

Call Centre Architecture and Solutions

Mag. Christian Dombacher (BDD)
Nikolaus Lenaugasse 8
A-2232 Deutsch-Wagram

22.06.1999

Call Centre Architecture and Solutions
1999 - 2016
(c) by Ing. DI Mag. Ing. Christian Dombacher

Call Centre Architecture and Solutions
is licensed under a
Creative Commons Attribution-NonCommercial 4.0 International License
<http://creativecommons.org/licenses/by-nc/4.0/>

Attribution-NonCommercial 4.0 International

=====

Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. More considerations for licensors:
wiki.creativecommons.org/Considerations_for_licensors

Considerations for the public: By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the licensor's permission is not necessary for any reason--for example, because of any applicable exception or limitation to copyright--then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More considerations for the public:
wiki.creativecommons.org/Considerations_for_licensees

=====

Creative Commons Attribution-NonCommercial 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-NonCommercial 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 -- Definitions.

- a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.
- b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.
- c. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.
- d. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.
- e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.
- f. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.
- g. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.
- h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.
- i. NonCommercial means not primarily intended for or directed towards

commercial advantage or monetary compensation. For purposes of this Public License, the exchange of the Licensed Material for other material subject to Copyright and Similar Rights by digital file-sharing or similar means is NonCommercial provided there is no payment of monetary compensation in connection with the exchange.

- j. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.
- k. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.
- l. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 -- Scope.

a. License grant.

- 1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:
 - a. reproduce and Share the Licensed Material, in whole or in part, for NonCommercial purposes only; and
 - b. produce, reproduce, and Share Adapted Material for NonCommercial purposes only.
- 2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.
- 3. Term. The term of this Public License is specified in Section 6(a).
- 4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.
- 5. Downstream recipients.

- a. Offer from the Licensor -- Licensed Material. Every

recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

- b. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

- 6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

- 1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
- 2. Patent and trademark rights are not licensed under this Public License.
- 3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties, including when the Licensed Material is used other than for NonCommercial purposes.

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

- 1. If You Share the Licensed Material (including in modified form), You must:
 - a. retain the following if it is supplied by the Licensor with the Licensed Material:
 - i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
 - ii. a copyright notice;

- iii. a notice that refers to this Public License;
 - iv. a notice that refers to the disclaimer of warranties;
 - v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
- b. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
 - c. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.
- 2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
 - 3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.
 - 4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

- a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database for NonCommercial purposes only;
- b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and
- c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 -- Disclaimer of Warranties and Limitation of Liability.

- a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS,

ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

- b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.
- c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 -- Term and Termination.

- a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.
- b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:
 - 1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
 - 2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

- c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.
- d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 -- Other Terms and Conditions.

- a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.
- b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

- a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose

conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

- b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.
- c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.
- d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

=====

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the "Licensor." The text of the Creative Commons public licenses is dedicated to the public domain under the CC0 Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.

Contents

1	Introduction and Concepts	9
1.1	Applications	10
1.2	Call Management	11
1.2.1	Inbound Call Management	11
1.2.2	Outbound Call Management	12
1.2.3	Call Blending	12
1.3	Call Routing and Distribution	13
1.3.1	Split Groups	13
1.3.2	Skill Based Routing	14
1.3.3	Agent vs. Call Selection	14
1.4	Call Centre Resources	15
1.4.1	Call Centre Agent	15
1.4.2	Call Centre Supervisor	18
1.5	Call Switching	18
1.5.1	Circuit Switching	19
1.5.2	Packet Switching	22
1.6	Call Centre Performance	27
2	Call Centre Architecture	29
2.1	Adjunct ACD	32
2.2	Integrated ACD	34
2.3	Switch Architecture	35
2.3.1	Functions of a Classic Switching System	35
2.3.2	Evolution of Switching Systems	36
2.3.3	Generic Central Switch Model	39
2.3.4	Distributed Communication Platform	43
2.3.5	Analysis of Switching Networks	47
2.4	Interactive Voice Response Integration	53

2.5	Additional Call Centre Adjuncts	55
2.5.1	Call Accounting System	55
2.5.2	Reader Boards	57
3	Queueing Theory	59
3.1	Introduction to Queueing Theory	59
3.1.1	History	59
3.1.2	Applications	60
3.1.3	Characterization	61
3.1.4	Use of Statistical Distributions in Queueing Systems	63
3.1.5	* Approximation of Arbitrary Distributions	72
3.1.6	* Renewal Processes	73
3.1.7	Performance Characteristics of Queueing Systems	74
3.1.8	* Little's Law for Distributions	78
3.1.9	Notation	79
3.2	Simple Queueing Models	81
3.2.1	* Idle and Busy Period	87
3.2.2	Capacity Constraints	87
3.2.3	Queueing Disciplines	89
3.3	Birth-Death Process	90
3.3.1	Multiserver Systems	92
3.3.2	Capacity Constraints in Multiserver Systems	95
3.3.3	* Erlang B revisited	97
3.3.4	Customer Impatience	98
3.3.5	Bounded Holding Times	104
3.3.6	Finite Population Models	105
3.3.7	Relation to Markov Chains	107
3.4	Extended Markovian Models	109
3.4.1	Some Useful Relations	109
3.4.2	General Impatience Distribution	111
3.4.3	Retrials and the Orbit Model	117
3.5	Arbitrary Service Processes	124
3.5.1	Single Server Systems	124
3.5.2	Finite Single Server Systems	132
3.5.3	Multiserver Systems	135
3.5.4	Customer Impatience	138
3.5.5	Retrials	139
3.6	Arbitrary Arrival Processes	142

3.6.1	Multiserver Systems	143
3.6.2	Single Server Systems	146
3.6.3	Capacity Constraints	148
3.7	Arbitrary Arrival and Service Processes	150
3.7.1	Single Server Systems	150
3.7.2	Multiserver Systems	152
3.7.3	Customer Impatience	153
3.8	Matrix Analytic Solutions	154
3.8.1	Distribution Theory	155
3.8.2	Single Server Systems	164
3.8.3	Multiserver Systems	173
3.8.4	Quasi Birth-Death Process	185
3.8.5	More General Models	189
3.8.6	Retrials	191
4	Model Parameter Estimation	195
4.1	Data Analysis and Modeling Life Cycle	195
4.2	Concepts in Parameter Estimation	196
4.3	Moment Estimation	198
4.3.1	Classic Approach	198
4.3.2	Iterative Methods for Phase Type Distributions	201
4.3.3	The EC_2 Method	202
4.3.4	Fixed Node Approximation	207
4.4	Maximum Likelihood Estimation	221
4.4.1	Classic Approach	221
4.4.2	EM Algorithm	223
4.5	Bayesian Analysis	226
4.6	Concluding Remarks	229
5	Interconnection of Call Centres	231
5.1	Types of Call Centre Integration	231
5.1.1	Distributed Call Centres	231
5.1.2	Virtual Call Centre	232
5.2	Signaling Systems	232
5.2.1	Traditional Signaling	235
5.2.2	ISDN	236
5.2.3	QSig	263
5.2.4	IP Networks with Binary Signaling	268

5.2.5	A New Millenium of Signaling	288
5.2.6	IP Networks with Text Based Signaling	289
5.2.7	IP Networks with XML Based Signaling	296
5.2.8	Notes on Performance Engineering	304
5.3	From Distributed to Virtual Call Centres	305
5.4	Signaling between Call Centres	309
5.4.1	Classic Telephony	309
5.4.2	Multimedia Call Centres	312
5.5	Call Centre to Network Signaling	314
5.5.1	Classic Telephony	314
5.5.2	Multimedia Call Centres	318
6	Analytical Call Centre Modeling	321
6.1	Call Centre Perspective	322
6.2	Stochastic Traffic Assessment	324
6.2.1	Data Sources and Measurement	324
6.2.2	Traffic Processes	326
6.3	Call Flow Model	335
6.4	Call Distribution	338
6.4.1	Classic Disciplines	338
6.4.2	Call Priorities	342
6.4.3	Multiple Skills	344
6.4.4	Call and Media Blending	346
6.5	Resource Modeling	347
6.5.1	Classic Models	348
6.5.2	Impact of Overflow Traffic	351
6.5.3	Markovian State Analysis	355
6.5.4	Matrix Exponential Approach	358
6.5.5	Comparison	373
6.5.6	Extensions and Open Issues	379
6.6	Remarks and Applications	381
7	Call Centre Solutions in Public Networks	385
7.1	The Classic Public Network	385
7.2	Common Channel Signaling	386
7.3	The Intelligent Network	386
7.4	IN and TCP/IP (optional)	387
7.5	Prerouting Revisited	387

7.6	Prerouting Extended	388
7.7	IN based Call Handling and Distribution	388
7.8	Performance Analysis	388
A	Stochastic Processes	391
A.1	Introduction	391
A.2	Markov Processes	393
A.3	Markov Chains	394
A.3.1	Homogenous Markov Chains in Discrete Time	394
A.3.2	Homogenous Markov Chains in Continuous Time	398
B	Computer Programs	403
B.1	Library Description	404
B.2	R	408
B.3	Classpad 300	426
B.4	Source Codes	429
C	CTI Protocols and APIs	431
C.1	Adjunct Switch Application Interface	432
C.1.1	Protocol Overview	433
C.1.2	ASAI and UUS	437
C.1.3	Sample ASAI message	439
C.2	Call Processing Language	439
C.3	Edge of CTI	442
D	System Administration Examples	445
D.1	Interconnections	446
D.1.1	Public ISDN access	446
D.1.2	Private Networking	448
D.1.3	Call Routing with AAR, ARS and UDP	453
D.1.4	Caller Privileges	457
D.2	CTI Administration	460
D.2.1	Configuration of the LAN Gateway	460
D.2.2	ASAI Station Administration	463
D.2.3	Shared UUS Configuration	465
E	List of Acronyms	467

Chapter 1

Introduction and Concepts

From a business point of view, a call centre is an entity that combines voice and data communications technology, which allows an organization to implement critical business strategies to reduce costs and increase revenue. It is merely a collection of resources capable of handling customer contacts by means of a telephone or similar device. Designed to accept or initiate a larger call volume, a call centre is typically installed for sales, marketing, technical support and customer service purposes. A group of agents handles incoming and outgoing calls using voice terminals almost always in combination with a personal computers. The latter either operates stand alone or in conjunction with a mainframe or server, which provides more advanced features like screen pop up and intelligent call routing. Such a setup is often referred to as *CTI-enabled call centres*, where CTI stands for *computer telephone integration*. If tightly integrated to serve customers via internet, one often speaks of an *internet call centre*. As a typical application consider a call-back-button available on a customer support website allowing the customer to get in touch with an agent. Such a scenario typically relies on voice over IP technology to establish the call.

The main resources of a call centre are *call centre agents*, *call centre supervisors* and *interactive voice response* ports. These and some other less familiar call centre resources will be discussed in section 1.4. In practice all of them are limited and therefore subject to performance and capacity engineering.

1.1 Applications

With respect to a classification of call centre segments and applications, a vertical/horizontal approach seems to be the most appropriate one. In the vertical direction we often encounter the following segments, which vary in extent and complexity:

- Government and Education - universities, government agencies and hospitals. In this scenario callers often request information and sometimes issue transactions, e.g. students getting registered for a test.
- Transportation - public transport and travel agencies. Information provided include schedules, reservations, billing- and account information.
- Retail and Wholesale - catalog service, ticket offices and hotels. Usage ranges from requesting product information up to purchase and warranty registration of products.
- Banking and Finance - perform tasks such as checking the account balance and initiating the transfer of funds and investments.
- Communication - newspapers, cable television and telephony service providers. Callers can activate or change services and subscriptions such as the prepaid service for cellphones.

After having identified the call centre segments, we now consider common business uses. Even if products or services change, the principle tasks performed by the call centre remain the same. The following list is not exclusive and due to the philosophy and terminology commonly met in practice, some items may overlap.

