Queueing Networks and Markov Chains

Modeling and Performance Evaluation with Computer Science Applications

Second Edition

Gunter Bolch
Stefan Greiner
Hermann de Meer
Kishor S. Trivedi

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Queueing networks and Markov chains : modeling and performance evaluation with computer science applications / Gunter Bolch … [et al.].—2nd rev. and enlarged ed.
p. cm.
"A Wiley-Interscience publication."
Includes bibliographical references and index.
ISBN-13 978-0-471-56525-3 (acid-free paper)
ISBN-10 0-471-56525-3 (acid-free paper)
1. Markov processes. 2. Queuing theory. I. Bolch, Gunter.

QA76.9E94Q48 2006
004.2'401519233—dc22 200506965

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

7

Queueing Networks

Queueing networks consisting of several service stations are more suitable for representing the structure of many systems with a large number of resources than models consisting of a single service station. In a queueing network at least two service stations are connected to each other. A station, i.e., a *node*, in the network represents a resource in the real system. Jobs in principle can be transferred between any two nodes of the network; in particular, a job can be directly returned to the node it has just left.

A queueing network is called *open* when jobs can enter the network from outside and jobs can also leave the network. Jobs can arrive from outside the network at every node and depart from the network from any node. A queueing network is said to be *closed* when jobs can neither enter nor leave the network. The number of jobs in a closed network is constant. A network in which a new job enters whenever a job leaves the system can be considered as a closed one.

In Fig. 7.1 an open queueing network model of a simple computer system is shown. An example of a closed queueing network model is shown in Fig. 7.2. This is the central-server model, a particular closed network that has been proposed by [Buze71] for the investigation of the behavior of multiprogramming system with a fixed degree of multiprogramming. The node with service rate μ_1 is the central-server representing the central processing unit (CPU). The other nodes model the peripheral devices: disk drives, printers, magnetic tape units, etc. The number of jobs in this closed model is equal to the degree of multiprogramming. A closed tandem queueing network with two nodes is shown in Fig. 7.3. A very frequently occurring queueing network is the machine repairman model, shown in Fig. 7.4.

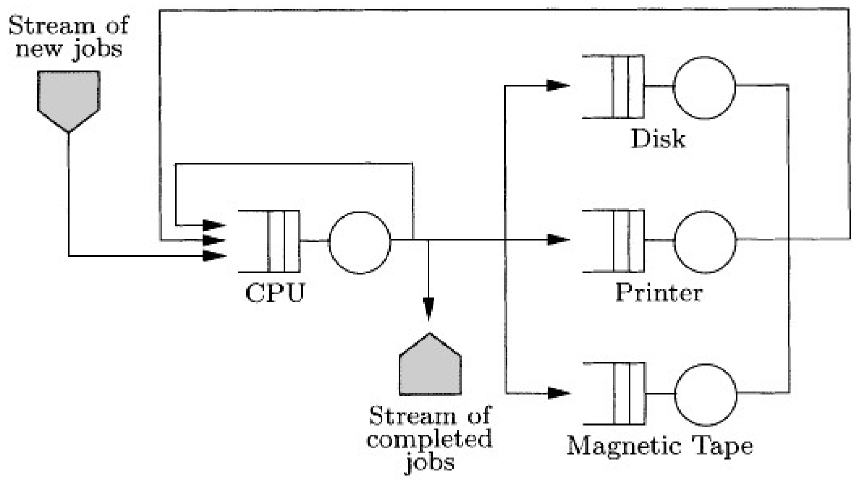

Fig. 7.1 Computer system shown as an open queueing network.

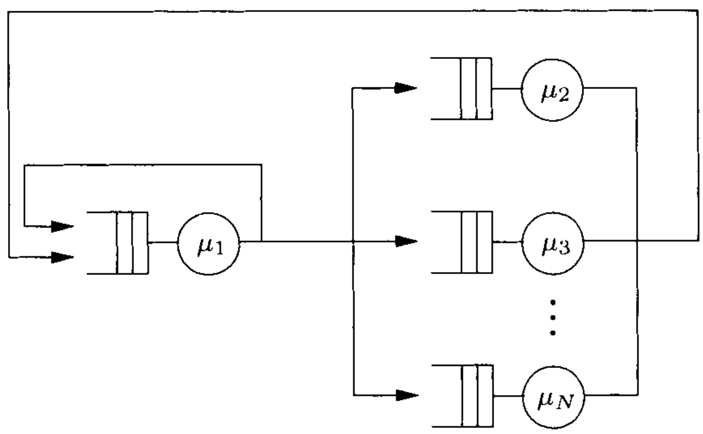

Fig. 7.2 Central-server model.

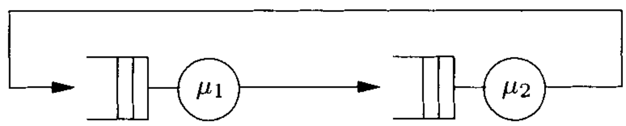

Fig. 7.3 Closed tandem network.

There are many systems that can be represented by the machine repairman model, for example, a simple terminal system where the M machines represent the M terminals and the repairman represents the computer. Another example is a system in which M machines operate independently of each other and are repaired by a single repairman if they fail. The M machines are modeled by a delay server or an infinite server node. A job does not have to wait; it is immediately accepted by one of the servers. When a machine fails, it sends a repair request to the repair facility. Depending on the service discipline, this request may have to wait until other requests have been dealt with. Since there are M machines, there can be at most M repair requests.

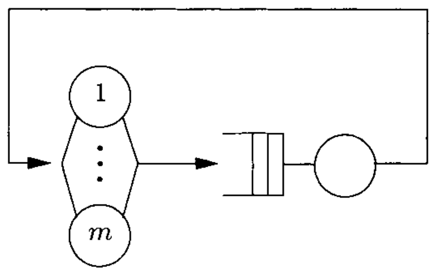

Fig. 7.4 Machine repairman model.

This is a closed two-node queueing network with M jobs that are either being processed, waiting, or are in the working machines.

7.1 DEFINITIONS AND NOTATION

We will consider both single class and multiclass networks.

7.1.1 Single Class Networks

The following symbols are used in the description of queueing networks:

N	Number of nodes
K	The constant number of jobs in a closed network
$(k_1, k_2, \ldots, k_N)$	The state of the network
k_i	The number of jobs at the ith node; for closed networks $\sum_{i=1}^{N} k_i = K$
m_i	The number of parallel servers at the ith node ($m_i \geq 1$)
μ_i	Service rate of the jobs at the ith node
$1/\mu_i$	The mean service time of the jobs at the ith node
p_{ij}	Routing probability, the probability that a job is transferred to the jth node after service completion at the ith node. (In open networks, the node with index 0 represents the external world to the network.)
p_{0j}	The probability that a job entering the network from outside first enters the jth node
p_{i0}	The probability that a job leaves the network just after completing service at node i ($p_{i0} = 1 - \sum_{j=1}^{N} p_{ij}$)
λ_{0i}	The arrival rate of jobs from outside to the ith node

λ	The overall arrival rate from outside to an open network $(\lambda = \sum_{i=1}^{N} \lambda_{0i})$
λ_i	the overall arrival rate of jobs at the ith node

The *arrival rate* λ_i for node $i = 1, \ldots, N$ of an open network is calculated by adding the arrival rate from outside and the arrival rates from all the other nodes. Note that in statistical equilibrium the rate of departure from a node is equal to the rate of arrival, and the overall arrival rate at node i can be written as:

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^{N} \lambda_j p_{ji}\,, \quad \text{for } i = 1, \ldots, N \tag{7.1}$$

for an open network. These are known as traffic equations. For closed networks these equations reduce to

$$\lambda_i = \sum_{j=1}^{N} \lambda_j p_{ji}\,, \quad \text{for } i = 1, \ldots, N\,, \tag{7.2}$$

since no jobs enter the network from outside.

Another important network parameter is the *mean number of visits* (e_i) of a job to the ith node, also known as the *visit ratio* or *relative arrival rate*:

$$e_i = \frac{\lambda_i}{\lambda}\,, \quad \text{for } i = 1, \ldots, N\,, \tag{7.3}$$

where λ is the overall throughput of the network (see Eqs. (7.24) and (7.25)). The visit ratios can also be calculated directly from the routing probabilities using Eqs. (7.3) and (7.1), or (7.3) and (7.2). For open networks, since $\lambda_{0i} = \lambda \cdot p_{0i}$, we have

$$e_i = p_{0i} + \sum_{j=1}^{N} e_j p_{ji}\,, \quad \text{for } i = 1, \ldots, N\,, \tag{7.4}$$

and for closed networks we have

$$e_i = \sum_{j=1}^{N} e_j p_{ji}\,, \quad \text{for } i = 1, \ldots, N. \tag{7.5}$$

Since there are only $(N - 1)$ independent equations for the visit ratios in closed networks, the e_i can only be determined up to a multiplicative constant. Usually we assume that $e_1 = 1$, although other possibilities are used as well. Using e_i, we can also compute the *relative utilization* x_i, which is given by

$$x_i = \frac{e_i}{\mu_i}\,. \tag{7.6}$$

It is easy to show that [ChLa74] the ratio of server utilizations is given by

$$\frac{x_i}{x_j} = \frac{\rho_i}{\rho_j}. \tag{7.7}$$

7.1.2 Multiclass Networks

The model type discussed in the previous section can be extended by including multiple job classes in the network. The job classes can differ in their service times and in their routing probabilities. It is also possible that a job changes its class when it moves from one node to another. If no jobs of a particular class enter or leave the network, i.e., the number of jobs of this class is constant, then the job class is said to be *closed.* A job class that is not closed is said to be *open.* If a queueing network contains both open and closed classes, then it is said to be a *mixed* network. Figure 7.5 shows a mixed network.

The following additional symbols are needed to describe queueing networks that contain multiple job classes, namely:

R The number of job classes in a network

k_{ir} The number of jobs of the rth class at the ith node; for a closed network:

$$\sum_{i=1}^{N}\sum_{r=1}^{R} k_{ir} = K \tag{7.8}$$

K_r The number of jobs of the rth class in the network; not necessarily constant, even in a closed network:

$$\sum_{i=1}^{N} k_{ir} = K_r \tag{7.9}$$

$\mathbf{K}$ The number of jobs in the various classes, known as the population vector ($\mathbf{K} = (K_1, \ldots, K_R)$)

$\mathbf{S}_i$ The state of the ith node ($\mathbf{S}_i = (k_{i1}, \ldots, k_{iR})$):

$$\sum_{i=1}^{N} \mathbf{S}_i = \mathbf{K} \tag{7.10}$$

$\mathbf{S}$ The overall state of the network with multiple classes ($\mathbf{S} = (\mathbf{S}_1, \ldots, \mathbf{S}_N)$)

μ_{ir} The service rate of the ith node for jobs of the rth class

$p_{ir,js}$ The probability that a job of the rth class at the ith node is transferred to the sth class and the jth node (routing probability)

$p_{0,js}$ The probability in an open network that a job from outside the network enters the jth node as a job of the sth class

$p_{ir,0}$ The probability in an open network that a job of the rth class leaves the network after having been serviced at the ith node, so

$$p_{ir,0} = 1 - \sum_{j=1}^{N} \sum_{s=1}^{R} p_{ir,js} \tag{7.11}$$

λ The overall arrival rate from outside to an open network

$\lambda_{0,ir}$ The arrival rate from outside to node i for class r jobs ($\lambda_{0,ir} = \lambda \cdot p_{0,ir}$)

λ_{ir} The arrival rate of jobs of the rth class at the ith node:

$$\lambda_{ir} = \lambda \cdot p_{0,ir} + \sum_{j=1}^{N} \sum_{s=1}^{R} \lambda_{js} \cdot p_{js,ir}\,; \tag{7.12}$$

for closed networks, $p_{0,ir} = 0$ $(1 < i < N\,,\ 1 < r < R)$ and we obtain

$$\lambda_{ir} = \sum_{j=1}^{N} \sum_{s=1}^{R} \lambda_{js} \cdot p_{js,ir}\,. \tag{7.13}$$

The mean number of visits e_{ir} of a job of the rth class at the ith node of an open network can be determined from the routing probabilities similarly to Eq. (7.4):

$$e_{ir} = p_{0,ir} + \sum_{j=1}^{N} \sum_{s=1}^{R} e_{js} p_{js,ir}\,, \quad \text{for } i = 1,\ldots,N\,,\ r = 1,\ldots,R\,. \tag{7.14}$$

For closed networks, the corresponding equation is

$$e_{ir} = \sum_{j=1}^{N} \sum_{s=1}^{R} e_{js} p_{js,ir}\,, \qquad \text{for } i = 1,\ldots,N\,,\ r = 1,\ldots,R\,. \tag{7.15}$$

Usually we assume that $e_{1r} = 1$, for $r = 1,\ldots,R$, although other settings are also possible.

7.2 PERFORMANCE MEASURES

7.2.1 Single Class Networks

Analytic methods to calculate state probabilities and other performance measures of queueing networks are described in the following sections. The determination of the *steady-state probabilities* $\pi(k_1,\ldots,k_N)$ of all possible states of the network can be regarded as the central problem of queueing theory. The mean values of all other important performance measures of the network can be calculated from these. There are, however, simpler methods for calculating these characteristics directly without using these probabilities.

Note that we use a slightly different notation compared with that used in Chapters 2–5. In Chapters 2-5, $\pi_i(t)$ denoted the transient probability of the CTMC being in state i at time t and π_i as the steady-state probability in state i. Since we now deal with multidimensional state spaces, $\pi(k_1, k_2, \ldots, k_N)$ will denote the steady-state probability of state $(k_1, k_2, \ldots, k_N)$.

