\right).$$

□

MVA Algorithm

In the convolution algorithm, the number of nodes increases in an iteration of the algorithm. The MVA algorithm is a kind of counterpart of the convolution algorithm in the sense that the MVA algorithm is also an iterative algorithm, but in this case the number of customers increases in an iteration step. According to this approach, we analyze the involved quantities as a function of the number of customers in the network.

In contrast with the convolution algorithm, the applicability of the MVA algorithm is limited to the case of single servers at the network nodes, i.e., $m_i = 1, i = 1, \ldots, N$, and the algorithm yields mean performance measures, hence its name.

The mean time a customer spends at node i during a visit to node i is

$$\mathbf{E}\left(T_i(K)\right) = \left(1 + \mathbf{E}\left(N_i^*(K)\right)\right)\frac{1}{\mu_i},$$

where $\mathbf{E}\left(N_i^*(K)\right)$ denotes the mean number of customers present at node i upon the arrival of an observed customer. According to the traffic theorem, $\mathbf{E}\left(N_i^*(K)\right)$ is identical to the stationary number of customers at node i when the number of customers in the network is $K-1$, i.e., $\mathbf{E}\left(N_i(K-1)\right)$, whence

$$\mathbf{E}\left(T_i(K)\right) = \left(1 + \mathbf{E}\left(N_i(K-1)\right)\right)\frac{1}{\mu_i}.$$

On the other hand, the mean number of customers at node i in equilibrium is

$$\mathbf{E}\left(N_i(K)\right) = K\,\frac{\lambda_i \mathbf{E}\left(T_i(K)\right)}{\sum_{j=1}^{N} \lambda_j \mathbf{E}\left(T_j(K)\right)}$$

because the arrival rate at node i is proportional to an arbitrary nonzero solution of the traffic equation $\hat{\lambda}_i = \lambda_i c$, according to Little's law $\mathbf{E}\left(N_i(K)\right) = \hat{\lambda}_i \mathbf{E}\left(T_i(K)\right)$ and

$$\begin{aligned} K\,\frac{\lambda_i \mathbf{E}\left(T_i(K)\right)}{\sum_{j=1}^{N} \lambda_j \mathbf{E}\left(T_j(K)\right)} &= K\,\frac{\hat{\lambda}_i \mathbf{E}\left(T_i(K)\right)}{\sum_{j=1}^{N} \hat{\lambda}_j \mathbf{E}\left(T_j(K)\right)} = K\,\frac{\mathbf{E}\left(N_i(K)\right)}{\sum_{j=1}^{N} \mathbf{E}\left(N_j(K)\right)} \\ &= K\,\frac{\mathbf{E}\left(N_i(K)\right)}{K} = \mathbf{E}\left(N_i(K)\right). \end{aligned}$$

Applying Little's law to another time we obtain

$$\hat{\lambda}_i = \frac{\mathbf{E}\left(N_i(K)\right)}{\mathbf{E}\left(T_i(K)\right)} = K\,\frac{\lambda_i}{\sum_{j=1}^{N} \lambda_j \mathbf{E}\left(T_j(K)\right)}.$$

With these expressions we have all the ingredients of the iterative algorithm:

Initial value:

$$\mathbf{E}\left(N_i(0)\right) = 0;$$

Iteration step:

$$\mathbf{E}\left(T_i(K)\right) = \left(1 + \mathbf{E}\left(N_i(K-1)\right)\right)\frac{1}{\mu_i},$$

$$\mathbf{E}\left(N_i(K)\right) = K\,\frac{\lambda_i \mathbf{E}\left(T_i(K)\right)}{\sum_{j=1}^{N} \lambda_j \mathbf{E}\left(T_j(K)\right)};$$

Closing step:

$$\hat{\lambda}_i = \frac{\mathbf{E}\left(N_i(K)\right)}{\mathbf{E}\left(T_i(K)\right)}.$$

The computational complexity and memory complexity of the algorithm are $O(KN^2)$ and $O(N)$. Compared to the convolution algorithm the MVA is more efficient when K is larger than N.