- Customer Service and Helpdesk Applications - Customer service is more focused on questions about products and order status tracking. As an example consider some tasks related to transportation such as checking travel schedules, booking tickets and reserving seats. All of them may be carried out easily by a customer service call centre. Helpdesk scenarios in turn are more concerned with the technical aspects of products and services, e.g. the support for assembly, setup, repair and replacement. For the latter it is a common policy to initiate a follow-up call to verify, that the replacement part or product has been received.

- Order Processing - Evolved from market needs, call centres provide a convenient way to purchase products and services directly from catalogs, advertising and other promotions. Due to the personal interaction offered by call centre agents, order processing call centres are often considered as being superior in quality compared to internet based order processing systems.
- Account Information and Transaction Processing - Often found in banking and finance, these call centres provide account balances and payment informations. Customers can call to complete transactions, transfer funds, initiate new and modify existing services.
- Telemarketing and Telesales - Any marketing or sales activity conducted over the telephone can be identified as telemarketing. Telesales products are often limited in their complexity, as it should be possible to sell them in an acceptable time frame. On the contrary the information gathered by telemarketers is often used to conduct a field study concerned with more complex products and solutions.
- Sales Support and Account Management - Whereas the former is concerned with the quality of leads and the support of field representatives, responsibility of the latter includes the introduction of special offers and promotions, as well as making proposals and dealing with questions about inventory, delivery schedules and back orders.

1.2 Call Management

1.2.1 Inbound Call Management

When calls arrive at the call centre to be served by a group of agents we are talking about *inbound traffic*. *Inbound call management* refers to methods and procedures for handling this traffic by means of work flow routing and call distribution. Whereas the latter relies on algorithms under human control, the former is concerned with the selection of an appropriate routing path based on parameters given by the *automatic call distributor (ACD)* or received from the telephone network. Common routing criteria include certain load levels, calling and called party telephone numbers (also referred to by the *automatic number identification (ANI)* and *dialed number identification service (DNIS)*). In a CTI-enabled call centre, the calling party number

can be used to perform a database lookup and display the retrieved data on the agents screen even before the call is answered. With respect to the applications mentioned in section 1.1, customer service and order processing are typical inbound call centre applications.

1.2.2 Outbound Call Management

If calls originate from the call centre, we are talking about *outbound traffic*. Accordingly, *outbound call management* relates to the ability to initiate new and maintain existing customer contacts by launching calls. It is also common to use the term *outbound dialing*, which exists in the following flavours:

- *Preview dialing* describes the task of dialing a telephone number on behalf of the agent, if he intends to do so. In either case, whether the call attempt has been successful or not, it is up to the agent to return control to the system by hanging up.
- *Predictive dialing* covers the entire dialing process and the treatment of call progress tones. The call is only connected to the agent, if a person picks up the phone. All non-productive calls including busy line, modems and faxes are screened out or handled separately. Usually less agents are staffed than calls are initiated by the system to maintain a high level of utilization.

With respect to the applications mentioned in section 1.1, telemarketing and telesales are typical outbound call centre applications.

1.2.3 Call Blending

A *blended call centre* is one, where agent groups are allowed to handle calls in both directions. The corresponding management tasks are more than the total of inbound and outbound call management procedures, as incoming traffic should be preferred due to its unpredictable nature. In accepting only a minimal effect in terms of delay for inbound traffic, one has to prevent the outbound dialer from blocking resources. This is best achieved by assigning a small group of agents to incoming calls only.

Very often the term *call blending* is confused with *media blending*, which describes the ability of an ACD to accept and route calls from different sources. So media blending is related to a multimedia call centre handling

emails, faxes, instant messages, video and telephone calls. Interestingly, the corresponding media management also has to define and execute rules for the appropriate handling of real time and non-real time traffic.

So both types of blending require an excessive set of methods and procedures for resource allocation leading to installations of higher complexity. Furthermore, the *blended agent* receives significantly more training than a standard call centre agent. Accordingly, some operators refused to establish call or media blending in their call centres.

1.3 Call Routing and Distribution

Call or workflow routing and *call distribution* relate to the set of rules, which are applied to isolate the most appropriate resource for a specific caller. Call routing is experienced by the customer as being guided through a decision tree. By progressing through that tree, the system provides information to and collects user inputs from the caller. The corresponding realization is often referred to as *routing path*. From a technical point of view, the interactive voice response unit provides the necessary resources. However, in having reached the leafs of the decision tree, the collected information is considered as being sufficiently complete. So call distribution methods take over to determine the most appropriate agent based on agent properties, user input, system load and environmental conditions. With respect to the organization of agents, up to two dimensions are commonly implemented, i.e. split groups and agent skills.

1.3.1 Split Groups

Based on the call centre application and the information collected, the call is routed to a designated group of agents. Such a group of agents is called a *split group* or *split*. The split group is associated with a call distribution algorithm, which is responsible to determine the most appropriate agent for the call under consideration. Each algorithm corresponds to a certain business objective, i.e. one wants the calls to be evenly distributed between all available agents or one is interested in delivering a new call to the most idle agent first. Besides its purpose for call distribution, the split group concept is also important for reporting, call management and workforce planning. At the time of writing there was a common understanding, that an agent

exclusively belongs to a certain split group. Joining another group means to log out from the existing split group. This fact has been very much welcome by workforce planning tools, as it warrants the independence of split groups allowing for unbiased predictions.

1.3.2 Skill Based Routing

Another dimension of routing commonly found in call centre products is based on the *skill* of an agent. Unfortunately there is no common understanding of the terms skill and skill based routing. Some vendors envision skill as a simple filtering criteria, whereas other vendors define an overlapping group concept. The latter has proven to be a very flexible scheme, but it also introduces difficulties to subsequent systems. Especially workforce systems are affected by the inability of some ACDs to provide the necessary association between skill and call in their data streams.

However, skill based routing is a flexible concept well suited to increase the quality of service experienced by the customer. Some vendors even allowed for differentiation within a certain skill and introduced the *skill level*. As an example consider a spanish customer calling a companies' customer service helpdesk. An ACD featuring skill based routing would then attempt to connect the caller to a technician with the highest level in skill "spanish".

1.3.3 Agent vs. Call Selection

By closely inspecting some of the call distribution algorithms mentioned above it turns out, that some are related to quality of service while others aim to balance the call centres load. For example, a fair distribution of calls is simply not relevant, if calls are queued for service. Such a situation is called a *call surplus* as opposed to an *agent surplus*. Consequently we may agree, that a well defined call distribution strategy consists of two methods, one for each surplus scenario, i.e.

- *Agent selection* occurs any time there are more available agents than calls arriving to a split group. There is certainly no queue and so one out of the group of available agents for that split group is selected to serve an arriving call according to some preconfigured distribution strategy. Examples include uniform call distribution, skill based rout-

ing and the allocation of agents with the longest idle time or the least number of answered calls.

- *Call selection* occurs any time there are more calls arriving to a split group than agents available to handle them. A queue has built up and when an agent becomes free again, a call is selected from the queue and assigned to that agent. The most obvious strategies are to select the call with the longest waiting time or the highest priority. If changes between agent and call surplus are very likely, one might also consider to implement agent queues and preassign calls according to an agents' skill.

There are only some exceptions to the rule of specifying two separate strategies, one for each situation. One example is the selection of the most idle agent first, which reveals some natural coincidence for both scenarios.

1.4 Call Centre Resources

Driven by economical factors, call centre operators aim to optimize the trade off between cost and quality of service. Human resources like call centre agent and call centre supervisor are considered to be more cost sensitive as interactive voice reponse ports and communication channels. However, all of them are limited in availability and therefore subject to capacity and performance engineering. Experience and mathematical modeling have shown, that an increased variation in traffic has a bad effect on the overall performance and raises cost. Therefore certain call centre features like queueing and announcements have been introduced leading to smooth traffic. For the current context smoothness means lower variation, and so the number of required agents decreases.

1.4.1 Call Centre Agent

At the time of writing, the call centre agent is the primary resource of a call centre. As for almost every organization, agents are assigned to groups according to criteria such as application, education state and skills. This also applies to other resources, which may then also be considered as agents. Examples include *interactive voice response (IVR) ports*, *world wide web (WWW) channels* and *electronic mails (EMAIL)*. Instead of being connected

to an agent, the caller sends an email or completes a voice/web form. However, all agents whether human or not are subject to classification and allocation.

With respect to the generation of reports suitable for management purposes, the current work state of an agent is tracked in terms of occurrence and time. This leads to a set of states configured as *agent state diagram* or *agent state machine*. The meaning of each state is best described by considering an example. We will assume a call centre active 24 hours a day with agents working in three shifts of 8 hours. When an agent is sick, on holiday or otherwise not available, the corresponding state is called *unstaffed*. At the beginning of his shift, the agent logs into the system putting him in an *auxiliary state (AUX)*. When ready to accept calls, the agent issues a state change to *available*. Given there are calls waiting, one of them is connected to that agent according to the rules previously defined for call distribution. As a consequence, the system initiates a transition to *active* state. After call termination, a call centre agent very often performs some wrap up work. The related agent state is called *after call work (ACW)*. If the end of wrap up work can be foreseen, the agent is put into available state after some predefined period of time. Otherwise the agent has to declare himself ready to accept a new call. In case of exceptional events, e.g. lunch time, breaks and meetings, the agent triggers a change to auxiliary state. This state transition is always carried out manually, because of their unpredictable nature. Recently a new type of exceptional event has been introduced by regulation. In case of CTI-enabled call centres, the agent is forced to take a break from his computerized work. Although time and occurrence are deterministic and therefore highly predictable, it is still an open challenge to incorporate this type of blocking into resource allocation algorithms. Being in an auxiliary state, the agent is allowed to complete the shift by logging out from the system.

All the states just described - unstaffed, AUX, available, active and ACW - are measured, processed and delivered to a reporting engine, which basically generates two types of reports. Real time reports provide an indication of the short run behaviour and historical reports help to identify bottlenecks and problems in the long run. Both are subject to management activities, but both are on a different level and scale.

Up to now only so called *primary agent states* have been discussed. Sometimes it is necessary to gain more information from the system, e.g. the duration an agent was available for other skills due to traffic overload in that

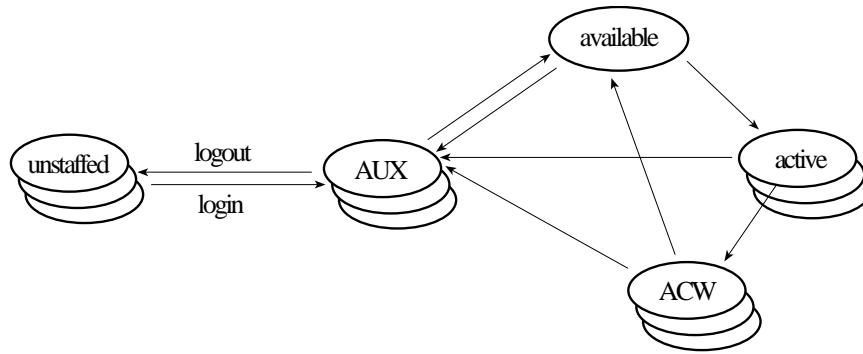


Figure 1.1: Primary and Secondary Agent States

skill. *Secondary agent states* allow for differentiation within a primary state and therefore enable more detailed reporting. The secondary agent states of the active and ACW states are strongly related. Considering a classification in routed calls and direct calls for the former, the same is likely to exist also for the latter. If the caller is aware of an extension to reach a dedicated agent without travelling along the whole call path, the call is referred to as *direct call*. Considering the auxiliary states, they are an appropriate choice for the representation of certain system, error and exceptional conditions in addition to operative requirements. If, for example, the handset is lifted and no speech activity is detected for a certain period, some systems may force a transition to an AUX state. This provides an indication, that something is wrong with the agent. By inspecting the corresponding real time report, the supervisor immediately becomes aware of it and initiates further actions. The same applies to non-human agents. For example, consider an electronic mail server suffering from a hardware failure.

