Chapter 15
Decomposition-Based Queueing Network Analysis with FiFiQueues

Ramin Sadre, Boudewijn R. Haverkort

Abstract In this chapter we present an overview of decomposition-based analysis techniques for large open queueing networks. We present a general decomposition-based solution framework, without referring to any particular model class, and propose a general fixed-point iterative solution method for it. We concretize this framework by describing the well-known QNA method, as proposed by Whitt in the early 1980s, in that context, before describing our FiFiQueues approach. FiFiQueues allows for the efficient analysis of large open queueing networks of which the inter-arrival and service time distributions are of phase-type; individual queues, all with single servers, can have bounded or unbounded buffers. Next to an extensive evaluation with generally very favorable results for FiFiQueues, we also present a theorem on the existence of a fixed-point solution for FiFiQueues.

15.1 Introduction

In this chapter we present an overview of the FiFiQueues method (and supporting tool) to evaluate large open queueing networks with non-Poissonian traffic streams

Ramin Sadre
Centre for Telematics and Information Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
P.O. Box 217, 7500 AE Enschede, The Netherlands,
e-mail: r.sadre@utwente.nl

Boudewijn R. Haverkort
Centre for Telematics and Information Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
P.O. Box 217, 7500 AE Enschede, The Netherlands,
e-mail: b.r.h.m.haverkort@utwente.nl;
Embedded Systems Institute
P.O. Box 513, 5600 MB Eindhoven, Netherlands,
e-mail: boudewijn.haverkort@esi.nl

and non-exponential services. FiFiQueues is an example of a decomposition-based queueing network evaluation approach, in which the overall evaluation is broken into per-queue evaluations, thus making the method highly scalable.

The earliest work on open networks of queues has probably been reported by Jackson [20]. The so-called Jackson queueing networks (JQNs) allow for the analysis of open networks of M|M|1 queues, in which jobs are routed according to fixed probabilities. The external arrival process forms a Poisson process; arrivals may be spread over more than one queue. Departures from the queueing network are also possible.

In the mid 1970s, Kühn developed an approximate evaluation approach for an extended class of models [27], including non-Poissonian arrivals, as well as service times that followed other than exponential distributions. As an extension of this approach, Whitt proposed the QNA method in the early 1980s [49, 50]; QNA can be seen as a full-fledged approach to evaluate networks of G|G|1 queues approximately. Since our FiFiQueues approach can be regarded as an extension of QNA, and still relies on some of the assumptions made in QNA, we concisely present QNA in Section 15.3.

We are not the only researchers who have worked on extensions of QNA, nor are the extensions of QNA that we describe here the only possible extensions. Schuba et al. [46] reported on work involving the inclusion of multicast communication using routing trees (instead of the usual routing chains). Heindl et al. proposed decomposition-based analysis techniques taking into account correlations in the traffic streams between the queueing nodes, e.g., by using MAPs and MMPPs as traffic descriptors, cf. [17, 19, 20, 21]. Kim et al. [24] proposed an extension of QNA to include correlations in the traffic streams. For two small networks that are studied in detail (with 2 and 3 nodes resp.) better results than with standard QNA are obtained. The question how well the method scales to larger and more complex queueing networks remains open. Finally, in 1990 Harrison and Nguyen proposed the QNET approach [12] which, however, appears impractical for large queueing networks. A simplification of QNET, called ΠNET (described in the same paper) appears more practical; however, its approach is very similar to that of (standard) QNA.

The aim of this chapter is to present in detail the complete set of extensions we have proposed, and that led to the approach now known as FiFiQueues. FiFiQueues extends QNA in two ways: first, it extends the model class, and secondly, it removes a number of approximation steps from it. In particular, we do not address general G|G|1 queues, but allow instead for both PH|PH|1 as well as PH|PH|1|K queues. That is, we allow for phase-type distributions as inter-arrival and service-times, but at the same time also allow for finite- and infinite-buffer queues. This choice has two implications. The restriction to phase-type distributions allows us to use exact analysis algorithms for the per-queue evaluations, e.g., based on matrix-geometric methods. Secondly, the introduction of finite queues allows us to model queueing networks with losses, which has a severe impact on the solution of the traffic equations and forces us to follow a fixed-point iterative algorithm to solve them.

The chapter is further organized as follows. In Section 15.3 we summarize the QNA method. In Section 15.4 we present the FiFiQueues algorithm and its implementation in an integrated tool. We then present a large number of cases to validate both basic FiFiQueues and its extensions (against simulation results) in Section 15.5. Finally, Section 15.6 presents some conclusions. To keep the chapter self-contained, we added appendices on Jackson queueing networks (Section 15.7), on Markovian arrival processes, phase-type distributions and quasi-birth-death processes (Section 15.8), as well as a proof of the existence of a fixed point for the models we study (Section 15.9).

We finally remark that we already worked on some further extensions on FiFiQueues. We extended our approach to also deal with closed queueing networks, as published in [44]. Furthermore, we developed extensions that deal with correlations in traffic streams, as well as with higher moments [40].

15.2 The decomposition approach

Sketch of the idea

A common approach to evaluate the performance of communication systems is to construct and analyze a large monolithic model, often via an underlying state-space-based representation (typically a Markov chain). However, analysis methods relying on an analysis of such a large state space usually suffer from the state space explosion phenomenon: If two models A and B with a resp. b states are composed to a new "product model" $A \times B$, this model has potentially $a \cdot b$ states (this assumes that there are no mutually exclusive states). For large systems models, the number of states quickly grows beyond what can be practically handled.

The decomposition approach aims to reduce the complexity of the analysis by decomposing the system into smaller components that are analyzed more or less independently, thus avoiding the analysis of the overall full state space. The basic idea is the following: If we have two submodels A and B, with a resp. b possible states, we avoid to construct and analyze the full "product model" $A \times B$. Instead we do the following:

1. We assume that the system has the structure $B(A)$ instead of $A \times B$, i.e., that in the resulting composition the submodel B depends on A but not vice versa.
2. Based on that assumption, we analyze model A independently of B and summarize its behavior in some so-called *descriptor* d_A.
3. The descriptor d_A is used to parameterize model B and we analyze the new model $B(d_A)$ with $b(d_A)$ states instead of $B(A)$ with $a \cdot b$ states. Hence, the decomposition approach reduces the number of states to analyze from $a \cdot b$ to $a + b(d_A)$.
4. Now, we know the behavior of A combined with B. The global behavior of the system can then be derived.

Of course, this approach only makes sense if $B(d_A)$ has fewer states than $B(A)$. In general, this requires that the descriptor d_A is only an approximation to A and, hence, the decomposition approach only provides an approximation to the original model.

In the context of queueing networks, the submodels are naturally equivalent to the individual queueing stations and the descriptor represents the inter-station traffic. For example, in a tandem queueing network with two stations A and B, the descriptor d_A is obtained by analyzing station A and actually is a description of the traffic stream that departs from station A and arrives at station B. Hence, we will often call d_A a *traffic descriptor* in the following.

Open questions

The approach described above leaves several open questions:

1. What do the traffic descriptors look like?
2. How are more complex systems analyzed? Note that in the example above, the assumed structure $B(A)$ would basically restrict the analysis to tandem queueing networks.
3. How are the individual stations analyzed?

These questions are addressed by various decomposition-based analysis methods in different ways, thus leading to different model classes. When we describe the Queueing Network Analyzer (Section 15.3), FiFiQueues (Section 15.4), and the analysis of Jackson queueing networks (Section 15.7), we have to address these three questions for each method separately. However, since we focus on the analysis of open queueing networks with feedback, we can already give some general answers to question 2 and 3 which are true for all methods.

The analysis of complex networks

In an open queueing network with N queueing stations, the traffic descriptor $desc_{i,j}$ describes the traffic stream from queueing station i to station j, with $1 \leq i, j \leq N$. The outside world is represented by a "virtual" station *ext*, hence, we denote the traffic arriving from outside to station i as $desc_{ext,i}$, and the traffic leaving the network from i as $desc_{i,ext}$. We rely on the fixed-point iteration algorithm presented in [14, 48] to analyze such networks:

```
 1 initialize all traffic descriptors desc_{i,j}^(0):
 2     set desc_{ij}^(0) to the null value if i ≠ ext
 3     set desc_{ij}^(0) to the specified value if i = ext
 4 n := 0
 5 do
 6     n := n + 1
 7     analyze each queueing station i with arrival traffic desc_{k,i}^(n), 1 ≤ k ≤ N,
 8     and compute departing traffic desc_{i,j}^(n), 1 ≤ j ≤ N.
 9 while dist(desc^(n), desc^(n−1)) > ε
10 compute network-wide performance results
```

In each iteration the queueing stations are analyzed using the available descriptions of the traffic arriving at the stations (line 7). The analysis allows to compute station-related performance measures, such as the mean queue length, and, more important, the description of the traffic leaving the stations (line 8). In this way, a new set of traffic descriptors $desc^{(n)} = \{desc_{i,j}^{(n)} | i, j\}$ is computed in each iteration.

When the algorithm starts only the descriptions of the traffic arriving from outside are known (they are part of the model specification). Hence, all other descriptors are initially set to the *null* value in line 2 and have to be ignored in line 7 until a first approximation is available.

The algorithm stops when the distance $dist(desc^{(n-1)}, desc^{(n)})$ between two successive sets of descriptors is smaller than or equal a given threshold ε (line 9). Once all traffic descriptors are known, network-wide performance results can be computed in line 10.

The analysis of individual stations

For the analysis of the single stations (in line 7 and 8 of the iteration algorithm), we define that a station specification consists of two components:

1. a queue with finite or infinite capacity,
2. one or more service entities that serve the jobs (served jobs leave the queue),

and two policies:

1. a policy that handles incoming jobs if the queue is full (only for finite queues),
2. a scheduling policy that describes how the service stations fetch new jobs from the queue.

Such queueing stations can be analyzed by different approaches. Since most analysis methods for queueing processes require that a queueing station has exactly *one* arrival traffic descriptor and *one* traffic departure descriptor, a *traffic merging* (or *traffic superpositioning*) and a *traffic splitting* step are required. The traffic merging step merges for a station its arrival descriptors into a single overall arrival descriptor whereas the traffic splitting step splits the overall departure descriptor into the required number of departure descriptors. Thus, every time when the fixed-point iteration algorithm analyzes a queueing station we have to perform the following

steps (as part of steps 7 and 8 in the algorithm above, and illustrated in Figure 15.1 below):

1. merge the incoming traffic;
2. analyze the service operation;
3. split the departure traffic.

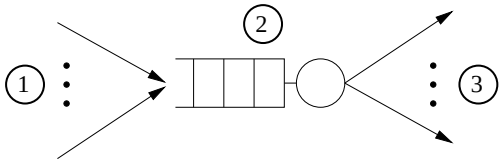


Fig. 15.1: The three key operations to be performed on a traffic stream: merging, queueing and splitting

Notice that for Jackson queueing networks (cf. Section 15.7), these three steps are extremely simple, hence, they are not often distinguished explicitly. Furthermore, for JQNs there appears to be no need for a fixed-point iteration. However, this is only partly true. One can argue that an iterative method to solve the traffic equations in JQNs (like Gauss-Seidel iterations) in fact forms a fixed-point iteration in itself. The distinguishing feature is then that for JQNs, no queueing analysis takes place within the fixed-point computation (only afterwards), whereas, in general, decomposition-based methods do require the intertwining of fixed-point iteration steps and queueing analysis steps, as will become clear in the following sections.

15.3 Whitt's Queueing Network Analyzer

In the early 1980s, Whitt presented the Queueing Network Analyzer (QNA) [49, 50], a software package developed at Bell Laboratories for the approximate analysis of open queueing networks. Unlike prior approaches which were based on Markovian models, QNA allows for the analysis of open queueing networks where the external arrival processes need not be Poissonian and the service times need not be negative exponentially distributed. Additionally, QNA is able to perform the analysis fast: due to the involved approximations and assumptions, the network traffic analysis is, in essence, reduced to the solution of a set of linear equations, comparable to those in JQNs (cf. Section 15.7).

In the following, we will give an overview of the functionality of QNA. The structure of our presentation slightly differs from Whitt's original paper [49], however, it follows the presentation of JQNs in Section 15.7.

15.3.1 Model class

QNA allows for the approximate analysis of open queueing networks fed by external arrival processes, in which the routing takes place according to fixed probabilities (like in JQNs). The nodes are GI|G|m multiserver queues without capacity constraints and with the FCFS service discipline. The external arrival processes as well as the service processes of the nodes are described by the first and the second moment of the inter-arrival, resp. service time distributions. The QNA approach allows for the separate analysis of the nodes, hence, QNA is well scalable to larger networks.

QNA's model class includes three features which we will not describe in detail in the following. First, QNA is able to analyze networks with multiple classes of customers, and secondly, networks with immediate feedback are allowed. Both features are "implemented" by adding a pre-processing and post-processing phase to the core QNA algorithms, that is, QNA treats multiple visits of a single job to one queue as one longer visit, and multiple classes are treated as one class with multimodal service times. The third feature, the customer multiplication factor of a node, only requires small modifications in the service operation equations. Although these features are interesting as such, they have not been implemented for FiFiQueues, however, also in that context they could be added via appropriate pre- and post-processing phases.

15.3.2 Traffic descriptors

The external arrival processes are specified by the first and second moment of the inter-arrival times. In fact, this representation is also applied to the traffic streams between the nodes. More specifically, QNA uses the traffic descriptor $\langle \lambda, c^2 \rangle$ to describe a traffic stream where λ is the arrival rate and c^2 is the squared coefficient of variation of the inter-arrival time.

Clearly, this allows the representation of non-Poissonian processes. However, neither higher moments nor correlations of the arrival stream are considered, which may influence the quality of the analysis. QNA employs fine-tuned heuristics deduced from simulation studies to reduce the errors introduced by this simplification.

15.3.3 Superposition of traffic streams

To merge n traffic streams specified by $\langle \lambda_1, c_1^2 \rangle, \ldots, \langle \lambda_n, c_n^2 \rangle$ into one traffic stream $\langle \lambda, c^2 \rangle$, QNA first computes the total arrival rate which is simply given by

$$\lambda = \sum_{i=1}^{n} \lambda_i.$$

QNA's efficiency is based on the fact that it computes the traffic descriptors from linear systems of equations. The above expression for λ is clearly linear in λ_i. For c^2 a linear equation can be found, too, by the asymptotic approximation method (AS):

$$c_{AS}^2 = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda} c_i^2.$$

However, the asymptotic method does not work well for a wide range of cases. It is therefore combined with the stationary-interval method (SI), resulting in the following hybrid approximation:

$$c^2 = w \cdot c_{AS}^2 + (1-w) \cdot c_{SI}^2.$$

The stationary-interval method does not provide a linear expression for c_{SI}^2, but experiments have shown that setting c_{SI}^2 to 1 (in the expression above) increases the average error only by 1 percent, so that we obtain

$$c^2 = w \cdot c_{AS}^2 + (1-w).$$

Simulations have shown that the above approximations do impact the quality of the analysis of a node which takes the merged traffic stream as input. To improve the results, QNA respects the utilization ρ of the node in the computation of the factor w. With $\rho = \lambda/\mu$ (where μ is the service rate of the queueing station), QNA sets

$$w = \left[1 + 4(1-\rho)^2(v-1)\right]^{-1} \text{ with } v = \left(\sum_{i=1}^{n} \left(\frac{\lambda_i}{\lambda} \right)^2 \right)^{-1}.$$

15.3.4 Splitting traffic streams

When splitting, QNA assumes that the involved processes are renewal processes. Under this assumption, an exact solution is available. For n splitting probabilities $p_1, \ldots, p_n$ and the traffic stream $\langle \lambda, c^2 \rangle$, we obtain the splitted streams

$$\langle \lambda_1, c_1^2 \rangle, \ldots, \langle \lambda_n, c_n^2 \rangle,$$

with

$$\lambda_i = p_i \cdot \lambda, \quad \text{and} \quad c_i^2 = p_i \cdot c^2 + (1-p_i), \qquad i = 1, \ldots, n.$$

15.3.5 Servicing jobs

Network nodes are analyzed as GI|G|m queues. Let $\langle \lambda_A, c_A^2 \rangle$ be the arrival traffic descriptor of the node and m the number of service entities. The service process is specified by the service rate μ and by the squared coefficient of variation c_S^2 of the service time distribution. We require the stability of all stations, i.e., $\lambda_A < \mu$. How does QNA compute the departure descriptor $\langle \lambda_D, c_D^2 \rangle$?

Since the queues are stable and have infinite capacity, no losses occur and we clearly have $\lambda_D = \lambda_A$. To compute c_D^2, Whitt combines Marshall's formula [33] with other approximations to obtain

$$c_D^2 = 1 + (1-\rho^2)(c_A^2 - 1) + \frac{\rho^2}{\sqrt{m}}(c_S^2 - 1). \tag{15.1}$$

The involved approximations may lead to large errors when c_S^2 is small, thus QNA uses the following extension of the above formula:

$$c_D^2 = 1 + (1-\rho^2)(c_A^2 - 1) + \frac{\rho^2}{\sqrt{m}}(\max\{c_S^2, 0.2\} - 1). \tag{15.2}$$

Note again the linearity of the expressions for λ_D and c_D^2 in the arrival traffic $\langle \lambda_A, c_A^2 \rangle$.

15.3.6 Node performance

QNA is able to compute results for the first and second moment of the waiting time W and the queue length N. Due to the complexity of the involved approximations, we limit our presentation only to the simplest one, i.e., the computation of $\mathrm{E}[W]$ in the case of single-server GI|G|1 queues. The required derivations for the other quantities can be found in [49, Eq. (46)–(71)]. For given arrival traffic $\langle \lambda_A, c_A^2 \rangle$, service descriptor $\langle \mu, c_S^2 \rangle$ and utilization ρ, $\mathrm{E}[W]$ is approximated as

$$\mathrm{E}[W] = \frac{\rho}{2(1-\rho)\mu}(c_A^2 + c_S^2)\, g(\rho, c_A^2, c_S^2), \tag{15.3}$$

where the function ρ is defined as

$$g(\rho, c_A^2, c_S^2) = \begin{cases} \exp\left(\frac{2(1-\rho)(1-c_A^2)^2}{3\rho(c_A^2+c_S^2)}\right), & c_A^2 < 1, \\ 1, & c_A^2 \geq 1. \end{cases}$$

Note that Equation (15.3) is exact for $c_A^2 = 1$, i.e., in the case of an M|G|1 queue. When $c_A^2 < 1$, it is equivalent to the Krämer and Langenbach-Belz approximation [26].

15.3.7 Network-wide performance

The results presented for network performance measures in Jackson queueing networks (see Section 15.7) can also be applied here, providing expressions for $\mathrm{E}[V_i]$, $\mathrm{E}[T_i]$, $\mathrm{E}[T_{total}]$ and $\mathrm{E}[N_{total}]$. Additionally, Whitt developed approximations for the variances of the above-stated measures [49, Eq. (80)–(84)].