The most important performance measures for queueing networks are:

Marginal Probabilities $\pi_i(k)$: For *closed* queueing networks, the *marginal probabilities* $\pi_i(k)$ that the ith node contains exactly $k_i = k$ jobs are calculated as follows:

$$\pi_i(k) = \sum_{\substack{\sum_{j=1}^{N} k_j = K \\ \& \ k_i = k}} \pi(k_1, \ldots, k_N). \tag{7.16}$$

Thus, $\pi_i(k)$ is the sum of the probabilities of all possible states $(k_1, \ldots, k_N)$, $0 \leq k_i \leq K$ that satisfy the condition $\sum_{j=1}^{N} k_j = K$ where a fixed number of jobs, k, is specified for the ith node. The normalization condition that the sum of the probabilities of all possible states $(k_1, \ldots, k_N)$ that satisfy the condition $\sum_{j=1}^{N} k_j = K$ with $(0 \leq k_j \leq K)$ must be 1, that is,

$$\sum_{\sum_{j=1}^{N} k_j = K} \pi(k_1, \ldots, k_N) = 1. \tag{7.17}$$

Correspondingly, for *open* networks we have

$$\pi_i(k) = \sum_{k_i = k} \pi(k_1, \ldots, k_N),$$

with the normalization condition

$$\sum \pi(k_1, \ldots, k_N) = 1.$$

Now we can use the marginal probabilities to obtain other interesting performance measures for open and closed networks. For closed networks we have to take into consideration that $\pi_i(k) = 0 \quad \forall k > K$.

Utilization ρ_i: The *utilization* ρ_i of a single server node with index i is given by

$$\rho_i = \sum_{k=1}^{\infty} \pi_i(k), \tag{7.18}$$

where ρ_i is the probability that the ith node is busy, that is,

$$\rho_i = 1 - \pi_i(0)\,. \tag{7.19}$$

For nodes with multiple servers we have

$$\rho_i = \frac{1}{m_i}\sum_{k=0}^{\infty}\min(m_i, k)\pi_i(k) = 1 - \sum_{k=0}^{m_i-1}\frac{m_i - k}{m_i}\cdot\pi_i(k)\,, \tag{7.20}$$

and if the service rate is independent of the load we get (see Eq. (6.4))

$$\rho_i = \frac{\lambda_i}{m_i\mu_i}\,. \tag{7.21}$$

Throughput λ_i: The *throughput* λ_i of an individual node with index i represents in general the rate at which jobs leave the node:

$$\lambda_i = \sum_{k=1}^{\infty}\pi_i(k)\mu_i(k)\,, \tag{7.22}$$

where the service rate $\mu_i(k)$ is, in general, dependent on the load, i.e., on the number of jobs at the node. For example, a node with multiple servers ($m_i > 1$) can be regarded as a single server whose service rate depends on the load $\mu_i(k) = \min(k, m_i)\cdot\mu_i$, where μ_i is the service rate of an individual server. It is also true for load-independent service rates that (see Eqs. (6.4), and (6.5))

$$\lambda_i = m_i\cdot\rho_i\cdot\mu_i\,. \tag{7.23}$$

We note that for a node in equilibrium, arrival rate and throughput are equal. Also note that when we consider nodes with finite buffers, arriving customers can be lost when the buffer is full. In this case, node throughput will be less than the arrival rate to the node.

Overall Throughput λ: The *overall throughput* λ of an open network is defined as the rate at which jobs leave the network. For a network in equilibrium, this departure rate is equal to the rate at which jobs enter the network, that is

$$\lambda = \sum_{i=1}^{N}\lambda_{0i}\,. \tag{7.24}$$

The overall throughput of a closed network is defined as the throughput of a particular node with index i for which $e_i = 1$. Then the overall throughput of jobs in closed networks is

$$\lambda = \frac{\lambda_i}{e_i}\,. \tag{7.25}$$

Mean Number of Jobs $\overline{K}_i$: The *mean number of jobs* at the ith node is given by

$$\overline{K}_i = \sum_{k=1}^{\infty} k \cdot \pi_i(k) \,. \tag{7.26}$$

From Little's theorem (see Eq. (6.9)), it follows that

$$\overline{K}_i = \lambda_i \cdot \overline{T}_i \,, \tag{7.27}$$

where $\overline{T}_i$ denotes the mean response time.

Mean Queue Length $\overline{Q}_i$: The *mean queue length* at the ith node is determined by

$$\overline{Q}_i = \sum_{k=m_i}^{\infty} (k - m_i) \cdot \pi_i(k) \,, \tag{7.28}$$

or, using Little's theorem,

$$\overline{Q}_i = \lambda_i \overline{W}_i \,, \tag{7.29}$$

where $\overline{W}_i$ is the *mean waiting time.*

Mean Response Time $\overline{T}_i$: The *mean response time* of jobs at the ith node can be calculated using Little's theorem (see Eq. (6.9)) for a given mean number of jobs $\overline{K}_i$:

$$\overline{T}_i = \frac{\overline{K}_i}{\lambda_i} \,. \tag{7.30}$$

Mean Waiting Time $\overline{W}_i$: If the service rates are independent of the load, then the *mean waiting time* at node i is

$$\overline{W}_i = \overline{T}_i - \frac{1}{\mu_i} \,. \tag{7.31}$$

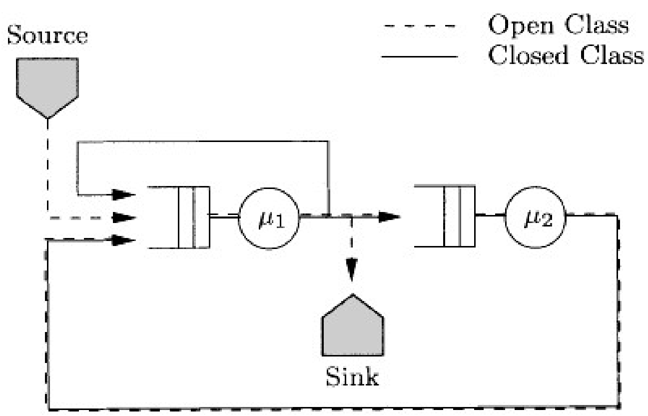

Fig. 7.5 A mixed network.

7.2.2 Multiclass Networks

The extension of queueing networks to include multiple job classes leads to a corresponding extension of the performance measures. The *state probability* of a network with multiple job classes is represented by $\pi(\mathbf{S}_1, \ldots, \mathbf{S}_N)$. The normalization condition that the sum of the probabilities of all possible states $(\mathbf{S}_1, \ldots, \mathbf{S}_N)$ is 1 must also be satisfied here.

Marginal Probability $\pi_i(\mathbf{k})$: For closed networks the *marginal probability*, i.e., the probability that the ith node is in the state $\mathbf{S}_i = \mathbf{k}$, is given by

$$\pi_i(\mathbf{k}) = \sum_{\substack{\sum_{j=1}^{N} \mathbf{S}_j = \mathbf{K} \\ \& \ \mathbf{S}_i = \mathbf{k}}} \pi(\mathbf{S}_1, \ldots, \mathbf{S}_N), \tag{7.32}$$

and for open networks we have

$$\pi_i(\mathbf{k}) = \sum_{s_i = k} \pi(\mathbf{S}_1, \ldots, \mathbf{S}_N). \tag{7.33}$$

The following formulae for computing the performance measures can be applied to open and closed networks.

Utilization ρ_{ir}: The *utilization* of the ith node with respect to jobs of the rth class is

$$\rho_{ir} = \frac{1}{m_i} \sum_{\substack{\text{all states } \mathbf{k} \\ \text{with } k_r > 0}} \pi_i(\mathbf{k}) \frac{k_{ir}}{k_i} \min(m_i, k_i), \quad k_i = \sum_{r=1}^{R} k_{ir}, \tag{7.34}$$

and if the service rates are independent on the load, we have

$$\rho_{ir} = \frac{\lambda_{ir}}{m_i \mu_{ir}}. \tag{7.35}$$

Throughput λ_{ir}: The *throughput* λ_{ir} is the rate at which jobs of the rth class are serviced and leave the ith node [BrBa80]:

$$\lambda_{ir} = \sum_{\substack{\text{all states } \mathbf{k} \\ \text{with } k_r > 0}} \pi_i(\mathbf{k}) \frac{k_{ir}}{k_i} \mu_i(k_i); \tag{7.36}$$

or if the service rates are independent on the load, we have:

$$\lambda_{ir} = m_i \cdot \rho_{ir} \cdot \mu_{ir}. \tag{7.37}$$

Overall Throughput λ_r: The *overall throughput* of jobs of the rth class in closed networks with multiclasses is

$$\lambda_r = \frac{\lambda_{ir}}{e_{ir}}, \tag{7.38}$$

and for open networks we obtain

$$\lambda_r = \sum_{i=1}^{N} \lambda_{0,ir}. \tag{7.39}$$

Mean Number of Jobs $\overline{K}_{ir}$: The *mean number of jobs* of the rth class at the ith node is

$$\overline{K}_{ir} = \sum_{\substack{\text{all states } \mathbf{k} \\ \text{with } k_r > 0}} k_r \cdot \pi_i(\mathbf{k}). \tag{7.40}$$

Little's theorem can also be used here:

$$\overline{K}_{ir} = \lambda_{ir} \cdot \overline{T}_{ir}. \tag{7.41}$$

Mean Queue Length $\overline{Q}_{ir}$: The *mean queue length* of class r jobs at the ith node can be calculated using Little's theorem as

$$\overline{Q}_{ir} = \lambda_{ir} \overline{W}_{ir}. \tag{7.42}$$

Mean Response Time $\overline{T}_{ir}$: The *mean response time* of jobs of the rth class at the ith node can also be determined using Little's theorem (see Eq. (7.41)):

$$\overline{T}_{ir} = \frac{\overline{K}_{ir}}{\lambda_{ir}}. \tag{7.43}$$

Mean Waiting Time $\overline{W}_{ir}$: If the service rates are load-independent, then the *mean waiting time* is given by

$$\overline{W}_{ir} = \overline{T}_{ir} - \frac{1}{\mu_{ir}}. \tag{7.44}$$

7.3 PRODUCT-FORM QUEUEING NETWORKS

In the rest of this chapter, we will discuss queueing networks that have a special structure such that their solutions can be obtained without generating their underlying state space. Such networks are known as product-form or separable networks.

7.3.1 Global Balance

The behavior of many queueing system models can be described using CTMCs. A CTMC is characterized by the transition rates between the states of the corresponding model (for more details on CTMC see Chapters 2 - 5). If the CTMC is ergodic, then a unique steady-state probability vector independent of the initial probability vector exists. The system of equations to determine the steady-state probability vector $\boldsymbol{\pi}$ is given by $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ (see Eq. (2.58)), where $\mathbf{Q}$ is the infinitesimal generator matrix of the CTMC. This equation says that for each state of a queueing network in equilibrium, the flux out of a state is equal to the flux into that state. This conservation of flow in steady state can be written as

$$\sum_{j\in S} \pi_j q_{ji} = \pi_i \sum_{j\in S} q_{ij}\,, \quad \forall i \in S\,, \tag{7.45}$$

where q_{ij} is the transition rate from state i to state j. After subtracting $\pi_i \cdot q_{ii}$ from both sides of Eq. (7.45) and noting that $q_{ii} = -\sum_{j\neq i} q_{ij}$, we obtain the global balance equation (see Eq. (2.61)):

$$\forall i \in S: \quad \sum_{j\neq i} \pi_j q_{ji} - \pi_i \sum_{j\neq i} q_{ij} = 0\,, \tag{7.46}$$

which corresponds to the matrix equation $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ (Eq. (2.58)).

In the following we use two simple examples to show how to write the global balance equations and use them to obtain performance measures.

Example 7.1 Consider the closed queueing network given in Fig. 7.6. The network consists of two nodes ($N = 2$) and three jobs ($K = 3$). The service times are exponentially distributed with mean values $1/\mu_1 = 5\,\text{sec}$ and $1/\mu_2 = 2.5\,\text{sec}$, respectively. The service discipline at each node is FCFS. The state space of the CTMC consists of the following four states:

$$\{(3,0),(2,1),(1,2),(0,3)\}\,.$$

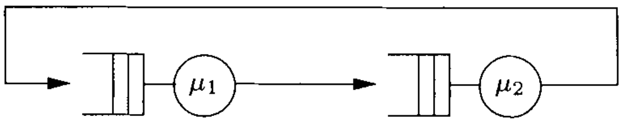

Fig. 7.6 A closed network.

The notation (k_1, k_2) says that there are k_1 jobs at node 1 and k_2 jobs at node 2, and $\pi(k_1, k_2)$ denotes the probability for that state in equilibrium. Consider state $(1,2)$. A transition from state $(1,2)$ to state $(0,3)$ takes place if the job at node 1 completes service (with corresponding rate μ_1). Therefore,

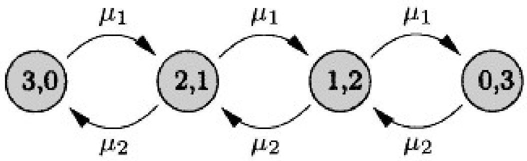

Fig. 7.7 State transition diagram for Example 7.1.