The entire set of states and state transitions as just described is depicted in figure 1.1. Note, that no secondary states have been assigned to the available state, as there are no differences in being available. It's also worth to mention, that some states might be skipped, e.g. in a fully loaded call centre, agents toggle between active and ACW states. Although the agent state model has been described in the context of a call centre it also applies to contact centres, which often implement additional media streams and task scheduling. So the model still remains valid in environments with media blending and channels featuring real time and batch requests. In the latter

case a secondary auxiliary state may be assigned to tasks such as paper work, fax and email handling.

1.4.2 Call Centre Supervisor

Groups of agents are often covered by a *call centre supervisor*, who regulates the call flow. This is usually done by reallocating agents between the groups he is in charge of. Therefore, he can be seen as a part of call centre management. Besides his standard tasks, he has also to deal with exceptional conditions like malicious calls or agent absences. The supervisor has the ability to monitor agents and assign them to other splits and skills. Some modern call centre installations allow for parts of his functionality to be taken over by certain statistical and adaptive methods. This automated *service level supervisor* traces real time reports to take the necessary actions, when one out of a set of predefined thresholds is exceeded. Such actions include the reallocation of existing agents between split or skill groups or the allocation of spare agents to split or skill groups in overload conditions. Accordingly the call centre supervisor is freed from recurring tasks allowing him to take over responsibility for a large group of agents. For non-human agents, a major part of exception handling is offloaded to the network management system and the IT departments in charge of the corresponding system. Using keep-alive packets, *simple network management protocol (SNMP)* together with alerting systems, manual and automatic recovery procedures can be invoked. These in turn may lead to an alternate call flow circumventing the malfunctioning resource.

1.5 Call Switching

In early days of telegraphy transmitting a message from A to C via B resulted in manual intervention by an operator at location B. After having identified the next station in line, he simply retransmitted the entire message. Technical developments in *message switching* led to an exchange unit, which stored messages electronically and forwarded them to the receiving location after the outgoing resource became available. This was the first application of a telecommunication system, which adopted *stored-program-control*. A similar mechanism, called *store-and-forward messaging*, is used in today's email communication. One of the major disadvantages of message switching was

and still is the uncertain delay introduced by the system. The same applied to packet switching, which can be considered a variant of message switching. With the invention of the telephone, it became necessary to connect the circuit of a calling telephone on demand and to maintain this connection for the entire duration of the call. This real time behaviour was covered by a technique called *circuit switching*. In opposite to message switching, circuit switched calls cannot be stored and are considered *lost* or *blocked*.

Whatever call centre architecture is assumed, both circuit and packet switching provide the foundation for any transmission carried out. Some people might argue, that in recent installations circuit switching is of no use anymore. But experience has shown, that circuit switching concepts are still substantial for the transport of real time traffic.

1.5.1 Circuit Switching

The evolution of circuit switched systems revealed various different systems, which can be grouped into manual, analog and digital switching systems. In the earliest form of switchboard, an operator made a connection by inserting a brass peg where appropriate vertical and horizontal bars crossed. As incoming circuits were connected to vertical metal bars and outgoing links were connected to horizontal metal bars, a connection was established. If more operators were involved in setting up a call, they used a separate line to communicate with each other. Known as *order-wire working*, this type of signaling can be seen as the forerunner of today's *common channel signaling*.

With the invention of the two motion selector by Almon B. Strowger operators were eliminated and the caller became responsible for providing address information by simply dialing a number. Each dialed digit was and in some countries still is represented by pulses, one pulse for digit one, two pulses for digit two up to ten pulses for zero. The two motion selector was able to address one out of hundred outgoing trunks by a two digit combination. As a network set up only by two motion selectors would have been too expensive, line *concentration* had to be introduced. Each line was provided with a much cheaper single motion switch called *uniselector* or *linefinder* choosing the next free two motion selector on receiving a seize signal. Dialing before this process had stopped did not have any effect, so a *dial tone* as a proceed signal was sent back to the caller.

Especially in large cities network designers were concerned about the raising complexity of linked numbering schemes. This led to invention of *reg-*

isters, which determined the route based on the dialed number. As registers were only used a short period of time at the beginning of a call, a few units were sufficient to serve a large number of customers.

The next generation of telephone systems introduced a central processing unit, which controlled a set of relays. One such example is the Bell No.1 ESS system. As every processing occurred centrally, more and more features were incorporated in these switching systems, e.g. repeating the last call, conference call and charge advice. The same approach of *stored-program control* is used in today's switches, but instead of switching analog signals by the use of relays, digital streams are being processed by *space division switches*, *time division switches* or a combination of these two. The same concept used in space division switching also occurred in the good old crossbar systems. Each input line is physically connected to an output link and this connection is held for its entire duration. With space division switching, transmission data can be of both digital and analog nature.

Time division switching is best described as making a connection over the same physical path, but at different instants of time. Before time division switching can occur, the analog data need to be sampled and converted to digital packets by some modulation scheme such as *pulse code modulation (PCM)* or *adaptive difference pulse code modulation (ADPCM)*. The latter is a recommendation of the ITU-T (G.722) designed to achieve a higher transmission efficiency than PCM. It is important, that the length of each packet is proportional to the duration of the part of transmission it represents. From a technical perspective, a time division switch or *time division multiplexer (TDM)* is commonly implemented as a hierarchy of multiplexing units operating synchronously in time. Equidistant packets also called *time slots* are arranged in *channels*, which are routed from the input line to the output link by some control mechanism. If k channels are multiplexed onto a single line, time slots $1, 1 + k, 1 + 2k, \dots, 1 + nk$ belong to channel 1, time slots $2, 2 + k, 2 + 2k, \dots, 2 + nk$ belong to channel 2, etc. In order to avoid collisions or queueing for output channels, time division switching is often combined with space division switching. According to [149], time division switching is realized by one of the following technologies:

1. *TDM Bus Switching* - All lines are connected to a common bus. Equidistant packets of data placed in an input buffer are taken out and sent to an output link frequently by a common multiplexer. Being the simplest form of time division switching, TDM bus switching is often im-

plemented in *private branch exchanges (PBX)* designed for enterprise usage.

2. *Time slot interchange (TSI)* - As suggested by the name, routine is achieved by simply reordering the time slots in the input stream. Working in both directions, this is a full duplex operation. TSI is often used as a building block in multistage switches.
3. *Time multiplex switching (TMS)* - This arrangement is already a combination of time and space division switching. Multiple input and output TSI blocks are interconnected using a space division multiplexer, which operates at TSI speed. Accordingly its configuration may change for each time slot. As a consequence, TMS avoids some problems of bandwidth and delay common to TSI.

The concept of TMS has been further developed leading to more general combinations of time and space division switching modules, which will be discussed in more detail in section 2.3.3. Getting technically less specific, we now attempt to identify the main characteristics of circuit switching:

- Channels of fixed bandwidth with low and constant delay are offered.
- In the connection setup phase the routing path is determined before data are actually transmitted.
- Data transmission is transparent to the network, e.g. the network does not attempt to correct transmission errors.
- Overload is handled by rejecting further connection requests to the network. Therefore no delay or bandwidth degradation occurs.
- Highest efficiency is achieved for connections with a long holding time compared to the setup time, and demanding a fixed bandwidth and a low delay.

Some applications might require simultaneous connections to be set up. One example is video conferencing featuring separate channels for voice, video and data transmission. If voice and video streams are conveyed on separate paths through a network of switches, a relative delay (*skew*) between the media streams might be the consequence. As a solution, *multi-rate circuit*

switching or *wideband switching* has been implemented in some switches. Channels are combined and transmitted over the network as a single connection with higher bandwidth. Usually, connections are offered at a fixed bandwidth called the *rate* in circuit switched networks. One example is the *integrated services digital network (ISDN)* operating 64 kbit/sec channels. These channels are grouped together allowing the overall bandwidth demand to be satisfied.

Unfortunately not all problems of *quality of service (QoS)* have been addressed by multi-rate circuit switching. In fact, there is neither a guarantee for the skew variation occurring in a transmission nor a way to handle bursty sources. To cope with the excessive bandwidth requirement of the latter, a new connection may be assigned to each burst provided the connection setup is fast enough. Another approach is suggested by *time assignment speech interpolation (TASI)*, a multiplexing technique used on expensive analogue transmission links such as overseas links. Transmission channels are only allocated during periods of speaker activity also called *talkspurts*. This results in an increase of the number of sources handled by a single transmission link. If sufficiently enough sources are multiplexed, the impact of bursts averages out. In relying on a large number of sources, TASI is nothing else than another instance of *statistical multiplexing* [39].

Things are not so clear for some recent technologies such as the *asynchronous transfer mode (ATM)*. Although highly related to circuit switching in more than one layer, ATM also employs concepts of packet switching. Therefore we decided to include a discussion on ATM in the next section. For a detailed introduction to telephone systems and circuit switching we refer to [79], [39] and [46].

1.5.2 Packet Switching

Packet switching originated in the late 1960s from communications research sponsored by the U.S. Department of Defense and was designed to connect user terminals to host computers. As suggested by the name, any information is conveyed across the network in form of *packets*. Communication devices are usually connected to a common shared circuit, a switch or a combination of them. Data being sent are partitioned into smaller pieces (*segmented*) whereby a *header* containing the address information is added. This address labeling also determines the routing through the packet switched network. As every packet might be routed on a different path over the network, a *sequence*

number is included in the header to determine the right sequence of packets at the receiving side. When more than one out of a group of devices attached to the same circuit attempt to send data at the same time, a collision occurs. In *Ethernet* networks, such a common shared circuit is implemented either by the use of *coax cabling* or within a *network hub*. The number of collisions could be significantly reduced by replacing the latter with an *ethernet switch*. The related network topology is star based as opposed to bus based common in coax networks. Furthermore these switched *twisted pair networks* may operate at a higher speed.

In the current section, we will occasionally mention the OSI seven layer model. Following a staged approach, it defines each layer as a service provider for the layer above and a service user for the layer below. For example, layer 1 (*physical layer*) deals with physical connections offering them as a service to layer 2 (*data link layer*). Layer 3 (*network layer*) is responsible for providing end-to-end connections to layer 4 relying on layer 2 to deliver the necessary link connections. Proceeding in that manner, the seven layer model can be constructed. Although several aspects of the OSI seven layer model have been implemented, it has been accepted as the standard reference for network models.

Packets are usually transmitted over the *packet switched network* in a *store-and-forward* manner, that is from node to node in sequence. Very often some type of *error control and recovery* is exercised by each node. Both concepts relate to the data link layer of the OSI model, but have been considered as unnecessary burden for the transmission over very reliable high speed links. As a consequence, *frame relay* has been developed. Frame relay is related to layer 1 and 2 of the OSI model. The protocol used in layer 2 is called *high level data link control (HDLC)*. A variant of HDLC known as *link access procedure for the d-channel (LAPD)* is used for the exchange of ISDN signaling information. Data are transmitted from node to node at high speed with a very limited error detection. If an error is detected, the erroneous packet is immediately discarded. It is left to the sending and receiving devices to perform end-to-end error control. As a result protocol processing has been significantly reduced in the frame relay nodes. However, frame relay is only suited for reliable links, in case of a high *bit error rate* another protocol should be used.

Very common to packet switching is the concept of a *virtual circuit (VC)*. This virtual circuit has to be established by means of a connection setup procedure, which allows for the negotiation of capabilities and an exchange of

address information. During data transmission, a VC tag is assigned to each packet. These *connection oriented services* save a lot of time, because instead of the entire address information only the VC identifier has to be processed at each node in the network. In fact, every packet with the same tag follows the same routing path leading to an in-sequence delivery of packets. If no connection setup occurs and the packet header conveys the entire address information, the corresponding service is called *connection-less*. In other words, each packet travels on its own. Both concepts relate to end-to-end connections and therefore to layer 3 and above of the OSI model. OSI layer 4 (*transport layer*) is responsible to provide transparent connection oriented and connection-less services to layer 5. In TCP/IP the relevant protocols are the *transmission control protocol TCP* for connection-oriented services and *user datagram protocol UDP* for connection-less services.