15.3.8 Complexity

In the above sections, we have repeatedly pointed out the linearity of the employed equations for the three traffic operations merging, splitting, and service. In fact, QNA exploits this linearity to efficiently evaluate the queueing network.

First, for the arrival rates of the traffic streams the system of equations derived for JQNs is also valid for QNA. Let $\left\langle \lambda_{A,i}, c^2_{A,i} \right\rangle$ be the traffic arriving at node i, $\left\langle \lambda_{D,i}, c^2_{D,i} \right\rangle$ the traffic leaving this node, and $\left\langle \lambda_{ext,i}, c^2_{ext,i} \right\rangle$ the external traffic. If $\Gamma = (r_{ij})$ is the routing matrix, the following traffic equation holds for each node $i = 1, \ldots, n$ of the network:

$$\lambda_{A,i} = \lambda_{ext,i} + \sum_{j=1}^{n} \lambda_{D,j} \cdot r_{ji}. \tag{15.4}$$

Again, QNA's model class implies $\lambda_{D,i} = \lambda_{A,i}$ and the traffic equations form a system of linear equations which can be expressed in vector/matrix notation as

$$\lambda_{\mathrm{A}} = \lambda_{\mathrm{ext}}(\mathrm{I} - \Gamma)^{-1}.$$

For the squared coefficients of variation of the traffic streams a system of equations can be set up, too. The synthesis of the superposition and the splitting operations yields

$$c^2_{A,i} = (1 - w_i) + w_i \left(p_{ext,j} c^2_{ext,i} + \sum_{j=1}^{n} p_{j,i}(r_{ji} c^2_{D,j} + 1 - r_{ji}) \right),$$

where $p_{j,i} = \lambda_{D,j} r_{j,i} / \lambda_{A,i}$ is the fraction of traffic arriving from node j to node i and $p_{ext,j} = \lambda_{ext,i} / \lambda_{A,i}$ is the fraction of external traffic arriving to node i. Finally, if we include the result of the service operation we obtain the following system of linear equations

$$\begin{aligned} c^2_{A,i} = (1 - w_i) + w_i \{ p_{ext,j} c^2_{ext,i} \; + \; & \sum_{j=1}^{n} p_{j,i}(r_{ji}(1 + (1 - \rho_i^2)(c^2_{A,j} - 1) \\ & + \frac{\rho_i^2}{\sqrt{m_i}}(\max(c^2_{S,i}, 0.2) - 1)) + 1 - r_{ji}) \}. \end{aligned} \tag{15.5}$$

Using the equations (15.4–15.5), the traffic descriptors can easily be computed. Thus, obviously QNA has the same time complexity as the Jackson network method. Note that, due to the linearity of the involved equations, QNA does not require the fixed-point iteration described in Section 15.2 (although, if an iterative solver is used to solve the linear equations, the fixed-point iteration can be regarded as hidden in the solver).

15.4 FiFiQueues

In the mid-1990's Haverkort and Weerstra, cf. [13, 14, 15, 48], extended Whitt's QNA approach by means of replacing the core of the analysis: the service operation. Unlike QNA, their new approach, called QNAUT, does not directly use the descriptor of the arrival traffic to compute the departure traffic descriptor, but assumes that the arrival traffic descriptor can be used to construct a phase-type (PH) renewal process (see Section 15.8.2) which approximates the "real" underlying arrival process. This allows for the inclusion of finite-buffer queueing stations as well as for the analysis of the queueing stations by matrix-geometric and general Markovian techniques, instead of the approximations originally used in QNA.

At the end of the 1990s, an extended version of the original approach was proposed, in which some approximate steps were removed and the model class was slightly enhanced [41, 42, 43]. In particular, this enhanced class provides:

- exact results for the the departure process based on the results of Bocharov [5] for PH|PH|1|K queues;
- efficient per-queue analysis;
- for each finite queueing station, a traffic stream is computed which consists of the customers rejected at a completely filled queue. This loss traffic stream can be used as arrival stream for other queueing stations like any other "regular" departure traffic stream.

This approach, as well as the analysis tool developed from it, is named *FiFiQueues* (for *Fi*xpoint-based analysis of networks with *Fi*nite *Queues*).

15.4.1 Model class

The external arrival processes are described, as in QNA, by the first and the second moment of the inter-arrival times. The main differences to QNA's model class are:

- the service processes are specified as PH renewal processes;
- the queueing stations can have *infinite* or *finite* queueing capacities. The nodes are analyzed as PH|PH|1(|K) queues with the FCFS service discipline. The customer multiplication factor known from QNA is also supported, but not described in the following;

- finite queues have two output streams: the "regular" departure traffic stream and the loss traffic stream which consists of the customers rejected by a full queue.

Seen from a single queue, customers arriving at a completely filled queue are simply lost. This form of blocking is common in communication networks (communication blocking) and has an important advantage: unlike other types of blocking (like back-blocking), it still allows the independent analysis of each of the queueing stations.

Just like the regular departure traffic of a queueing station with finite capacity, the loss traffic is not known a priori and is computed by the analysis of the station. The "reuse" of loss traffic streams as arrival streams to other nodes requires an auxiliary routing matrix. Its handling will not be discussed further in the following sections, since, once the traffic descriptors of the loss streams are known, they can easily be processed like the regular departure traffic. However, note that loss traffic streams should only be used very carefully in feedback networks: if a loss traffic stream is fed back directly or indirectly to the node which produced the stream, it can prevent the iteration algorithm (see Section 15.2) to terminate because the arrival rate to the node increases in each iteration step.

15.4.2 Traffic descriptor

As in QNA, the external arrival processes as well as the inter-node traffic streams are described by the first and second moment of the inter-arrival times. The traffic descriptor $\langle \lambda, c^2 \rangle$ contains the arrival rate λ and the squared coefficient of variation c^2 of the inter-arrival time.

15.4.3 Superposition of traffic streams

To merge n traffic streams specified by $\langle \lambda_1, c_1^2 \rangle, \ldots, \langle \lambda_n, c_n^2 \rangle$ into one traffic stream $\langle \lambda, c^2 \rangle$, we adopt the hybrid approximation of QNA, i.e.,

$$\lambda = \sum_{i=1}^{n} \lambda_i, \tag{15.1}$$

$$c^2 = w \cdot \sum_{i=1}^{n} \frac{\lambda_i}{\lambda} c_i^2 + (1-w), \tag{15.2}$$

with

$$w = \left[1 + 4(1-\rho)^2(v-1)\right]^{-1}, \quad \text{and} \quad v = \left(\sum_{i=1}^{n} \left(\frac{\lambda_i}{\lambda} \right)^2 \right)^{-1},$$

where ρ is the utilization of the node receiving the resulting traffic stream. It should be emphasized that these formulae were originally designed in the context of QNA's

model class, i.e., *not* for finite queues. Thus, their usage in FiFiQueues introduces auxiliary errors to the computation, in addition to the errors inherent to the hybrid approximation method.

One may wonder if we could obtain better results by not following QNA's linear approximation ($c_{SI}^2 = 1$) but in actually computing the correct value for c_{SI}^2. Our experiments have shown that nearly the same results are obtained by doing so. This is consistent with Whitt's observation that fixing c_{SI}^2 at 1 increases the average error by only 1 percent.

15.4.4 Splitting traffic streams

When splitting, we assume that the involved processes are renewal processes. Under this assumption, an exact solution is available. For n splitting probabilities $p_1, \ldots, p_n$ and the traffic stream $\langle \lambda, c^2 \rangle$, we obtain the splitted streams $\langle \lambda_1, c_1^2 \rangle, \ldots, \langle \lambda_n, c_n^2 \rangle$ with

$$\lambda_i = p_i \cdot \lambda, \quad \text{and} \quad c_i^2 = p_i \cdot c^2 + (1 - p_i), \qquad i = 1, \ldots, n. \tag{15.3}$$

15.4.5 Servicing jobs

We have already stated that the nodes are analyzed as PH|PH|1(|K) queues. Thus, before a queueing station can be analyzed we need to find a PH distribution that fits the two moments given in the arrival traffic descriptor. In the following we will explain the fitting step and the actual queueing analysis procedure, thereby treating PH|PH|1 and PH|PH|1|K queues separately. We require that the PH|PH|1 queues are stable, i.e., the total arrival rate at a PH|PH|1 station should be smaller than its service rate.

15.4.5.1 Phase-type representation of the arrival processes

Let $\langle \lambda, c^2 \rangle$ be the arrival traffic descriptor. We write $\mathrm{E}[X] = 1/\lambda$ for the corresponding mean inter-arrival time. Clearly, having only two moments allows us some freedom to select an appropriate PH distribution. We require that the chosen PH distribution, represented by (α, A)

1. matches the two moments exactly (at least for a certain range; see below), and
2. is as compact as possible, i.e., has the smallest number of transient states m.

Additionally, we want that the employed fitting procedure does not consume too much time since it has to be executed every time when a node is analyzed. In FiFiQueues, we use the following approach, first presented in [14]. Two cases are distinguished:

- In case $c^2 \leq 1$, we use a hypo-exponential distribution with $m = \left\lceil \frac{1}{c^2} \right\rceil$ phases and initial probability vector $\alpha = (1, 0, \cdots, 0)$. The matrix A is then given as

$$\mathrm{A} = \begin{pmatrix} -\lambda_0 & \lambda_0 & & & \\ & -\lambda_1 & \lambda_1 & & \\ & & \ddots & \ddots & \\ & & & -\lambda_{m-2} & \lambda_{m-2} \\ & & & & -\lambda_{m-1} \end{pmatrix}, \tag{15.4}$$

  where $\lambda_i = m/\mathrm{E}[X]$, for $0 \leq i < m-2$ and where

$$\lambda_{m-1} = \frac{2m\left(1 + \sqrt{\frac{1}{2}m(mc^2-1)}\right)}{\mathrm{E}[X](m+2-m^2c^2)} \quad \text{and} \quad \lambda_{m-2} = \frac{m\lambda_{m-1}}{2\lambda_{m-1}\mathrm{E}[X] - m} \ .$$

  For small c^2, PH distributions with a large number of states will be obtained. To limit the computational requirements in the analysis process we do not allow c^2 to be smaller than $\frac{1}{10}$. This approximation corresponds to an Erlang-10 distribution and produces generally good results, also as approximation for deterministic distributions.
- In case $c^2 > 1$, we take a hyper-exponential distribution with $m = 2$ phases. Such a distribution has three free parameters: the choice probability p between the two possible phases and the rates μ_1 and μ_2 of the two phases. Fitting the first two moments thus leaves one degree of freedom. We resolve this by assuming so-called "balanced means", meaning that the ratios p/μ_1 and $(1-p)/\mu_2$ should be equal. This then yields $\alpha = (p, 1-p)$ and

$$\mathrm{A} = \begin{pmatrix} -\frac{2p}{\mathrm{E}[X]} & 0 \\ 0 & -\frac{2(1-p)}{\mathrm{E}[X]} \end{pmatrix} \quad \text{with} \quad p = \frac{1}{2} + \frac{1}{2}\sqrt{\frac{c^2-1}{c^2+1}} \ .$$

15.4.5.2 Analysis of PH|PH|1|K queues

The underlying CTMC Let (α, A) be the arrival PH renewal process with l states as obtained by the fitting step and (β, B) the service PH renewal process with m states. Then we can describe the behavior of a node with queueing capacity K by a QBD process [37] (see Section 15.8.4) with $K+1$ levels, where level 0 consists of l states and where levels 1 through K consist of $l \cdot m$ states each.

The i-th level represents the state of the system when it contains i customers. A step from level i to level $i+1$ ($i < K$) stands for an arrival and a step from level i to level $i-1$ ($i > 0$) stands for a departure. The $l \cdot m$ states of a level $i > 0$ describe the current state of the arrival and of the service processes (level 0 contains only l states because the queue is empty and the service process has not yet started; it only records the state of the arrival process). This leads to the following generator matrix

of the Markov chain:

$$\mathrm{Q} = \begin{pmatrix} \mathrm{A} & \mathrm{A}^0\alpha\otimes\beta & & \\ \mathrm{I}\otimes\mathrm{B}^0 & \mathrm{A}\oplus\mathrm{B} & \mathrm{A}^0\alpha\otimes\mathrm{I} & 0 \\ 0 & \mathrm{I}\otimes\mathrm{B}^0\beta & \mathrm{A}\oplus\mathrm{B} & \mathrm{A}^0\alpha\otimes\mathrm{I} \\ & \ddots & \ddots & \ddots \\ & & \mathrm{I}\otimes\mathrm{B}^0\beta & (\mathrm{A}+\mathrm{A}^0\alpha)\oplus\mathrm{B} \end{pmatrix},$$

where $\mathrm{A}^0 = -\mathrm{A}\cdot 1$, $\mathrm{B}^0 = -\mathrm{B}\cdot 1$, $\mathrm{L}\oplus\mathrm{M} = \mathrm{L}\otimes\mathrm{I}+\mathrm{I}\otimes\mathrm{M}$, and $\otimes$ is the Kronecker product operator (also known as tensor or matrix direct product operator).

The steady-state solution v of the Markov chain with generator Q can be obtained by solving the global balance equation (see Section 15.8.4):

$$\mathrm{v}\cdot\mathrm{Q} = 0 \quad \text{and} \quad \mathrm{v}\cdot 1 = 1.$$

The vector v is of size $l + K\cdot l\cdot m$. In the following we write v_0 for the vector $(v_1,\ldots,v_l)$ which contains the steady-state probabilities of level 0 and we write v_i for the vector $(v_{l+1+(i-1)\cdot l\cdot m},\ldots,v_{l+i\cdot l\cdot m})$ which contains the steady-state probabilities of level $i = 1,\ldots,K$.

The departure traffic The steady-state solution vector v now allows us to compute the departure traffic descriptor $\langle\lambda_D, c_D^2\rangle$. To this end, we use the results of Bocharov presented in [5] which we will briefly describe in the following.

We begin with the computation of the blocking probability π, i.e., the probability that an arriving customer encounters a full queue and, hence, is lost. The vector $\mathrm{v}_{\mathrm{A,K}}$ gives for this situation the state probabilities and it holds that

$$\mathrm{v}_{\mathrm{A,K}} = \frac{1}{\lambda_A}\mathrm{v}_\mathrm{K}(\mathrm{A}^0\otimes\mathrm{I}),$$

where λ_A stands for the arrival rate to the node and K stands for the queueing capacity of the node. This leads to the blocking probability π:

$$\pi = \mathrm{v}_{\mathrm{A,K}}\cdot 1.$$

With π, we easily find the departure rate of served customers as

$$\lambda_D = \lambda_A(1-\pi). \tag{15.5}$$

Higher moments of the inter-departure time can be computed using the following consideration. If the queue is not empty after a departure took place, the distribution of the time up to the next departure is equal to the distribution of the service time. Otherwise, it is equal to the distribution of the sum of the time until the next customer arrival and its service time (which are independent). The probability to leave an empty queue at departure instant $t+\varepsilon$ is

$$\mathrm{v}_{\mathrm{D},0} = \frac{1}{\lambda_D}\mathrm{v}_1(\mathrm{I}\otimes\mathrm{B}^0). \tag{15.6}$$

This leads Bocharov to the expression for the i-th moment d_i of the inter-departure time distribution:

$$d_i = b_i + \mathrm{v}_{\mathrm{D},0}\sum_{j=1}^{i}(-1)^j\frac{i!}{(i-j)!}\mathrm{A}^{-j}1b_{i-j}, \tag{15.7}$$

where b_i is the i-th moment of the service time distribution. Thus, one can easily verify that the variance σ_D^2 of the departure process is

$$\sigma_D^2 = \sigma_S^2 + \sigma_0^2, \tag{15.8}$$

where σ_S^2 is the variance of the service time distribution and σ_0^2 equals

$$\sigma_0^2 = 2\mathrm{v}_{\mathrm{D},0}\mathrm{A}^{-2}1 - (\mathrm{v}_{\mathrm{D},0}\mathrm{A}^{-1}1)^2. \tag{15.9}$$

The squared coefficient of variation is then given by $c_D^2 = \lambda_D^2\sigma_D^2$.

The loss traffic The rate of loss λ_L is given by $\lambda_L = \lambda_A \cdot \pi$, where π is the loss probability. In oder to obtain higher moments of the inter-loss time we describe the loss process by the MAP $(\mathrm{L}_0, \mathrm{L}_1)$ with

$$\mathrm{L}_0 = \begin{pmatrix} \mathrm{A} & \mathrm{A}^0\alpha\otimes\beta & & \\ \mathrm{I}\otimes\mathrm{B}^0 & \mathrm{A}\oplus\mathrm{B} & \mathrm{A}^0\alpha\otimes\mathrm{I} & \\ 0 & \mathrm{I}\otimes\mathrm{B}^0\beta & \mathrm{A}\oplus\mathrm{B} & \mathrm{A}^0\alpha\otimes\mathrm{I} \\ & \ddots & \ddots & \ddots \\ & & \mathrm{I}\otimes\mathrm{B}^0\beta & \mathrm{A}\oplus\mathrm{B} \end{pmatrix}, \mathrm{L}_1 = \begin{pmatrix} 0 & & \\ & \ddots & \\ & & \mathrm{A}^0\alpha\otimes\mathrm{I} \end{pmatrix}.$$

The underlying CTMC of this MAP is the CTMC of the QBD where arrivals in the last level K have been marked. Naturally, it has the same steady-state probability vector v. The i-th moment of the inter-loss time is given by

$$\mathrm{E}[L^i] = \frac{i!}{\lambda_D}\mathrm{v}(-\mathrm{L}_0)^{-(i-1)}1, \tag{15.10}$$

hence, its second moment equals

$$\mathrm{E}[L^2] = \frac{2}{\lambda_L}\mathrm{v}(-\mathrm{L}_0)^{-1}1.$$

15.4.5.3 Analysis of PH|PH|1 queues

The underlying CTMC Let (α, A) be the arrival PH renewal process with l states and (β, B) the service PH renewal process with m states. Again, the behavior of the

queue can be described by a QBD process with a generator matrix similar to the one of the PH|PH|1|K; the only difference is the fact that it has repeating columns ad infinitum:

$$\mathrm{Q} = \begin{pmatrix} \mathrm{A} & \mathrm{A}^0\alpha\otimes\beta & & \\ \mathrm{I}\otimes\mathrm{B}^0 & \mathrm{A}\oplus\mathrm{B} & \mathrm{A}^0\alpha\otimes\mathrm{I} & \\ 0 & \mathrm{I}\otimes\mathrm{B}^0\beta & \mathrm{A}\oplus\mathrm{B} & \mathrm{A}^0\alpha\otimes\mathrm{I} \\ & \ddots & \ddots & \ddots \end{pmatrix},$$

with the infinite steady-state probability vector v fulfilling

$$\mathrm{v}\cdot\mathrm{Q} = 0 \quad \text{and} \quad \mathrm{v}\cdot 1 = 1.$$

We refer to Section 15.8.3 for an overview of solution techniques.