10.7 BCMP Networks: Multiple Customer and Service Types

The Jackson-type and Gordon–Newell-type queueing networks have a product-form stationary distribution. Thus, efficient computational methods are applicable for the analysis of systems modeled by this kind of network. For a long time, the performance analysis and the development of efficient computer systems were based on these kinds of simple and computable models. The analysis of increasingly complex system behavior required the introduction of more complex queueing behavior and the analysis of the obtained queueing network models. This resulted in fertile research in an effort to find the most general set of queueing networks with a product-form stationary distribution. The results of this effort are summarized in [9], and the set of most general queueing networks with a product-form solution is commonly referred to as BCMP networks, whose abbreviation comes from the initials of the coauthors: Baskett, Chandy, Muntz, and Palacios [9].

The set of BCMP networks generalizes the previous queueing networks in two main directions. In the previously discussed queueing networks, customers are indistinguishable and the service discipline is first come, first served (FCFS). In BCMP networks, customers belong to customer classes that are distinguished by the system because customers of different classes might arrive from the environment at the nodes at different rates, might obtain different services (service time distribution and service discipline) at the nodes, and might follow a different traffic routing probability upon completion of a service. Still, customers of the same class are indistinguishable.

The arrival of class r customers at node i occurs at rate γ_{ir}. When a class r customer is rendered a service at node i, the customer gets in the queue at node j as a class s customer with probability $P_{ir,js}$, i.e., customers might change their class right after the completion of a service. Let the number of customer classes be C. Then

$$\sum_{j=0}^{N}\sum_{s=1}^{C} P_{ir,js} = 1, \quad \forall i = 1,\ldots,N,\ r = 1,\ldots,C,$$

$P_{ir,0s}$ denotes the probability of departure to the environment.

A wide range of traffic models can be defined with an appropriate setting of the arrival rate γ_{ir} and traffic routing probability $P_{ir,js}$. Some examples are listed below.

- Customer classes are independent, and some classes behave as in open queueing networks and others as in closed queueing networks: $P_{ir,js} = 0$ if $r \neq s$, i.e., there is no class change. $\gamma_{ir} = 0$ if $r \leq C_z$, and for all $r > C_z$ there exists i such that $\gamma_{ir} > 0$, i.e., the first C_z classes of customers behave as in closed queueing networks and the rest as in open ones. The probability of departure to the environment is as follows, $P_{ir,0s} = 0$ for $r \leq C_z$, and for all $r > C_z$ there exists i such that $P_{ir,0s} > 0$.
- Background traffic at a subset of the network: Let $\gamma_{ir} = 0$ if $i > N_z$, $r \leq C_z$, and $P_{ir,js} = 0$ if $i \leq N_z$, $j > N_z$, $r, s \leq C_z$. In this case the class $r \leq C_z$ customers load only node $i \leq N_z$ and form a kind of background traffic for customers of class $r > C_z$ in that part of the network.
- Multiple service at a node: Customer classes can be used to obtain a fixed number of services, u, at node i during a single visit to node i by customers of class v. For example, if for $r = v, \ldots, v + u - 2$ we let $P_{ir,js} = 1$ if $s = r + 1$, $j = i$, and $P_{ir,js} = 0$ otherwise, and for $r = v + u - 1$ we let $P_{ir,js} \geq 0$ if $s = r$, $j \neq i$, and $P_{ir,js} = 0$ otherwise, then we have the following behavior. A class v customer arrives at node i and gets served sooner as a class v customer than as a class $v + 1$ customer and so on, while it departs as a class $v + u - 1$ customer from node i and goes to node j as a class v customer.

The service disciplines at a node of a BCMP network can be one of the following disciplines:

1. FCFS (first come, first served): Customers arrive at the server in the same order in which they arrived at the node. With this service discipline the service time of all customers is exponentially distributed with the same parameter, which is common to all customer classes. The service intensity might depend on the number of all customers at the node.
2. Processor sharing (PS): In this case, the service capacity of the server is divided into as many equal parts as there are customers at the node, and each part of the server capacity is assigned to a customer. That is, when there are n customers at the node, all of them are served by a $1/n$ portion of the full service capacity. In this case (if there are n customers at the node during the complete service of a customer), the service time of the customer is n times longer than it would

have been had the full service capacity been assigned to this customer. With this service discipline the service time distribution of different customer classes might be different and can be more general than exponentially distributed. Service time distributions with rational Laplace transforms (matrix exponential distributions) are allowed in this case.