Another approach to reduce the total transmission time was the elimination of collisions occurring on common shared circuits by introducing the packet switch. The technique used is very similar to the concept of time division switching, except that packets can vary in length and queueing mechanisms are often applied to prevent packet loss. Accordingly, packets are switched to the output link as suggested by address information contained in the packet header. A combination of packet and circuit switching concepts led to the development of *cell switching*, which is mainly used in *asynchronous transfer mode (ATM)* networks. ATM technology is designed to meet the needs of heterogeneous, high-speed networking. The unit of communication is a fixed-length packet called a *cell*. With a total length of 53 octets, only 5 octets have been assigned to the header. Dedicated communication channels with a certain quality of service are supported by the concept of a virtual circuit. From the viewpoint of a network user, ATM is equipped with the advantages of circuit and packet switching, which are the provision of fixed and variable bandwidth connections. However, ATM is truly a connection oriented technology and so many people were not satisfied with its connectionless services. This resulted in many contributions to ATM and most of them have been put aside years later. Over the years ATM has become a mature technology, which has been widely applied.

In almost every packet switched protocol, signaling information and user data are transmitted over the same physical link. This type of signaling is called *in-band signaling*. If both are conveyed over separate links, one speaks about *out-of-band signaling*. As an example consider ISDN, where signaling information (*D-channel*) is separated from the user data (*B-channels*). The

former is handled by a 3-layered connectionless packet switching protocol. The high level signaling messages fit into a single packet, which leads to the common description *message switching* protocol.

In summarizing what has been said up to now, we are able to pin down some features of packet switching. These are:

- Flexible packet lengths improve the effect of statistical multiplexing and are well suited to varying bandwidth demands.
- With connection oriented services packet header processing becomes simpler and faster.
- With connection-less services packets are less vulnerable to node failures.
- Due to the synchronous transmission of fixed length packets in ATM processing time has been significantly reduced. The feature of statistical multiplexing has been preserved.
- The highest efficiency is achieved for connections with a short holding time and varying bandwidth demands.

In the last years a lot of new protocols dealing with the transmission of multimedia information over packet switched networks have emerged. In case of ATM, several quality of service mechanisms have been proposed and implemented to overcome the problems of transmitting data with different quality of service needs over packet switched networks. For example, voice is very sensible to delays, but very stable with respect to transmission errors. Exactly the opposite is true for pure data. As ATM grew up with multimedia handling requirements, the corresponding capabilities have been integrated in lower layers as well. Some older protocols have only been adapted by adding layers on top of the existing ones. One such example is TCP/IP in conjunction with a H.323 or SIP protocol stack. H.323 deals with the transmission of multimedia traffic over LANs that provide a non-guaranteed quality of service. The *session initiation protocol (SIP)* is only capable of handling connections, which are referred to as sessions. It is not an umbrella standard like H.323 and has to be supplemented by other protocols such as the *real time transport protocol (RTP)* and the *media gateway control protocol (MGCP)*.

In order to achieve bandwidth guarantees on an IP network, the *resource reservation protocol (RSVP)* can be used to allocate resources on the entire transmission path requiring all units on that path to support RSVP. A more transparent approach has been offered by *differentiated services (DiffServ)*, which assigns a different meaning to the *type of service (TOS)* bits of the IP header. However, both methods rely on in-band signaling to convey flow control information. Without the possibility of an end-to-end expedited delivery the transmission of signaling information will suffer in overload situations. This is one reason for implementing *voice over IP* architectures based on SIP or H.323 in closed IP networks rather than across the internet. There is still some work to be done with respect to real time traffic management. That does not necessarily mean, that packet switching is unsuitable for transmitting delay sensitive traffic. In fact, ATM has been very successful in the past. Furthermore we can expect voice over IP to profit from recent developments such as carrier ethernet (CE) and ethernet in the first mile (EFM).

Another example common to public network implementations is the *digital cross connect system* providing packetization and compression services for standard 1544 kbit/sec and 2048 kbit/sec links. AT&T [33] describes the system as packetized circuit multiplication equipment conforming to ITU standards. Voice is compressed by means of *digital speech interpolation (DSI)*, which is similar to TASI described in section 1.5.1.

We have covered many issues related to packet switching. Accordingly, we will now present some references of interest. A general introduction to networking is given in [209]. More information about ATM, frame relay including traffic and overload control can be found in [35]. Additional references for ATM are [169] and [67]. A treatment of fast packet switching and gigabit networking appears in [164]. For a detailed description of interconnection units, bridges, routers and switches refer to [94] and [166]. More about the OSI model can be found in [120], [192] and [42]. An introduction to TCP/IP is given in [84] and [76], although more recent protocols such as RSVP, RTP and the *real time control protocol (RTCP)* have not been covered. For the latter refer to [54]. An introduction to Ethernet and related protocols is given in [56]. Further protocols of interest have been omitted here, as are *X.25* [47][193] and *Token Ring* [87][152]. A presentation of carrier ethernet and ethernet in the first mile appears in [118] and [38].

1.6 Call Centre Performance

In order to consider appropriate actions, call centre management relies on the feedback delivered by the call centre system in terms of reports. It became very popular to express certain aspects of call centre performance by a single value only called a *key performance indicator (KPI)*. Unfortunately there is no unique description available, so key performance indicators may differ from system to system. However, we will attempt to provide a rather comprehensive list also featuring a short description, if necessary.

- *number of calls abandoned (NCA)*, a call is considered as being abandoned, if the caller hangs up before receiving an answer.
- *number of successful call attempts (NSCA)*
- *number of calls answered* or *number of calls handled (NCH)* by an agent, *number of calls processed* by resources other than agents.
- *number of calls waiting (NCW)* in queue.
- *number of calls offered (NCO)* is the sum of NSCA and NCA.
- *number of calls transferred, held, consulted* or similar.
- *number of contacts* is the number of customers an outbound agent has been able to get in touch.
- *average time to abandon* or *average delay to abandon (ADA)*
- *average after call work time (AAWT)* describes the mean time an agent is occupied by wrap up work.
- *average talk time (ATT)* describes the mean time an agent is on the phone.
- *average inbound time (AIT)* describes the mean time an agent is on the phone serving inbound calls.
- *average outbound time (AOT)* describes the mean time an agent is on the phone serving outbound calls.

- *average speed of answer (ASA)* or *average delay to handle (ADH)* is the mean time a caller has to wait before being connected to an agent.
- *average service time (AST)* or *average handle time (AHT)* is the sum of ATT and AWT.
- *average auxiliary time (AXT)* is related to the time an agent spends in the AUX state.
- *oldest call waiting (OCW)* may be considered as an estimate for the current wait time. In fact, this estimate is biased, as it always underestimates the current wait time. Therefore some vendors suggested the *expected wait time (EWT)*, which is a more suitable estimate in a certain statistical sense.
- *service level (SL)* is the percentage of calls being answered within a certain period of time. Sometimes the SL is specified in the form 80/20 meaning that 80% of all calls have been answered within 20 seconds.
- *agent occupancy* or *agent productivity* as a measure for the work performed by an agent. Unfortunately there are as many definitions as call centre vendors. Therefore we refuse to contribute another one.

We decided to relate all key performance indicators to classic voice calls instead of cluttering the list by cloning the entries for each new media. The extension to video, email, instant messages and others should be evident. Especially the service level has been an item of discussion for years. Different metrics to measure customer satisfaction have been proposed, but most of them in a local setting only. More general considerations relevant for all call centre installations have been carried out in [130]. It is a common practice to define target values and thresholds for the relevant key performance indicators. Combined with other objectives, these are compiled into a single document called the *service level agreement (SLA)*. For most organizations, the SLA is realized as a contract between departments or companies.

Chapter 2

Call Centre Architecture

Basically, the heart of a call centre is formed by a standard telephone switch augmented by an intelligent call distribution mechanism. This enhancement is called the *automatic call distributor* - ACD. With respect to the direction of the call, two basic strategies have been identified before, i.e.

- Inbound call management, which is usually a core functionality of the ACD, and
- Outbound call management, which is often implemented on an adjunct system attached to a CTI server

Simple interactions with the system such as touch tone detection are often handled by the telephone switch itself. When becoming more complex, as is the case for speech recording, text-to-speech and voice form processing, the necessary services are provided by an *interactive voice response* system. Many call centres are able to record performance values and call statistics. These may be displayed on the agents voice terminal or on a *readerboard* (also called a *wallboard*). The latter is used, when information has to be presented to a larger group of people.

Taking a closer look on the ACD, we are able to identify several functional components. Calls arriving at the call centre are held in a *queue* until the requested resource becomes available. During queueing time, the customer follows the steps of a predefined program while being monitored by the *call progress control*. These programs range from simple music being played to complex branching scenarios composed of music, announcements and touch tone interaction routines. Announcement units, tone detectors, CD players,

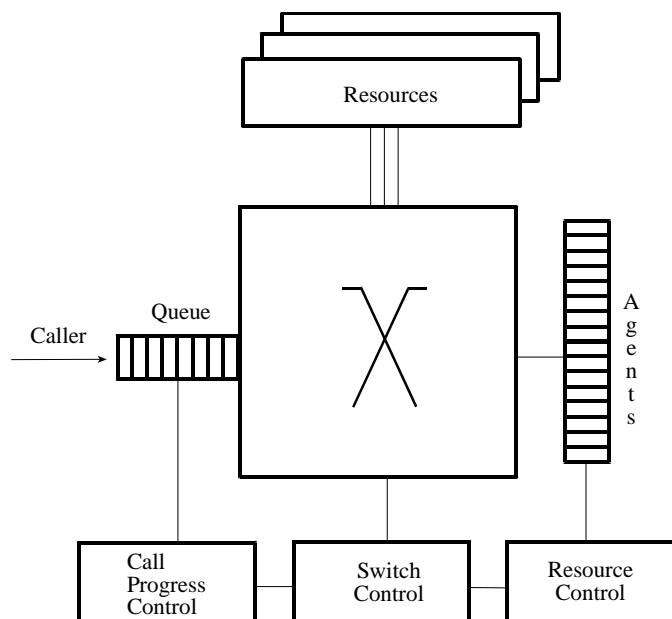


Figure 2.1: Generic Call Centre

interactive voice responses, mail and advertisement servers are only some examples for the non-human resources indicated in figure 2.1. The most flexible and often the most limited resource of a call centre is the call centre agent. The *resource control* in agreement with the call progress control is responsible for reassigning a waiting call from the call progress queue to an available resource. It has to be noted, that the definition of an endpoint in the call path varies with the implementation of a call centre. For example, many call centres do not allow the call to stay on track after an assignment to a life agent or an interactive voice response. It is up to the resource to reassign the call to an appropriate call path using standard telephony features such as call transfer. If the call is maintained in the queue during the entire conversation, the call may proceed along the call path after the resource hangs up. This type of loose interaction is a common technique in public networks, but not for enterprise systems. Using the information collected by telephone switch and adjunct nodes, the call progress control determines an optimal path through the call flow, while the resource control identifies the appropriate resources along that path. Sources of information are caller inputs, call

state and environment conditions. Examples for caller input are touch tone strings and spoken words, whereas the call state is like a container for call related variables such as arrival time, time in queue and priority status. The priority status changes, when exceeding a predefined threshold. Some call centres even implement a two level threshold scheme based on the average queueing time. A split group in overload condition is assigned additional resources in the form of reserve agents or agents from other split groups. Environment conditions include time of the day, the number dialed by the customer (DNIS or called party number provided by the local exchange with ISDN) or an email address. The latter only occurs in contact centres or multimedia call centre with mail capability.