The departure traffic Since infinite queues produce no loss, we have

$$\lambda_D = \lambda_A, \tag{15.11}$$

where λ_A is the arrival rate to the node. The variance of the output stream is calculated using the same approach as in the case of finite-buffer queues and the equations (15.6), (15.8), and (15.9) still hold.

15.4.6 Node performance

FiFiQueues computes the first and second moment of the waiting time W and the queue length N. Again, queues with finite and infinite buffer capacity are treated separately.

15.4.6.1 Node performance of PH|PH|1|K queues

The j-moment $\mathrm{E}\left[N^j\right]$ of the queue length distribution (including the job in service) is given by

$$\mathrm{E}\left[N^j\right] = \sum_{i=1}^{K} i^j \mathrm{v_i} 1. \tag{15.12}$$

Hence, mean and variance of the queue length N are:

$$\mathrm{E}[N] = \sum_{i=1}^{K} i\cdot \mathrm{v_i} 1 \quad \text{and } \mathrm{Var}[N] = \sum_{i=1}^{K} i^2\cdot \mathrm{v_i} 1 - \mathrm{E}[N]^2.$$

Equation (4.4) in [5] gives the Laplace-Stieltjes transform of the waiting time probability density function. From this equation, any desired moment of the waiting time can be derived. For the mean and the variance we obtain [5, Eq. (4.5)–(4.7)]:

$$\mathrm{E}[W] = \frac{1}{\lambda_D}(\mathrm{E}[N] - 1 + \mathrm{v}_0 1),$$
$$\mathrm{Var}[W] = \frac{2}{\lambda_D}\left(\mu \cdot \mathrm{q}_2 1 - \left(\mathrm{q}_1\left(1 \otimes \mathrm{B}^{-1}1\right)\right)\right) - \mathrm{E}[W]^2,$$

where μ is the service rate. The components of the vector q_1 resp. q_2 give the first, resp. second binomial moment of the number of jobs in the queue as a function of the system state. For $j > 0$, the j-th binomial moment q_j is defined as [5, Eq. (3.1)]:

$$\mathrm{q}_j = \sum_{i=j+1}^{K} \binom{i-1}{j} \mathrm{v}_i.$$

15.4.6.2 Node performance of PH|PH|1 queues

In the case of infinite buffer capacity, the expressions presented for the PH|PH|1|K queue in the previous section can still be applied, provided that the steady-state probability vectors v_i are available in a form that allows to calculate the, now infinite, sums. For example, if we assume that a matrix-geometric solution method (see Section 15.8.3) is employed to compute the steady-state probabilities, the vectors v_i have the so-called matrix geometric form

$$\mathrm{v}_i = \mathrm{v}_1 \mathrm{R}^{i-1}, \quad \mathrm{R} \in \mathbb{R}^{lm \times lm}, \quad i = 1, 2, \ldots,$$

where R is the entry-wise smallest non-negative solution of the matrix-quadratic equation

$$\mathrm{A}^0 \alpha \otimes \beta + \mathrm{R}(\mathrm{A} \oplus \mathrm{B}) + \mathrm{R}^2(\mathrm{I} \otimes \mathrm{B}^0 \beta) = 0.$$

The j-th moment of the queue length distribution is then given by

$$\mathrm{E}[N^j] = \sum_{i=1}^{\infty} i^j \mathrm{v}_i 1 = \sum_{i=1}^{\infty} i^j \mathrm{v}_1 \mathrm{R}^{i-1} 1, \tag{15.13}$$

which yields in case $j = 1$:

$$\mathrm{E}[N] = \mathrm{v}_1 (\mathrm{I} - \mathrm{R})^{-2} 1.$$

Similarly, the other node performance measures can be obtained.

15.4.7 Network-wide performance

Many results for the network performance measures developed by Whitt for QNA (see Section 15.3.7) can also be applied to FiFiQueues when respecting the fact that, due to losses at finite queues, the departure rate of a node may differ from the total

arrival rate to that node. Additionally, one has to decide how loss traffic streams should be treated in the computation of network-wide performance results. For example, the following question has to be answered: should the expected number of visits $\mathrm{E}[V_i]$ also include rejections due to full buffers? As this is only a problem of "interpretation" of the results, we will not discuss it further here.

15.4.8 Complexity

15.4.8.1 Traffic computation

In FiFiQueues the traffic descriptor of the outgoing traffic depends in a complex, non-linear way on the incoming traffic. Thus, unlike the QNA method, FiFiQueues clearly requires an iterative computation scheme to compute the descriptors of the internal traffic streams. A deeper discussion of FiFiQueues' iteration behavior is given in Section 15.5. Here, we will analyze the complexity of the operations that have to be performed for each node during each iteration.

First, we can safely neglect the traffic merging and splitting steps in our discussion. They only consist of a small number of additions and multiplications. The most time and space consuming operation is the service operation. It can be divided into three phases:

1. fitting of the PH distribution to the arrival traffic,
2. computation of the steady-state probability vector of the underlying CTMC, and
3. computation of the departure traffic descriptor (and, if needed, of the loss traffic descriptor).

Again, we can neglect the first phase since its time complexity is $O(1)$. For the second phase, we distinguish between finite and infinite queueing stations.

If the queueing capacity is finite, so is the CTMC. Let l be the size of the arrival PH process, i.e., the number of states of its CTMC representation, m the size of the service PH process and K the queueing capacity. Then, the generator matrix is of size $(l+lmK)\times(l+lmK)$. This corresponds to a finite QBD with $N_0 = l$ and $N = lm$ (see Section 15.8.4). The latest implementation of FiFiQueues uses for finite capacities the Cyclic Reduction method [3] which has time complexity $O((l+m)^3 \log K + (l+m)^2 K)$. If the descriptor of the loss traffic is required, additional operations have to be performed to compute the product $\mathrm{v}(-\mathrm{L}_0)^{-1}$. For unbounded queueing capacity, the LR algorithm [28] is used.

Once the steady-state solution is known, the departure traffic descriptor can be computed. Both for finite and infinite queueing stations, this only requires a small number of matrix vector multiplications. Note that the moments b_i of the service process needed by Equation (15.7) are constant for a given network and hence can be precomputed once.

15.4.8.2 Node performance and network performance computation

Since the network performance computation is comparable to that of the QNA method, we only discuss the complexity of the node performance computation here.

Concerning finite queueing stations, the computation of the mean and variance of the queue length requires the summation over the $lm(K-1)$ entries of the steady-state probability vector. For the moments of the waiting time distribution, we have to invert matrix B of size $m \times m$ which can be seen as a constant time operation even for very complex PH representations of the service process (say, $m = 50$).

In case of infinite queueing capacity, the complexity depends on the employed solution method. Assuming a matrix-geometric solution method, the expression $\mathrm{E}[N] = \mathrm{v}_1(\mathrm{I}-\mathrm{R})^{-2}1$ we gave for the mean queue length in Section 15.4.6, requires the vector $\mathrm{X} = \mathrm{v}_1(\mathrm{I}-\mathrm{R})^{-2}$ which can be obtained by solving the linear system $\mathrm{X}(\mathrm{I}-\mathrm{R})^2 = \mathrm{v}_1$ of order lm.

15.4.9 The FiFiQueues network designer

The FiFiQueues approach has proven to be stable and reliable enough for end users. In this section we present an integrated tool environment, the FiFiQueues network designer, that allows an easy access to the underlying algorithms. The tool also contains a simulator for the steady-state simulation of queueing networks. The FiFiQueues network designer consists of a graphical user interface written in Java, a numerical analysis module, and a simulation module. The latter two have been written in C++.

15.4.9.1 The graphical user interface

The graphical user interface allows to construct, edit and study open and closed queueing networks of arbitrary topology. The networks can be evaluated by numerical analysis or by simulation. Figure 15.2 shows a screenshot of the main window. The lower part of the window shows the edited network and the properties of the currently selected node. The upper part displays the results of the numerical analysis (left section) and the results of the simulation (middle section, including the 95% confidence intervals) as well as a comparison of both methods (right section).

Every object in the network has properties that can be edited via the user interface. Figure 15.3(a) shows the properties of a finite queueing station while the user is selecting a service distribution. The global-properties panel (see Figure 15.3(b)) allows to control the length of the simulation and the parameters specific to closed networks.

The user interface communicates with the numerical analyzer and the simulator via text files. As an example, the network shown in the screenshot is translated into the following textual description in order to evaluate it.

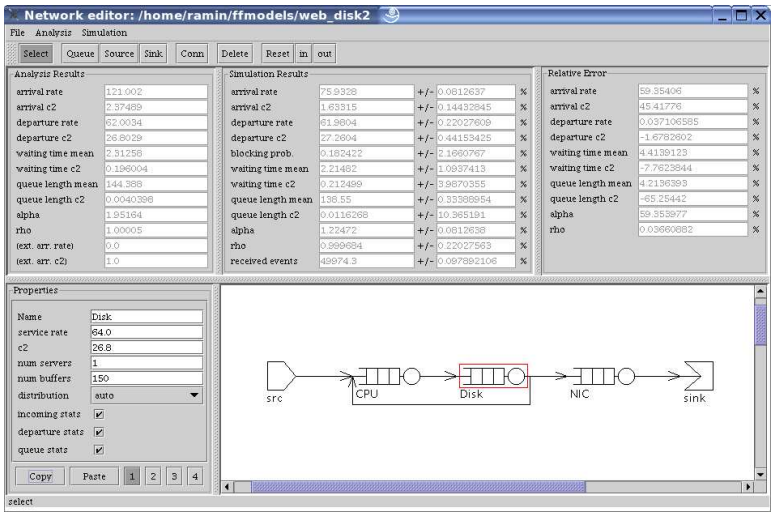
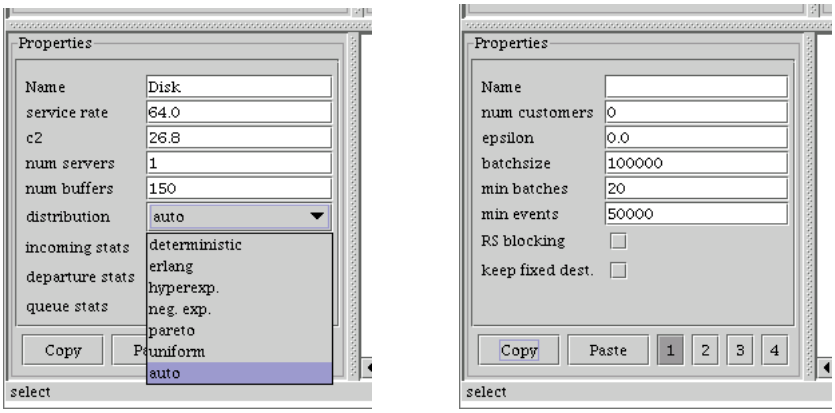


Fig. 15.2: Main window of the graphical user interface



(a) Properties of a network node (b) Global properties of the network

Fig. 15.3: Property editor

```
# Queue mapping
# 0 CPU
# 1 NIC
# 2 Disk
network_props
1 3 1 0 100000 20 50000 0  0   0.0
source_props
90.0 1.2 0 6
```

```
queue_props
1200.0 26.8 1 150 6 1  1  1
1401.0 26.8 1 150 6 1  1  1
64.0 26.8 1 150 6 1  1  1
counter_dest
-1
r
0.0 0.0 1.0 0.0
0.0 0.0 0.0 1.0
0.5 0.5 0.0 0.0
b
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
```

15.4.9.2 The numerical analysis module

The numerical analysis module is the core of the implementation. It incorporates the FiFiQueues algorithms as discussed and the extension for closed queueing networks as described in [44].

15.4.9.3 The simulation module

The simulation module offers the discrete-event simulation of open and closed queueing networks. It is described in detail in [40].

15.5 Performance of FiFiQueues

In this section we evaluate the performance of the FiFiQueues algorithm with regard to the quality of the numerical results. This evaluation consists of

- tests with the FiFiQueues algorithm on some representative queueing networks (Section 15.5.1),
- a case study of a web server (Section 15.5.2).

The results of the numerical analysis are compared to results determined using discrete-event simulation. The relative half-width of the 95%-confidence intervals is smaller than 1% for all the simulation results. If not stated otherwise, arrival time and service time distributions specified by their rate and the squared coefficient of variation (SCV) are always mapped to PH distributions. Relative errors between numerical analysis and simulation are always computed relative to the latter. We conclude with Section 15.5.3.

15.5.1 Evaluation of FiFiQueues

In this section we evaluate FiFQueues' performance with some typical networks. We begin with a single queue in Section 15.5.1.1, then continue with some complex networks in Section 15.5.1.2. The presented tests cover a wide range of input parameters, including (nearly) deterministic processes, and complex networks with finite queueing capacities.

15.5.1.1 Single queues

In the case of queueing networks that consist of only one queueing station, FiFiQueues always produces exact results, provided that the selected arrival and service PH renewal processes match the actual arrival and service processes of the real system. Hence, results of single-queue systems are not very interesting. At this place, we will only discuss the special case of deterministic distributions.

As explained, FiFiQueues limits the number of phases in hypo-exponential PH distributions to 10, which corresponds to a minimum SCV of 0.1. As a consequence, deterministic distributions can only be approximated. To evaluate the effect of this restriction we have analyzed a queueing station with negative exponential services and deterministic arrival process at different loads. Table 15.1 compares the thus obtained mean queue lengths with results found by simulation. It shows that the relative error between analysis and simulation increases with the load. Errors of comparable magnitude can also be observed for other performance measures and for hypo-exponential and hyper-exponential service distributions.

load	analysis	simulation	rel. error
0.1	0.10	0.10	0.0%
0.2	0.20	0.20	0.0%
0.4	0.47	0.45	4.4%
0.6	0.95	0.89	6.7%
0.8	2.34	2.18	7.3%
0.95	10.60	9.26	14.4%

Table 15.1: Mean queue length for a queueing station with deterministic arrival traffic

15.5.1.2 Queueing networks with feedback

3-node queueing network We first address three queueing nodes in series, with a feedback from the last to the first queue, as shown in Figure 15.4. The external Poisson source has rate 1.3 and the service times are Erlang-5 distributed with rate

1.5; the node capacity is 10 (not including the service station) at all queues. The feedback probability is 25%.

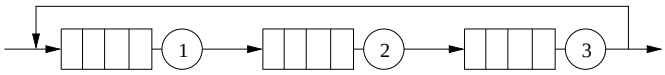


Fig. 15.4: 3-node queueing network with feedback

The results are shown in Table 15.2. The first two rows show the characteristics of the traffic leaving the queueing network from node 3. The middle six rows show the rate and SCV of the arrival traffic at each node, and the last three rows show the expected queue length at each node.

		analysis	simulation	rel. error
Output traffic	$\lambda_{netd,3}$	1.08	1.08	0.0%
	$c^2_{netd,3}$	0.41	0.41	0.0%
Arrival traffic	$\lambda_{a,1}$	1.65	1.66	-0.6%
	$c^2_{a,1}$	0.96	0.96	0.0%
	$\lambda_{a,2}$	1.47	1.47	0.0%
	$c^2_{a,2}$	0.23	0.23	0.0%
	$\lambda_{a,3}$	1.45	1.45	0.0%
	$c^2_{a,3}$	0.21	0.22	-4.5%
Queue length	$\mathrm{E}[N_1]$	6.47	6.47	0.0%
	$\mathrm{E}[N_2]$	4.43	4.45	-0.4%
	$\mathrm{E}[N_3]$	3.96	3.90	1.5%

Table 15.2: Results for the 3-node network with Poisson source

The good results of the analysis can be explained by the fact that the resulting arrival traffic to node 1 (i.e., where the traffic superposition operation happens) is near to Poisson as indicated by $c^2_{a,1}$=0.96. If we replace the external source distribution by a hyper-exponential distribution with $c^2 = 4.0$ we obtain the results shown in Table 15.3. As expected, larger errors can be observed this time for the SCV of the arrival traffic. Interestingly, node 2 does not seem to be affected. This is because node 2 is fed by node 1 which is overloaded and hence reduces short-range correlations in the traffic stream.

Figure 15.5 shows the incoming traffic to node 1 as a function of the number of iterations in the fixed-point procedure for both kind of external sources. As can be observed, the fixed-point is reached after a very small number of iterations. This behavior has been typical for all queueing networks we have analyzed so far.

Kühn's nine-node network As a larger queueing network we evaluated a modified version of Kühn's nine-node network [27], as shown in Figure 15.6 (the numbers at the edges specify the routing probabilities). A similar network has been ex-

		analysis	simulation	rel. error
Output traffic	$\lambda_{netd,3}$	0.99	0.99	0.0%
	$c^2_{netd,3}$	0.45	0.69	-34.8%
Arrival traffic	$\lambda_{a,1}$	1.63	1.63	0.0%
	$c^2_{a,1}$	3.33	2.35	41.7%
	$\lambda_{a,2}$	1.33	1.33	0.0%
	$c^2_{a,2}$	0.79	0.79	0.0%
	$\lambda_{a,3}$	1.32	1.33	-0.8%
	$c^2_{a,3}$	0.35	0.65	-46.2%
Queue length	$\mathrm{E}[N_1]$	5.57	5.59	-0.4%
	$\mathrm{E}[N_2]$	3.30	3.16	4.4%
	$\mathrm{E}[N_3]$	2.38	2.76	-13.8%

Table 15.3: Results for the 3-node network with hyper-exponential source

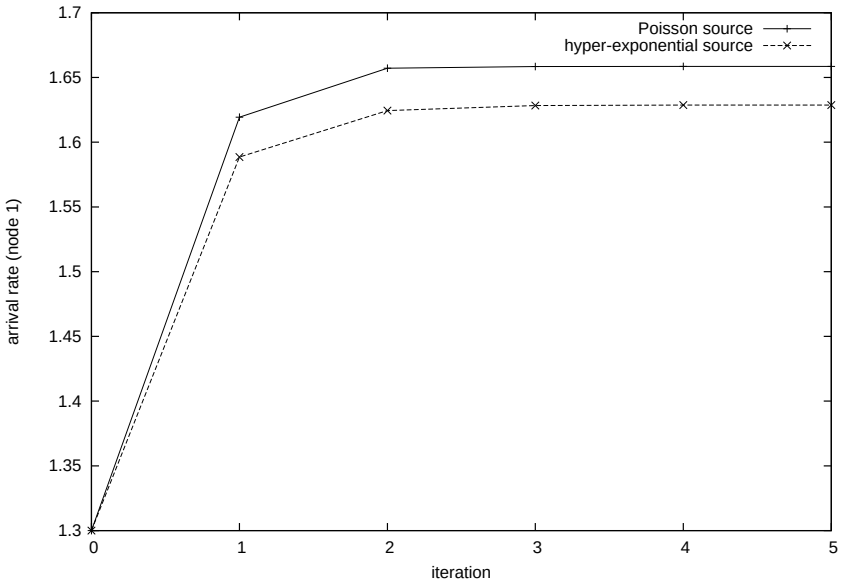


Fig. 15.5: Incoming traffic (arrival rate) at node 1 as a function of the number of iterations in the fixed-point procedure for the 3-node network

amined in [15, 48]. The external arrival rate to nodes 1–3 equals 0.8 and $c^2_{ext} = 4.0$. The service rate at each node is 1.0 (except for node 5 where $\mu_5 = 0.5$), and the SCV of all service processes is $c^2_s = 0.5$. All nodes have a finite queueing capacity of 25. Hence, without decomposition the underlying CTMC would comprise $2^3 \cdot (1+25\cdot 2)^9 \approx 1.86 \times 10^{16}$ states. For all nodes, we observe excellent agreement with the simulation results.