μ_1 is the transition rate from state $(1,2)$ to state $(0,3)$ and, similarly, μ_2 is the transition rate from state $(1,2)$ to state $(2,1)$. The flux into a state of the model is just given by all arcs into the corresponding state, and the flux out of that state is determined from the set of all outgoing arcs from the state. The corresponding state transition diagram is shown in Fig. 7.7. The global balance equations for this example are

$$\begin{aligned}\pi(3,0)\mu_1 &= \pi(2,1)\mu_2\,,\\ \pi(2,1)(\mu_1+\mu_2) &= \pi(3,0)\mu_1 + \pi(1,2)\mu_2\,,\\ \pi(1,2)(\mu_1+\mu_2) &= \pi(2,1)\mu_1 + \pi(0,3)\mu_2\,,\\ \pi(0,3)\mu_2 &= \pi(1,2)\mu_1\,.\end{aligned}$$

Rewriting this system of equations in the form $\boldsymbol{\pi}\mathbf{Q}=\mathbf{0}$, we have

$$\mathbf{Q} = \begin{pmatrix} -\mu_1 & \mu_1 & 0 & 0\\ \mu_2 & -(\mu_1+\mu_2) & \mu_1 & 0\\ 0 & \mu_2 & -(\mu_1+\mu_2) & \mu_1\\ 0 & 0 & \mu_2 & -\mu_2 \end{pmatrix}$$

and the steady-state probability vector $\boldsymbol{\pi} = (\pi(3,0), \pi(2,1), \pi(1,2), \pi(0,3))$. Once the steady-state probabilities are known, all other performance measures and marginal probabilities $\pi_i(k)$ can be computed. If we use $\mu_1 = 0.2$ and $\mu_2 = 0.4$, then the generator matrix $\mathbf{Q}$ has the following values:

$$\mathbf{Q} = \begin{pmatrix} -0.2 & 0.2 & 0 & 0\\ 0.4 & -0.6 & 0.2 & 0\\ 0 & 0.4 & -0.6 & 0.2\\ 0 & 0 & 0.4 & -0.4 \end{pmatrix}.$$

Using one of the steady-state solution methods introduced in Chapter 3, the steady-state probabilities are computed to be

$$\pi(3,0) = \underline{0.5333}\,,\quad \pi(2,1) = \underline{0.2667}\,,\quad \pi(1,2) = \underline{0.1333}\,,\quad \pi(0,3) = \underline{0.0667}$$

and are used to determine all other performance measures of the network, as follows:

- Marginal probabilities (see Eq. (7.16)):

$$\pi_1(0) = \pi_2(3) = \pi(0,3) = \underline{0.0667}\,,\quad \pi_1(1) = \pi_2(2) = \pi(1,2) = \underline{0.133}\,,$$

$$\pi_1(2) = \pi_2(1) = \pi(2,1) = \underline{0.2667}\,, \quad \pi_1(3) = \pi_2(0) = \pi(3,0) = \underline{0.5333}\,.$$

- Utilizations (see Eq. (7.20)):

$$\rho_1 = 1 - \pi_1(0) = \underline{0.9333}\,, \quad \rho_2 = 1 - \pi_2(0) = \underline{0.4667}\,.$$

- Throughput (see Eq. (7.23)):

$$\lambda = \lambda_1 = \lambda_2 = \rho_1\mu_1 = \rho_2\mu_2 = \underline{0.1867}\,.$$

- Mean number of jobs (see Eq. (7.26)):

$$\overline{K}_1 = \sum_{k=1}^{3} k\pi_1(k) = \underline{2.2667}\,, \quad \overline{K}_2 = \sum_{k=1}^{3} k\pi_2(k) = \underline{0.7333}\,.$$

- Mean response time of the jobs (see Eq. (7.30)):

$$\overline{T}_1 = \frac{\overline{K}_1}{\lambda_1} = \underline{12.1429}\,, \quad \overline{T}_2 = \frac{\overline{K}_2}{\lambda_2} = \underline{3.9286}\,.$$

Example 7.2 Now we modify the closed network of Example 7.1 so that the service time at node 1 is Erlang-2 distributed, while the service time at node 2 remains exponentially distributed. The modified network is shown in Fig. 7.8. Service rates of the different phases at node 1 are given by $\mu_{11} = \mu_{12} = 0.4\ \text{sec}^{-1}$, and the service rate of node 2 is $\mu_2 = 0.4$. There are $K = 2$ jobs in the system. A state $(k_1, l; k_2)$, $l = 0, 1, 2$ of the network is now not

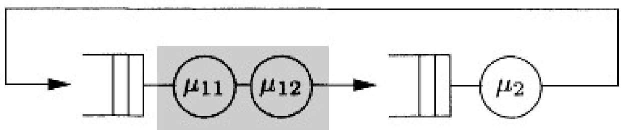

Fig. 7.8 Network with Erlang-2 distributed server.

only given by the number k_i of jobs at the nodes, but also by the phase l in which a job is being served at node 1. This state definition leads to exactly five states in the CTMC underlying the network.

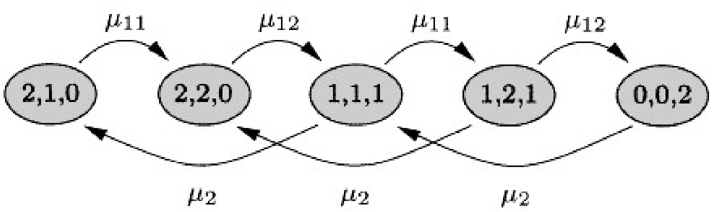

Fig. 7.9 State transition diagram for Example 7.2.

The state transition diagram of the CTMC is shown in Fig. 7.9. The following global balance equations can be derived:

$$
\begin{aligned}
\pi(2,1;0)\mu_{11} &= \pi(1,1;1)\mu_2\,, \\
\pi(2,2;0)\mu_{12} &= \pi(2,1;0)\mu_{11} + \pi(1,2;1)\mu_2\,, \\
\pi(1,1;1)(\mu_{11}+\mu_2) &= \pi(2,2;0)\mu_{12} + \pi(0,0;2)\mu_2\,, \\
\pi(1,2;1)(\mu_{12}+\mu_2) &= \pi(1,1;1)\mu_{11}\,, \\
\pi(0,0;2)\mu_2 &= \pi(1,2;1)\mu_{12}\,.
\end{aligned}
$$

The generator matrix is

$$
\mathbf{Q} = \begin{pmatrix}
-\mu_{11} & \mu_{11} & 0 & 0 & 0 \\
0 & -\mu_{12} & \mu_{12} & 0 & 0 \\
\mu_2 & 0 & -(\mu_{11}+\mu_2) & \mu_{11} & 0 \\
0 & \mu_2 & 0 & -(\mu_{12}+\mu_2) & \mu_{12} \\
0 & 0 & \mu_2 & 0 & -\mu_2
\end{pmatrix}.
$$

After inserting the values $\mu_{11} = \mu_{12} = \mu_2 = 0.4$, $\mathbf{Q}$ becomes

$$
\mathbf{Q} = \begin{pmatrix}
-0.4 & 0.4 & 0 & 0 & 0 \\
0 & -0.4 & 0.4 & 0 & 0 \\
0.4 & 0 & -0.8 & 0.4 & 0 \\
0 & 0.4 & 0 & -0.8 & 0.4 \\
0 & 0 & 0.4 & 0 & -0.4
\end{pmatrix}.
$$

Solving the system of equations $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$, we get

$$
\begin{aligned}
&\pi(2,1;0) = \underline{0.2219}\,, \quad \pi(2,2;0) = \underline{0.3336}\,, \\
&\pi(1,1;1) = \underline{0.2219}\,, \quad \pi(1,2;1) = \underline{0.1102}\,, \\
&\pi(0,0;2) = \underline{0.1125}\,.
\end{aligned}
$$

These state probabilities are used to determine the marginal probabilities:

$$
\begin{aligned}
\pi_1(0) = \pi_2(2) &= \pi(0,0;2) = \underline{0.1125}\,, \\
\pi_1(1) = \pi_2(1) &= \pi(1,1;1) + \pi(1,2;1) = \underline{0.3321}\,, \\
\pi_1(2) = \pi_2(0) &= \pi(2,1;0) + \pi(2,2;0) = \underline{0.5555}\,.
\end{aligned}
$$

The computation of other performance measures is done in the same way as in Example 7.1.

7.3.2 Local Balance

Numerical techniques based on the solution of the global balance equations (see Eq. (2.61)) can in principle always be used, but for large networks this technique is very expensive because the number of equations can be extremely large. For such large networks, we therefore look for alternative solution techniques.

In this chapter we show that efficient and exact solution algorithms exist for a large class of queueing networks. These algorithms avoid the generation and solution of global balance equations. If all nodes of the network fulfill certain assumptions concerning the distributions of the interarrival and service times and the queueing discipline, then it is possible to derive local balance equations, which describe the system behavior in an unambiguous way. These local balance equations allow an essential simplification with respect to the global balance equations because each equation can be split into a number of single equations, each one related to each individual node.

Queueing networks that have an unambiguous solution of the local balance equations are called *product-form networks*. The steady-state solution to such networks' state probabilities consist of multiplicative factors, each factor relating to a single node. Before introducing the different solution methods for product-form networks, we explain the local balance concept in more detail. This concept is the theoretical basis for the applicability of analysis methods.

Consider global balance equations (Eq. (2.61)) for a CTMC:

$$\forall i \in S: \quad \sum_{j \in S} \pi_j q_{ji} = \pi_i \sum_{j \in S} q_{ij}$$

or

$$\boldsymbol{\pi} \cdot \mathbf{Q} = \mathbf{0}\,,$$

with the normalization condition

$$\sum_{i \in S} \pi_i = 1\,.$$

Chandy [Chan72] noticed that under certain conditions the global balance equations can be split into simpler equations, known as *local balance equations*.

Local balance property for a node means: The departure rate from a state of the queueing network due to the departure of a job from node i equals the arrival rate to this state due to an arrival of a job to this node.

This can also be extended to queueing networks with several job classes in the following way:

The departure rate from a state of the queueing network due to the departure of a *class r-job* from node i equals the arrival rate to this state due to an arrival of a *class r-job* to this node.

In the case of nonexponentially distributed service times, arrivals and departures to phases, instead of nodes, have to be considered.

Example 7.3 Consider a closed queueing network (see Fig. 7.10) consisting of $N = 3$ nodes with exponentially distributed service times and the

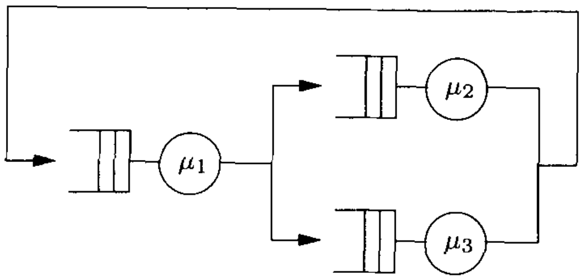

Fig. 7.10 A closed queueing network.

following service rates: $\mu_1 = 4\ \text{sec}^{-1}$, $\mu_2 = 1\ \text{sec}^{-1}$, and $\mu_3 = 2\ \text{sec}^{-1}$. There are $K = 2$ jobs in the network, and the routing probabilities are given as: $p_{12} = 0.4$, $p_{13} = 0.6$, and $p_{21} = p_{31} = 1$. The following states are possible in the network:

$$(2,0,0), \quad (0,2,0), \quad (0,0,2), \quad (1,1,0), \quad (1,0,1), \quad (0,1,1).$$

The state transition diagram of the underlying CTMC is shown in Fig. 7.11.

We set the overall flux into a state equal to the overall flux out of the state for each state to get global balance equations:

$$\begin{aligned}
&(1) & \pi(2,0,0)(\mu_1 p_{12} + \mu_1 p_{13}) &= \pi(1,0,1)\mu_3 p_{31} + \pi(1,1,0)\mu_2 p_{21}\,, \\
&(2) & \pi(0,2,0)\mu_2 p_{21} &= \pi(1,1,0)\mu_1 p_{12}\,, \\
&(3) & \pi(0,0,2)\mu_3 p_{31} &= \pi(1,0,1)\mu_1 p_{13}\,, \\
&(4) & \pi(1,1,0)(\mu_2 p_{21} + \mu_1 p_{13} + \mu_1 p_{12}) &= \pi(0,2,0)\mu_2 p_{21} + \pi(2,0,0)\mu_1 p_{12} \\
& & &\quad + \pi(0,1,1)\mu_3 p_{31}\,, \\
&(5) & \pi(1,0,1)(\mu_3 p_{31} + \mu_1 p_{12} + \mu_1 p_{13}) &= \pi(0,0,2)\mu_3 p_{31} + \pi(0,1,1)\mu_2 p_{21} \\
& & &\quad + \pi(2,0,0)\mu_1 p_{13}\,, \\
&(6) & \pi(0,1,1)(\mu_3 p_{31} + \mu_2 p_{21}) &= \pi(1,1,0)\mu_1 p_{13} + \pi(1,0,1)\mu_1 p_{12}\,.
\end{aligned}$$

To determine the local balance equations we start, for example, with the state $(1,1,0)$. The departure rate out of this state, because of the departure of a job from node 2, is given by $\pi(1,1,0)\cdot \mu_2 \cdot p_{21}$. This rate is equated to the arrival rate into this state $(1,1,0)$ due to the arrival of a job at node 2; $\pi(2,0,0)\cdot \mu_1 \cdot p_{12}$. Therefore, we get the local balance equation:

$$(4') \quad \pi(1,1,0)\cdot \mu_2 \cdot p_{21} = \pi(2,0,0)\cdot \mu_1 \cdot p_{12}\,.$$

Correspondingly, the departure rate of a serviced job at node 1 from state $(1,1,0)$ equals the arrival rate of a job, arriving at node 1 into state $(1,1,0)$:

$$(4'') \quad \pi(1,1,0)\cdot \mu_1 \cdot (p_{13} + p_{12}) = \pi(0,1,1)\cdot \mu_3 \cdot p_{31} + \pi(0,2,0)\cdot \mu_2 \cdot p_{21}\,.$$

By adding these two local balance equations, $(4')$ and $(4'')$, we get the global balance Eq. (4). Furthermore, we see that the global balance Eqs. (1),

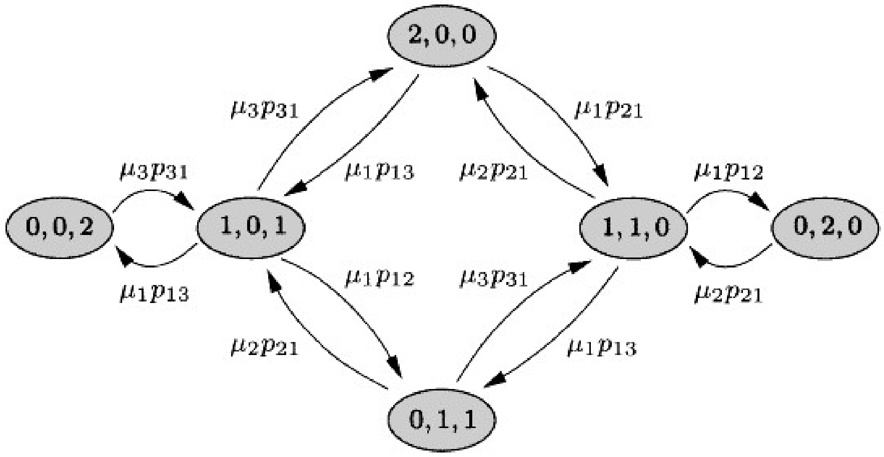

Fig. 7.11 The CTMC for Example 7.3.

(2), and (3) are also local balance equations at the same time. The rest of the local balance equations are given by

$$\begin{array}{ll} (5') & \pi(1,0,1)\mu_1(p_{12}+p_{13}) = \pi(0,1,1)\mu_2 p_{21} + \pi(0,0,2)\mu_3 p_{31}\,, \\ (5'') & \pi(1,0,1)\mu_3 p_{31} = \pi(2,0,0)\mu_1 p_{13}\,, \\ (6') & \pi(0,1,1)\mu_2 p_{21} = \pi(1,0,1)\mu_1 p_{12}\,, \\ (6'') & \pi(0,1,1)\mu_3 p_{31} = \pi(1,1,0)\mu_1 p_{13}\,, \end{array}$$

with $(5') + (5'') = (5)$ and $(6') + (6'') = (6)$, respectively.