In addition to the information provided by the call progress control, the resource control has to monitor agent availability and status. Accordingly, the resource control on request determines a proper routing path to the resource best matching the callers requirements. In any case, the *switch control* is involved with and responsible for the interconnection of caller and requested resource. There is no difference between connecting the caller to a tone detector, an announcement, an advertisement server, an agent or to an interactive voice response - the switch control obeys the commands directed by call progress control and resource control.

Provided calls are being queued for service and a demanded resource becomes available, the caller may experience resource allocation in the following ways. Either the call is dragged out of the call progress queue or the customer is allowed to finish his current transaction step. The former is called an *interruptive assignment*, whereas the latter is known as *non-interruptive assignment*. Although interruptive assignment may stress the patience of certain customers, it is definitely the better choice with respect to performance engineering. Furthermore it is common practice, where uncomfortable effects are avoided by the right choice of waiting entertainment.

As can be seen from the text above, it is hard to separate the call progress control from the resource control. When designing a call centre according to figure 2.1, a major part of work has to be allocated to the interface specification between call progress control, resource control and switch control. Although on a different level the same applies to call centre solutions for public networks, when based on the concepts introduced by *intelligent network (IN)* standards. In fact, only the switch control interface is defined in terms of these standards. So the necessity to provide the appropriate interface mappings remains.

2.1 Adjunct ACD

In this scenario, the ACD is separated from the switch. To allow for control of switch resources by the ACD, an open application interface is agreed upon. This enables an ACD to be built from standard components. The ACD then becomes just another CTI component. One advantage of separating the ACD and the switch is vendor independence. With respect to the configuration shown in figure 2.1, this means an externalization of call progress and resource control. In other words, the switch does not know anything about queues or agents, as it is concerned with calls and stations only. This provides a first indication on how to design an open application interface. It has to bridge the gap between these two worlds. From a technical perspective, call control is entirely passed to the adjunct ACD for processing. So the adjunct ACD is responsible for the determination of an appropriate routing path and the transfer to an agent or to an interactive voice response. This does not mean, that the adjunct ACD exercises some type of switching functionality. The execution of call control directives is still up to the switch. The entire set of directives constitutes the CTI application interface, which is expected to provide

- *Call control through a third party* - e.g. call answer, call disconnect.
- *Set and Query value* - examples are trunk status, time of the day, station and call status information (answered, busy, transferred and conferenced are some examples)
- *Event notification* - includes digit input and changes in station and call status.
- *Call routing* - choose target as indicated by the adjunct ACD

A set of capabilities as listed above allows an adjunct ACD to interact with the switch. It is required, that both units follow the same interface specification. If the switch does not support the ACD interface, it is a common practice to install a gateway called *telephony server*, which translates the proprietary switch interface to a standardized application interface. Common choices include

- SCAI - *Switch Computer Application Interface* - an interface standardized by ANSI T1, which introduces a close relationship to the ISDN standard. For more information refer to [89].

- CSTA - *Computer Support Telecommunications Applications* - an interface standardized by ECMA. Its most recent flavours have been specified in terms of the *Extensible Markup Language (XML)* [70] and the *Web Services Description Language (WSDL)* [69] to allow for inclusion of new media and SIP support [68].
- TSAPI - *Telephony Services Application Programming Interface* - an interface specified by Apple, IBM, Lucent Technologies and Siemens ROLM Communications. The resulting compilation, called the Versit Archives (refer to [206]) were adopted by the ECTF. The TSAPI concept was implemented by Novell in their telephony server called TSAPI Server.
- TAPI - *Telephony Application Programming Interface* - an interface specified by Microsoft. TAPI in its current version 3.0 became a de-facto standard for most applications [199].
- JTAPI - *Java Telephony Application Programming Interface* - a portable, object-oriented application interface for JAVA-based computer telephony applications defined by Sun Microsystems. Implementations of JTAPI are available for existing integration platforms including TAPI, TSAPI and IBM's Callpath [112][113].
- CPL - *Call Processing Language* - a scripting language for distributed communication platforms implementing the SIP or H.323 protocol suite.
- ASAI - *Adjunct Switch Application Interface* - an interface specified by Avaya Communications former AT&T. ASAI is strongly related to TSAPI. Several vendors, e.g. Genesys, Hewlett Packard and IBM implemented the ASAI protocol stack in their own products to be compatible to Lucent Technologies telephone switches.
- CSA - *Callpath Services Architecture* - IBM's computer to switch link.
- *Meridian Link* - Nortel's host to switch link, mainly to connect the Nortel Meridian switches.

This list is far from being complete, as nearly every switch manufacturer defined its own standard to connect to the outside world. When separating

the ACD from the switch, one has to be concerned about the overall system reliability. If one does not purchase the entire system from a single source, he is left with the judgement of system reliability. If an adjunct ACD built from standard components breaks down and the switching system is not designed to provide appropriate fallback mechanisms, this might become a critical issue. Very often the CTI application protocol provides indications for the existence of fallback mechanisms and should be closely inspected. A lot can be done by considering the use of duplex units and standard failover mechanisms for off-the-shelf hardware. Whereas duplex units are often exposed to implement the same software bug as the main system, failover mechanisms provided by the operating system or a device driver often lack application awareness. One solution to these problems is to consider heterogeneous fallback mechanisms, which might be subject to service degradation. In case of an IVR breakdown, some simpler call flows may be provided by the switch. Customers would miss speech recognition and some of the warm sounding announcements, but at least they would reach an agent. Even this little example shows, that reliability is a staged rather than a binary concept, which has to be treated with care.

Some manufacturers decided to combine telephony server, ACD and further CTI functionality into a single unit called *contact management system*. The contact management system typically provides integration with client computers and access to databases. The latter allows to establish some mapping between customer information and the data from the ACD or the telephone network. As an example, consider a database lookup based on the received ANI digits. Some External ACDs and contact management systems also include the possibility for call management and reporting functions. In the current context the term call management is understood from a user perspective and not to be mistaken with the call management performed by the system and described in section 1.2.

2.2 Integrated ACD

In order to avoid problems with the interface specification between adjunct and switch, some vendors have chosen to integrate ACD functionality into the switch. This often results in enhanced functionality and increased reliability. Even for this scenario, some components still reside outside the switch and communicate with the integrated ACD. These include call management,

statistics and reporting platforms. Whereas telephone switch and integrated ACD are often configured using a textbased terminal, call centre operation and reporting is carried out on graphical workstations. Due to the rapid change in workstation capabilities, a tight integration of call management and reporting platforms with the ACD would result in a loss of reliability.

Although the late 1990s faced a trend of consolidation, i.e. vendors aimed to combine switching, IVR, ACD, voice mail, web and fax functionality into a single communication server, the new millenium heads for the opposite direction. It is expected, that the complexity of a single unit cannot be afforded due to the ever changing nature of its components. By inspecting the problem more closely, it turns out, that only a shift in architecture has occured. However, the new trend certainly fits the requirement of object oriented development demanding reusable and often abstract components with exact interface specification. As an example, consider the routing logic applied to an abstract call, which can assume a classic voice call, a video call, an instant messaging call or an email.

2.3 Switch Architecture

With the ACD as the brain of a call centre, the switch is readily identified as its body performing all the physical work such as providing connections. But there is more to do, as will be seen in the subsequent description of the functions of a classic switching system. We will then move on describing the evolution of switching systems from past until now. Last but not least we will attempt to provide a rather generic description of a central switching system and a distributed communication platform. While the former is better suited for circuit switched telephony, the latter is more appropriate for packet based telephony and multimedia transmissions.

2.3.1 Functions of a Classic Switching System

According to [79], telephone switching systems have to perform the following basic functions

- *Attending* - Lines are monitored to detect an incoming call request. The corresponding signal is known under the name *seize signal*.

- *Information receiving and processing* - The information received is processed by the telephone switch to determine, execute and control any necessary actions to be performed. Typical information elements include the called party address, the calling party address and the *class of service*.
- *Busy testing* - Before a connection can be made, the system has to test, if the called party is engaged in another call. If so, a busy signal is delivered to the calling party. In case, a call is made to a group of lines, testing proceeds until a free one is found. Such a group is also called a *hunt group*.
- *Interconnection* - An end-to-end connection across the network is established between the two terminals after both have been connected to the network. Sometimes the initial connection is released in favor of a new one. This is known as *call-back* or *crank-back*.
- *Alerting* - After having established the connection, the called customer is alerted using a dedicated signal such as the ringing current. For digital communication systems, the signal is often represented by an appropriate protocol element, which upon receipt triggers some audible feedback to the user.
- *Supervision* - In order to tear down the connection, the system steadily monitors for the occurrence of a disconnect signal. The disconnect signal is usually triggered by a telephone going on-hook. Although more interesting for public telephone switches, charging information has to be collected. This can be done by operating a meter and sending metering pulses. For digital communication systems, metering pulses have been replaced by suitable protocol elements.
- *Information forwarding* - If more telephone switches are involved with a single connection, it may become necessary to convey additional information to all of them. As an example, consider the address of the called party.

2.3.2 Evolution of Switching Systems

According to the developments in circuit switching, we are able to identify five generations of switches. As switches were primarily concerned with pure

voice switching in the past, they more and more have become feature loaded communication systems with multimedia capability. Recently the concept of a communication platform has undergone a significant change from the centralized to the distributed paradigm. Architectures built of separate units with dedicated tasks have been proposed. Where the communication from one component to another has been unclear, new protocols have been defined by standards organizations. At the time of writing, development for large enterprise switches has already stopped, whereas classic telephone switches remain the dominant structure in public networks. We should prevent ourselves from making a final judgement, as distributed platforms are still an active part of research and development.

Early circuit switched systems introduced an annoying effect called *blocking*. Blocking occurs, when a desired resource is not available. An input line, which cannot be connected to an engaged output link, is blocked and the corresponding call is lost. Classic telephone switches are loss systems as opposed to queueing systems. As an example for the latter consider requests queueing for access in a local area network. Most switching platforms available today have completely eliminated the effect of blocking. Their switching network is said to be *non-blocking*. Unfortunately effects of blocking are not limited to calls only, moreover they can occur as a result of contention for almost any limited switch resource. As an example consider the *tone detectors* and *tone generators* responsible for detection and generation of various tones such as modem tone, fax tone, busy and interception tone. It is a common practice to install less tone detectors/generators as are required, because they are assigned for a relatively short time period to a certain call rendering the effects of blocking negligible. It would be too costly to add further tone detectors/generators until the resource supply is considered non-blocking. But there is a certain chance for unavailability of a dial tone, when the handset goes off-hook. The fraction of time a certain resource is not available, is also called the *blocking factor*, which is often subject to minimization by the methods of *queueing* and *optimization theory*. Similar methods are also applied to problems of variation occurring in packet based communication scenarios. Especially the medium of voice exhibits a certain sensibility to delay and packet variation.

Coming back to the classification in generations of communication platforms showing their role and importance in history, we may identify the following stages of development:

- *First generation switches* include any type of manual and hardwired electromechanical switching facility. (Up to 1950s)
- *Second generation switches* were computer controlled, programmable systems introducing stored program control. Furthermore the first telephony features appeared on the scene. Blocking effects have not been eliminated for second generation switches. Data were usually transmitted by use of *modems*, converting digital signals to tones (MODulation) and converting them back at the receiving side (DEModulation). (The 1960s to 1970s).
- *Third generation systems* introduced the use of digital switching. Switching became more efficient and cheaper allowing for non-blocking switching networks. As today, voice is converted to digital streams by means of certain modulation schemes such as PCM (refer to section 1.5.1). Implementing *integrated services digital network (ISDN)* capabilities into the switches was not possible at that time due to the excessive processing power required to handle the signaling part of ISDN. The time frame for third generation switches ranges approximately from 1975 to 1985.
- *Fourth generation switches* introduced the integration of high level concepts into switching platforms. Examples include ISDN capabilities, call centre solutions and the handling of *local area network (LAN)* traffic using dedicated *router* modules. The latter is a combination of existing technologies, as the router board only encapsulated LAN traffic to be carried by standard telephony protocols. In the public network, *common channel signaling system no. 7 (SS7)* led to the development of the first *intelligent network application* in the United States - the toll free calling service. With respect to cellular mobile telephone, standards such as *group speciale mobile (GSM)* in Europe or *personal communication system (PCS)* have been integrated with SS7. (1985 to 1995).
- *Fifth generation systems* are tighter integrated with different networks. Some private branch exchanges already showed improved public network integration by using SS7 modules. This trend is also emphasized in the development of interoperability standards for LAN and telephony protocols. Packet switched telephone systems appeared on the market

around 1995. Another point of interest has been the integration of radio connectivity with telephone switches based on the *digital european cordless telephone (DECT)* standard. (1995 up to now).