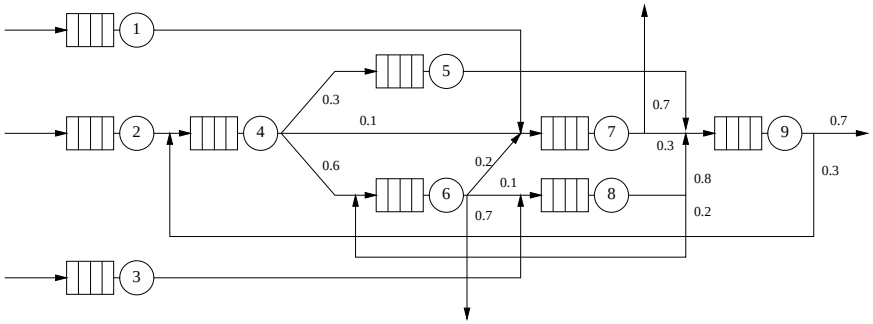


Fig. 15.6: Kühn's nine-node network

Table 15.4 shows the results obtained by FiFiQueues and by simulation for the mean queue length and the offered load at each station. Note that the results for the (identical) nodes 1–3 are only stated once.

node		analysis	simulation	rel. error
1–3	$E[N_i]$	6.39	6.39	0.0%
	offered load	0.8	0.8	0.0%
4	$E[N_4]$	16.84	16.74	0.6%
	offered load	1.09	1.09	0.0%
5	$E[N_5]$	1.14	1.13	0.9%
	offered load	0.59	0.59	0.0%
6	$E[N_6]$	2.31	2.28	1.3%
	offered load	0.77	0.76	1.3%
7	$E[N_7]$	14.67	14.86	-1.3%
	offered load	1.04	1.04	0.0%
8	$E[N_8]$	6.36	6.63	-4.0%
	offered load	0.87	0.87	0.0%
9	$E[N_9]$	22.41	21.88	2.4%
	offered load	1.28	1.27	0.8%

Table 15.4: Results for the departure rates in Kühn's nine-node network

15.5.2 Performance evaluation of a web server

In this section we will use FiFiQueues for the performance evaluation of a web server. The employed parameters in the models have been derived from measurements made at a test system.

This section is structured as follows. First, we describe the test system in Section 15.5.2.1. Then we present a QN model for a web server without disk access

(cache only) in Section 15.5.2.2, followed of the model of a web server with disk access in Section 15.5.2.3. These two models are then combined to a model of a server group in Section 15.5.2.4. We compare the results obtained by analysis with the results obtained by simulation, and, where available, with the data collected at the test system.

15.5.2.1 Description of the test system

The test system consists of a computer running the Apache web server [2]. The server load is generated by two client systems that send HTTP/1.0 GET requests to the server in a 100 MBit Ethernet LAN. The request times as well as the sizes of the requested files have been extracted from traces (access logs) collected at the UC Berkeley Home IP Service [11] in 1996. For our tests we have used a part of the original trace file: it consists of 35541 requests for static files (i.e., pictures, HTML pages, etc.) sent over 4 hours by different users. This corresponds to a request rate of 2.468 requests per second. The SCV of the inter-request time is 1.2. The requested files have a mean size of 8510 bytes where the smallest file has a size of 2 bytes and the largest file a size of about 4.5 MBytes. The size distribution has a SCV as large as 26.8.

The web server of the test system has been configured to use not more than 150 server threads. This implies that the number of requests that can be processed concurrently is limited to 150. Since connection requests are not queued the clients will experience a connection rejection if they try to exceed this number. In addition, the request time-out has been set to 8 seconds. More details concerning the test system can be found in [25]; please note that the QN models presented in the following differ from the models discussed there.

15.5.2.2 Web server without disk access

For the first model, we assume that the server holds all requested files in the file cache and, as a consequence, no disk access is performed. This is a typical situation in intranets where the number of often requested files is limited. In this scenario the performance of the web server is only limited by the CPU, the main memory, and the network interface controller (NIC).

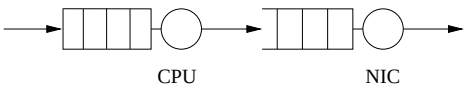


Fig. 15.7: QN model for the web server without disk access

We model the web server by two queueing stations in series as shown in Figure 15.7. Both stations have a finite queueing capacity; we comment on how the

buffer capacity is chosen below. The first station is fed by an external source that represents the clients sending the HTTP requests. The SCV of 1.2 for the source is equal to the corresponding value of the trace file.

The first station models the CPU. Measurements at the test system have shown that the CPU of the test server is able to process up to around 1200 requests per second. We adopt this value for the service rate of the first queueing station. Concerning the SCV of the CPU's service time distribution, we observe that the CPU service time is dominated by the time to handle the HTTP protocol and by the management of the cache data structures. Since the NIC accesses the main memory via DMA (Direct Memory Access), the CPU service time exhibits nearly no dependency on the size of the requested file. Hence, we choose a (nearly) deterministic service time distribution with a SCV of 0.1. The second queue represents the NIC. Measurements have shown a network load between 90% and 95% for a response rate of 1100 responses per second. This leads to a NIC service rate of approximately 1200. For the SCV of the NIC's service time distribution, we assume a direct dependency of the service time on the file size and we set the SCV to 26.8, i.e., to the SCV of the file size distribution.

The most problematic aspect of the test system is the limitation to 150 simultaneously connected clients. This cannot be easily modeled by the FCFS-scheduling used by FiFiQueues. To approximate the limit, we have first analyzed the network at a request rate of 1500 requests per second. Using a Newton-iteration, we have determined the queueing capacity at which the total mean number of jobs in the network equals 150. The thus found buffer capacity of 106 has then been used for *all* other request rates (we have chosen the same capacity for both queues; the jobs are distributed evenly over both stations at high request rates).

Figure 15.8 shows the number of responses per second as a function of the number of requests sent per second as measured at the test system and as computed by FiFiQueues. Simulation results are not shown since they are nearly identical to the analytical results (relative error $< 1\%$). It shows that the QN model is able to predict the response rate quite well. The total mean response times are shown in Figure 15.9. The results are acceptable, but we can see that the model is not able to reproduce the sharp jump of the response time at 1000 requests/s. A model with more complex behavior, for example non-FCFS scheduling, would be required in order to obtain better results.

15.5.2.3 Web server with disk access

The second model assumes that all requested files have to be loaded from the disk of the server system. Measurements have shown that the test system only achieves a maximum response rate of 63.5 files/s at a CPU load of 9%. Clearly, the disk transfer is the bottleneck.

We model the influence of the disk access through an additional queueing station. Figure 15.10 shows the resulting model. The first station represents the CPU. For the SCV of the CPU service time, we have kept the value of 0.1 of the previous model.

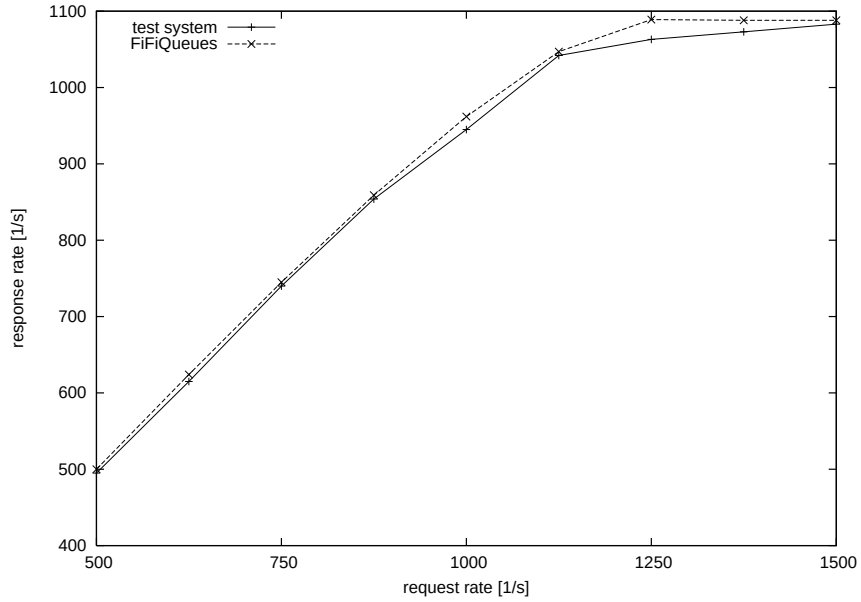


Fig. 15.8: Response rate as function of the request rate for the web server without disk access

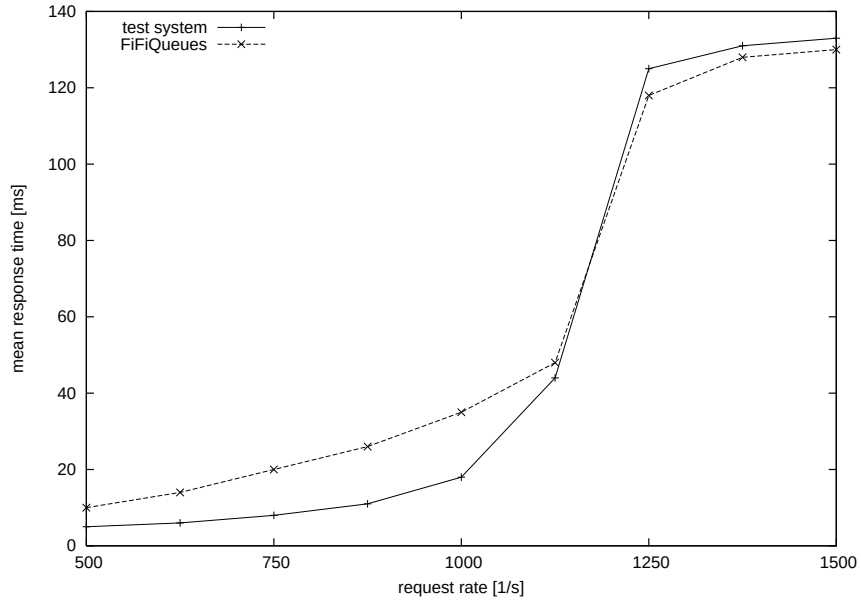


Fig. 15.9: Mean response time as function of the request rate for the web server without disk access

However, the service rate has now been set to a value of 706 ($= \frac{63.5}{0.09}$) to reflect the higher CPU demand of the single disk-based request. The service rate of the disk station has been set to 63.5. For the SCV, we have assumed a direct dependency of the service time on the size of the requested file *measured in blocks of 4 KBytes* since this corresponds to the organization of the data on the disk. This leads to a SCV of 16.5 instead of 26.8. The NIC in this model has the same service rate and SCV as in the previous model.

Again, the problem of the bounded number of simultaneously connected clients remains. Since the disk station clearly is the bottleneck, we have limited its queueing capacity to 150 while the CPU and the NIC station now have infinite queueing capacity. Note that, in spite of the large differences between the service rates, the CPU and the NIC should not be removed from the model since they have a small but measurable influence on the SCV of the traffic stream.

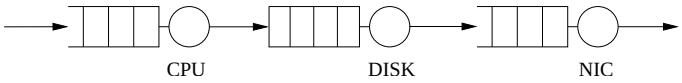


Fig. 15.10: QN model for the web server with disk access

Figure 15.11 shows the number of responses per second as a function of the number of requests sent per second as measured at the test system and as computed by FiFiQueues. Again, simulation results are not shown since their are nearly identical to the analytical results (relative error $< 1\%$). Again, it shows that the QN model is able to predict the response rate quite well. The total mean response times are shown in Figure 15.12. We observe that the QN model underestimates the response time, especially at request rates near to the maximum response rate of the disk. Our experiments with more complex QN models have shown that an improvement of the results cannot be easily achieved by using the type of queueing stations offered by FiFiQueues. For example, a more appropriate model would have to consider that the seek time of the disk becomes a significant part of the disk's response time at high file reqest rates since the disk has to reposition its read/write heads more often. Detailed models like the one presented in [39] simulate axial and rotational head positions, seek, rotation and transfer times, and provide separate submodels for the disk mechanism, the cache and the DMA engine.

15.5.2.4 Group of servers

In this section we evaluate a group of servers as shown in Figure 15.13. In our model, the client requests HTML pages from the main server of a web site. An HTML file refers to, on average, three other objects (company logo, images,...) that are also located on the main server. In addition, the HTML file refers to an object located on one of the five data servers. We assume that the HTML file and the three referred files located on the main server are frequently requested and, hence, the main

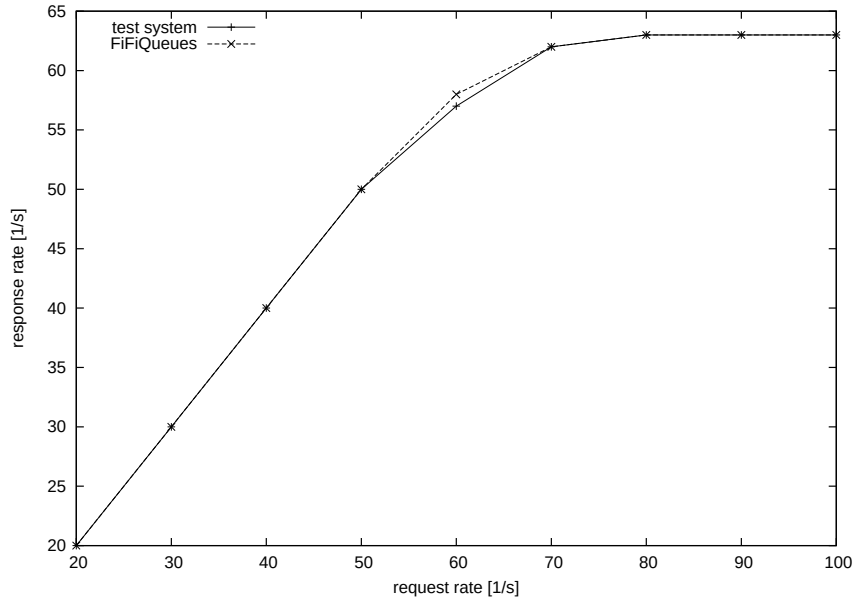


Fig. 15.11: Response rate as function of the request rate for the web server with disk access

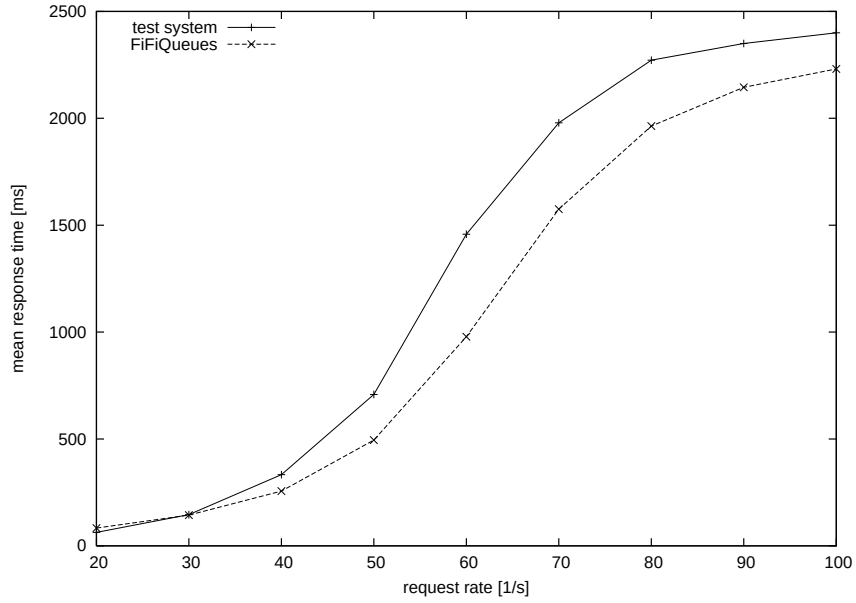


Fig. 15.12: Mean response time as function of the request rate for the web server with disk access

server mainly operates on the cache. Concerning the data servers, we assume that they store large amounts of infrequently requested files, for example files specific to the requesting user, media files, et cetera. The client uses the HTTP/1.0 protocol [35], i.e., the five files that constitute the requested HTML page are sequentially requested.

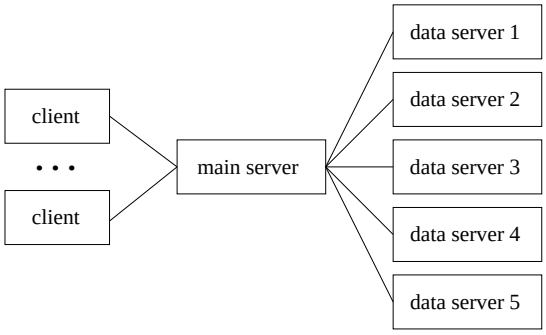


Fig. 15.13: Group of Web servers

The QN model is shown in Figure 15.14. The QN of the server without disk access (representing the main server) is combined with five copies of the QN of the server with disk access (representing the data servers). Jobs leaving the main server are fed back to it with a probability of 0.75, thus resulting in four visits to the main server in average. The jobs finally leaving the main server are distributed evenly on the data servers. The service processes and the capacities of the stations remain unchanged.

We have evaluated the QN model by FiFiQueues and by simulation. The results for the response rate (for one data server) and the mean response time are shown in Figure 15.15 and, respectively, Figure 15.16. The vertical bars in the latter show the 95%-confidence intervals of the simulation results. FiFiQueues provides good results for request rates smaller than 250. At larger request rates, FiFiQueues overestimates the losses in the main server because it ignores the correlations caused by the feedback. As a consequence, the load of the data servers is underestimated which leads to a smaller mean response time in comparison with the results obtained by simulation. To make this very clear: the differences we observe here show shortcomings of our analysis approach, as for both curves, the same model is being used.