3. LCFS–PR (last come first served–preemptive resume): The server serves one customer at a time, but in such a way that the last arrived customer interrupts the service of the customer currently being served (if any) and starts being served. If during this customer's service time a new customer arrives, the first customer is interrupted and waits while all of the customers arriving later get served. At this point, the first cusomter goes to the server again and resumes the service process starting at the point at which it was interrupted.

   Similar to the PS case, with this service discipline the service time distribution of different customer classes might be different and can be more general than exponentially distributed. Service time distributions with rational Laplace transforms (matrix exponential distributions) are allowed with this service discipline.
4. Infinite server (IS): There are infinitely many servers in this service discipline, and thus all arriving customers go to an idle server upon arrival. Similar to the PS and LCFS–PR cases, with this service discipline the service time distributions of different customer classes might be different and can be more general than exponentially distributed. Service time distributions with rational Laplace transforms (matrix exponential distributions) are allowed with this service discipline.

With the introduction of customer classes, the traffic equation only slightly modifies,

$$\lambda_{ir} = \gamma_{ir} + \sum_{j=1}^{N} \sum_{s=1}^{C} \lambda_{js} \, P_{js,ir}, \tag{10.12}$$

but to describe the product-form solution of BCMP networks, we need to introduce further cumbersome notations. To avoid this, we restrict our attention to exponentially distributed service times instead of matrix exponentially distributed ones, but we allow all other generalizations of BCMP service disciplines.

Let N_{ir} denote the number of class r customers at node i and define the vectors $\mathbf{N_i} = \{N_{i1}, \ldots, N_{iC}\}$ and $\mathbf{N} = \{\mathbf{N_1}, \ldots \mathbf{N_N}\}$. Thus, vector $\mathbf{N}$ defines the distribution of the different classes of customers at the network nodes. With this notation the stationary distribution has the form

$$p_{\mathbf{N}} = \frac{1}{G} \prod_{i=1}^{N} h^{(i)}_{\mathbf{N_i}}, \tag{10.13}$$

where

$$h_{\mathbf{N_i}}^{(i)} = \begin{cases} \dfrac{N_i!}{\mu_i^{N_i}} \displaystyle\prod_{r=1}^{C} \dfrac{1}{N_{ir}!} \lambda_{ir}^{N_{ir}} & \text{if node } i \text{ is FCFS type,} \\ N_i! \displaystyle\prod_{r=1}^{C} \dfrac{1}{N_{ir}!} \left(\dfrac{\lambda_{ir}}{\mu_{ir}}\right)^{N_{ir}} & \text{if node } i \text{ is PS or IS type,} \\ \displaystyle\prod_{r=1}^{C} \dfrac{1}{N_{ir}!} \left(\dfrac{\lambda_{ir}}{\mu_{ir}}\right)^{N_{ir}} & \text{if node } i \text{ is LCFS-PR type,} \end{cases}$$

and $N_i = \sum_{r=1}^{C} N_{ir}$. μ_{ir} denotes the service rate of a class r customer at node i.

10.8 Non-Product-Form Queueing Networks

Despite the fact that BCMP networks allow for a wide range of node behaviors, there are practical examples whose stationary solutions do not exhibit product-form solutions. The most common reasons for non-product-form solutions are

- Non-Poisson customer arrival process,
- Different exponentially distributed service time at FCFS-type node for different customer classes,
- Nonexponentially distributed service time at FCFS-type node,
- Nonmatrix exponentially distributed service time,
- Queueing nodes with finite buffer.

In general queueing networks, the stochastic behavior of the number of (different classes of) customers at the nodes is not a Markov chain (e.g., in the case of general interarrival or service time distributions). There are also cases where the number of (different classes of) customers at the nodes is a Markov chain but the stationary solution of this Markov chain does not possess product form (e.g., in the case of a Poisson arrival process and exponentially distributed service time distributions and finite-capacity FCFS-type nodes). In these cases no exact analysis methods are available, and we must resort to approximate analysis methods.

The majority of the approximate analysis methods are somewhat based on a product-form solution. They analyze a system as if its solution were of product form and adjust the result obtained from the product-form assumptions to better satisfy system equations.

From the set of approximate analysis methods of queueing networks we summarize traffic-based decomposition.

10.9 Traffic-Based Decomposition

One way to interpret the product-form solution is that the network nodes are independently analyzed based on the traffic load given by the solution of the traffic equation and the known service process (discipline and service time) of the node.