- *Next generation networks (NGN)* are composed by network nodes performing tasks such as control, media and signal processing. Advocated by wireless standard bodies such as the *third generation partner project (3GPP)*, NGN proponents aim to replace circuit switched telephony by packet switched communication based on the *internet protocol (IP)*. (1995 up to now).

More information on SS7 is available from [177], whereas for DECT and GSM we refer to [208]. An introduction to IP is provided by [76], although some recent aspects such as quality of service and security have not been covered.

2.3.3 Generic Central Switch Model

Basically a switch can be described as a unit interconnecting input and output links (or channels) according to a set of rules provided by a certain control mechanism. To be more specific, these rules determine the routing path through the switching network. If considered as a single unit, the switching network is usually called *switching fabric*. Connections to the outside world are provided by dedicated *interface modules*. According to the type of interface, these modules often implement alarming, signaling and protocol handling functions. As an example, consider the Italtel UT 10/3 Switching System. Each interface module consists of several interface units and a time slot interchanger. The interface unit maps voice and data channels to internal time slots and performs analog-to-digital or digital-to-analog conversion of analog subscriber lines and trunks. A module processor responsible for call handling, call processing and call routing is included into each interface module. The switching modules are connected to a space division switch and a central control. Switching occurs in a three-stage process, as is the interface module with time slot interchange - the main space division switch - the interface module with time slot interchange. Such an arrangement is called a *switching network* of the T-S-T type. Each letter relates to a switching stage, either T for a time division switch or S for a space division switch. Following that notation, a network of the T-S-S-S-T type would consist of

5 stages in total consisting of time division switch units in the first and last stage and space division switch units in the other stages.

The Italtel UT 10/3 Switching System is an example for a circuit switched public exchange. Services delivered differ from those provided by packet switches. This is mainly reflected in the service logic of the control system and the interface modules. As the error rate rather than the packet delay is the sensitive parameter, the design of interface modules for packet based switches has to cope with the need for error detection, error control and buffering. Similar considerations also become necessary in the context of distributed communication platforms, which will be discussed later.

In considering switching architectures, one has to be very careful about the terms synchronous and asynchronous. For example, *asynchronous transfer mode (ATM)* switches implement a synchronous switching fabric. In fact, the term asynchronous refers to the way, how ATM cells are selected for processing. The synchronous nature of ATM is also underlined by the fact, that packets get segmented into ATM cells of fixed length.

Adopting a similar approach here, we will now attempt to describe a rather generic switch architecture. Bearing in mind, that such a structure reflects an ideal szenario, some properties of switching systems may be derived. Reviewing figure 2.2, the *switching fabric* can be identified as the heart of the entire unit. It performs the major work, that is connecting input lines to output lines. From a technical perspective, the switching network is implemented as a space division multiplexer, a high speed bus system, a shared memory architectures or a combination of them. When dealing with asynchronous data traffic, the required queueing buffer is usually built into the interface modules rather than the switching network. The switching fabric is controlled and monitored by a *central control module*, which in turn communicates with the interface control units. In practice, the controller is part of the interface module as shown by the example of mid-sized to large *private branch exchanges (PBX)*. On a rather general level, we are able to identify the following types of interface modules:

- *Subscriber modules* provide dedicated connections to end users. Protocols to be handled by the subscriber module include the ones used for analog signaling, *integrated services digital network (ISDN)*, *digital subscriber line (DSL)* and wireless access. Signaling information is processed by the control attached to the interface module. In case of analog subscribers, the interface module has to support analog/digital

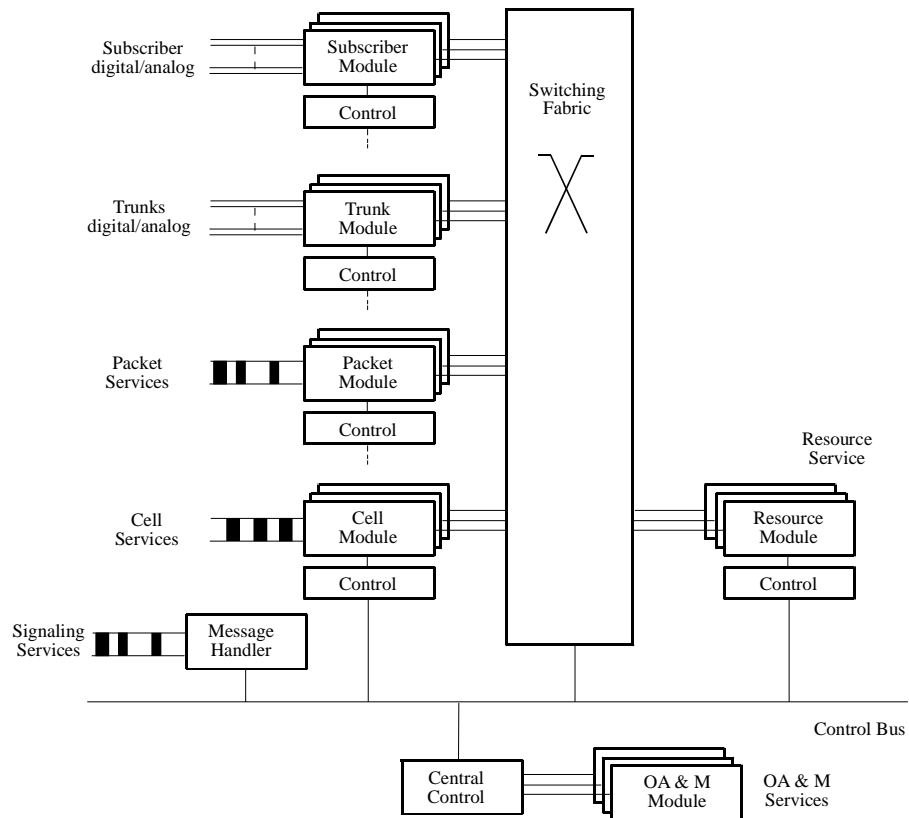


Figure 2.2: Generic Switch Model

conversion and echo cancellation.

- *Trunk modules* provide access to the telephone network. Protocols include analog protocols, ISDN, *broadband ISDN (B-ISDN)* and *asynchronous transfer mode (ATM)*. Different to subscriber access, it is not required to select a single line or channel within a trunk. In case of *channel associated signaling (CAS)*, the signaling information is processed by the control module attached to the interface module. When attached to the public switched telephone network, it is a common choice to use a separate signaling channel, which is provided by the message handler.
- The *packet module* provides access to packet switched services, e.g. TCP/IP and X.25. Inband signaling information is extracted to be processed by the control module. User data are segmented and buffered in the interface module. If enhanced by high level telephony signaling protocols such as H.323 or SIP, the packet module may be used for transmission of *voice over packet* networks.
- The *cell module* provides access to ATM cell services. Signaling information is often separated from user data as in the B-ISDN/ATM model and therefore processed by the message handler. Cell based transport includes applications on the subscriber side as well as the carrier side. It is displayed separately in the figure to point out the importance of cell services.
- The *message handler* processes all types of outband signaling information. Protocols include *common channel signaling system no. 7 (SS7)* and *QSig*. QSig is a global signaling standard for corporate networking and has been adopted by the standard bodies International Standards Organization (ISO), European Computer Manufacturer Association (ECMA) and the European Telecommunication Standards Institute (ETSI).
- The *resource module* provides access to generic resources including but not limited to speech synthesis, speech recognition, tone generators and detectors. The resources are either integrated or implemented as an adjunct system. In the latter case the resource module often acts as a protocol converter between the internal control bus and the adjunct

node. Adjunct link protocols usually range from a *RS232/V.24 serial interface* [47] to a TCP/IP connection over Ethernet. In the public network, communication between database resources and the *signal control point* have been established by the use of the *X.25 protocol suite* [47][193]. In order to provide faster access, these links are replaced by faster Ethernet links [56].

- The *OA&M module* provides access to operations and maintenance information. Attached to the central control, certain tasks such as administration, maintenance, exception handling and software uploads may be performed. In the public switched network, a similar functionality is provided by the *system management platform (SMP)*.

Although our exposition is based on an abstract concept, some properties are commonly found in switches used for telecommunication and datacommunication purposes. In the world of telephony, message handler modules, subscriber and trunk interfaces play a major role. Some private branch exchanges also implement packet modules with data routing capability. Data delivered to the packet module are handled the same way as by a router attached to a PBX. Cell modules have also been integrated into PBXs to establish communication between the nodes of a distributed PBX architecture. With respect to the communication in data networks, a great effort has been undertaken to improve *voice over network* technology perhaps completely eliminating the need for subscriber and trunk modules in the near future. This led to the adoption of distributed communication architecture as will be described next.

2.3.4 Distributed Communication Platform

Motivated by classic IT architectures and the excessive increase in processing power a distributed design has also been adopted for communication platforms recently. Another contributing factor has been the rapid development of the internet, which also explains the choice of IP in favour of other protocols better suited for transport of real time traffic. Therefore we will accept the IP in voice over IP technology as granted, which leads to a more specific architecture as the ones considered so far. Although the term *softswitch* has not been defined in a strict sense and bears many different interpretations, we have decided to include it for sake of a more realistic viewpoint.