Table 15.5 shows the runtimes (in seconds) of the FiFiQueues algorithm and of the discrete-event simulation for the evaluation of the server group model with various request rates. For FiFiQueues, we have recorded the runtimes for two different implementations of the finite queue analysis. The original implementation uses a Gauss-Seidel iteration, whereas the latest version uses the Cyclic Reduction method [3]. As observed in [40], the runtime of the Gauss-Seidel iteration increases

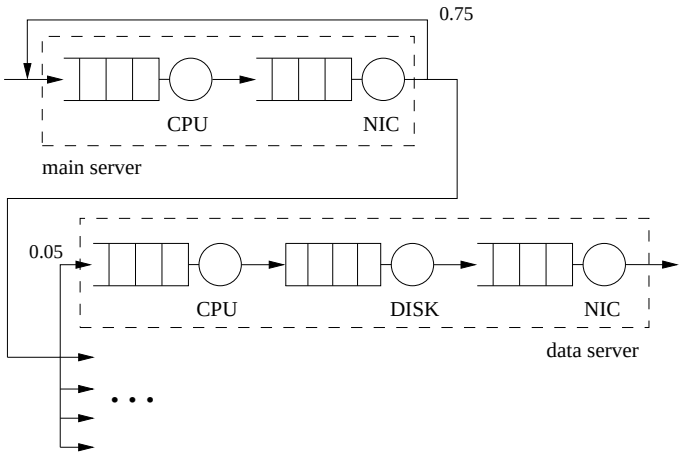


Fig. 15.14: QN model for the server group

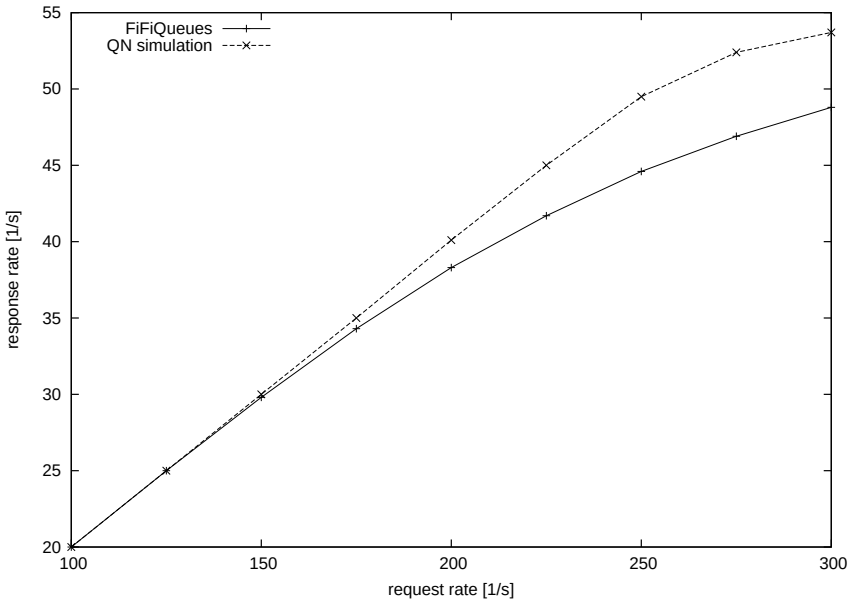


Fig. 15.15: Response rate as function of the request rate for the server group

	FiFiQueues		
request rate	Gauss-Seidel	Cyclic Red.	simulation
100	7	2	11
200	15	3	11
300	19	3	11

Table 15.5: Runtimes (in seconds) for the evaluation of the server group model

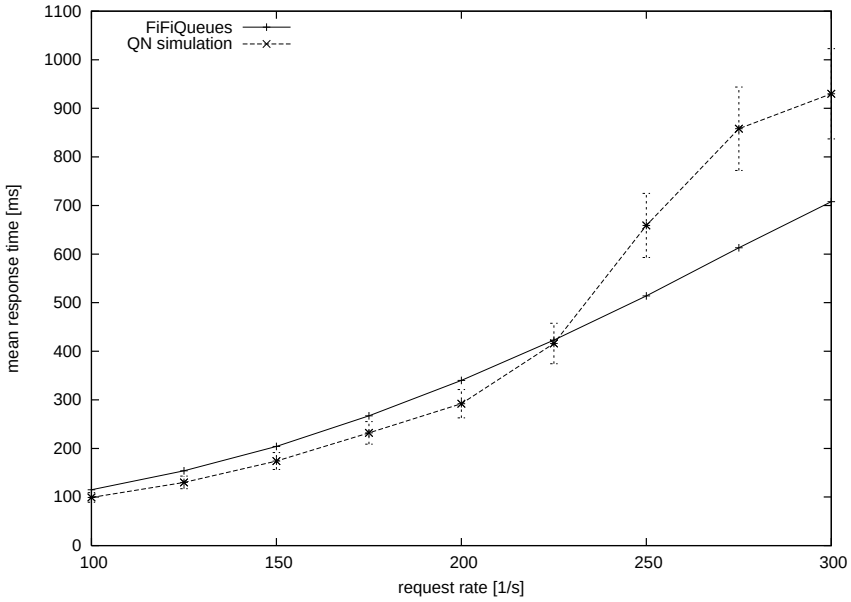


Fig. 15.16: Mean response time as function of the request rate for the server group

with the load of the stations. The Cyclic Reduction method is clearly faster than the Gauss-Seidel iteration and the simulation.

15.5.3 Summary

In this section, we have evaluated the performance of the FiFiQueues algorithm. Our experiments have shown that FiFiQueues provides very good results for important performance measures, like mean queue length, if the involved arrival times in the queueing network are hypo-exponentially or nearly (negative-)exponentially distributed. In such situations, we can generally expect relative errors less than 5%, even if the network has a complex structure. In case of hyper-exponential arrival processes, especially in queueing networks with feedback, relative errors up to 10%, rarely up to 20%, have been observed.

15.6 Summary and conclusions

In this chapter we have presented an overview of decomposition-based analysis techniques for large open queueing networks. We first presented the decomposition-

based approach in general terms, without referring to any particular model class, and proposed a general fixed-point iterative solution method for it. We concretized this framework by describing the well-known QNA method, as proposed by Whitt in the early 1980s, in that context, before describing our FiFiQueues approach. It should be noted that the work on FiFiQueues has been performed by a group of people over the last (almost) 15 years. To keep this chapter self-contained, we have added appendices on various underlying building blocks. In addition to an extensive evaluation with generally very favorable results for FiFiQueues, we also present a theorem on the existence of a fixed-point solution for FiFiQueues (which has not been published before).

In [40], we have also experimented with three-moment traffic descriptors, as well as with traffic descriptors taking into account correlations in the traffic streams. However, in our experiments, the three-moment descriptors have not significantly improved the results for queueing networks with feedback in comparison to the two-moment descriptors used by FiFiQueues. Since three-moment descriptors considerably increase the runtime of the analysis, we currently refrain from using them. Incorporating correlations in the traffic descriptors does hold promise, however, this should be investigated further before it can be made into a daily practice.

15.7 Appendix: Jackson queueing networks

The simplest open queueing networks allowing feedback are the so-called Jackson queueing networks (JQNs). Their analytical performance evaluation was developed by J.R. Jackson [20] in the 1950s.

15.7.1 Model class

In JQNs, all nodes are assumed to be infinite-buffer M|M|1 queues with the First-Come-First-Served (FCFS) service discipline. In many modeling applications, the restriction to Poisson arrival and service processes cannot be justified.

15.7.2 Traffic descriptor

In JQNs, all traffic processes (including the external arrival processes) are assumed to be Poisson, hence a sufficient traffic descriptor only contains the arrival rate λ of the traffic stream, denoted as $\langle\lambda\rangle$.

15.7.3 Superposition of traffic streams

Merging two (possibly dependent) traffic streams does not necessarily yield a new Poisson stream. However, it can be shown that the nodes of a JQN still can be described by M|M|1 queues even when traffic merging occurs. Thus, to merge n traffic streams specified by $\langle\lambda_1\rangle,\ldots,\langle\lambda_n\rangle$ into one traffic stream $\langle\lambda\rangle$, one simply adds the rates:

$$\lambda = \sum_{i=1}^{n} \lambda_i.$$

15.7.4 Splitting traffic streams

The Markovian splitting of a Poisson stream $\langle\lambda\rangle$ again results in n Poisson streams. Let $p_1,\ldots,p_n$ be the splitting probabilities, then the resulting streams $\langle\lambda_1\rangle,\ldots,\langle\lambda_n\rangle$ are given by

$$\lambda_i = p_i \cdot \lambda, \qquad i = 1,\ldots,n.$$

15.7.5 Servicing jobs

Let $\langle\lambda_A\rangle$ be the arrival traffic descriptor of the node, and μ its service rate. We require that $\lambda_A < \mu$, otherwise the station is not stable. Burke [7] proved that the departure process for a stable single server M|M|1 queue is a Poisson process with rate λ_A, hence, the departure process can be described as $\langle\lambda_D\rangle$ with $\lambda_D = \lambda_A$.

15.7.6 Node performance

Let $\langle\lambda_A\rangle$ be the arrival traffic descriptor, and μ the service rate of the node. Then, $\rho = \lambda_A/\mu$ is the utilization of the node. Since the node is an M|M|1 queue, the steady-state probability p_j to find j customers in the queue can be easily derived from the underlying birth-death Markov chain [16]:

$$p_j = (1-\rho)\rho^j, \qquad j = 0,1,\ldots$$

Having computed the steady-state probabilities, quantities like the expected number of jobs in the queueing station $\mathrm{E}[N]$ can be calculated as

$$\mathrm{E}[N] = \sum_{j=0}^{\infty} j \cdot p_j = \frac{\rho}{1-\rho}.$$

Then, Little's law can be applied to compute the expected waiting time $\mathrm{E}[W]$.

Similarly, higher moments of measures can be computed too, e.g., the variance of the number of customers in the node:

$$\mathrm{Var}[N] = \sum_{j=0}^{\infty} (j - \mathrm{E}[N])^2 \cdot p_j = \frac{\rho}{(1-\rho)^2}.$$

15.7.7 Network-wide performance

Since no losses occur and all nodes are required to be stable, the total throughput λ_{thr} of the network, i.e., the average number of customers passing through the network per time unit, is simply the sum of arrival rates $\lambda_{ext,i}$ of the external arrival processes:

$$\lambda_{thr} = \sum_{i=1}^{n} \lambda_{ext,i}$$

where $\langle \lambda_{ext,i} \rangle$ is the external traffic arriving at node i and n is the number of nodes. Other performance measures may be derived from the node performance measures. If $\lambda_{A,i}$ is the total amount of traffic arriving at node i, the expected number of visits $\mathrm{E}[V_i]$ of a customer at node i is given by [49, Eq. (77)]:

$$\mathrm{E}[V_i] = \lambda_{A,i} / \lambda_{thr}.$$

The expected total sojourn time $\mathrm{E}[T_{total}]$, i.e., the time a customer spends in the network, defined as the sum of the expected sojourn times $\mathrm{E}[T_i]$ at each node i, thus equals

$$\mathrm{E}[T_{total}] = \sum_{i=1}^{n} \mathrm{E}[T_i] = \sum_{i=1}^{n} \mathrm{E}[V_i] \left(\frac{1}{\mu_i} + \mathrm{E}[W_i] \right).$$

Since the total number of customers N_{total} in the network is the sum of customers present in each queueing station, we have

$$\mathrm{E}[N_{total}] = \sum_{i=1}^{n} \mathrm{E}[N_i],$$

where $\mathrm{E}[N_i]$ is the expected number of jobs in node i.

15.7.8 Complexity

If $\Gamma = (r_{ij})$ is the routing matrix, the traffic $\langle \lambda_{A,i} \rangle$ arriving at node i is given by the so-called *first-order traffic equation*:

$$\lambda_{A,i} = \lambda_{ext,i} + \sum_{j=1}^{n} \lambda_{D,j} \cdot r_{ji}.$$

Since $\lambda_{D,i} = \lambda_{A,i}$, the traffic equations form a system of linear equations which can be expressed in vector/matrix notation as $\lambda_{\mathrm{A}} = \lambda_{\mathrm{ext}} + \lambda_{\mathrm{A}} \cdot \Gamma$, or, after transformation, as

$$\lambda_{\mathrm{A}} = \lambda_{\mathrm{ext}} (\mathrm{I} - \Gamma)^{-1}.$$

Thus, to find λ_{A} we solve the linear system

$$\lambda_{\mathrm{A}} (\mathrm{I} - \Gamma) = \lambda_{\mathrm{ext}}.$$

This system of equations can be solved by direct methods like Gaussian elimination, resulting in a time complexity of $O(n^3)$, or by iterative methods like Gauss-Seidel. This implies that, due to the linearity of the involved equations, the analysis of JQNs does not require the fixed-point iteration described in Section 15.2 (although, if an iterative solver is used to solve the linear system, the fixed-point iteration can be regarded as hidden in the solver).

For very large networks, we can make use of the fact that the routing matrix typically is a sparse matrix. In this way, the time complexity of an iterative solver such as Gauss-Seidel can be reduced to about $O(c \cdot n)$ where c is the average number of outgoing connections per station.

The expressions given in Section 15.7.6 for the node performance measures can be computed in constant time for each node. For the network performance, most results require summation over the number of nodes in the network which yields a time complexity of $O(n)$.

15.8 Appendix: MAPs, PH-distributions and QBDs

In this appendix we introduce the fundamental mathematical structures and notation used throughout this chapter. We begin with an important class of stochastic processes, the Markovian Arrival Processes (MAP) in Section 15.8.1. Phase-type (PH) renewal processes, which can be seen as special cases of MAPs, are introduced in Section 15.8.2. The queueing processes that we have discussed in Section 15.4 have underlying Markov chains that belong to the well-known class of continuous-time Quasi-Birth-and-Death (QBD) processes. We give the formal definition of QBD processes as well as methods to compute their steady-state solution. We first discuss infinite QBDs in Section 15.8.3 and continue with finite QBDs in Section 15.8.4.

15.8.1 Markovian Arrival Processes (MAPs)

15.8.1.1 Definition and notation

Markovian Arrival Processes (MAPs) [29, 30, 36] belong to the general class of point processes and can be seen as special cases of Matrix Exponential Point Processes (which, in turn, form a subset of the class of Semi-Markov Processes [16]). MAPs cover many interesting processes including the Markov-Modulated Poisson Processes (MMPPs) [10] and the phase-type (PH) renewal processes (see below).

A MAP can be described by a finite irreducible continuous-time Markov chain (CTMC) with generator matrix Q where some transitions are "marked". Every time when the process passes through such a marked transition an event is triggered. The time instants of these events form the point process. We follow the notation of [31] and split the generator matrix into two matrices Q_0 and Q_1 as follows:

$$\mathrm{Q}_0 = \begin{pmatrix} -q_1 & q_{12} & \dots & q_{1m} \\ q_{21} & -q_2 & \dots & q_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ q_{m1} & q_{m2} & \dots & -q_m \end{pmatrix}, \quad \mathrm{Q}_1 = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{pmatrix},$$

with $\mathrm{Q}_0 + \mathrm{Q}_1 = \mathrm{Q}$ where $q_i = a_{ii} + \sum_{j=1, j\neq i}^{m} (q_{ij} + a_{ij})$. The elements of the matrix Q_1 give the transition rates of the marked transitions.[1] In the following, we denote a MAP by the pair $(\mathrm{Q}_0, \mathrm{Q}_1)$ and call *m* the *size* of the MAP.

15.8.1.2 Characteristics

Some general results of the Markov-modulated Poisson process [10] can be easily adapted to the MAP. In order to compute the behavior of a MAP $(\mathrm{Q}_0, \mathrm{Q}_1)$ we first need to choose the initial probability vector p of the MAP. In analogy to phase-type renewal processes we start the MAP at an "arbitrary" arrival epoch by choosing

$$\mathrm{p} = \frac{1}{\pi \mathrm{Q}_1 1} \pi \mathrm{Q}_1,$$

where π is the steady-state probability vector of the MAP, i.e., $\pi(\mathrm{Q}_0 + \mathrm{Q}_1) = 0$. The thus-obtained process is said to be *interval-stationary*. The inter-arrival time distribution function of the interval-stationary process is given by

$$F(t) = 1 - \mathrm{p} \exp(\mathrm{Q}_0 t) 1, \tag{15.1}$$

[1] This definition allows the following interpretation of the matrices Q_0 and Q_1: passing through a transition given as entry of Q_1 triggers the generation of *one* event. Batch Markovian Arrival Processes (BMAPs) generalize this viewpoint by introducing matrices Q_i with $i > 1$ whose entries describe transitions with batch arrivals of size *i*.

which leads to the following expression for the kth moment of the inter-arrival time:

$$\mathrm{E}[T^k] = k!\mathrm{p}(-\mathrm{Q}_0)^{-(k+1)}\mathrm{Q}_1\mathrm{1}. \tag{15.2}$$

Hence, the first moment of the inter-arrival time is given by

$$\mathrm{E}[T] = \frac{1}{\pi\mathrm{Q}_1\mathrm{1}}\pi\mathrm{Q}_1(-\mathrm{Q}_0)^{-2}\mathrm{Q}_1\mathrm{1}.$$

This equation can be further simplified by using the equations $\pi\mathrm{Q}_1 = -\pi\mathrm{Q}_0$ and $\mathrm{Q}_1\mathrm{1} = -\mathrm{Q}_0\mathrm{1}$ which follow from the definition of π, respectively, from the fact that $\mathrm{Q}_0 + \mathrm{Q}_1$ is a stochastic matrix. We find that the arrival rate λ of a MAP (the inverse of the first moment) is

$$\lambda = \pi\mathrm{Q}_1\mathrm{1}$$

which yields

$$\mathrm{E}[T^k] = \frac{k!}{\lambda}\pi(-\mathrm{Q}_0)^{-(k-1)}\mathrm{1}.$$

Let T_i be the time between the ith and the $(i+1)$st arrival in a MAP. Then, the autocovariance function $R(k)$ for T_1 and T_{k+1} with $k \geq 1$ is given by

$$\begin{aligned} R(k) &= \mathrm{E}\left[(T_1 - \mathrm{E}[T_1])(T_{k+1} - \mathrm{E}[T_{k+1}])\right] \\ &= \mathrm{p}(-\mathrm{Q}_0)^{-2}\mathrm{Q}_1\left\{\left[(-\mathrm{Q}_0)^{-1}\mathrm{Q}_1\right]^{k-1} - \mathrm{1p}\right\}(-\mathrm{Q}_0)^{-1}\mathrm{1}. \end{aligned}$$

The limiting index of dispersion I of a MAP is given by [18]

$$I = \lim_{t\to\infty}\frac{\mathrm{Var}[N(t)]}{E[N(t)]} = 1 + 2\left(\lambda - \frac{1}{\lambda}\pi\mathrm{Q}_1(\mathrm{Q}_0 + \mathrm{Q}_1 + \mathrm{1}\pi)^{-1}\mathrm{Q}_1\mathrm{1}\right),$$

where $N(t)$ is the counting process of the MAP.