Noting that $p_{12} + p_{13} = 1$ and $p_{21} = p_{31} = 1$, the following relations can be derived from these local balance equations:

$$\pi(1,0,1) = \pi(2,0,0)\frac{\mu_1}{\mu_3}p_{13}\,, \qquad \pi(1,1,0) = \pi(2,0,0)\frac{\mu_1}{\mu_2}p_{12}\,,$$

$$\pi(0,0,2) = \pi(2,0,0)\left(\frac{\mu_1}{\mu_3}p_{13}\right)^2, \qquad \pi(0,2,0) = \pi(2,0,0)\left(\frac{\mu_1}{\mu_2}p_{12}\right)^2,$$

$$\pi(0,1,1) = \pi(2,0,0)\frac{\mu_1^2}{\mu_2\mu_3}p_{12}p_{13}\,.$$

Imposing the normalization condition, that the sum of probabilities of all states in the network is 1, for $\pi(2,0,0)$ we get the following expression:

$$\pi(2,0,0) = \left[1 + \mu_1\left(\frac{p_{13}}{\mu_3} + \frac{p_{12}}{\mu_2} + \frac{\mu_1 p_{13}^2}{\mu_3^2} + \frac{\mu_1 p_{12}^2}{\mu_2^2} + \frac{\mu_1 p_{12}p_{13}}{\mu_2\mu_3}\right)\right]^{-1}.$$

After inserting the values, we get the following results:

$$\pi(2,0,0) = \underline{0.103}\,, \quad \pi(0,0,2) = \underline{0.148}\,, \quad \pi(1,0,1) = \underline{0.123}\,,$$
$$\pi(0,2,0) = \underline{0.263}\,, \quad \pi(1,1,0) = \underline{0.165}\,, \quad \pi(0,1,1) = \underline{0.198}\,.$$

As can be seen in this example, the structure of the local balance equations is much simpler than the global balance equations. However, not every network has a solution of the local balance equations, but there always exists a solution of the global balance equations. Therefore, local balance can be considered as a sufficient (but not necessary) condition for the global balance. Furthermore, if there exists a solution for the local balance equations, the model is then said to have the *local balance property.* Then this solution is also the unique solution of the system of global balance equations.

The computational effort for the numerical solution of the local balance equations of a queueing network is still very high but can be reduced considerably with the help of a characteristic property of local-balanced queueing networks: For the determination of the state probabilities it is not necessary to solve the local balance equations for the whole network. Instead, the state probabilities of the queueing network in these cases can be determined very easily from the state probabilities of individual single nodes of the network. If each node in the network has the local balance property, then the following two very important implications are true:

- The overall network also has the local balance property as proven in [CHT77].
- There exists a product-form solution for the network, that is,

$$\pi(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_N) = \frac{1}{G} \left[\pi(\mathbf{S}_1)\cdot \pi(\mathbf{S}_2)\cdot \ldots \cdot \pi(\mathbf{S}_N)\right] , \tag{7.47}$$

  in the sense that the expression for the state probability of the network is given by the product of marginal state probabilities of each individual node. The proof of this fact can be found in [Munt73]. The normalization constant G is chosen in such a way that the sum of probabilities over all states in the network equals 1.

Equation (7.47) says that in networks having the local-balance property, the nodes behave as if they were single queueing systems. This characteristic means that the nodes of the network can be examined in isolation from the rest of the network. Networks of the described type belong to the class of so-called *separable networks* or *product-form networks.* Now we need to examine for which types of elementary queueing systems a solution of the local balance equation exists. If the network consists of only these types of nodes then we know, because of the preceding conclusion, that the whole network has the local-balance property and the network has a product-form solution. The local-balance equations for a single node can be presented in a simplified form as follows [SaCh81]:

$$\pi(\mathbf{S})\cdot \mu_r(\mathbf{S}) = \pi(\mathbf{S} - \mathbf{1}_r)\cdot \lambda_r\,. \tag{7.48}$$

In this equation, $\mu_r(\mathbf{S})$ is the rate with which class-r jobs in state $\mathbf{S}$ are serviced at the node, λ_r is the rate at which class-r jobs arrive at the node, and $(\mathbf{S} - \mathbf{1}_r)$ describes the state of the node after a single class-r job leaves it.

It can be shown that for the following types of queueing systems the local balance property holds [Chan72]:

Type 1: **M/M/m–FCFS**. The service rates for different job classes must be equal. Examples of Type 1 nodes are input/output (I/O) devices or disks.

Type 2: **M/G/1–PS**. The CPU of a computer system can very often be modeled as a Type 2 node.

Type 3: **M/G/∞ (infinite server)**. Terminals can be modeled as Type 3 nodes.

Type 4: **M/G/1–LCFS PR**. There is no practical example for the application of Type 4 nodes in computer systems.

For Type 2, Type 3, and Type 4 nodes, different job classes can have *different general service time distributions*, provided that these have rational Laplace transform. In practice, this requirement is not an essential limitation as any distribution can be approximated as accurately as necessary using a Cox distribution.

In the next section we consider product-form solutions of separable networks in more detail.

Problem 7.1 Consider the closed queueing network given in Fig. 7.12. The network consists of two nodes ($N = 2$) and three jobs ($K = 3$). The

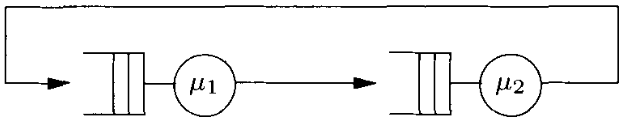

Fig. 7.12 A simple queueing network example for Problem 7.1.

service times are exponentially distributed with mean values $1/\mu_1 = 5\,\text{sec}$ and $1/\mu_2 = 2.5\,\text{sec}$. The service discipline at each node is FCFS.

(a) Determine the local balance equations.

(b) From the local balance equations, derive the global balance equations.

(c) Determine the steady-state probabilities using the local balance equations.

7.3.3 Product-Form

The term *product-form* was introduced by [Jack63] and [GoNe67a], who considered open and closed queueing networks with exponentially distributed interarrival and service times. The queueing discipline at all stations was assumed to be FCFS. As the most important result for the queueing theory, it is shown that for these networks the solution for the steady-state probabilities can be expressed as a product of factors describing the state of each node. This solution is called *product-form solution.* In [BCMP75] these results were extended to open, closed, and mixed networks with several job classes, non-exponentially distributed service times and different queueing disciplines. In this section we consider these results in more detail and give algorithms to compute performance measures of product-form queueing networks.

A necessary and sufficient condition for the existence of product-form solutions is given in the previous section but repeated here in a slightly different way:

Local Balance Property: Steady-state probabilities can be obtained by solving steady-state (global) balance equations. These equations balance the rate at which the CTMC leaves that state with the rate at which the CTMC enters it. The problem is that the number of equations increases exponentially in the number of states. Therefore, a new set of balance equations, the so-called *local balance equations*, is defined. With these, the rate at which jobs enter a *single* node of the network is equated to the rate at which they leave it. Thus, local balance is concerned with a local situation and reduces the computational effort.

Moreover, there exist two other characteristics that apply to a queueing network with product-form solution:

M ⇒ M-Property (Markov Implies Markov): A service station has the M ⇒ M-property if and only if the station transforms a Poisson arrival process into a Poisson departure process. In [Munt73] it is shown that a queueing network has a product-form solution if all nodes of the network have the M ⇒ M-property.

Station-Balance Property: A service discipline is said to have station-balance property if the service rates at which the jobs in a position of the queue are served are proportional to the probability that a job enters this position. In other words, the queue of a node is partitioned into positions and the rate at which a job enters this position is equal to the rate with which the job leaves this position. In [CHT77] it is shown that networks that have the station-balance property have a product-form solution. The opposite does not hold.

The relation between station balance (SB), local balance (LB), product-form property (PF), and Markov implies Markov property ($M \Rightarrow M$) is shown in Fig. 7.13.

7.3.4 Jackson Networks

The breakthrough in the analysis of queueing networks was achieved by the works of Jackson [Jack57, Jack63]. He examined open queueing networks and

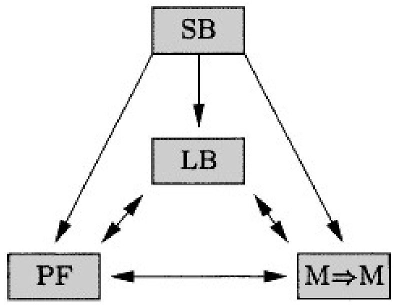

Fig. 7.13 Relation between SB, LB, PF, and $M \Rightarrow M$.

found product-form solutions. The networks examined fulfill the following assumptions:

- There is only one job class in the network.
- The overall number of jobs in the network is unlimited.
- Each of the N nodes in the network can have Poisson arrivals from outside. A job can leave the network from any node.
- All service times are exponentially distributed.
- The service discipline at all nodes is FCFS.
- The ith node consists of $m_i \geq 1$ identical service stations with the service rates μ_i, $i = 1, \ldots, N$. The arrival rates λ_{0i}, as well as the service rates, can depend on the number k_i of jobs at the node. In this case we have *load-dependent service rates* and *load-dependent arrival rates.*

Note: A service station with more than one server and a constant service rate μ_i is equivalent to a service station with exactly one server and load-dependent service rates:

$$\mu_i(k) = \begin{cases} k_i \cdot \mu_i\,, & k_i \leq m_i\,, \\ m_i \cdot \mu_i\,, & k_i \geq m_i\,. \end{cases} \tag{7.49}$$

Jackson's Theorem: If in an open network ergodicity ($\lambda_i < \mu_i \cdot m_i$) holds for all nodes $i = 1, \ldots, N$ (the arrival rates λ_i can be computed using Eq. (7.1)), then the steady-state probability of the network can be expressed as the product of the state probabilities of the individual nodes, that is,

$$\pi(k_1, k_2, \ldots, k_N) = \pi_1(k_1) \cdot \pi_2(k_2) \cdot \ldots \cdot \pi_N(k_N)\,. \tag{7.50}$$

The nodes of the network can be considered as independent M/M/m queues with arrival rate λ_i and service rate μ_i. To prove this theorem, [Jack63] has shown that Eq. (7.50) fulfills the global balance equations. Thus, the marginal probabilities $\pi_i(k_i)$ can be computed with the well-known formulae for M/M/m systems (see Eqs. (6.26), (6.27)):

$$\pi_i(k_i) = \begin{cases} \pi_i(0)\dfrac{(m_i\rho_i)^{k_i}}{k_i!}, & k_i \le m_i, \\ \pi_i(0)\dfrac{m_i^{m_i}\rho_i^{k_i}}{m_i!}, & k_i > m_i, \end{cases} \tag{7.51}$$

where $\pi_i(0)$ is given by the condition $\sum\limits_{k_i=0}^{\infty} \pi_i(k_i) = 1$:

$$\pi_i(0) = \left(\sum_{k_i=0}^{m_i-1} \frac{(m_i\rho_i)^{k_i}}{k_i!} + \frac{(m_i\rho_i)^{m_i}}{m_i!(1-\rho_i)} \right)^{-1}, \quad \rho_i = \frac{\lambda_i}{m_i\mu_i} < 1. \tag{7.52}$$

Proof: We verify that Eq. (7.50) fulfills the following global balance equations:

$$\begin{aligned} &\left(\sum_{i=1}^{N} \lambda_{0i} + \sum_{i=1}^{N} \alpha_i(k_i)\mu_i \right) \pi(k_1, \ldots, k_N) = \\ &= \sum_{i=1}^{N} \lambda_{0i}\gamma(k_i)\pi(k_1, \ldots, k_i - 1, \ldots, k_N) \\ &\quad + \sum_{i=1}^{N} \alpha_i(k_i + 1)\mu_i \left(1 - \sum_{j=1}^{N} p_{ij} \right) \pi(k_1, \ldots, k_i + 1, \ldots, k_N) \\ &\quad + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_j(k_j + 1)\mu_j p_{ji}\pi(k_1, \ldots, k_j + 1, \ldots, k_i - 1, \ldots, k_N). \end{aligned} \tag{7.53}$$

The indicator function $\gamma(k_i)$ is given by

$$\gamma(k_i) = \begin{cases} 0, & k_i = 0, \\ 1, & k_i > 0. \end{cases} \tag{7.54}$$

The function

$$\alpha_i(k_i) = \begin{cases} k_i, & k_i \le m_i, \\ m_i, & k_i \ge m_i \end{cases} \tag{7.55}$$

gives the load-dependent service rate multiplier.

For the proof we use the following relations:

$$\begin{aligned}
&\frac{\pi(k_1,\ldots,k_i+1,\ldots,k_N)}{\pi(k_1,\ldots,k_i,\ldots,\ldots,\ldots,k_N)} = \frac{\pi_1(k_1)\cdots\pi_i(k_i+1)\cdots\pi_N(k_N)}{\pi_1(k_1)\cdots\pi_i(k_i)\cdots\pi_N(k_N)} \\
&\quad = \frac{\lambda_i}{\mu_i\alpha_i(k_i+1)}, \\
&\frac{\pi(k_1,\cdots,k_i-1,\cdots,k_N)}{\pi(k_1,\cdots,k_i,\cdots,k_N)} = \frac{\mu_i\alpha_i(k_i)}{\lambda_i}, \\
&\frac{\pi(k_1,\ldots,k_j+1,\ldots,k_i-1,\ldots,k_N)}{\pi(k_1,\ldots,k_j,\ldots,k_i,\ldots,k_N)} = \frac{\lambda_j\mu_i\alpha_i(k_i)}{\lambda_i\mu_j\alpha_j(k_j+1)}.
\end{aligned} \tag{7.56}$$

If we divide Eq. (7.53) by $\pi(k_1,\ldots,k_N)$ and insert Eq. (7.56), then, by using the relation $\gamma(k_i)\cdot\alpha_i(k_i)\equiv\alpha_i(k_i)$, we get

$$\begin{aligned}
\sum_{i=1}^{N}\lambda_{0i} + \sum_{i=1}^{N}\alpha_i(k_i)\mu_i &= \sum_{i=1}^{N}\left(1-\sum_{j=1}^{N}p_{ij}\right)\lambda_i \\
&\quad + \sum_{i=1}^{N}\frac{\lambda_{0i}\mu_i\alpha_i(k_i)}{\lambda_i} + \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\mu_i\alpha_i(k_i)}{\lambda_i}p_{ji}\lambda_j.
\end{aligned} \tag{7.57}$$

The first term can be rewritten as

$$\sum_{i=1}^{N}\left(1-\sum_{j=1}^{N}p_{ij}\right)\lambda_i = \sum_{i=1}^{N}\lambda_{0i},$$

and the last one as

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\mu_i\alpha_i(k_i)}{\lambda_i}p_{ji}\lambda_j = \sum_{i=1}^{N}\mu_i\alpha_i(k_i) - \sum_{i=1}^{N}\frac{\lambda_{0i}\mu_i\alpha_i(k_i)}{\lambda_i}. \tag{7.58}$$

Substituting these results on the right side of Eq. (7.57), we get the same result as on the left-hand side. *q.e.d.*

The algorithm based on Jackson's theorem for computing the steady-state probabilities can now be described in the following three steps:

STEP 1 For all nodes, $i = 1,\ldots,N$, compute the arrival rates λ_i of the open network by solving the traffic equations, Eq. (7.1).