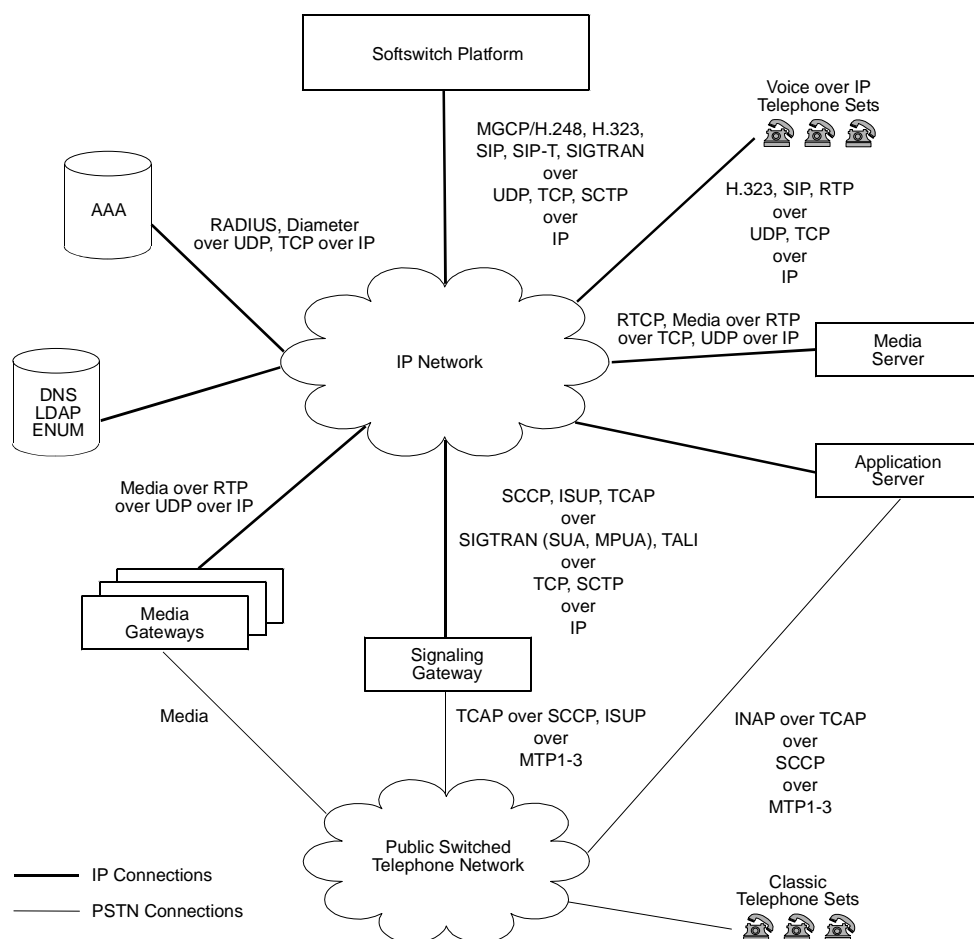


Figure 2.3: Distributed Communication Platform

Our discussion will be based on the architecture shown in figure 2.3. We also made reference to some of the protocols used in softswitch controlled networks. They have been included for completeness and will not be discussed in detail. Instead the reader is asked to pay attention to the components shown in figure 2.3. As mentioned above, the transport is provided by an IP network shown in the center. Due to the stringent performance constraints imposed by real time traffic, these networks are often isolated from a companies' intranet or the internet. Some vendors of existing systems even recommend a separation of hardware. In not taking any precautions with respect to the quality of service, even a *virtual LAN (VLAN)* would not provide any satisfying results. With respect to road traffic, a VLAN would only render the other vehicles invisible rather than providing an additional lane. Obviously we do not face an ordinary IP network, instead it has to obey to appropriate service level agreements. At the time of writing, new approaches such as *ethernet in the first mile (EFM)* [38] and *carrier ethernet (CE)* [118] are promising concepts with respect to quality-of-service assurance. We are now ready to identify the remaining components shown in figure 2.3, as are

- The *softswitch* is the central controller of this architecture. It implements a *media gateway controller (MGC)* to exercise control over the *media gateways (MG)* also attached to the network. Furthermore the softswitch is responsible for communication with application servers (AS), signaling gateways (SG) and other components.
- The *authentication and accounting center (AAA)* collects any data necessary for billing and authentication purposes. As no dedicated subscriber lines exist, a separate authentication mechanism has become necessary for packet based communication. Prior to admitting a conversation to be engaged, the softswitch requests the calling party to be validated. For that purpose, the *Remote Authentication Dial In User Service (RADIUS)* and the more recent *diameter* protocol have been specified.
- Directory services such as the extended *domain name service (DNS)*, the *lightweight directory access protocol (LDAP)* and *electronic numbering (ENUM)* are mainly used for address translation. Especially when interfacing to the PSTN, an appropriate translation mechanism becomes necessary, as classic telephone numbers have to be mapped to valid *unified resource identifier (URI)* entities.

- The media gateway is responsible for frame adaption and transcoding of media streams. Control is exercised by the use of a *media gateway control protocol*. Currently there are two versions available, which are *MGCP* defined by the IETF and its extension specified in recommendation H.248 of the ITU.
- The *signaling gateway (SG)* provides translation services for signaling protocols. In figure 2.3, we have chosen to show a more complex version connecting to the public network. As indicated by the protocols used, the softswitch has to be aware of that type of communication. When attached to an ISDN access network, the signaling gateway is acting on behalf of the telephone attached to the PSTN. Accordingly, standard protocols such as H.323 and the *session initiation protocol (SIP)* are sufficient avoiding a separate treatment of PSTN calls by the softswitch.
- The *application server (AS)* implements services provided by the distributed communication platform. At the time of writing a unified service architecture does not exist even for the recent *IP Multimedia Subsystem (IMS)*. In fact, services are still network dependent. Whereas telephony services such as call transfer, consultation and conferencing are common to the PSTN and H.323 networks, some of them have to be provided by an external entity in SIP based networks. On the contrary, *intelligent network (IN)* services have always been provided by adjunct systems. Accordingly, we can safely assign them to the domain of the application server.
- The *media server* can be seen as the multimedia counterpart of an announcement module built into a classic telephone switch. It simply provides multimedia streams on demand.

It is evident from the above description, that the specification of interoperability standards has become a major part in the overall design. This allows for vendor independence and the reuse of common concepts. On the contrary it is far more difficult to assess the reliability of a distributed communication platform. A thorough reliability analysis is based on architecture details of each component, which has been clouded by the black box approach common in recent designs. As a consequence the testing period has to be significantly extended. On the other hand the availability of open interface standards provides an increased chance for the introduction of new components.

For a thorough treatment of SIP and the SS7 protocol stack we refer to the books [177] and [178] by T. Russel. One of the best references in the field with respect to the design of softswitches is [66]. An introductory account on the IMS is provided by [117].

2.3.5 Analysis of Switching Networks

As the switching fabric introduces a common factor to both circuit and packet switching technology, it should be discussed in more detail. The simplest form of switching network is described by a crossbar arrangement with N inlets and N outlets. Obviously, it is always possible to connect an idle input line to an idle output line. As no *blocking* can occur in such an arrangement, the switching network is referred to as *non-blocking*. Blocking can be easily introduced by changing the number of inlets or outlets. If the number of outlets is reduced, i.e. M inlets and N outlets (called an $M \times N$ switch) with $M > N$, it is not always possible to connect an idle input line to an idle output line. The same holds for the oppsite, i.e. when $M < N$. These two switching systems are called *concentrator* and *expander* in the order of their appearance.

Coming back to the $N \times N$ arrangement, it is easy to calculate the number of crosspoints $C = N^2$. The number of crosspoints is treated as a cost indication for a switching network arrangement. A reduced number of crosspoints results in reduced production costs. This is not true for modern switch designs due to the evolution in electronics and the introduction of VLSI technology. In lack of an appropriate cost indicator, the number of crosspoints is still used as cost index for a switching network bearing in mind its limited significance.

From the above discussion it is appearant, that there exists a trade off between the number of crosspoints and the *probability of blocking*. More than fifty years of research activity have revealed some non-blocking switching network arrangements with less than N^2 crosspoints. Before describing some of these arrangements, a classification of non-blocking switching networks according to A. Pattavina [165] will be presented:

- A switching network is called *strict-sense non-blocking*, if an idle input line can always be connected to an arbitrary idle output line independent of the current network permutation.

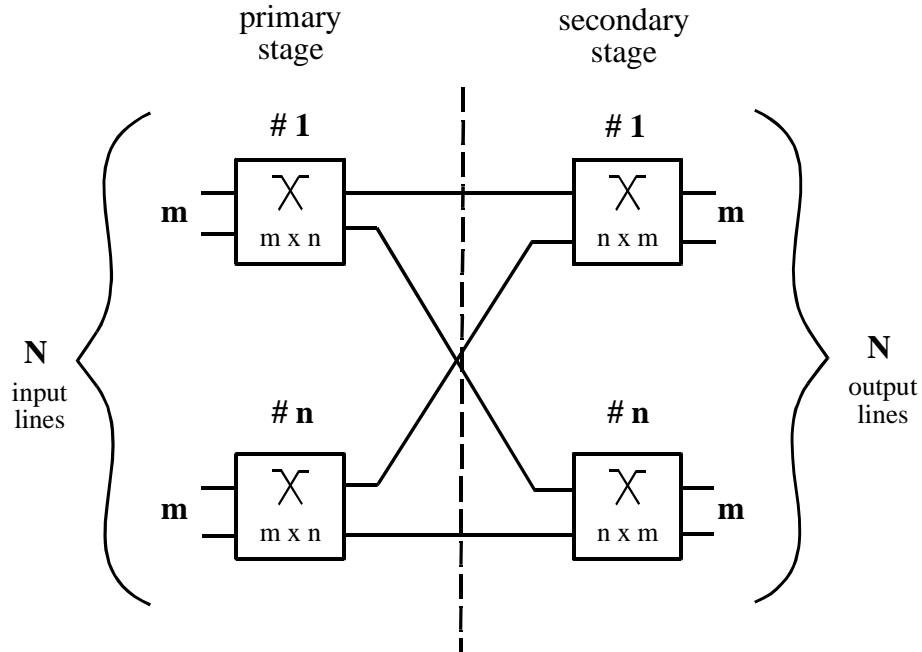


Figure 2.4: Two-Stage Switching Network

- A switching network is called *wide-sense non-blocking*, if it can always connect an idle input line to an arbitrary idle output line by preventing blocking states in the network through a policy of allocating the connections.
- A switching network is called *rearrangeable non-blocking*, if an interconnection between idle input and output lines is achieved by applying internal rearrangements of the connections already established, if blocking states are to occur.

The $N \times N$ crossbar arrangement from the above example is non-blocking in the strict-sense and therefore also satisfies the definitions of wide-sense and rearrangeable non-blocking. Giving a closer look at the definitions raises the question, how a switch has to be designed to allow for internal rearrangements or allocation policies. The first big step was done by interconnecting smaller switching elements to a switching network thus introducing the concept of network stages. Naturally, the $N \times N$ crossbar arrangement can be described

as *single stage switching network*, because it only consists of a single stage. A two-stage switching network is shown in figure 2.4. It has N input lines and N output lines. The number n of $n \times m$ secondary switch units is given by the number of outlets of each primary switch unit, i.e. $n = N/m$. Each switch unit in the primary stage is interconnected with all switch units of the secondary stage. As the number of crosspoints is the same for both stages, i.e. $C_1 = C_2 = nN = \frac{N^2}{m}$, the total number of crosspoints is given by $C = 2\frac{N^2}{m}$. Primary and secondary stage are interconnected by $n^2 = (N/m)^2$ links. In designing an optimal configuration, a trade-off between the number of links and the number of crosspoints has to be made. If the number of crosspoints is reduced by increasing m , there will be not enough links between primary and secondary stage to carry the traffic, i.e. $n < m$. On the other hand, increasing the number of links by raising n would result in a waste of resources. Therefore, a reasonable choice might be to use $m \times m$ switch units. The number of links is then given by $n^2 = N$ leading to the number of crosspoints $C = 2N^{\frac{3}{2}}$. In such an arrangement an idle input line can always be connected to an idle output line resulting in a non-blocking switching network. That does not necessarily mean, that a connection can be established to a specific output line, as the output channel may be busy. It is possible to find two stage arrangements, which differ slightly from those stated in figure 2.4 by interconnecting $m \times n$ primary switch units to $l \times m$ secondary switch units, where l must be an integer factor of n or n must be an integer factor of l . One example is a switching network with twenty 10×10 primary stage switch units interconnected with ten 20×20 secondary stage switch units supporting 200 input and output channels. Such an architecture requires 200 links and 6000 crosspoints, but it still gives the closest match for 200 input and output lines, as a two stage network made of similar switch units is not possible due to the fact, that 200 does not possess an integer root [79]. A more general case of two-stage switching networks taking into account concentration and expansion can also be found there.