15.8.1.3 Superposition and Markovian splitting

The class of MAPs is closed under superposition and Markovian splitting. The superposition of two MAPs $(\mathrm{A}_0, \mathrm{A}_1)$ and $(\mathrm{B}_0, \mathrm{B}_1)$ is a new MAP $(\mathrm{C}_0, \mathrm{C}_1)$ with

$$\mathrm{C}_0 = \mathrm{A}_0 \oplus \mathrm{B}_0, \quad \mathrm{C}_1 = \mathrm{A}_1 \oplus \mathrm{B}_1,$$

where $\mathrm{L} \oplus \mathrm{M} = \mathrm{L} \otimes \mathrm{I} + \mathrm{I} \otimes \mathrm{M}$, and $\otimes$ is the Kronecker product operator (also known as tensor or matrix direct product operator).

The Markovian splitting of a MAP $(\mathrm{A}_0, \mathrm{A}_1)$ with probability r gives two MAPs $(\mathrm{B}_0, \mathrm{B}_1)$ and $(\mathrm{C}_0, \mathrm{C}_1)$ with

$$\begin{aligned} (\mathrm{B}_0, \mathrm{B}_1) &= (\mathrm{A}_0 + (1-r)\mathrm{A}_1, r\mathrm{A}_1), \\ (\mathrm{C}_0, \mathrm{C}_1) &= (\mathrm{A}_0 + r\mathrm{A}_1, (1-r)\mathrm{A}_1). \end{aligned}$$

15.8.1.4 Markov-Modulated Poisson Processes (MMPPs)

The MMPP is the doubly stochastic Poisson process whose arrival rate depends on the state of an irreducible Markov process. Thus, MMPPs can be seen as MAPs where the matrix Q_1 is restricted to the form

$$\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{mm} \end{pmatrix}.$$

15.8.2 Phase-type (PH) renewal processes

15.8.2.1 Definition and notation

A continuous phase-type renewal process can be seen as a special MAP $(\mathrm{A}, \mathrm{A}^0\alpha)$ where A^0 is a $n \times 1$ column vector with entries and α is a $1 \times n$ row probability vector. Consequently, it holds $\mathrm{A}^0 = -\mathrm{A}1$.

We adopt the notation of [37] and denote PH renewal processes by the pair (α, A) which can be interpreted as follows: the $n \times n$ matrix A describes the transitions from the n transient states of a CTMC with $n+1$ states. The last state $n+1$ is an absorbing state and any transition (given by A^0) from the transient state to the absorbing state will trigger an arrival. After the arrival, the process will restart in the transient state i with probability α_i. Furthermore, PH inter-event time distributions form a dense subset of all distributions with support on $[0;\infty)$, i.e., any distribution can be approximated arbitrarily closely by a PH distribution [23].

15.8.2.2 Inter-event time characteristics

Obviously, the vector α of the PH renewal process (α, A) is identical to the interval-stationary probability vector p of the corresponding MAP. Hence, expressions for the distribution function of the inter-event time and the k-th moment directly follow from Equations (15.1) and (15.2) and we have

$$F(t) = 1 - \alpha \exp(\mathrm{A}t)1,$$

respectively

$$\mathrm{E}[T^k] = k!\alpha(-\mathrm{A})^{-k}1.$$

Note that the matrix A is nonsingular, so that all moments are finite. From this follows that the MAP $(\mathrm{Q}_0, \mathrm{Q}_1)$ and the PH renewal process $(\mathrm{p}, \mathrm{Q}_0)$ have the same inter-event time distribution.

15.8.2.3 Superposition and Markovian splitting

The superposition of two PH renewal processes (α, A) and (β, B) is a MAP $(\mathrm{C}_0, \mathrm{C}_1)$ with

$$\mathrm{C}_0 = \mathrm{A} \oplus \mathrm{B}, \quad \mathrm{C}_1 = \mathrm{A}^0\alpha \oplus \mathrm{B}^0\beta.$$

Note that the class of PH renewal processes is not closed under superposition.

The Markovian splitting of a PH renewal processes (α, A) with probability r gives two PH renewal processes $(\alpha, \mathrm{A} + (1-r)\mathrm{A}^0\alpha)$ and $(\alpha, \mathrm{A} + r\mathrm{A}^0\alpha)$.

15.8.3 Infinite QBDs

This section is based on Chapter 4 of [38]. Note that we use a simplified notation.

15.8.3.1 Definition

QBD processes [37] can be described as a generalization of the queueing process of M|M|1 queueing stations. In the underlying Markov chain of such a queue we can identify an infinite number of states where state *i* describes that *i* jobs are in the system. The transition from state i to $i+1$ resp. from $i+1$ to i is marked by the arrival rate resp. the service rate of the queueing station.

In QBDs, these states are replaced by so-called *levels*: level i still stands for i jobs in system but in QBDs each level may consist of more than one state. Usually, a two-dimensional addressing scheme is used for the states where (i, j) addresses state j of level i. Note that in the QBD the number of levels is unbounded whereas the number of states per level is required to be finite. Moreover, the levels $1, 2, \ldots$ (the *repeating levels*) have to contain the same number of states N. Level 0 is called *boundary level* and may contain a different number of states N_0.

In QBDs, two adjacent levels i and $i+1$ are not connected by one single transition. Instead, arbitrary transitions between the states of two adjacent levels and between states of the same level are allowed. Consequently, the transition rates are specified by matrices:

- the entry (i, j) of the $N_0 \times N$ matrix $\mathrm{B}_{0,1}$ gives the transition rate from state $(0, i)$ to state $(1, j)$. The opposite direction (from level 1 to level 0) is given similarly by the $N \times N_0$ matrix $\mathrm{B}_{1,0}$.
- the entry (i, j) of matrix A_0 gives the transition rate from state (l, i) to state $(l+1, j)$ where $l = 1, 2, \ldots$. The opposite direction (from level $l+1$ to l) is given by matrix A_2. Both matrices are of size $N \times N$.
- transitions inside level 0 are specified by the $N_0 \times N_0$ matrix $\mathrm{B}_{0,0}$. Entry (i, j) gives the transition rate from state $(0, i)$ to state $(0, j)$. Correspondingly, transitions inside repeating level l (with $l = 1, 2, \ldots$) are specified by the $N \times N$ matrix A_1.

As can be seen, all repeating levels have a similar transition structure. The above described matrices directly lead to the generator matrix of the QBD Markov chain. If we sort the states lexicographically, i.e., in the sequence

$$(0,1),\ldots,(0,N_0),(1,1),\ldots,(1,N),(2,1),\ldots$$

we obtain the tri-diagonal block generator matrix Q of infinite size:

$$\mathrm{Q} = \begin{pmatrix} \mathrm{B}_{0,0} & \mathrm{B}_{0,1} & & & \\ \mathrm{B}_{1,0} & \mathrm{A}_1 & \mathrm{A}_0 & & \\ & \mathrm{A}_2 & \mathrm{A}_1 & \mathrm{A}_0 & \\ & & \mathrm{A}_2 & \mathrm{A}_1 & \mathrm{A}_0 \\ & & & \ddots & \ddots & \ddots \end{pmatrix}. \tag{15.3}$$

15.8.3.2 Steady-state solution

The infinite steady-state probability vector v of the QBD Markov chain with generator matrix Q fulfills the global balance equation

$$\mathrm{v} \cdot \mathrm{Q} = 0, \tag{15.4}$$

and the normalization condition

$$\mathrm{v} \cdot 1 = 1. \tag{15.5}$$

In the following we write v_0 for the vector $(v_1,\ldots,v_{N_0})$ which contains the steady-state probabilities for the states of level 0 and we write v_i for the vector $(v_{N_0+1+(i-1)\cdot N},\ldots,v_{N_0+i\cdot N})$ which contains the steady-state probabilities of level $i = 1,2,\ldots$. With this notation we can rewrite Equations (15.4) and (15.5) as

$$\mathrm{v}_0\mathrm{B}_{0,0} + \mathrm{v}_1\mathrm{B}_{1,0} = 0, \tag{15.6}$$

$$\mathrm{v}_0\mathrm{B}_{0,1} + \mathrm{v}_1\mathrm{B}_{1,1} + \mathrm{v}_2\mathrm{A}_2 = 0, \tag{15.7}$$

$$\mathrm{v}_\mathrm{i}\mathrm{A}_0 + \mathrm{v}_{\mathrm{i}+1}\mathrm{A}_1 + \mathrm{v}_{\mathrm{i}+2}\mathrm{A}_2 = 0, \qquad \text{for } i = 1,2,\ldots, \tag{15.8}$$

$$\sum_{i=0}^{\infty} \mathrm{v}_\mathrm{i} 1 = 1. \tag{15.9}$$

The regular structure of Equation (15.8) is the key to the efficient solution of the QBD. Two different classes of solution techniques can be distinguished: matrix-geometric solution methods and transform methods. We will describe them briefly in the following.

15.8.3.3 Matrix-geometric solution methods

The main idea in this class of solution methods is that the solution vector v has a matrix-geometric form, i.e., there exists a matrix R of size $N \times N$ with

$$\mathrm{v_i} = \mathrm{v_1 R}^{i-1}, \qquad i = 1, 2, \ldots \tag{15.10}$$

In [37], it is shown that R is the entry-wise smallest non-negative solution of the quadratic matrix equation

$$\mathrm{A_0} + \mathrm{RA_1} + \mathrm{R}^2\mathrm{A_2} = 0. \tag{15.11}$$

The methods belonging to this class of solution methods try to solve this equation as efficiently as possible. Once R has been determined, the complete stationary vector v can be computed using Equations (15.6)–(15.10). Examples for such methods are the Successive Substitution method [37], the Logarithmic Reduction method [28] and its improvement [34].

15.8.3.4 Transform methods

Unlike the matrix-geometric solution methods the transform methods do not aim to directly solve Equation (15.11). Instead, they first transform the problem into some other domain in order to derive the solution of Equation (15.8). Three well known methods belonging to this class are the Cyclic Reduction method [4], the Invariant Subspace method [1], and the Spectral Expansion method [37, 8].

15.8.4 Finite QBDs

15.8.4.1 Definition

Similar to infinite QBDs, finite QBDs can be seen as the generalization of the queueing process of a bounded M|M|1|K queue. Finite QBD processes result in QBD Markov chains with a finite number $K+1$ of levels, hence two boundary levels can be identified: the *lower boundary level* 0 and the *upper boundary level* K.

In the following we will only treat a quite restricted class of finite QBDs that is sufficient for the queueing process discussed in this chapter: The upper boundary level has the same number of states N as the repeating levels 1 through $K-1$. Additionally, the transition rates between levels $K-1$ and K are the same as between the repeating levels — only one new matrix C is introduced that specifies the transition rates inside level K. The finite generator matrix of the QBD Markov chain then has the following form:

$$Q = \begin{pmatrix} B_{0,0} & B_{0,1} & & & & & \\ B_{1,0} & A_1 & A_0 & & & & \\ & A_2 & A_1 & A_0 & & & \\ & & A_2 & A_1 & A_0 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & A_2 & A_1 & A_0 \\ & & & & & A_2 & C \end{pmatrix}. \tag{15.12}$$

15.8.4.2 Steady-state solution

We search the finite steady-state probability vector v of the QBD Markov chain with generator matrix Q that fulfills the global balance equation

$$v \cdot Q = 0 \tag{15.13}$$

and the normalization condition

$$v \cdot 1 = 1. \tag{15.14}$$

As in the infinite case, we partition the vector v into subvectors v_i where $v_0 = (\nu_1, \ldots, \nu_{N_0})$ and $v_i = (\nu_{N_0+1+(i-1)\cdot N}, \ldots, \nu_{N_0+i\cdot N})$, for $i = 1, \ldots, K$. It is important to note that the solution of Equation (15.13) is uncritical with respect to space complexity. Due to the special structure of the Markov chain it is not necessary to hold the whole matrix Q in memory but only the matrices $B_{0,0}$, $B_{1,0}$, $B_{0,1}$, $B_{1,1}$, A_0, A_1, A_2 and C. In terms of these matrices, Equation (15.13) becomes:

$$v_0 B_{0,0} + v_1 B_{1,0} = 0 \tag{15.15}$$

$$v_0 B_{0,1} + v_1 B_{1,1} + v_2 A_2 = 0 \tag{15.16}$$

$$v_i A_0 + v_{i+1} A_1 + v_{i+2} A_2 = 0, \qquad \text{for } i = 1, \ldots, K-2, \tag{15.17}$$

$$v_{K-1} A_0 + v_K C = 0 \tag{15.18}$$

Since Q is of finite size, Equation (15.13) can be solved by an ordinary Gauss-Seidel-iteration which performs very efficiently due to the band-structure of Q. More sophisticated algorithms have been developed on the basis of the solution methods for infinite QBDs; most of the algorithms presented in Section 15.8.3 have been extended to the treatment of finite QBDs.

In addition to these methods some authors have developed solution methods especially adapted to QBDs arising from PH|PH|1|K queues (see Section 15.4). Such methods are the Bocharov-Naoumov method [6], two methods proposed by Chakravarthy and Neuts in [9], and the Cyclic-Reduction method [3].

15.9 Appendix: Existence of the fixed point

In general, it is not known for the fixed-point iteration algorithm described in Section 15.2 whether the searched fixed point exists, is unique or will be reached. However, some intermediate results are available for FiFiQueues which we will present here. In the following we give a proof that the fixed point exists for a modified version of the original FiFiQueues algorithm.

15.9.1 Notation and Brouwer's theorem

Given a queueing network with n stations, we define $D \subset \mathbb{R}^{2n}$ where the tuple

$$\left(\left\langle \lambda_{a,1}, c_{a,1}^2 \right\rangle, \ldots, \left\langle \lambda_{a,n}, c_{a,n}^2 \right\rangle\right) \in D$$

gives for each node $i \in \{1,\ldots,n\}$ the traffic descriptor $\left\langle \lambda_{a,i}, c_{a,i}^2 \right\rangle$ of its arrival traffic. Then the operations performed by FiFiQueues during step $k+1$ of the fixed-point iteration can be expressed as a function $H : D \to D$ [47] which computes from the traffic descriptor d^k obtained from step k the new traffic descriptor d^{k+1}, that is,

$$\mathrm{d}^{k+1} = H(\mathrm{d}^k),$$

where d^0 is the initial traffic descriptor used in the iteration. We use the *Brouwer fixed-point theorem* [45] to prove the existence of the fixed point for the function H. It states:

Let $D \subset \mathbb{R}^m$ be a non-empty, closed, convex, and bounded set, and $H : D \to D$ continuous. Then H has a fixed point.

A first proof of the existence of the fixed point has been discussed in [47] for special service processes. The proof presented in the following applies to arbitrary PH renewal service processes. We first show in Section 15.9.2 that the requirements to the set D are met. The continuity of H is shown in Sections 15.9.3–15.9.5.

15.9.2 Properties of D

Lower and upper bounds for the arrival rate $\lambda_{a,i}$ of a node i exist. It holds that

$$0 \leq \lambda_{a,i} \leq \lambda_{max,i}$$

where the upper bound $\lambda_{max,i}$ is the maximum arrival rate that will only be reached if all queueing stations operate with a load of 100%. It is given by

$$\lambda_{\max} = \lambda_{\mathrm{ext}}(\mathrm{I} - \Gamma)^{-1},$$

where Γ is the routing matrix and $\lambda_{ext,i}$ is the rate of the external traffic arriving at node i. As previously explained, FiFiQueues limits the squared coefficient of variation to $\frac{1}{10}$ to prevent the generation of PH distributions with more than 10 states. Originally, no upper bound is provided for the coefficients but we can safely define

$$c_{a,i}^2 := \min(c_{max}^2, c_{a,i}^2), \qquad i = 1, \ldots, n,$$

with $c_{max}^2 = 1000$ without affecting the analysis. We thus obtain that D is the non-empty, closed and convex interval

$$\left[\left(\left\langle 0, \tfrac{1}{10}\right\rangle, \ldots, \left\langle 0, \tfrac{1}{10}\right\rangle\right), \left(\left\langle \lambda_{max,1}, c_{max}^2\right\rangle, \ldots, \left\langle \lambda_{max,1}, c_{max}^2\right\rangle\right)\right] \subset \mathbb{R}^{2n},$$

as required.

15.9.3 Continuity of H

The function H performs for each node the following operations to compute the traffic descriptors for the next iteration in the algorithm:

1. the service operation;
2. the traffic splitting;
3. the traffic merging.

The traffic merging step is a function of the traffic descriptors generated during the splitting operation. The traffic splitting, in turn, is a function of the departure-traffic descriptor as computed by the service operation. An inspection of the involved terms for the traffic merging (Equation (15.1) and (15.2)), the traffic splitting (Equation (15.3)), and the service operation (Equations (15.5), (15.8), and (15.11)) shows that the proof of continuity reduces to the question whether, for a given node, the loss probability π (in case of a finite queue) and the variance σ_0^2 are continuous functions of the arrival traffic $\langle \lambda_a, c_a^2 \rangle$. Since π and σ_0^2 depend on the stationary distribution v of the underlying CTMC we can make use of the following theorem [32] to prove this continuity:

The stationary distribution of a CTMC as function of the transition rates $\lambda_1, \ldots, \lambda_n$ of the generator matrix is continuous for all $\lambda_i > 0, i = 1, \ldots, n$ if the CTMC has exactly one irreducible set of states.

The underlying CTMC of a queueing station has exactly one irreducible set of states since it is a QBD. Then, the question is, how do the transition rates of the generator matrix depend on $\langle \lambda_a, c_a^2 \rangle$? FiFiQueues uses the traffic descriptor to determine a PH renewal process that represents the arrival traffic. This arrival PH process is then combined with the service PH process of the node to construct the generator matrix. For fixed c_a^2 the transition rates of the generator matrix are a continuous function of the arrival rate λ_a. The theorem then yields the continuity of v as function of λ_a. However, varying c_a^2 may cause FiFiQueues to change the size and structure of the

PH representation. Such a change also influences the size and structure of the QBD. As a consequence, the theorem can only be applied for values of c_a^2 that do not cause such a change. We obtain:

1. v is continuous for $c_a^2 > 1$, since then the PH distribution always takes the same, hyper-exponential, form.
2. v is continuous for $c_a^2 \in \left(\frac{1}{m+1}, \frac{1}{m}\right)$, for all $m \in \{1,\ldots,9\}$.

The other cases, i.e., $c_a^2 = \frac{1}{m}, m \in \{1,\ldots,10\}$, have to be separately discussed. In the following we only show the continuity of π. The proof for σ_0^2 is done in a similar way.

15.9.4 Continuity for $c_a^2 = 1$

We show that

$$\lim_{c_a^2 \nearrow 1} \pi(c_a^2) = \pi(c_a^2 = 1) = \lim_{c_a^2 \searrow 1} \pi(c_a^2)$$

which yields the continuity of π around $c_a^2 = 1$.