STEP 2 Consider each node i as an M/M/m queueing system. Check the ergodicity, Eq. (6.5), and compute the state probabilities and performance measures of each node using the formulae given in Section 6.2.3.

STEP 3 Using Eq. (7.50), compute the steady-state probabilities of the overall network.

We note that for a Jackson network with feedback input, processes to the nodes will, in general, not be Poisson and yet the nodes behave like independent M/M/m nodes. Herein lies the importance of Jackson's theorem [BeMe78]. We illustrate the procedure with the following example.

Example 7.4 Consider the queueing network given in Fig. 7.14, which consists of $N = 4$ single server FCFS nodes. The service times of the jobs at

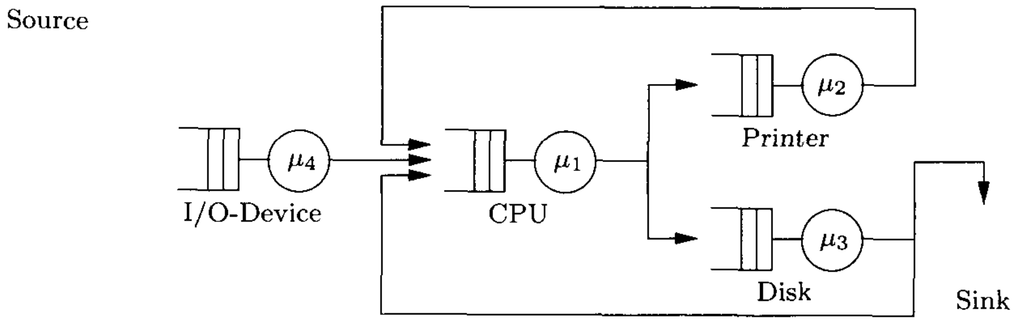

Fig. 7.14 Open queueing network model of a computer system.

each node are exponentially distributed with respective means:

$$\frac{1}{\mu_1} = 0.04\,\text{sec}\,, \quad \frac{1}{\mu_2} = 0.03\,\text{sec}\,, \quad \frac{1}{\mu_3} = 0.06\,\text{sec}\,, \quad \frac{1}{\mu_4} = 0.05\,\text{sec}\,.$$

The interarrival time is also exponentially distributed with the parameter:

$$\lambda = \lambda_{04} = 4\ \text{jobs/sec}\,.$$

Furthermore, the routing probabilities are given as follows:

$$p_{12} = p_{13} = 0.5, \quad p_{41} = p_{21} = 1, \quad p_{31} = 0.6, \quad p_{30} = 0.4.$$

Assume that we wish to compute the steady-state probability of state $(k_1, k_2, k_3, k_4) = (3, 2, 4, 1)$ with the help of the Jackson's method. For this we follow the three steps given previously:

STEP 1 Compute the arrival rates from the traffic equations, Eq. (7.1):

$$\lambda_1 = \lambda_2 p_{21} + \lambda_3 p_{31} + \lambda_4 p_{41} = \underline{20}\,, \quad \lambda_2 = \lambda_1 p_{12} = \underline{10}\,,$$
$$\lambda_3 = \lambda_1 p_{13} = \underline{10}\,, \quad \lambda_4 = \lambda_{04} = \underline{4}\,.$$

STEP 2 Compute the state probabilities and important performance measures for each node. For the *utilization* of a single server we use Eq. (6.3):

$$\rho_1 = \frac{\lambda_1}{\mu_1} = \underline{0.8}\,, \quad \rho_2 = \frac{\lambda_2}{\mu_2} = \underline{0.3}\,, \quad \rho_3 = \frac{\lambda_3}{\mu_3} = \underline{0.6}\,, \quad \rho_4 = \frac{\lambda_4}{\mu_4} = \underline{0.2}\,.$$

Thus, ergodicity ($\rho_i < 1$) is fulfilled for all nodes. The *mean number of jobs* at the nodes is given by Eq. (6.13)

$$\overline{K}_1 = \frac{\rho_1}{1-\rho_1} = \underline{4}\,, \quad \overline{K}_2 = \underline{0.429}\,, \quad \overline{K}_3 = \underline{1.5}\,, \quad \overline{K}_4 = \underline{0.25}\,.$$

Mean response times, from Eq. (6.15):

$$\overline{T}_1 = \frac{1/\mu_1}{1-\rho_1} = \underline{0.2}\,, \quad \overline{T}_2 = \underline{0.043}\,, \quad \overline{T}_3 = \underline{0.15}\,, \quad \overline{T}_4 = \underline{0.0625}\,.$$

The *mean overall response time* of a job is given by using Little's theorem in the following way:

$$\overline{T} = \frac{\overline{K}}{\lambda} = \frac{1}{\lambda}\sum_{1=1}^{4}\overline{K}_i = \underline{1.545}\,.$$

Mean waiting times, from Eq. (6.16):

$$\overline{W}_1 = \frac{\rho_1/\mu_1}{1-\rho_1} = \underline{0.16}\,, \quad \overline{W}_2 = \underline{0.013}\,, \quad \overline{W}_3 = \underline{0.09}\,, \quad \overline{W}_4 = \underline{0.0125}\,.$$

Mean queue lengths, from Eq. (6.17):

$$\overline{Q}_1 = \frac{\rho_1^2}{1-\rho_1} = \underline{3.2}\,, \quad \overline{Q}_2 = \underline{0.129}\,, \quad \overline{Q}_3 = \underline{0.9}\,, \quad \overline{Q}_4 = \underline{0.05}\,.$$

The necessary marginal probabilities can be computed using Eq. (6.12):

$$\pi_1(3) = (1-\rho_1)\rho_1^3 = \underline{0.1024}\,, \quad \pi_2(2) = (1-\rho_2)\rho_2^2 = \underline{0.063}\,,$$
$$\pi_3(4) = (1-\rho_3)\rho_3^4 = \underline{0.0518}\,, \quad \pi_4(1) = (1-\rho_4)\rho_4 = \underline{0.16}\,.$$

STEP 3 Computation of the state probability $\pi(3,2,4,1)$ using Eq. (7.50):

$$\pi(3,2,4,1) = \pi_1(3)\cdot\pi_2(2)\cdot\pi_3(4)\cdot\pi_4(1) = \underline{0.0000534}\,.$$

7.3.5 Gordon–Newell Networks

Gordon and Newell [GoNe67a] considered closed queueing networks for which they made the same assumptions as in open queueing networks, except that no job can enter or leave the system ($\lambda_{0i} = \lambda_{i0} = 0$). This restriction means that the number K of jobs in the system is always constant:

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

Thus, the number of possible states is finite, and it is given by the binomial coefficient:

$$\binom{N+K-1}{N-1}\,,$$

which describes the number of ways of distributing K jobs on N nodes. The theorem of Gordon and Newell says that the probability for each network state in equilibrium is given by the following product-form expression:

$$\pi(k_1,\ldots,k_N) = \frac{1}{G(K)}\prod_{i=1}^{N} F_i(k_i). \tag{7.59}$$

Here $G(K)$ is the so-called *normalization constant.* It is given by the condition that the sum of all network state probabilities equals 1:

$$G(K) = \sum_{\sum_{i=1}^{N} k_i = K} \prod_{i=1}^{N} F_i(k_i)\,. \tag{7.60}$$

The $F_i(k_i)$ are functions that correspond to the state probabilities $\pi_i(k_i)$ of the ith node and are given by:

$$F_i(k_i) = \left(\frac{e_i}{\mu_i}\right)^{k_i} \cdot \frac{1}{\beta_i(k_i)}\,, \tag{7.61}$$

where the visit ratios e_i are computed using Eq. (7.5). The function $\beta_i(k_i)$ is given by

$$\beta_i(k_i) = \begin{cases} k_i!\,, & k_i \le m_i\,, \\ m_i! \cdot m_i^{k_i - m_i}\,, & k_i \ge m_i\,, \\ 1\,, & m_i = 1\,. \end{cases} \tag{7.62}$$

For various applications a more general form of the function $F_i(k_i)$ is advantageous. In this generalized function, the service rates depend on the number of jobs at the node. For this function we have

$$F_i(k_i) = \frac{e_i^{k_i}}{A_i(k_i)}\,, \tag{7.63}$$

with

$$A_i(k_i) = \begin{cases} \prod_{j=1}^{k_i} \mu_i(j)\,, & k_i > 0\,, \\ 1\,, & k_i = 0\,. \end{cases} \tag{7.64}$$

With relation (7.49) it can easily be seen that the case of constant service rates, Eq. (7.61), is a special case of Eq. (7.63).

Proof: Gordon and Newell have shown in [GoNe67a] that Eq. (7.59) fulfills the global balance equations (see Eq. (7.53)) that have now the following form:

$$\left(\sum_{i=1}^{N} \gamma(k_i)\alpha_i(k_i)\mu_i\right)\pi(k_1,\ldots,k_N) = \\ \sum_{j=1}^{N}\sum_{i=1}^{N} \gamma(k_i)\alpha_j(k_j+1)\mu_j p_{ji}\pi(k_1,\ldots,k_i-1,\ldots,k_j+1,\ldots,k_N)\,, \tag{7.65}$$

where the left-hand side describes the departure rate out of state $(k_1,\ldots,k_N)$ and the right-hand side describes the arrival rate from successor states into this state. The function $\gamma(k_i)$ and $\alpha_i(k_i)$ are given by Eqs. (7.54) and (7.55).

We define now a variable transformation as follows:

$$\pi(k_1,\ldots,k_N) = \frac{Q(k_1,\ldots,k_N)}{\prod\limits_{i=1}^{N} \beta_i(k_i)}\,, \tag{7.66}$$

and

$$\pi(k_1,\ldots,k_i-1,\ldots,k_j+1,\ldots,k_N) = \\ \frac{\frac{\alpha_i(k_i)}{\alpha_j(k_j+1)} Q(k_1,\ldots,k_i-1,\ldots,k_j+1,\ldots,k_N)}{\prod\limits_{i=1}^{N} \beta_i(k_i)}\,. \tag{7.67}$$

If we substitute these equations into Eq. (7.65), we get

$$\frac{\sum\limits_{i=1}^{N} \gamma(k_i)\alpha_i(k_i)\mu_i Q(k_1,\ldots,k_N)}{\prod\limits_{i=1}^{N} \beta_i(k_i)} = \\ \frac{\sum\limits_{j=1}^{N}\sum\limits_{i=1}^{N} \gamma(k_i)\alpha_j(k_j+1)\mu_j p_{ji}\frac{\alpha_i(k_i)}{\alpha_j(k_j+1)} Q(k_1,\ldots,k_i-1,\ldots,k_j+1,\ldots,k_N)}{\prod\limits_{i=1}^{N} \beta_i(k_i)}\,.$$

This equation can be simplified using the relation $\gamma(k_i)\alpha_i(k_i) \equiv \alpha_i(k_i)$:

$$\sum_{i=1}^{N} \alpha_i(k_i)\mu_i Q(k_1,\ldots,k_N) = \\ \sum_{j=1}^{N}\sum_{i=1}^{N} \alpha_i(k_i)\mu_j p_{ji} Q(k_1,\ldots,k_i-1,\ldots,k_j+1,\ldots,k_N)\,, \tag{7.68}$$

and $Q(k_1,\ldots,k_N)$ can be written in the form

$$Q(k_1,\ldots,k_N) = \left\{\prod_{i=1}^{N} x_i^{k_i}\right\}\cdot c\,, \tag{7.69}$$

with the relative utilization $x_i = e_i/\mu_i$ (Eq. (7.6)).

Here c is a constant. Substituting this into (7.68), we have

$$\begin{aligned}
&\sum_{i=1}^{N}\alpha_i(k_i)\mu_i(x_1^{k_1}\cdots x_N^{k_N})\cdot c\\
&= \sum_{j=1}^{N}\sum_{i=1}^{N}\alpha_i(k_i)\mu_j p_{ji}(x_1^{k_1}\cdots x_i^{k_i-1}\cdots x_j^{k_j+1}\cdots x_N^{k_N})\cdot c\\
&= \sum_{j=1}^{N}\sum_{i=1}^{N}\alpha_i(k_i)\mu_j p_{ji}(x_1^{k_1}\cdots x_N^{k_N})\frac{x_j}{x_i}\cdot c\,,\\
\sum_{i=1}^{N}\alpha_i(k_i)\mu_i &= \sum_{j=1}^{N}\sum_{i=1}^{N}\alpha_i(k_i)\mu_j p_{ji}\frac{x_j}{x_i}\,.
\end{aligned}$$

This expression can be rewritten as

$$\sum_{i=1}^{N}\alpha_i(k_i)\left(\mu_i - \sum_{j=1}^{N}\mu_j p_{ji}\frac{x_j}{x_i}\right) = 0\,.$$

Since at least one $\alpha_i(k_i)$ will be nonzero, it follows that the factor in the square brackets must be zero. Thus, we are led to consider the system of linear algebraic equations for x_i:

$$\mu_i x_i = \sum_{j=1}^{N}\mu_j x_j p_{ji}\,,$$

where $x_i = e_i/\mu_i$ (Eq. (7.6)) and

$$e_i = \sum_{j=1}^{N} e_j p_{ji}\,.$$

This is the traffic equation for closed networks (Eq. (7.5)). That means that Eq. (7.69) is correct and with Eqs. (7.66) and (7.61) we obtain Eq. (7.59).*q.e.d.*

Thus, the Gordon–Newell theorem yields a product-form solution. In the general form it says that the state probabilities $\pi(k_1,k_2,\cdots,k_N)$ are given as the product of the functions $F_i(k_i)$, $i = 1\ldots,N$, defined for single nodes. It is interesting to note that if we substitute in Eq. (7.59) $F_i(k_i)$ by $L_i \cdot F_i(k_i), 1 \leq i \leq N$, then this has no influence on the solution of the state probabilities $\pi(k_1, k_2, \ldots, k_N)$ as long as L_i is a positive real number. Furthermore, the use of $\lambda \cdot e_i$, $i = 1, \ldots, N$, with an arbitrary constant $\lambda > 0$, has no influence on the results because the visit ratios e_i are relative values [ShBu77]. (Also see Problems 2 and 3 on page 444 of [Triv01].)