Traffic-based decomposition is an iterative procedure that analyzes the nodes of a network independently, and the traffic load of the node under evaluation is determined based on the departure processes of the network nodes previously analyzed.

The advantages of the procedure are its flexibility and low computational cost, while its disadvantages are the potential inaccuracy of the results and the lack of evidence about the convergence of the procedure. Despite its disadvantages, this is a very often applied approximate analysis method in practice because in the majority of cases it converges and gives reasonable agreement with simulation results.

The traffic-based decomposition procedure iteratively goes through all nodes of the network and performs the following steps for all nodes:

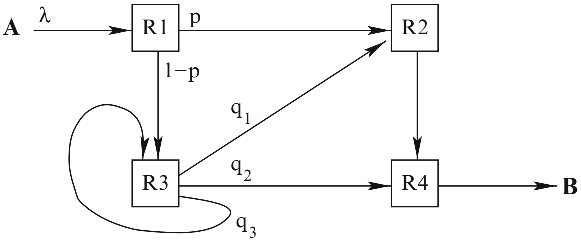
- Traffic aggregation: aggregates the traffic coming from the environment and from the departure processes of the other nodes (based on the preceding iterations).
- Node analysis and departure process computation: a single queueing system analysis step in which the parameters of the departure process are also computed.
- Departure process filtering: computation of traffic components going to other network nodes.

The complexity of an iteration step and the accuracy of the results depend on the applied traffic descriptors. The flexibility of the procedure is due to the wide range of potentially applicable traffic descriptors. The most commonly used traffic descriptor is the average intensity of the traffic such that a Poisson arrival process is assumed with a given intensity. Using this traffic model with more than one traffic class results in a nontrivial analysis problem itself. If a more sophisticated traffic model is applied to, e.g., higher moments or correlation parameters of the interarrival time distribution are considered, then the complexity of the analysis steps increases and the overall accuracy improves.

10.10 Exercises

Exercise 10.1. In the depicted queueing network the requests of input A are forwarded to output B according to the following traffic routing probabilities: $p = 0.3, q_1 = 0.2, q_2 = 0.5, q_3 = 0.3$.

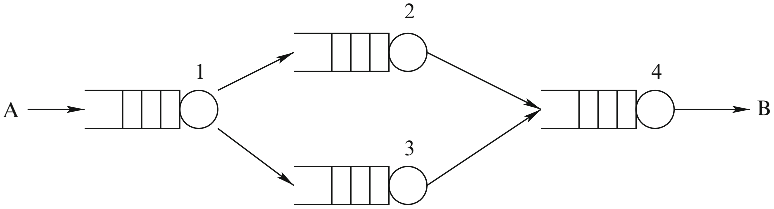
Requests from input A arrive according to a Poisson process at a rate $\lambda = 50$. The service times are exponentially distributed in nodes R1, R2, and R3 with the parameters $\mu_1 = 90$, $\mu_2 = 35$, and $\mu_3 = 100$, respectively. The service time in



R4 is composed of two phases. The first phase is exponentially distributed with the parameter $\mu_4 = 400$, and the second phase is deterministic with $D = 0.01$.

- Compute the traffic load of the nodes.
- Compute the mean and the coefficient of variation of the service time at node R4.
- Compute the system time at each node.
- Compute $\lambda_{\max}$ at which the system is at the limit of stability.

Exercise 10.2. In the depicted queueing network the requests of input A are forwarded to output B according to the following traffic routing probabilities: $p_{12} = 0.3$, $p_{13} = 0.7$.



The requests from input A arrive according to a Poisson process at a rate $\lambda = 50$. In nodes 1, 2, and 3 there are single servers and infinite buffers, and the service times are exponentially distributed with the parameters $\mu_1 = 80$, $\mu_2 = 45$, and $\mu_3 = 50$, respectively. There are two servers and two additional buffers at node R4. Both servers can serve requests with exponentially distributed service time with the parameter $\mu_4 = 40$.

- Characterize the nodes using Kendall's notation.
- Compute the traffic load of the nodes.
- Compute the system time at each node.
- Compute the server utilization at node 4.
- Compute the packet loss probability.
- Compute the mean time of a request from A to B.
- Which node is the bottleneck of the system? Which node saturates first when λ increases?