15.9.4.1 Case $c_a^2 = 1$

If $c_a^2 = 1$, the arrival PH distribution is a negative-exponential distribution with rate λ_a. Following the notation used in Section 15.4.5, we obtain the steady-state probability distribution $\mathrm{v}^{=}$ by solving the global balance equations

$$\left.\begin{aligned} \mathrm{v}_0^{=}(-\lambda_a) + \mathrm{v}_1^{=}\mathrm{B}^0 &= 0 \\ \mathrm{v}_0^{=}\lambda_a\boldsymbol{\beta} + \mathrm{v}_1^{=}(-\lambda_a\mathrm{I} + \mathrm{B}) + \mathrm{v}_2^{=}\mathrm{B}^0\boldsymbol{\beta} &= 0 \\ \mathrm{v}_1^{=}\lambda_a\mathrm{I} + \mathrm{v}_2^{=}(-\lambda_a\mathrm{I} + \mathrm{B}) + \mathrm{v}_3^{=}\mathrm{B}^0\boldsymbol{\beta} &= 0 \\ &\cdots \\ \mathrm{v}_{\mathrm{K}-1}^{=}\lambda_a\mathrm{I} + \mathrm{v}_{\mathrm{K}}^{=}\mathrm{B} &= 0 \end{aligned}\right\} \tag{15.1}$$

and

$$\mathrm{v}^{=} \cdot 1 = 1,$$

where $(\boldsymbol{\beta}, \mathrm{B})$ is the service PH process and K is the queueing capacity. The loss probability $\pi^{=}$ is then given as

$$\pi^{=} = \pi(c_a^2 = 1) = \frac{1}{\lambda_a}\mathrm{v}_{\mathrm{K}}^{=}(\lambda_a \otimes \mathrm{I}) \cdot 1 = \mathrm{v}_{\mathrm{K}}^{=} \cdot 1.$$

15.9.4.2 Case $c_a^2 \searrow 1$

If $c_a^2 > 1$, FiFiQueues selects the PH renewal process (α, A) as representation of the arrival traffic with

$$\mathrm{A} = \begin{pmatrix} -\lambda_0 & 0 \\ 0 & -\lambda_1 \end{pmatrix} \quad \text{and} \quad \alpha = (p, 1-p),$$

where $p = \frac{1}{2} + \frac{1}{2}\sqrt{\frac{c_a^2-1}{c_a^2+1}}$, $\lambda_0 = 2p\lambda_a$ and $\lambda_1 = 2(1-p)\lambda_a$.

Let $\mathrm{v}^{>}$ be the steady-state probability distribution of the resulting QBD. To ease the following calculations we split the components $\mathrm{v}_\mathrm{i}^{>}$, i$= 0, \ldots, K$, of the probability distribution vector into two parts $\mathrm{v}_\mathrm{i}^{>} = (\mathrm{v}_{\mathrm{i}1}^{>}, \mathrm{v}_{\mathrm{i}2}^{>})$ where $\mathrm{v}_{\mathrm{i}1}^{>}$ and $\mathrm{v}_{\mathrm{i}2}^{>}$ are associated with the first resp. the second state of the arrival PH process. The vector $\mathrm{v}^{>}$ is then determined by the following equations:

$$\begin{aligned}
\mathrm{v}_{01}^{>}(-\lambda_0) + \mathrm{v}_{11}^{>}\mathrm{B}^0 &= 0, && (15.2)\\
\mathrm{v}_{02}^{>}(-\lambda_1) + \mathrm{v}_{12}^{>}\mathrm{B}^0 &= 0, && (15.3)\\
\mathrm{v}_{01}^{>}p\lambda_0\beta + \mathrm{v}_{02}^{>}p\lambda_1\beta + \mathrm{v}_{11}^{>}(-\lambda_0\mathrm{I}+\mathrm{B}) + \mathrm{v}_{21}^{>}\mathrm{B}^0\beta &= 0, && (15.4)\\
\mathrm{v}_{01}^{>}(1-p)\lambda_0\beta + \mathrm{v}_{02}^{>}(1-p)\lambda_1\beta + \mathrm{v}_{12}^{>}(-\lambda_1\mathrm{I}+\mathrm{B}) + \mathrm{v}_{22}^{>}\mathrm{B}^0\beta &= 0, && (15.5)\\
\mathrm{v}_{11}^{>}p\lambda_0\mathrm{I} + \mathrm{v}_{12}^{>}p\lambda_1\mathrm{I} + \mathrm{v}_{21}^{>}(-\lambda_0\mathrm{I}+\mathrm{B}) + \mathrm{v}_{31}^{>}\mathrm{B}^0\beta &= 0, && (15.6)\\
\mathrm{v}_{11}^{>}(1-p)\lambda_0\mathrm{I} + \mathrm{v}_{12}^{>}(1-p)\lambda_1\mathrm{I} + \mathrm{v}_{22}^{>}(-\lambda_1\mathrm{I}+\mathrm{B}) + \mathrm{v}_{32}^{>}\mathrm{B}^0\beta &= 0, && (15.7)\\
&\ldots\\
\mathrm{v}_{\mathrm{K}-1,1}^{>}p\lambda_0\mathrm{I} + \mathrm{v}_{\mathrm{K}-1,2}^{>}p\lambda_1\mathrm{I} + \mathrm{v}_{\mathrm{K}1}^{>}((p-1)\lambda_0\mathrm{I}+\mathrm{B}) + \mathrm{v}_{\mathrm{K}2}^{>}p\lambda_1\mathrm{I} &= 0, && (15.8)\\
\mathrm{v}_{\mathrm{K}-1,1}^{>}(1-p)\lambda_0\mathrm{I} + \mathrm{v}_{\mathrm{K}-1,2}^{>}(1-p)\lambda_1\mathrm{I} + & \\
\mathrm{v}_{\mathrm{K}1}^{>}(1-p)\lambda_0\mathrm{I} + \mathrm{v}_{\mathrm{K}2}^{>}(-p\lambda_1\mathrm{I}+\mathrm{B}) &= 0, && (15.9)
\end{aligned}$$

and

$$\mathrm{v}^{>} \cdot 1 = 1.$$

The loss probability $\pi^{>}$ of the station is then given as

$$\pi^{>} = \frac{1}{\lambda_a}\mathrm{v}_\mathrm{K}^{>}(\mathrm{A}^0 \otimes \mathrm{I}) \cdot 1 = \frac{1}{\lambda_a}(\mathrm{v}_{\mathrm{K}1}^{>}\lambda_0 + \mathrm{v}_{\mathrm{K}2}^{>}\lambda_1)\mathrm{I} \cdot 1. \qquad (15.10)$$

Summing Equations (15.2) and (15.3), (15.4) and (15.5), …, gives:

$$\left.\begin{aligned}
s_0 + t_1\mathrm{B}^0 &= 0\\
-s_0\beta + s_1\mathrm{I} + t_1\mathrm{B} + t_2\mathrm{B}^0\beta &= 0\\
-s_1\mathrm{I} + s_2\mathrm{I} + t_2\mathrm{B} + t_3\mathrm{B}^0\beta &= 0\\
&\ldots\\
-s_{K-1}\mathrm{I} + t_K\mathrm{B} &= 0
\end{aligned}\right\} \qquad (15.11)$$

where $s_i = \mathrm{v}_{\mathrm{i}1}^{>}(-\lambda_0) + \mathrm{v}_{\mathrm{i}2}^{>}(-\lambda_1)$ and $t_i = \mathrm{v}_{\mathrm{i}1}^{>} + \mathrm{v}_{\mathrm{i}2}^{>}$. For $c_a^2 \searrow 1$ we have $p \to \frac{1}{2}$. From this, it follows

$$\lim_{c_a^2 \searrow 1} \lambda_0 = \lim_{c_a^2 \searrow 1} \lambda_1 = \lambda_a,$$

and

$$\lim_{c_a^2 \searrow 1} s_i = -\lambda_a \lim_{c_a^2 \searrow 1} t_i, \qquad i = 0, \ldots, k.$$

By applying these limits to Equation (15.11) we observe their correspondence with Equation (15.1). Hence, we obtain for $c_a^2 \searrow 1$:

$$\lim_{c_a^2 \searrow 1} t_i = \lim_{c_a^2 \searrow 1} (\mathrm{v}_{i1}^{>} + \mathrm{v}_{i2}^{>}) = \mathrm{v}_{i}^{=},$$

which provides, with Equation (15.10), the desired relationship

$$\lim_{c_a^2 \searrow 1} \pi^{>} = \pi^{=}.$$

15.9.4.3 Case $c_a^2 \nearrow 1$

If $0.5 < c_a^2 < 1$, the arrival PH distribution is a modified hypo-exponential distribution (α, A) as defined in Section 15.4.5 where

$$\mathrm{A} = \begin{pmatrix} -\lambda_0 & \lambda_0 \\ 0 & -\lambda_1 \end{pmatrix} \quad \text{and} \quad \alpha = (1, 0).$$

Let $\mathrm{v}^{<}$ be the steady-state probability distribution of the resulting QBD. Again, we split the components $\mathrm{v}_i^{<}$, i$= 0, \ldots, K$, of the probability distribution vector into two parts $\mathrm{v}_i^{<} = (\mathrm{v}_{i1}^{<}, \mathrm{v}_{i2}^{<})$, where $\mathrm{v}_{i1}^{<}$ and $\mathrm{v}_{i2}^{<}$ are associated with the first resp. the second state of the arrival PH distribution. The vector $\mathrm{v}^{<}$ is then determined by the following equations:

$$\mathrm{v}_{01}^{<}(-\lambda_0) + \mathrm{v}_{11}^{<}\mathrm{B}^0 = 0, \tag{15.12}$$
$$\mathrm{v}_{01}^{<}\lambda_0 + \mathrm{v}_{02}^{<}(-\lambda_1) + \mathrm{v}_{12}^{<}\mathrm{B}^0 = 0, \tag{15.13}$$
$$\mathrm{v}_{02}^{<}\lambda_1\beta + \mathrm{v}_{11}^{<}(-\lambda_0\mathrm{I} + \mathrm{B}) + \mathrm{v}_{21}^{<}\mathrm{B}^0\beta = 0, \tag{15.14}$$
$$\mathrm{v}_{11}^{<}\lambda_0\mathrm{I} + \mathrm{v}_{12}^{<}(-\lambda_1\mathrm{I} + \mathrm{B}) + \mathrm{v}_{22}^{<}\mathrm{B}^0\beta = 0, \tag{15.15}$$
$$\mathrm{v}_{12}^{<}\lambda_1\mathrm{I} + \mathrm{v}_{21}^{<}(-\lambda_0\mathrm{I} + \mathrm{B}) + \mathrm{v}_{31}^{<}\mathrm{B}^0\beta = 0, \tag{15.16}$$
$$\mathrm{v}_{21}^{<}\lambda_0\mathrm{I} + \mathrm{v}_{22}^{<}(-\lambda_1\mathrm{I} + \mathrm{B}) + \mathrm{v}_{32}^{<}\mathrm{B}^0\beta = 0, \tag{15.17}$$
$$\cdots$$
$$\mathrm{v}_{\mathrm{K}-1,2}^{<}\lambda_1\mathrm{I} + \mathrm{v}_{\mathrm{K}1}^{<}(-\lambda_0\mathrm{I} + \mathrm{B}) + \mathrm{v}_{\mathrm{K}2}^{<}\lambda_1\mathrm{I} = 0, \tag{15.18}$$
$$\mathrm{v}_{\mathrm{K}1}^{<}\lambda_0\mathrm{I} + \mathrm{v}_{\mathrm{K}2}^{<}(-\lambda_1\mathrm{I} + \mathrm{B}) = 0, \tag{15.19}$$

and

$$\mathrm{v}^{<} \cdot 1 = 1.$$

The loss probability $\pi^{<}$ of the station is then given as

$$\pi^{<} = \frac{1}{\lambda_a} \mathrm{v}_{\mathrm{K}}^{<} (\mathrm{A}^0 \otimes \mathrm{I}) \cdot 1 = \frac{1}{\lambda_a} \mathrm{v}_{\mathrm{K2}}^{<} \lambda_1 \mathrm{I} \cdot 1. \tag{15.20}$$

From Equation (15.19) we obtain

$$\mathrm{v}_{\mathrm{K2}}^{<} \lambda_1 \mathrm{I} = \mathrm{v}_{\mathrm{K1}}^{<} \lambda_0 \mathrm{I} + \mathrm{v}_{\mathrm{K2}}^{<} \mathrm{B},$$

which yields with Equation (15.20):

$$\pi^{<} = \frac{1}{\lambda_a} (\mathrm{v}_{\mathrm{K1}}^{<} \lambda_0 \mathrm{I} + \mathrm{v}_{\mathrm{K2}}^{<} \mathrm{B}) \cdot 1. \tag{15.21}$$

Using simple substitutions we derive from Equations (15.12)–(15.19):

$$\left.\begin{aligned} \mathrm{v}_{01}^{<}(-\lambda_0) + \mathrm{v}_{11}^{<} \mathrm{B}^0 &= 0 \\ (\mathrm{v}_{01}^{<} \lambda_0 + \mathrm{v}_{12}^{<} \mathrm{B}^0)\beta + \mathrm{v}_{11}^{<}(-\lambda_0 \mathrm{I} + \mathrm{B}) + \mathrm{v}_{21}^{<} \mathrm{B}^0 \beta &= 0 \\ (\mathrm{v}_{11}^{<} \lambda_0 \mathrm{I} + \mathrm{v}_{12}^{<} \mathrm{B} + \mathrm{v}_{22}^{<} \mathrm{B}^0 \beta) + \mathrm{v}_{21}^{<}(-\lambda_0 \mathrm{I} + \mathrm{B}) + \mathrm{v}_{31}^{<} \mathrm{B}^0 \beta &= 0 \\ (\mathrm{v}_{\mathrm{K}-1,1}^{<} \lambda_0 \mathrm{I} + \mathrm{v}_{\mathrm{K}-1,2}^{<} \mathrm{B} + \mathrm{v}_{\mathrm{K2}}^{<} \mathrm{B}^0 \beta) + \mathrm{v}_{\mathrm{K1}}^{<} \mathrm{B} + \mathrm{v}_{\mathrm{K2}}^{<} \mathrm{B} &= 0 \end{aligned}\right\} \tag{15.22}$$

For $c_a^2 \nearrow 1$ we have $\lambda_0 \to \lambda_a$ and $\lambda_1 \to \infty$. Solving Equations (15.13), (15.15), …, (15.19) for $\mathrm{v}_{\mathrm{i2}}^{<}$ then gives

$$\mathrm{v}_{\mathrm{i2}}^{<} \to 0, \quad i = 0, \ldots, K.$$

By applying these limits to Equation (15.22) we observe their correspondence with Equation (15.1). Hence we obtain for $c_a^2 \nearrow 1$:

$$\mathrm{v}_{\mathrm{i1}}^{<} \to \mathrm{v}_{\mathrm{i}}^{=} \quad \text{and} \quad \mathrm{v}_{\mathrm{i2}}^{<} \to 0, \qquad i = 0, \ldots, K,$$

which gives, applied to Equation (15.21):

$$\lim_{c_a^2 \nearrow 1} \pi^{<} = \pi^{=},$$

as required.

15.9.5 Continuity for $c_a^2 = \frac{1}{m}, m \in \{2, \ldots, 10\}$

The transition rates of the hypo-exponential PH distributions for $c_a^2 < 1$ are defined as functions of the number of phases $m = \left\lceil \frac{1}{c_a^2} \right\rceil$. The inherent discontinuity suggests that the steady-state distribution of the resulting QBD is discontinuous, too. To improve the behavior of the arrival distribution with regard to the continuity, [47] proposes to modify FiFiQueues' PH fitting procedure for $c_a^2 < 1$ as follows. Given the mean inter-arrival time $\mathrm{E}[X] = 1/\lambda_a$ and the squared coefficient of variation c_a^2, we fit the PH distribution (α, A) with $m = \left\lceil \frac{1}{c_a^2} \right\rceil$ phases and initial probability vector

$\alpha = (1,0,\ldots,0)$. The matrix A is given as

$$\mathrm{A} = \begin{pmatrix} -\lambda_0 & \lambda_0 & & & & \\ & -\lambda_1 & \lambda_1 & & & \\ & & \ddots & \ddots & & \\ & & & -\lambda_{m-2} & \lambda_{m-2} & \\ & & & & -\lambda_{m-1} & \end{pmatrix}, \tag{15.23}$$

where

$$\begin{aligned} \lambda_i &= 1/(c_a^2 \mathrm{E}[X]), \text{ for } 0 \le i < m-2, \\ \lambda_{m-2} &= \frac{\lambda_{m-1}}{\mathrm{E}[X]\lambda_{m-1}(1-(m-2)c_a^2)-1}, \\ \lambda_{m-1} &= \frac{1-(m-2)c_a^2+\sqrt{(c_a^2)^2(2m-m^2)+2c_a^2(m-1)-1}}{\mathrm{E}[X]((m-1)(m-2)(c_a^2)^2+c_a^2(3-2m)+1)} \end{aligned}$$

As can be seen, the transition rates λ_i for $i < m-2$ are now continuous functions of c_a^2. This causes that important statistics of this new PH renewal process, e.g., the third moment of the inter-arrival time, are now continuous at values of c_a^2 where a size change happens. Note that this is not true for the original PH distribution. Figure 15.17 illustrates this by comparing the third moment of both PH distributions for values of c_a^2 around 0.5 (i.e., when the size m changes from 3 to 2).