The Gordon–Newell method for computing the state probabilities can be summarized in the following four steps:

STEP 1 Compute the visit ratios e_i for all nodes $i = 1, \ldots, N$ of the closed network using Eq. (7.5).

STEP 2 For all $i = 1, \ldots, N$, compute the functions $F_i(k_i)$ using Eq. (7.61) or Eq. (7.63) (in the case of load-dependent service rates).

STEP 3 Compute the normalization constant $G(K)$ using Eq. (7.60).

STEP 4 Compute the state probabilities of the network using Eq. (7.59). From the marginal probabilities, which can be determined from the state probabilities using Eq. (7.16), all other required performance measures can be determined.

An example of the application of the Gordon–Newell theorem follows:

Example 7.5 Consider the closed queueing network shown in Fig. (7.15) with $N = 3$ nodes and $K = 3$ jobs. The queueing discipline at all nodes is FCFS. The routing probabilities are given as follows:

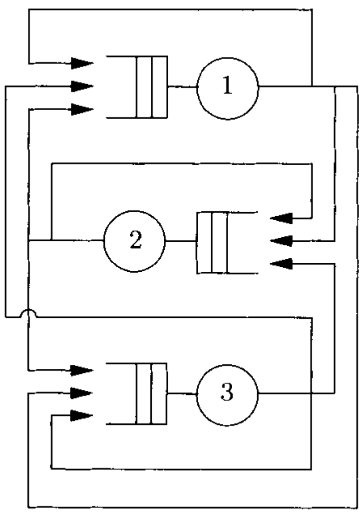

Fig. 7.15 Closed queueing network.

$$p_{11} = 0.6\,, \quad p_{21} = 0.2\,, \quad p_{31} = 0.4\,,$$
$$p_{12} = 0.3\,, \quad p_{22} = 0.3\,, \quad p_{32} = 0.1\,,$$

$$p_{13} = 0.1\,, \quad p_{23} = 0.5\,, \quad p_{33} = 0.5\,.$$

The service time at each node is exponentially distributed with the rates:

$$\mu_1 = 0.8\,\text{sec}^{-1}\,, \quad \mu_2 = 0.6\,\text{sec}^{-1}\,, \quad \mu_3 = 0.4\,\text{sec}^{-1}\,. \tag{7.70}$$

This network consists of

$$\binom{N+K-1}{N-1} = \underline{10}$$

states, namely,

$$\begin{array}{lllll} (3,0,0)\,, & (2,1,0)\,, & (2,0,1)\,, & (1,2,0)\,, & (1,1,1)\,, \\ (1,0,2)\,, & (0,3,0)\,, & (0,2,1)\,, & (0,1,2)\,, & (0,0,3)\,. \end{array}$$

We wish to compute the state probabilities using the Gordon–Newell theorem. For this we proceed in the following four steps:

STEP 1 Determine the visit ratios at each node using Eq. (7.5):

$$\begin{aligned} e_1 &= e_1p_{11} + e_2p_{21} + e_3p_{31} = \underline{1}\,, \\ e_2 &= e_1p_{12} + e_2p_{22} + e_3p_{32} = \underline{0.533}\,, \\ e_3 &= e_1p_{13} + e_2p_{23} + e_3p_{33} = \underline{0.733}\,. \end{aligned}$$

STEP 2 Determine the functions $F_i(k_i)$ for $i = 1, 2, 3$ using Eq. (7.61):

$$\begin{aligned} F_1(0) &= (e_1/\mu_1)^0 = \underline{1}\,, & F_1(1) &= (e_1/\mu_1)^1 = \underline{1.25}\,, \\ F_1(2) &= (e_1/\mu_1)^2 = \underline{1.5625}\,, & F_1(3) &= (e_1/\mu_1)^3 = \underline{1.953}\,, \end{aligned}$$

and correspondingly:

$$\begin{array}{llll} F_2(0) = \underline{1}\,, & F_2(1) = \underline{0.889}\,, & F_2(2) = \underline{0.790}\,, & F_2(3) = \underline{0.702}\,, \\ F_3(0) = \underline{1}\,, & F_3(1) = \underline{1.833}\,, & F_3(2) = \underline{3.361}\,, & F_3(3) = \underline{6.162}\,. \end{array}$$

STEP 3 Determine the normalization constant using Eq. (7.60):

$$\begin{aligned} G(3) = {} & F_1(3)F_2(0)F_3(0) + F_1(2)F_2(1)F_3(0) + F_1(2)F_2(0)F_3(1) \\ & + F_1(1)F_2(2)F_3(0) + F_1(1)F_2(1)F_3(1) + F_1(1)F_2(0)F_3(2) \\ & + F_1(0)F_2(3)F_3(0) + F_1(0)F_2(2)F_3(1) + F_1(0)F_2(1)F_3(2) \\ & + F_1(0)F_2(0)F_3(3) = \underline{24.733}\,. \end{aligned}$$

STEP 4 Determine the state probabilities using the Gordon–Newell theorem, Eq. (7.59):

$$\pi(3,0,0) = \frac{1}{G(3)} F_1(3)\cdot F_2(0)\cdot F_3(0) = \underline{0.079}\,,$$

$$\pi(2,1,0) = \frac{1}{G(3)} F_1(2)\cdot F_2(1)\cdot F_3(0) = \underline{0.056}\,.$$

In the same way we compute

$$\pi(2,0,1) = \underline{0.116}\,,\quad \pi(1,2,0) = \underline{0.040}\,,\quad \pi(1,1,1) = \underline{0.082}\,,\quad \pi(1,0,2) = \underline{0.170}\,,$$
$$\pi(0,3,0) = \underline{0.028}\,,\quad \pi(0,2,1) = \underline{0.058}\,,\quad \pi(0,1,2) = \underline{0.121}\,,\quad \pi(0,0,3) = \underline{0.249}\,.$$

Using Eq. (7.16), all marginal probabilities can now be determined:

$$\begin{aligned}
\pi_1(0) &= \pi(0,3,0) + \pi(0,2,1) + \pi(0,1,2) + \pi(0,0,3) = \underline{0.457}\,,\\
\pi_1(1) &= \pi(1,2,0) + \pi(1,1,1) + \pi(1,0,2) = \underline{0.292}\,,\\
\pi_1(2) &= \pi(2,1,0) + \pi(2,0,1) = \underline{0.172}\,,\\
\pi_1(3) &= \pi(3,0,0) = \underline{0.079}\,,
\end{aligned}$$

$$\begin{aligned}
\pi_2(0) &= \pi(2,0,1) + \pi(1,0,2) + \pi(0,0,3) + \pi(3,0,0) = \underline{0.614}\,,\\
\pi_2(1) &= \pi(2,1,0) + \pi(1,1,1) + \pi(0,1,2) = \underline{0.259}\,,\\
\pi_2(2) &= \pi(1,2,0) + \pi(0,2,1) = \underline{0.098}\,,\\
\pi_2(3) &= \pi(0,3,0) = \underline{0.028}\,,
\end{aligned}$$

$$\begin{aligned}
\pi_3(0) &= \pi(3,0,0) + \pi(2,1,0) + \pi(1,2,0) + \pi(0,3,0) = \underline{0.203}\,,\\
\pi_3(1) &= \pi(2,0,1) + \pi(1,1,1) + \pi(0,2,1) = \underline{0.257}\,,\\
\pi_3(2) &= \pi(1,0,2) + \pi(0,1,2) = \underline{0.291}\,,\\
\pi_3(3) &= \pi(0,0,3) = \underline{0.249}\,.
\end{aligned}$$

For the other performance measures, we get

- Utilization at each node, Eq. (7.19):

$$\rho_1 = 1 - \pi_1(0) = \underline{0.543}\,,\quad \rho_2 = \underline{0.386}\,,\quad \rho_3 = \underline{0.797}\,.$$

- Mean number of jobs at each node, Eq. (7.26):

$$\overline{K}_1 = \sum_{k=1}^{3} k\cdot \pi_1(k) = \underline{0.873}\,,\quad \overline{K}_2 = \underline{0.541},\quad \overline{K}_3 = \underline{1.585}\,.$$

- Throughputs at each node, Eq. (7.23):

$$\lambda_1 = m_1\rho_1\mu_1 = \underline{0.435}\,,\quad \lambda_2 = \underline{0.232}\,,\quad \lambda_3 = \underline{0.319}\,.$$

- Mean response time at each node, Eq. (7.43):

$$\overline{T}_1 = \frac{\overline{K}_1}{\lambda_1} = \underline{2.009}\,,\quad \overline{T}_2 = \underline{2.337}\,,\quad \overline{T}_3 = \underline{4.976}\,.$$

7.3.6 BCMP Networks

The results of Jackson and Gordon–Newell were extended by Baskett, Chandy, Muntz, and Palacios in their classic article [BCMP75], to queueing networks with several job classes, different queueing strategies, and generally distributed service times. The considered networks can be open, closed, or mixed.

Following [BrBa80], an allowed state in a queueing model without class switching is characterized by four conditions:

1. The number of jobs in each class at each node is always non-negative, i.e.,
$$k_{ir} \geq 0\,, \quad 1 \leq r \leq R\,, \quad 1 \leq i \leq N\,.$$

2. For all jobs the following condition must hold:
$$k_{ir} > 0 \quad \text{if there exists a way for class-}r\text{ jobs to node } i \text{ with a nonzero probability.}$$

3. For a closed network, the number of jobs in the network is given by
$$K = \sum_{r=1}^{R} \sum_{i=1}^{N} k_{ir}\,.$$

4. The sum of class-r jobs in the network is constant at any time, i.e.,
$$K_r = \sum_{i=1}^{N} k_{ir} = \text{const.} \quad 1 \leq r \leq R\,.$$

If class switching is allowed, then conditions 1–3 are fulfilled, but condition 4 can not be satisfied because the number of jobs within a class is no longer constant but depends on the time when the system is looked at and can have the values $k \in \{0, \ldots, K\}$. In order to avoid this situation, the concept of chains is introduced.

7.3.6.1 The Concept of Chains Consider the routing matrix $\mathbf{P} = [p_{ir,js}]$, $i, j = 1, \ldots, N$, and $r, s = 1, \ldots, R$ of a closed queueing network. The routing matrix $\mathbf{P}$ defines a finite-state DTMC whose states are pairs of form (i, r) (node number i, class index r). We call (i, r) a *node-class pair.* This state space can be partitioned into disjoint sets Γ_i, $i = 1, \ldots, U$:

$$\Gamma = \Gamma_1 + \Gamma_2 + \ldots + \Gamma_U\,,$$

where Γ_i is a closed communicating class of recurrent states of the DTMC.

With a possible relabeling of nodes, we get the routing matrix of Fig. 7.16, where submatrices P_i contains the transition probabilities in set Γ_i and each

$$\begin{pmatrix} \boxed{P_1} & & & \\ & \boxed{P_2} & & 0 \\ & 0 & \ddots & \\ & & & \boxed{P_U} \end{pmatrix}$$

Fig. 7.16 Modified routing matrix.

P_i is disjoint and assumed to be ergodic. Let chain C_i denote the set of job classes in Γ_i. Because of the disjointness of the chains C_i, it is impossible for a job to switch from one chain to another. If a job starts in a chain, it will never leave this chain. With the help of this partitioning technique, the number of jobs in each chain is always constant in closed networks. Therefore, Condition 4 for queueing networks is fulfilled for the chains. Muntz [Munt73] proved that a closed queueing network with U chains is equivalent to a closed queueing network with U job classes. If there is no class switching allowed, then the number of chains equals the number of classes. But if class switching is allowed, the number of chains is smaller than the number of classes. This means that the dimension of the population vector is reduced. An extension of the chains concept to open networks is possible.

The procedure to find the chains from a given solution matrix is the following:

STEP 1 Construct R sets that satisfy the following condition:

$$E_r := \left\{ s : \begin{array}{l} \text{node-class pair } (j,s) \text{ can be reached from} \\ \text{pair } (i,r) \text{ in a finite number of steps} \end{array} \right\},$$

$$1 \le r \le R, \quad 1 \le s \le R.$$

STEP 2 Eliminate all subsets and identical sets so that we have

$$U \le R \quad \text{sets to obtain the chains} \quad C_1, \ldots, C_U.$$

STEP 3 Compute the number of jobs in each chain:

$$K_q^* = \sum_{r \epsilon C_q} K_r, \quad 1 \le q \le U.$$

The set of all possible states in a closed multiclass network is given by the following binomial coefficient:

$$\prod_{q=1}^{U} \binom{N \cdot |C_q| + K_q^* - 1}{N \cdot |C_q| - 1},$$

where $|C_q|$ is the number of elements in C_q, $\quad 1 \le q \le U$.

For open networks, the visit ratios in a chain are given by

$$2e_{ir} = p_{0,ir} + \sum_{\substack{s \in C_q \\ j=1,\dots,N}} e_{js} p_{js,ir} \qquad \text{for} \quad \begin{array}{l} r \in C_q\,, \\ i = 1, \dots N\,, \\ 1 \le q \le U\,, \end{array} \tag{7.71}$$

and for closed networks we have

$$e_{ir} = \sum_{\substack{s \in C_q \\ j=1,\dots,N}} e_{js} p_{js,ir} \qquad \text{for} \quad \begin{array}{l} r \in C_q\,, \\ i = 1, \dots N\,, \\ 1 \le q \le U\,. \end{array} \tag{7.72}$$

Example 7.6 Let the following routing matrix $\mathbf{P}$ be given:

$$\mathbf{P} = \begin{array}{r} \\ (1,1) \\ (1,2) \\ (1,3) \\ (2,1) \\ (2,2) \\ (2,3) \\ (3,1) \\ (3,2) \\ (3,3) \end{array} \begin{array}{c} \begin{array}{ccccccccc} (1,1) & (1,2) & (1,3) & (2,1) & (2,2) & (2,3) & (3,1) & (3,2) & (3,3) \end{array} \\ \begin{pmatrix} 0 & 0.4 & 0 & 0 & 0.3 & 0 & 0 & 0.3 & 0 \\ 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0.7 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0.3 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0.7 \\ 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0 & 0 & 0.6 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

By using the chaining technique, we get

$$E_1 = \{1,2\}, \quad E_2 = \{1,2\}, \quad E_3 = \{3\},$$

meaning that $R = 3$ classes are reduced to $U = 2$ chains. By eliminating the subsets and identical sets, we get two chains:

$$C_1 = \{1,2\}, \quad C_2 = \{3\}.$$

Then the reorganized routing matrix $\mathbf{P}'$ has the form

$$\mathbf{P}' = \begin{array}{c} \\ (1,1) \\ (1,2) \\ (2,1) \\ (2,2) \\ (3,1) \\ (3,2) \\ (1,3) \\ (2,3) \\ (3,3) \end{array} \begin{array}{c} \begin{array}{ccccccccc} (1,1) & (1,2) & (2,1) & (2,2) & (3,1) & (3,2) & (1,3) & (2,3) & (3,3) \end{array} \\ \begin{pmatrix} 0 & 0.4 & 0 & 0.3 & 0 & 0.3 & 0 & 0 & 0 \\ 0.3 & 0 & 0 & 0 & 0 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0.3 & 0 & 0 & 0 & 0.7 & 0 & 0 & 0 & 0 \\ 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 & 0 \end{pmatrix} \end{array} .$$

If a job starts in one of the chains C_1 or C_2, then it cannot leave them.

Because of the switch from classes to chains, it is necessary to transfer class measures into chain measures [BrBa80] (chain measures are marked with a *):

- Number of visits: In a chain a job can reach different node-class pairs (i, j) (i: node index, j: class index) where

$$e_{iq}^* = \frac{\sum\limits_{r \in C_q} e_{ir}}{\sum\limits_{r \in C_q} e_{1r}} . \tag{7.73}$$

- The number of jobs per class is constant in a chain, but within the chain jobs can change their class. Therefore, if a job of chain q visits node i, it is impossible to know to which class it belongs because we consider a chain to be a single entity. If we make the transformation

$$\text{class} \quad \longrightarrow \quad \text{chain} \,,$$

the information about the class of a job is lost. Because different job classes have different service times, this missing information is exchanged by the scale factor α. For the service time in a chain we get

$$s_{iq}^* = \frac{1}{\mu_{iq}^*} = \sum_{r \in C_q} s_{ir} \cdot \alpha_{ir} \,, \tag{7.74}$$

$$\alpha_{ir} = \frac{e_{ir}}{\sum\limits_{s \in C_q} e_{is}} . \tag{7.75}$$

In Chapter 8 we introduce several algorithms to calculate the performance measures of single and multiclass queueing networks without class switching (such as mean value analysis or convolution). Using the concept of chains, it is possible to also use these algorithms for queueing networks with class switching. To do so, we proceed in the following steps:

STEP 1 Calculate the number of visits e_{ir} in the original network.

STEP 2 Determine the chains $C_1 \dots C_U$, and calculate the number of jobs K_q^* in each chain.

STEP 3 Compute the number of visits e_{iq}^* for each chain, Eq. (7.73).

STEP 4 Determine the scale factors α_{ir}, Eq. (7.75).

STEP 5 Calculate the service times s_{iq}^* for each chain, Eq. (7.74).

STEP 6 Derive the performance measures per chain [BrBa80] with one of the algorithms introduced later (mean value analysis, convolution, etc.).

STEP 7 Calculate the performance measures per class from the performance measures per chain:

$$\begin{aligned}
\overline{T}_{ir}(\mathbf{K}^*) &= s_{ir} \cdot \left(1 + \overline{K}_i(\mathbf{K}^* - \mathbf{1}_q)\right) , \quad r \in C_q \\
\lambda_{ir}(\mathbf{K}^*) &= \alpha_{ir} \cdot \lambda_{iq}^* , \\
\rho_{ir}(\mathbf{K}^*) &= s_{ir} \cdot \lambda_{ir}(\mathbf{K}^*) ,
\end{aligned}$$

where

$\mathbf{K}^* = (K_1^*, \dots, K_U^*)$:	Population vector containing the number of jobs in each chain,
$\mathbf{K}^* - \mathbf{1}_q$:	$\mathbf{K}^*$ with one job less in chain q,
$\overline{K}_i(\mathbf{K}^* - \mathbf{1}_q)$:	Mean number of jobs at node i if the number of jobs in the chains is given by $(\mathbf{K}^* - \mathbf{1}_q)$.

All other performance measures can easily be obtained using the formulae of Section 7.2.

In the following section we introduce the BCMP theorem, which is the basis of all analysis techniques to come. If class switching is not allowed in the network, then the BCMP theorem can be applied directly. In the case of class switching, it needs to be applied in combination with the concept of chains.

7.3.6.2 BCMP Theorem The theorems of Jackson and of Gordon and Newell have been extended by [BCMP75] to networks with several job classes and different service strategies and interarrival/service time distributions, and also to mixed networks that contain open and closed classes. The networks considered by BCMP must fulfill the following assumptions:

- Queueing discipline: The following disciplines are allowed at network nodes: FCFS, PS, LCFS-PR, IS (infinite server).

- Distribution of the service times: The service times of an FCFS node must be exponentially distributed and class-independent (i.e., $\mu_{i1} = \mu_{i2} = \ldots = \mu_{iR} = \mu_i$), while PS, LCFS-PR and IS nodes can have any kind of service time distribution with a rational Laplace transform. For the latter three queueing disciplines, the mean service time for different job classes can be different.

- Load-dependent service rates: The service rate of an FCFS node is only allowed to depend on the number of jobs at this node, whereas in a PS, LCFS-PR and IS node the service rate for a particular job class can also depend on the number of jobs of that class at the node but not on the number of jobs in another class.

- Arrival processes: In open networks two kinds of arrival processes can be distinguished from each other.

  Case 1: The arrival process is Poisson where all jobs arrive at the network from one source with an overall arrival rate λ, where λ can depend on the number of jobs in the network. The arriving jobs are distributed over the nodes in the network in accordance to the probability $p_{0,ir}$ where:

$$\sum_{i=1}^{N} \sum_{r=1}^{R} p_{0,ir} = 1\,.$$

  Case 2: The arrival process consists of U independent Poisson arrival streams where the U job sources are assigned to the U chains. The arrival rate λ_u from the uth source can be load dependent. A job arrives at the ith node with probability $p_{0,ir}$ so that

$$\sum_{\substack{r \in C_u \\ i=1,\ldots,N}} p_{0,ir} = 1\,, \quad \text{for all } u = 1,\ldots,U\,.$$

These assumptions lead to the four product-form node types and the local balance conditions for BCMP networks (see Section 7.3.2), that is,

Type 1:	-/M/m – FCFS	Type 2:	-/G/1 – PS
Type 3:	-/G/∞ (IS)	Type 4:	-/G/1 – LCFS PR

Note that we use -/M/m notation since we know that, in general, the arrival process to a node in a BCMP network will not be Poisson.

The BCMP Theorem says that networks with the characteristics just described have product-form solution:

For open, closed, and mixed queueing networks whose nodes consist only of the four described node types, the steady-state probabilities have the following product-form:

$$\pi(\mathbf{S}_1,\ldots,\mathbf{S}_N) = \frac{1}{G(\mathbf{K})} d(\mathbf{S}) \prod_{i=1}^{N} f_i(\mathbf{S}_i), \tag{7.76}$$

where,

$G(\mathbf{K})$ = the normalization constant,

$d(\mathbf{S})$ = a function of the number of jobs in the network, $\mathbf{S} = (\mathbf{S}_1,\ldots,\mathbf{S}_N)$, and

$$d(\mathbf{S}) = \begin{cases} \displaystyle\prod_{i=0}^{K(\mathbf{S})-1} \lambda(i)\,, & \text{open network with arrival process 1,} \\ \displaystyle\prod_{u=1}^{U} \prod_{i=0}^{K_u(\mathbf{S})-1} \lambda_u(i)\,, & \text{open network with arrival process 2,} \\ 1\,, & \text{closed networks.} \end{cases}$$

$f_i(\mathbf{S}_i)$ = a function which depends on the type and state of each node and

$$f_i(S_i) = \begin{cases} \displaystyle\left(\frac{1}{\mu_i}\right)^{k_i} \prod_{j=1}^{k_i} e_{is_{ij}}\,, & \text{Type 1,} \\ \displaystyle k_i! \prod_{r=1}^{R} \prod_{l=1}^{u_{ir}} \frac{1}{k_{irl}!} \left(\frac{e_{ir} A_{irl}}{\mu_{irl}}\right)^{k_{irl}}\,, & \text{Type 2,} \\ \displaystyle \prod_{r=1}^{R} \prod_{l=1}^{u_{ir}} \frac{1}{k_{irl}!} \left(\frac{e_{ir} A_{irl}}{\mu_{irl}}\right)^{k_{irl}}\,, & \text{Type 3,} \\ \displaystyle \prod_{j=1}^{k_i} \frac{e_{ir_j} A_{ir_j m_j}}{\mu_{ir_j m_j}}\,, & \text{Type 4.} \end{cases} \tag{7.77}$$

Variables in for Eq. (7.77) have the following meanings:

s_{ij}: Class of the job that is at the jth position in the FCFS queue.

μ_{irl}: Mean service rate in the lth phase ($l = 1,\ldots,u_{ir}$) in a Cox distribution (see Chapter 1).

u_{ir}: Maximum number of exponential phases.

A_{irl}: $\prod_{j=0}^{l-1} a_{irj}$, probability that a class-r job at the ith node reaches the lth service phase ($A_{ir1} = 1$ because of $a_{ir0} = 1$).

a_{irj}: Probability that a class-r job at the ith node moves to the $(j+1)$th phase.

k_{irl}: Number of class-r jobs in the lth phase of node i.

For the load-dependent case, $f_i(\mathbf{S}_i)$ is of the form

$$f_i(S_i) = \left(\frac{1}{\mu_i}\right)^{k_i} \prod_{j=1}^{k_i} e_{is_{ij}}.$$

Proof: The proof of this theorem is very complex and therefore only the basic idea is given here (for the complete proof see [Munt72]). In order to find a solution for the steady-state probabilities $\pi(\mathbf{S})$, the following global balance equations have to be solved:

$$\pi(\mathbf{S}) \begin{bmatrix} \text{state transition rate} \\ \text{from state } \mathbf{S} \end{bmatrix} = \sum_{\tilde{\mathbf{S}}} \pi(\tilde{\mathbf{S}}) \begin{bmatrix} \text{state transition rate from} \\ \text{state } \tilde{\mathbf{S}} \text{ to state } \mathbf{S} \end{bmatrix}, \tag{7.78}$$

with the normalization condition

$$\sum_{\mathbf{S}} \pi(\mathbf{S}) = 1. \tag{7.79}$$

Now we insert Eq. (7.76) into Eq. (7.78) to verify that Eq. (7.78) can be written as a system of balance equations that [Chan72] calls *local balance equations.* All local balance equations can be transformed into a system of $N-1$ independent equations. To get unambiguity for the solution the normalization condition, Eq. (7.79), has to be used.

Now we wish to give two simplified versions of the BCMP theorem for open and closed networks.

BCMP Version 1: For a *closed* queueing network fulfilling the assumptions of the BCMP theorem, the steady-state state probabilities have the form

$$\pi(\mathbf{S}_1, \ldots, \mathbf{S}_N) = \frac{1}{G(\mathbf{K})} \prod_{i=1}^{N} F_i(\mathbf{S}_i), \tag{7.80}$$

where the normalization constant is defined as

$$G(\mathbf{K}) = \sum_{\sum_{i=1}^{N} \mathbf{S}_i = \mathbf{K}} \prod_{i=1}^{N} F_i(\mathbf{S}_i), \tag{7.81}$$

and the function $F_i(\mathbf{S}_i)$ is given by

$$F_i(\mathbf{S}_i) = \begin{cases} k_i!\dfrac{1}{\beta_i(k_i)}\cdot\left(\dfrac{1}{\mu_i}\right)^{k_i}\cdot\displaystyle\prod_{r=1}^{R}\dfrac{1}{k_{ir}!}\mathrm{e}_{ir}^{k_{ir}}, & \text{Type 1,}\\ k_i!\displaystyle\prod_{r=1}^{R}\dfrac{1}{k_{ir}!}\cdot\left(\dfrac{e_{ir}}{\mu_{ir}}\right)^{k_{ir}}, & \text{Type 2,4,}\\ \displaystyle\prod_{r=1}^{R}\dfrac{1}{k_{ir}!}\cdot\left(\dfrac{e_{ir}}{\mu_{ir}}\right)^{k_{ir}}, & \text{Type 3.}\end{cases} \tag{7.82}$$

The quantity $k_i = \sum_{r=1}^{R} k_{ir}$ gives the overall number of jobs of all classes at node i. The visit ratios e_{ir} can be determined with Eq. (7.72), while the function $\beta_i(k_i)$ is given in Eq. (7.62).