For $c_a^2 \nearrow \frac{1}{m}$, $m \in \{1,\ldots,9\}$, the new PH distribution (with $m+1$ phases) yields

$$\lambda_i \longrightarrow m\lambda_a, \qquad \text{for } 0 \le i < m, \tag{15.24}$$

$$\lambda_m \longrightarrow \infty. \tag{15.25}$$

Hence, the proof of $\lim_{c_a^2 \nearrow 1} \pi(c_a^2) = \pi(c_a^2 = 1)$ for $m = 1$, as given in the previous section, is still valid and we only have to prove that

$$\lim_{c_a^2 \nearrow \frac{1}{m}} \pi(c_a^2) = \pi(c_a^2 = \frac{1}{m})$$

for $m \in \{2,\ldots,9\}$. For $c_a^2 = \frac{1}{m}$ the arrival PH distribution $(\alpha_=, \mathrm{A}_=)$ is an Erlang-m distribution. Again, let $\mathrm{v}^=$ be the steady-state probability vector of the resulting QBD with $c_a^2 = \frac{1}{m}$, $m \in \{2,\ldots,10\}$. We split the components $\mathrm{v}_\mathrm{i}^=$, i$= 0,\ldots,K$, of the probability distribution vector into m parts $\mathrm{v}_\mathrm{i}^= = (\mathrm{v}_{\mathrm{i}1}^=,\ldots,\mathrm{v}_{\mathrm{i}m}^=)$ where $\mathrm{v}_{\mathrm{i}j}^=$ is associated with the j-th state of the arrival PH distribution. The probabilities are determined by $\mathrm{v}^= \cdot 1 = 1$ and by the global balance equations of the QBD. For level 0 of the QBD, when no customers are in the queueing station, we obtain

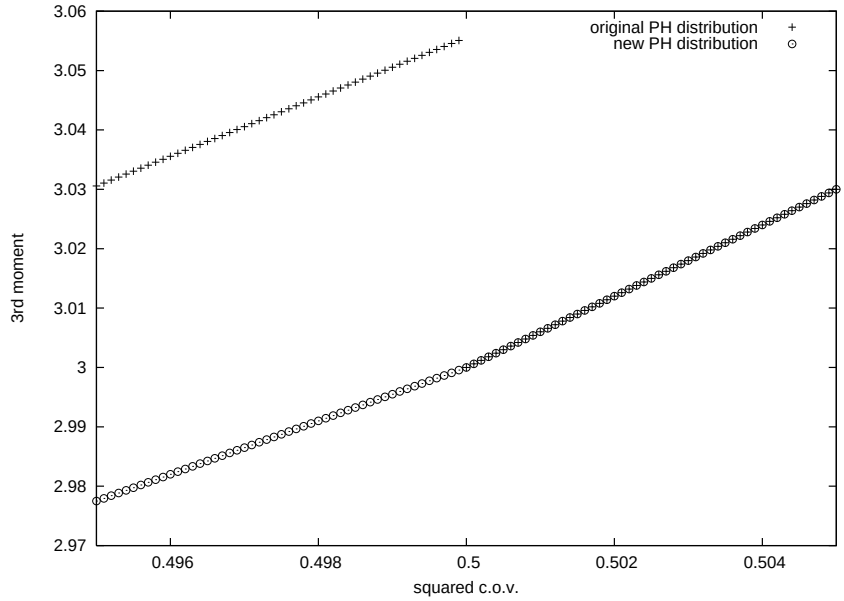


Fig. 15.17: Third moment of the original (Equation (15.4)) and the modified (Equation (15.23)) PH distribution as function of c_a^2

$$\begin{aligned} \bar{\bar{v}}_{01}(-\bar{\bar{\lambda}}) + \bar{\bar{v}}_{11}B^0 &= 0, \\ \bar{\bar{v}}_{01}\bar{\bar{\lambda}} + \bar{\bar{v}}_{02}(-\bar{\bar{\lambda}}) + \bar{\bar{v}}_{12}B^0 &= 0, \\ &\dots \\ \bar{\bar{v}}_{0,m-1}\bar{\bar{\lambda}} + \bar{\bar{v}}_{0m}(-\bar{\bar{\lambda}}) + \bar{\bar{v}}_{1m}B^0 &= 0, \end{aligned}$$

where $\bar{\bar{\lambda}} = m\lambda_a$. For level $1 \leq i < K$, we have

$$\begin{aligned} \bar{\bar{v}}_{i-1,m}\bar{\bar{\lambda}}\beta + \bar{\bar{v}}_{i1}(-\bar{\bar{\lambda}}I + B) + \bar{\bar{v}}_{i+1,1}B^0\beta &= 0, \\ \bar{\bar{v}}_{i1}\bar{\bar{\lambda}}I + \bar{\bar{v}}_{i2}(-\bar{\bar{\lambda}}I + B) + \bar{\bar{v}}_{i+1,2}B^0\beta &= 0, \\ &\dots \\ \bar{\bar{v}}_{i,m-1}\bar{\bar{\lambda}}I + \bar{\bar{v}}_{im}(-\bar{\bar{\lambda}}I + B) + \bar{\bar{v}}_{i+1,m}B^0\beta &= 0, \end{aligned}$$

and, finally, for level K:

$$\begin{aligned} \bar{\bar{v}}_{K-1,m}\bar{\bar{\lambda}}I + \bar{\bar{v}}_{K1}(-\bar{\bar{\lambda}}I + B) + \bar{\bar{v}}_{Km}\lambda_{m-1}I &= 0, \\ \bar{\bar{v}}_{K1}\bar{\bar{\lambda}}I + \bar{\bar{v}}_{K2}(-\bar{\bar{\lambda}}I + B) &= 0, \\ &\dots \\ \bar{\bar{v}}_{K,m-1}\bar{\bar{\lambda}}I + \bar{\bar{v}}_{Km}(-\bar{\bar{\lambda}}I + B) &= 0. \end{aligned}$$

The loss probability $\pi^{=}$ of the station is given by

$$\pi^{=} = \frac{1}{\lambda_a} \mathrm{v}_{\mathrm{K}}^{=} (\mathrm{A}_{=}^{0} \otimes \mathrm{I}) \cdot 1 = \frac{1}{\lambda_a} \mathrm{v}_{\mathrm{Km}}^{=} \lambda^{=} \mathrm{I} \cdot 1.$$

For $\frac{1}{m+1} \leq c_a^2 < \frac{1}{m}$ the resulting arrival PH process $(\alpha_<, \mathrm{A}_<)$ has $m+1$ states and the steady-state probability vector $\mathrm{v}^<$ of the QBD has additional components $\mathrm{v}_{\mathrm{i,m+1}}^{<}, i = 0, \ldots, K$. Let $\lambda_0, \ldots, \lambda_{m+1}$ be the transition rates of the PH process. The global balance equations for level 0 are

$$\begin{aligned} \mathrm{v}_{01}^{<}(-\lambda_0) + \mathrm{v}_{11}^{<}\mathrm{B}^0 &= 0, \\ \mathrm{v}_{01}^{<}\lambda_0 + \mathrm{v}_{02}^{<}(-\lambda_1) + \mathrm{v}_{12}^{<}\mathrm{B}^0 &= 0, \\ &\cdots \\ \mathrm{v}_{0\mathrm{m}}^{<}\lambda_{m-1} + \mathrm{v}_{0,\mathrm{m+1}}^{<}(-\lambda_m) + \mathrm{v}_{1,\mathrm{m+1}}^{<}\mathrm{B}^0 &= 0. \end{aligned}$$

For level $1 \leq i < K$, we have

$$\begin{aligned} \mathrm{v}_{\mathrm{i-1,m+1}}^{<}\lambda_m \boldsymbol{\beta} + \mathrm{v}_{\mathrm{i1}}^{<}(-\lambda_0 \mathrm{I} + \mathrm{B}) + \mathrm{v}_{\mathrm{i+1,1}}^{<}\mathrm{B}^0\boldsymbol{\beta} &= 0, \\ \mathrm{v}_{\mathrm{i1}}^{<}\lambda_0 \mathrm{I} + \mathrm{v}_{\mathrm{i2}}^{<}(-\lambda_1 \mathrm{I} + \mathrm{B}) + \mathrm{v}_{\mathrm{i+1,2}}^{<}\mathrm{B}^0\boldsymbol{\beta} &= 0, \\ &\cdots \\ \mathrm{v}_{\mathrm{im}}^{<}\lambda_{m-1}\mathrm{I} + \mathrm{v}_{\mathrm{i,m+1}}^{<}(-\lambda_m \mathrm{I} + \mathrm{B}) + \mathrm{v}_{\mathrm{i+1,m+1}}^{<}\mathrm{B}^0\boldsymbol{\beta} &= 0, \end{aligned}$$

and for level K:

$$\mathrm{v}_{\mathrm{K-1,m+1}}^{<}\lambda_m \mathrm{I} + \mathrm{v}_{\mathrm{K1}}^{<}(-\lambda_0 \mathrm{I} + \mathrm{B}) + \mathrm{v}_{\mathrm{K,m+1}}^{<}\lambda_m \mathrm{I} = 0 \tag{15.26}$$

$$\mathrm{v}_{\mathrm{K1}}^{<}\lambda_0 \mathrm{I} + \mathrm{v}_{\mathrm{K2}}^{<}(-\lambda_1 \mathrm{I} + \mathrm{B}) = 0 \tag{15.27}$$

$$\cdots$$

$$\mathrm{v}_{\mathrm{Km}}^{<}\lambda_{m-1}\mathrm{I} + \mathrm{v}_{\mathrm{K,m+1}}^{<}(-\lambda_m \mathrm{I} + \mathrm{B}) = 0. \tag{15.28}$$

The loss probability $\pi^<$ is given by

$$\pi^{<} = \frac{1}{\lambda_a} \mathrm{v}_{\mathrm{K}}^{<} (\mathrm{A}_{<}^{0} \otimes \mathrm{I}) \cdot 1 = \frac{1}{\lambda_a} \mathrm{v}_{\mathrm{K,m+1}}^{<} \lambda_m \mathrm{I} \cdot 1. \tag{15.29}$$

From Equation (15.28) we obtain

$$\mathrm{v}_{\mathrm{K,m+1}}^{<}\lambda_m \mathrm{I} = \mathrm{v}_{\mathrm{Km}}^{<}\lambda_{m-1}\mathrm{I} + \mathrm{v}_{\mathrm{K,m+1}}^{<}\mathrm{B},$$

which yields with Equation (15.29):

$$\pi^{<} = \frac{1}{\lambda_a}(\mathrm{v}_{\mathrm{Km}}^{<}\lambda_{m-1}\mathrm{I} + \mathrm{v}_{\mathrm{K,m+1}}^{<}\mathrm{B}) \cdot 1. \tag{15.30}$$

Solving the global balance equations for $\mathrm{v}_{\mathrm{i,m+1}}^{<}$ and applying the limits from (15.24) and (15.25) gives

$$\mathrm{v}_{\mathrm{i,m+1}}^{<} \to 0, \quad i = 0, \ldots, K.$$

Similar to the case $c_a^2 \nearrow 1$ of the original PH distribution (see Equation (15.22)), transforming the global balance equations finally yields

$$\begin{aligned} \mathrm{v}_{\mathrm{ij}}^{<} &\to \mathrm{v}_{\mathrm{ij}}^{=}, \\ \mathrm{v}_{\mathrm{i,m+1}}^{<} &\to 0, \qquad\qquad i = 0,\ldots,K \text{ and } j = 1,\ldots,m, \end{aligned}$$

which gives, applied to Equation (15.30):

$$\lim_{c_a^2 \nearrow \frac{1}{m}} \pi^{<} = \pi^{=},$$

as required.

We now have shown that the loss probability π is a continuous function of c_a^2. Together with the continuity of the variance σ_0^2 (not shown here), this yields the continuity of the the function H, and, as consequence, the existence of the fixed point.

Note that experiments have shown that the original FiFiQueues and the modified version using the new hypo-exponential PH distributions compute nearly identical results with relative errors of less than 10^{-4}.

References

1. Akar, N., Sohraby, K.: An invariant subspace approach in M|G|1 and G|M|1 type Markov chains. Communications in Statistics: Stochastic Models **13(3)**, 251–257 (1997)
2. Apache Software Foundation: Apache HTTP Server Project. http://httpd.apache.org/
3. Bini, D., Chakravarthy, S., Meini, B.: A new algorithm for the design of finite capacity service units. In: Numerical Solution of Markov Chains (NSMC'99), pp. 247–260. Prensas Universitarias de Zaragoza (1999)
4. Bini, D., Meini, B.: On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems. In: Proceedings of the Second International Workshop on Numerical Solution of Markov Chains, pp. 21–38. Raleigh, North Carolina (1995)
5. Bocharov, P.: Analysis of the queue length and the output flow in single server with finite waiting room and phase type distributions. Problems of Control and Information Theory **16**(3), 211–222 (1987)
6. Bocharov, P., Naoumov, V.: Matrix-geometric stationary distribution for the PH/PH/1/r queue. Elektronische Informationsverarbeitung und Kybernetik **22(4)**, 179–186 (1986)
7. Burke, P.: The output of a queueing system. Operations Research **4**, 699–704 (1956)
8. Chakka, R.: Performance and reliability modelling of computing systems using spectral expansion. Ph.D. thesis, University of Newcastle upon Tyne (1995)
9. Chakravarthy, S., Neuts, M.: Algorithms for the design of finite-capacity service units. Naval Research Logistics **36**, 147–165 (1989)
10. Fischer, W., Meier-Hellstern, K.: The Markov-modulated Poisson process (MMPP) cookbook. Performance Evaluation **18**, 149–171 (1992)
11. Gribble, S.: UC Berkeley Home IP HTTP Traces. http://www.acm.org/sigcomm/ITA/

12. Harrison, J., Nguyen, V.: The QNET method for two-moment analysis of open queueing networks. Queueing Systems: Theory and Applications **6**(1), 1–32 (1990)
13. Haverkort, B.: Approximate analysis of networks of PH|PH|1|K queues: Theory & tool support. In: H. Beilner, F. Bause (eds.) MMB, *Lecture Notes in Computer Science*, vol. 977, pp. 239–253. Springer (1995)
14. Haverkort, B.: QNAUT: Approximately analyzing networks of PH|PH|1|K queues. Proceedings of the 1996 International Computer Performance and Dependability Symposium p. 57 (1996)
15. Haverkort, B.: Approximate analysis of networks of PH|PH|1|K queues with customer losses: Test results. Annals of Operations Research **79**, 271–291 (1998)
16. Haverkort, B.: Performance of Computer Communication Systems — A Model-Based Approach. John Wiley & Sons (1998)
17. Heindl, A.: Decomposition of general tandem queueing networks with mmpp input. Performance Evaluation **44**(1-4), 5–23 (2001)
18. Heindl, A.: Traffic-based decomposition of general queueing networks with correlated input processes. Ph.D. thesis, Institut für Technische Informatik, Technische Universität Berlin (2001)
19. Heindl, A.: Decomposition of general queueing networks with MMPP inputs and customer losses. Performance Evaluation **51**(2-4), 117–136 (2003)
20. Heindl, A., Mitchell, K., van de Liefvoort, A.: Correlation bounds for second-order MAPs with application to queueing network decomposition. Performance Evaluation **63**(6), 553 – 577 (2006). Modelling Techniques and Tools for Computer Performance Evaluation
21. Heindl, A., Telek, M.: Output models of MAP/PH/1(/K) queues for an efficient network decomposition. Performance Evaluation **49**(1/4), 321–339 (2002)
22. Jackson, J.: Networks of waiting lines. Operations Research **5**, 518–521 (1957)
23. Johnson, M., Taaffe, M.: Matching moments to phase distributions: Mixtures of Erlang distributions of common order. Communications in Statistics: Stochastic Models **5**(4), 711–743 (1989)
24. Kim, S., Muralidharan, R., O'Cinneide, C.: Taking account of correlations between streams in queueing network approximations. Queueing Systems: Theory and Applications **49**(3–4), 261–281 (2005)
25. Koschel, T.: Modellierung und Bewertung von Verteilten Web-Servern. Diploma thesis, Lehr- und Forschungsgebiet Informatik 4, RWTH Aachen (2002)
26. Krämer, W., Langenbach-Belz, M.: Approximate Formulae for the Delay in the Queueing System GI/G/1. In: Proceedings of the 8th International Teletraffic Congress, pp. 235–1/8 (1976)
27. Kühn, P.: Approximate analysis of general queueing networks by decomposition. IEEE Transactions on Communications **27**(1), 113–126 (1979)
28. Latouche, G., Ramaswami, V.: A logarithmic reduction algorithm for quasi birth and death processes. Journal of Applied Probability **30**, 650–674 (1993)
29. Lucantoni, D.: New results on the single server queue with a batch Markovian arrival process. Commun. Statist.- Stochastic Models **7**(1), 1–46 (1991)
30. Lucantoni, D., Choudhury, G., Whitt, W.: Computing transient distributions in general single-server queues. In: Global Telecommunications Conference (GLOBECOM '93), pp. 1045–1050. IEEE (1993)
31. Lucantoni, D., Meier-Hellstern, K., Neuts, M.: A single server queue with server vacations and a class of non-renewal arrival processes. Advances in Applied Probability **22**, 676–705 (1990)
32. Mainkar, V., Trivedi, K.: Sufficient conditions for existence of a fixed point in stochastic reward net-based iterative models. IEEE Transactions of Software Engineering **22**(9), 640–653 (1996)
33. Marshall, K.: Some inequalities in queueing. Operations Research **16**, 651–665 (1968)
34. Naoumov, V., Krieger, U., Wagner, D.: Analysis of a multi-server delay-loss system with a general Markovian arrival process. Matrix-Analytical Methods in Stochastic Models **183**, 43–66 (1996)

35. Network Working Group: RFC 1945. Hypertext Transfer Protocol – HTTP/1.0. `http://www.w3.org/Protocols/rfc1945/rfc1945` (1996)
36. Neuts, M.: A versatile Markovian point process. Journal of Applied Probability **16**(2), 764–779 (1979)
37. Neuts, M.: Matrix-Geometric Solutions in Stochastic Models — An Algorithmic Approach. Dover Publications, Inc. (1994)
38. Ost, A.: Performance of communication systems – a model-based approach with matrix-geometric methods. Ph.D. thesis, Lehr- und Forschungsgebiet Informatik 4, RWTH Aachen (2001)
39. Ruemmler, C., Wilkes, J.: An introduction to disk drive modeling. IEEE Computer **27**(3), 17–29 (1994)
40. Sadre, R.: Decomposition-based analysis of queueing networks. Ph.D. thesis, University of Twente (2006)
41. Sadre, R., Haverkort, B.: FiFiQueues: fixed-point analysis of queueing networks with finite-buffer stations. In: MMB (Kurzvorträge), vol. 99-16, pp. 77–80. Universität Trier (1999)
42. Sadre, R., Haverkort, B.: FiFiQueues: fixed-point analysis of queueing networks with finite-buffer stations. In: Computer Performance Evaluation. Modelling Techniques and Tools: 11th International Conference, TOOLS 2000, *Lecture Notes in Computer Science*, vol. 1786, pp. 324–327. Springer (2000)
43. Sadre, R., Haverkort, B., Ost, A.: An efficient and accurate decomposition method for open finite- and infinite-buffer queueing networks. In: W. Stewart, B. Plateau (eds.) Proc. 3rd Int. Workshop on Numerical Solution of Markov Chains, pp. 1–20. Zaragosa University Press (1999)
44. Sadre, R., Haverkort, B., Reinelt, P.: A fixed-point algorithm for closed queueing networks. In: K. Wolter (ed.) Formal Methods and Stochastic Models for Performance Evaluation, Fourth European Performance Engineering Workshop, EPEW 2007, *Lecture Notes in Computer Science*, vol. 4748, pp. 154–170. Springer (2007)
45. Samelson, H.: On the Brouwer Fixed Point Theorem. Portugaliae mathematica **22**(4), 189–191 (1963)
46. Schuba, M., Haverkort, B., Schneider, G.: Performance evaluation of multicast communication in packet-switched networks. Performance Evaluation **39**(1-4), 61–80 (2000)
47. Tartemann, D.: Untersuchung der Existenz eines Fixpunktes in einem iterativen Verfahren zur Warteschlangenanalyse. Diploma thesis, Lehr- und Forschungsgebiet Informatik 4, RWTH Aachen (2002)
48. Weerstra, A.: Using matrix-geometric methods to enhance the QNA method for solving large queueing networks. Diploma thesis, Department of Computer Science, University of Twente (1994)
49. Whitt, W.: The Queueing Network Analyzer. The Bell System Technical Journal **62**(9), 2779–2815 (1983)
50. Whitt, W.: Performance of The Queueing Network Analyzer. The Bell System Technical Journal **62**(9), 2817–2843 (1983)