For FCFS nodes ($m_i = 1$) with load-dependent service rates $\mu_i(j)$ we get

$$F_i(\mathbf{S}_i) = \frac{k_i!}{\prod_{j=1}^{k_i}\mu_i(j)}\prod_{r=1}^{R}\frac{1}{k_{ir}!}\mathrm{e}_{ir}^{k_{ir}}. \tag{7.83}$$

BCMP Version 2: For an *open* queueing network fulfilling the assumptions of the BCMP theorem and load-independent arrival and service rates, we have

$$\pi(k_1,\ldots,k_N) = \prod_{i=1}^{N}\pi_i(k_i), \tag{7.84}$$

where

$$\pi_i(k_i) = \begin{cases}(1-\rho_i)\rho_i^{k_i}, & \text{Type 1,2,4 } (m_i = 1),\\ \mathrm{e}^{-\rho_i}\dfrac{\rho_i^{k_i}}{k_i!}, & \text{Type 3,}\end{cases} \tag{7.85}$$

and

$$k_i = \sum_{i=i}^{R} k_{ir},$$

$$\rho_i = \sum_{r=1}^{R}\rho_{ir},$$

with

$$\rho_{ir} = \begin{cases}\lambda_r\dfrac{e_{ir}}{\mu_i}, & \text{Type 1 } (m_i = 1),\\ \lambda_r\dfrac{e_{ir}}{\mu_{ir}}, & \text{Type 2,3,4.}\end{cases} \tag{7.86}$$

Furthermore, we have

$$\overline{K}_{ir} = \frac{\rho_{ir}}{1-\rho_i}. \tag{7.87}$$

For Type 1 nodes with more than one service unit ($m_i > 1$), Eq. (6.26) can be used to compute the probabilities $\pi_i(k_i)$. For the existence of the steady-state probabilities $\pi_i(k_i)$, ergodicity condition ($\rho_i < 1$) has to be fulfilled for all i, $i = 1, \ldots, N$.

The algorithm to determine the performance measures using the BCMP theorem can now be given in the following five steps:

STEP 1 Compute the visit ratios e_{ir} for all $i = 1, \ldots, N$ and $r = 1, \ldots, R$ using Eq. (7.71).

STEP 2 Compute the utilization of each node using Eq. (7.86).

STEP 3 Compute the other performance measures with the equations given in Section 7.2.

STEP 4 Compute the marginal probabilities of the network using Eq. (7.85).

STEP 5 Compute the state probabilities using Eq. (7.84).

Example 7.7 Consider the open network given in Fig. 7.17 with $N = 3$ nodes and $R = 2$ job classes. The first node is of Type 2 and the second and third nodes are of Type 4. The service times are exponentially distributed with the rates

$$\mu_{11} = 8\,\text{sec}^{-1}, \quad \mu_{21} = 12\,\text{sec}^{-1}, \quad \mu_{31} = 16\,\text{sec}^{-1},$$
$$\mu_{12} = 24\,\text{sec}^{-1}, \quad \mu_{22} = 32\,\text{sec}^{-1}, \quad \mu_{32} = 36\,\text{sec}^{-1}.$$

The interarrival times are also exponentially distributed with the rates

$$\lambda_1 = \lambda_2 = 1 \;\; \text{job/sec}.$$

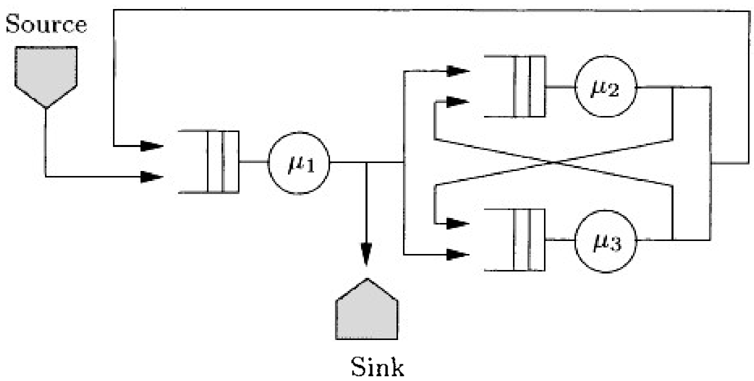

Fig. 7.17 Open queueing network.

The routing probabilities are given by

$$
\begin{array}{llll}
p_{0,11} = 1\,, & p_{21,11} = 0.6\,, & p_{0,12} = 1\,, & p_{22,12} = 0.7\,, \\
p_{11,21} = 0.4\,, & p_{21,31} = 0.4\,, & p_{12,22} = 0.3\,, & p_{22,32} = 0.3\,, \\
p_{11,31} = 0.3\,, & p_{31,11} = 0.5\,, & p_{12,32} = 0.6\,, & p_{32,12} = 0.4\,, \\
p_{11,0} = 0.3\,, & p_{31,21} = 0.5\,, & p_{12,0} = 0.1\,, & p_{32,22} = 0.6\,,
\end{array}
$$

which means that class switching is not allowed in the network. We wish to compute the probability for the state $(k_1, k_2, k_3) = (3, 2, 1)$ by using the BCMP theorem, Eq. (7.84).

STEP 1 Compute the visit ratios e_{ir} for all $i = 1, \ldots, N$ and $r = 1, \ldots, R$ using Eq. (7.71).

$$
\begin{aligned}
e_{11} &= p_{0,11} + e_{11}p_{11,11} + e_{21}p_{21,11} + e_{31}p_{31,11} = \underline{3.333}\,, \\
e_{21} &= p_{0,21} + e_{11}p_{11,21} + e_{21}p_{21,21} + e_{31}p_{31,21} = \underline{2.292}\,, \\
e_{31} &= p_{0,31} + e_{11}p_{11,31} + e_{21}p_{21,31} + e_{31}p_{31,31} = \underline{1.917}\,.
\end{aligned}
$$

In the same way we get

$$
e_{12} = \underline{10}, \quad e_{22} = \underline{8.049}, \quad e_{32} = \underline{8.415}.
$$

STEP 2 Compute the utilization of each node using Eq. (7.86):

$$
\rho_1 = \lambda_1 \frac{e_{11}}{\mu_{11}} + \lambda_2 \frac{e_{12}}{\mu_{12}} = \rho_{11} + \rho_{12} = \underline{0.833}\,,
$$

$$
\rho_2 = \lambda_1 \frac{e_{21}}{\mu_{21}} + \lambda_2 \frac{e_{22}}{\mu_{22}} = \rho_{21} + \rho_{22} = \underline{0.442}\,,
$$

$$
\rho_3 = \lambda_1 \frac{e_{31}}{\mu_{31}} + \lambda_2 \frac{e_{32}}{\mu_{32}} = \rho_{31} + \rho_{32} = \underline{0.354}\,.
$$

STEP 3 Compute the other performance measures of the network. In our case we use Eq. (7.87) to determine the mean number of jobs at each node:

$$
\overline{K}_{11} = \frac{\rho_{11}}{1-\rho_1} = \underline{2.5}, \quad \overline{K}_{21} = \frac{\rho_{21}}{1-\rho_2} = \underline{0.342}, \quad \overline{K}_{31} = \frac{\rho_{31}}{1-\rho_3} = \underline{0.186},
$$

$$
\overline{K}_{12} = \underline{2.5}, \quad \overline{K}_{22} = \underline{0.5}, \quad \overline{K}_{32} = \underline{0.362}.
$$

STEP 4 Determine the marginal probabilities using Eq. (7.85):

$$
\begin{aligned}
&\pi_1(3) = (1-\rho_1)\rho_1^3 = \underline{0.0965}, \quad \pi_2(2) = (1-\rho_2)\rho_2^2 = \underline{0.1093}, \\
&\pi_3(1) = (1-\rho_3)\rho_3 = \underline{0.2287}.
\end{aligned}
$$

STEP 5 Compute the state probabilities for the network using Eq. (7.84):

$$\pi(3,2,1) = \pi_1(3) \cdot \pi_2(2) \cdot \pi_3(1) = \underline{0.00241}\,.$$

An example of the BCMP theorem for closed networks, Eq. (7.80), is not given in this chapter. As shown in Example 7.7, the direct use of the BCMP theorem will require that all states of the network have to be considered in order to compute the normalization constant. This is a very complex procedure and only suitable for small networks because for bigger networks the number of possible states in the network becomes prohibitively large. In Chapter 8 we provide efficient algorithms to analyze closed product-form queueing networks. In these algorithms the states of the queueing network are not explicitly involved in the computation and therefore these algorithms provide much shorter computation time.

Several researchers have extended the class of product-form networks to nodes of the following types:

- In [Spir79] the SIRO (service **i**n **r**andom **o**rder) is examined and it is shown that -/M/1-SIRO nodes fulfill the local balance property and therefore have a product-form solution.

- In [Noet79] the LBPS (**l**ast **b**atch **p**rocessor **s**haring) strategy is introduced and it is show that -/M/1-LBPS node types have product-form solutions. In this strategy the processor is assigned between the two last batch jobs. If the last batch consists only of one job, then we get the LCFS-PR strategy and if the batch consists of all jobs, we get PS.

- In [ChMa83] the WEIRDP-strategy (**weird**, **p**arameterized strategy) is considered, where the first job in the queue is assigned $100 \cdot p\,\%$ of the processor and the rest of the jobs are assigned $100 \cdot (1-p)\,\%$ of the processor. It is shown that -/M/1-WEIRDP nodes have product-form solution.

- In [CHT77] it is shown that -/G/1-PS, -/G/∞-IS, and -/G/1-LCFS-PR nodes with arbitrary differentiable service time distribution also have product-form solution.

Furthermore, in [Tows80] and [Krze87], the class of product-form networks is extended to networks where the probability to enter a particular node depends on the number of jobs at that node. In this case, we have so-called *load-dependent routing probabilities.*

Problem 7.2 Consider an open queueing network with $N = 3$ nodes, FCFS queueing discipline, and exponentially distributed service times with the mean values

$$\frac{1}{\mu_1} = 0.08\,\text{sec}, \qquad \frac{1}{\mu_2} = 0.06\,\text{sec}, \qquad \frac{1}{\mu_3} = 0.04\,\text{sec}\,.$$

Only at the first node do jobs arrive from outside with exponentially distributed interarrival times and the rate $\lambda_{01} = 4$ jobs/sec. Node 1 is a multiple server node with $m_1 = 2$ server, nodes 2 and 3 are single server nodes. The routing probabilities are given as follows:

$$\begin{aligned} p_{11} &= 0.2, & p_{21} &= 1, & p_{31} &= 0.5, \\ p_{12} &= 0.4, & p_{30} &= 0.5, \\ p_{13} &= 0.4. \end{aligned}$$

(a) Draw the queueing network.

(b) Determine the steady-state probability for the state $\mathbf{k} = (4, 3, 2)$.

(c) Determine all performance measures.

Note: Use the Jackson's theorem.

Problem 7.3 Determine the CPU utilization and other performance measures of a central server model with $N = 3$ nodes and $K = 4$ jobs. Each node has only one server and the queueing discipline at each node is FCFS. The exponentially distributed service times have the respective means

$$\frac{1}{\mu_1} = 2 \text{ msec}, \quad \frac{1}{\mu_2} = 5 \text{ msec}, \quad \frac{1}{\mu_3} = 5 \text{ msec},$$

and the routing probabilities are

$$p_{11} = 0.3, \quad p_{12} = 0.5, \quad p_{13} = 0.2, \quad p_{21} = p_{31} = 1.$$

Problem 7.4 Consider a closed queueing network with $K = 3$ nodes, $N = 3$ jobs, and the service discipline FCFS. The first node has $m_1 = 2$ servers and the other two nodes have one server each. The routing probabilities are given by

$$\begin{aligned} p_{11} &= 0.6, & p_{21} &= 0.5, & p_{31} &= 0.4, \\ p_{12} &= 0.3, & p_{22} &= 0.0, & p_{32} &= 0.6, \\ p_{12} &= 0.1, & p_{23} &= 0.5, & p_{33} &= 0.0. \end{aligned}$$

The service times are exponentially distributed with the rates

$$\mu_1 = 0.4 \,\text{sec}^{-1}, \quad \mu_2 = 0.6 \,\text{sec}^{-1}, \quad \mu_3 = 0.3 \,\text{sec}^{-1}.$$

(a) Draw the queueing network.

(b) What is the steady-state probability that there are two jobs at node 2?

(c) Determine all performance measures.

(d) Solve this problem also with the local and global balance equations respectively and compare the solution complexity for the two different methods.

Note: Use the Gordon–Newell theorem for parts (*a*), (*b*), and (*c*).

Problem 7.5 Consider a closed queueing network with $N = 3$ nodes, $K = 3$ jobs, and exponentially distributed service times with the rates

$$\mu_1 = 0.72\,\text{sec}^{-1}, \quad \mu_2 = 0.64\,\text{sec}^{-1}, \quad \mu_3 = 1\,\text{sec}^{-1},$$

and the routing probabilities

$$p_{31} = 0.4, \quad p_{32} = 0.6, \quad p_{13} = p_{23} = 1.$$

Determine all performance measures.

Problem 7.6 Consider the following routing matrix $\mathbf{P}$:

P	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(2,4)
(1,1)	0.5	0	0	0	0.25	0	0.25	0
(1,2)	0	0.5	0	0.5	0	0	0	0
(1,3)	0.5	0	0	0	0	0	0.5	0
(1,4)	0	0	0	0.5	0	0.25	0	0.25
(2,1)	0.5	0	0	0	0.5	0	0	0
(2,2)	0	0.5	0	0	0	0.5	0	0
(2,3)	0	0	1	0	0	0	0	0
(2,4)	0	0	0	1	0	0	0	0

At the beginning, the jobs are distributed as follows over the four different classes:

$$K_1' = K_2' = K_3' = K_4' = 5.$$

(a) Determine the disjoint chains.

(b) Determine the number of states in the CTMC underlying the network.

(c) Determine the visit ratios.

Problem 7.7 Consider an open network with $N = 2$ nodes and $R = 2$ job classes. Node 1 is of Type 2 and node 2 of Type 4. The service rates are

$$\mu_{11} = 4\,\text{sec}^{-1}, \quad \mu_{12} = 5\,\text{sec}^{-1}, \quad \mu_{21} = 6\,\text{sec}^{-1}, \quad \mu_{22} = 2\,\text{sec}^{-1},$$

and the arrival rates per class are

$$\lambda_1 = \lambda_2 = 1\,\text{sec}^{-1}.$$

The routing probabilities are given as

$$p_{0,11} = p_{0,12} = 1, \quad p_{21,11} = p_{22,12} = 1,$$

$$p_{11,21} = 0.6, \quad p_{12,22} = 0.4 \quad p_{11,0} = 0.4, \quad p_{12,0} = 0.6.$$

With the BCMP theorem, Version 2, determine

(a) All visit ratios.

(b) The utilizations of node 1 and node 2.

(c) The steady-state probability for state (2,1).

This Page Intentionally Left Blank