Further improvements may be achieved by enhancing the network for another stage thus forming a *three-stage switching network*. A general network of this type is shown in figure 2.5 and is referred to as Clos Network due to C. Clos, who derived basic design criteria for non-blocking switching networks. As given in figure 2.5, the network has M input lines, N output lines and three stages consisting of r_1 switch units $m \times r_2$ in the primary stage, r_2 switch units $r_1 \times r_3$ in the secondary stage and r_3 switch units $r_2 \times n$ in the

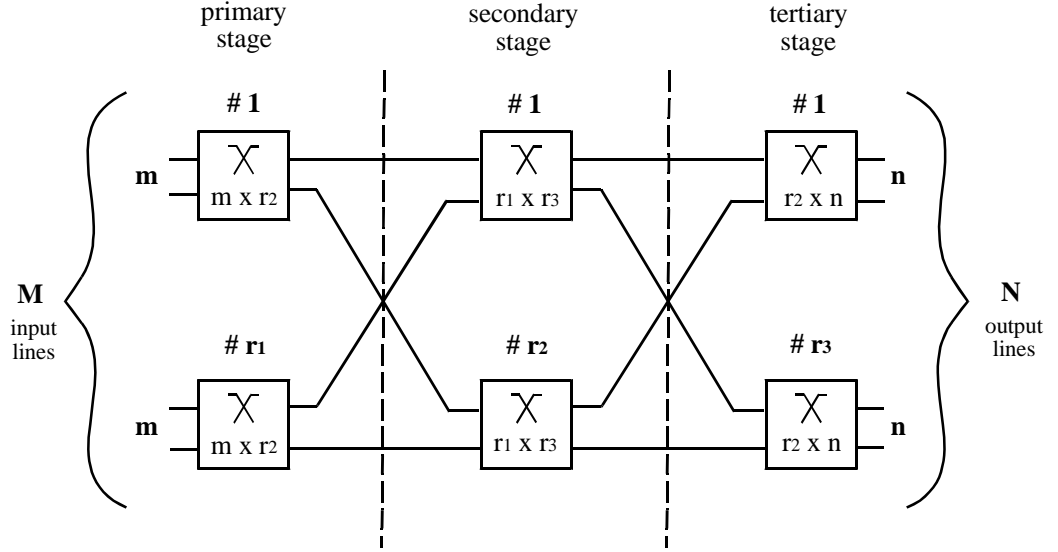


Figure 2.5: Three-Stage Switching Network

tertiary stage.

Theorem 1 (Clos) *A three-stage Clos network as shown in figure 2.5 is strict sense non-blocking if and only if*

$$r_2 \geq m + n - 1$$

To keep readability, the theorem was stated without proof. The interested reader may consult [165] or [101]. The latter presents a proof based on Paull's matrix. In the symmetric case, i.e. $M = N$, $r_1 = r_3 = \frac{N}{n}$ and $r_2 = n$, the number of crosspoints is given as

$$C = 2nr_1r_2 + r_1r_2r_3 = N(2n + \frac{N}{n}).$$

Such an arrangement introduces blocking to the switching network, as the Clos theorem is violated. In order to design a non-blocking arrangement, the number of switches in the secondary stage r_2 has to be set to $r_2 = (2n - 1)$ according to the theorem. As a result, the number of crosspoints increases to

$$C = 2nr_1r_2 + r_1r_2r_3 = (2n - 1)(2N + \frac{N^2}{n^2}).$$

An optimal arrangement similar to the two-stage case is found by differentiating the number of crosspoints with respect to n and is given by $n = \sqrt{\frac{N}{2}}$ thus providing the minimum number of crosspoints in a three-stage network, i.e.

$$C = 4\sqrt{2}N^{\frac{3}{2}} - 4N.$$

Again it should be taken into account, that the optimal n has to be approximated by integer values close to n . More information on three-stage Clos networks can be found in [165] and [101]. Three-stage concentrators and expanders are discussed in [79].

Further reductions in the number of crosspoints are achieved by rearranging existing connections in case of blocking conditions introduced by the switching network itself. This approach is due to Benes, who has shown, that rearrangement is able to eliminate blocking. Different to non-blocking networks, such networks do not naturally possess a zero blocking probability, it is moreover introduced by the rearrangement algorithm. Rearrangeable networks are not used in space division telephone switches, clicks caused by interruptions in current would severely affect quality of service. In order to avoid blocking in a $N \times N$ rearrangeable network, there must be at least N links at the intermediate stage. Before addressing the *Benes network*, some theorems useful for the design of rearrangeable networks are stated. The related proofs may be found in [165] and [101].

Theorem 2 (Slepian-Duguit) *A three-stage network is rearrangeable, if and only if*

$$r_2 \geq \max(m, n)$$

Theorem 3 (Paull) *The number of circuits, that need to be rearranged in a Slepian-Duguid network is at most $\min(r_1, r_3) - 1$.*

The basic element of a Benes network is a 2×2 switch unit arranged in $2 \log_2 N - 1$ stages. As every switching element accounts for four crosspoints, the total number of crosspoints is given by

$$C = 4N \log_2 N - 2N.$$

As shown in figure 2.6, the network may be developed following a recursive approach. Each primary switching element is connected to an upper and a lower switching network. The last stage is connected in a similar fashion.

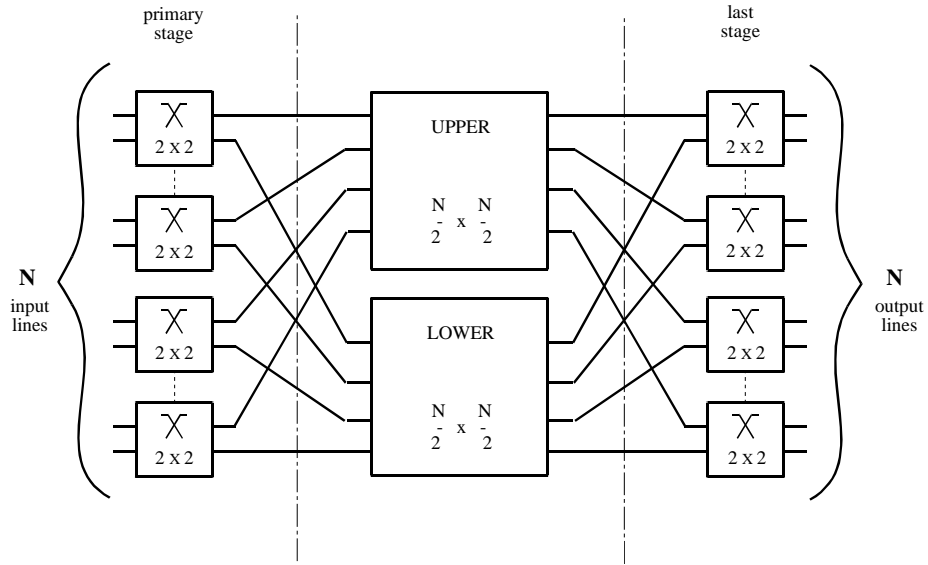


Figure 2.6: Recursive Factorization for the Benes Network

Both subnetworks either consist of a standard non-blocking network or follow the divide-and-conquer approach applied before. In a Benes network, the latter approach was chosen finally resulting in a switching network with an odd number of stages as stated above. The result for an 8×8 switching network is shown in figure 2.7.

In order to establish a new connection in a Benes network, the existing connections have to be rearranged. This can be achieved by the loop algorithm. This algorithm again follows a recursive approach by trying to interconnect the affected switch elements of the first and the last stage using either the upper or the lower subnetwork. If such a connection is not possible, an existing connection has to be rearranged to allow for the desired connection. The task of rearranging the existing connection is delegated to the subnetworks, while the switching elements in the first and last stage are configured for the new routing path. Rearranging an existing connection may be blocked by another connection. Such a connection has then to be rearranged too. At worst case, the process stops at the limit given by the Paull's theorem. Within the subnetworks the same procedure is applied to establish new routing paths within the subnetwork. For a detailed description of the loop algorithm, please refer to [165] and [101]. Even when a parallel

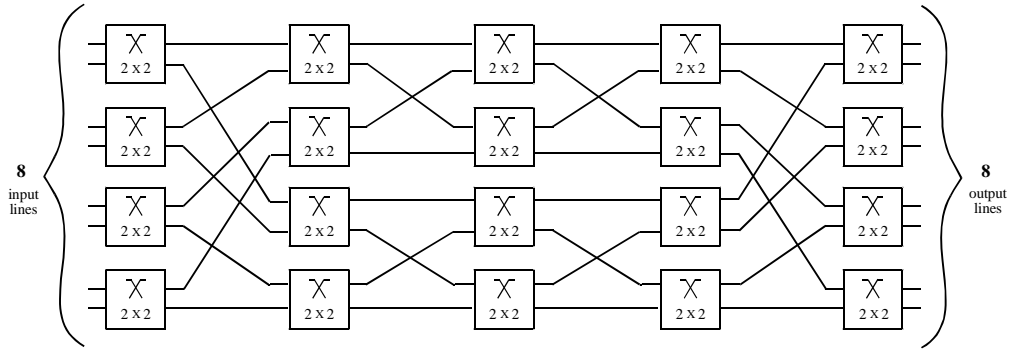


Figure 2.7: The complete Benes network for 8 inlets and 8 outlets

implementation is used for the loop algorithm, it is not sufficiently fast for packet switches. In circuit switching, rearrangement only occurs once per call.

Based on the mappings between stages called *network permutations*, various network architectures are possible. Some exploit the so called *self-routing* capability allowing the routing path to be selected based on binary address information. Such networks are built of 2×2 switching elements and belong to the classes of *banyan* and *delta networks*. These networks are generalized to sorting networks, which form the basic building block in non-blocking self-routing switches for ATM interconnection networks. In case of point to multipoint connections and multicast applications, so called *distribution* and *copy networks* are used. The latter allows for duplication of data within the network to deliver the same information to several output ports. More information about these networks may be found in [165], [101] and [155].

2.4 Interactive Voice Response Integration

An *interactive voice response system (IVR)* is a system that acts as an automated agent in a call centre. The caller interacts with the system through touch-tones or spoken words. The IVR replies with prerecorded statements or a synthesized voice. Advanced systems include voice mail, internet, *speech recognition* and *speech synthesis* capabilities. Some IVRs have the ability to use information residing in a database to screen and route calls. Call screening and call routing requires CTI features and interfaces to be implemented

in the IVR. The database either resides on the IVR itself or on an external host computer. Common database applications include account inquiries, information dissemination, order entry and transaction processing.

When used as a front-end for an ACD, callers may use the IVR to perform automated tasks while they wait in queue for a live agent. More calls can be handled at the call centre, because recurrent tasks have been offloaded to the IVR, while agents focus on non-routine tasks. IVRs enable customers to conduct business in their most convenient time, 24 hours a day, 7 days a week. The benefits of an IVR in a call centre can be summarized as follows:

1. Improved call processing
2. Increased call volume
3. Improved employee productivity and customer satisfaction
4. Increased revenue and reduced cost

The IVR can be configured as a stand alone unit or it can be attached to a switch. In the latter case, calls arriving at the switch are then transferred to the IVR. The IVR greets the caller, prompts for more information and transfers the call to the desired destination. IVRs can provide an *intelligent transfer*, a *blind transfer*, or both. During the intelligent transfer, the call is held in the IVR while being processed by an outside entity. Upon completion, the call is resumed and the IVR proceeds to interact with the calling customer. For a blind transfer call control is temporarily submitted to the target entity.

Signaling between the switch or ACD and the IVR is very often implemented by multifrequency tones conveying the signaling information. Some IVRs have digital signaling capabilities utilizing one of the CTI interface specifications described in section 2.1. Such IVRs are often manufactured for the private market and connect to a *private branch exchange (PBX)*. In public networks, IVRs communicate with a *SS7 service control point (SCP)* to provide services as part of the *intelligent network (IN)*. However, the IVR voice bearer channels terminate at the switching node, while signaling information is conveyed across the signaling network to the SCP.

2.5 Additional Call Centre Adjuncts

2.5.1 Call Accounting System

A *call accounting system* provides a cost management tool for monitoring call activity. Information is gathered from the PBX including time of the call, caller and calling identification, trunks used and routing information. Based on the duration and the destination of a call, the system creates a cost estimate for the call. In combination with an ACD, the call accounting system can also create call centre specific reports. In detail, a call accounting system provides the following services:

- Controlling telephone abuse and misuse
- Identification of the most cost-effective lines and connections
- Allocation of connection costs among departments, divisions and other organizational groups
- Statistics and evaluation of productivity
- Billing of customers at hotels and hospitals
- Quantification of cost and profitability of marketing campaigns
- Diagnosis of attached lines and trunks

Call accounting systems are usually connected to the switch using a *RS232/V.24 serial interface* [47] or an ethernet connection. If a single call accounting system supports more than one site, modems are used to bridge the distance. Data are delivered by the switch on a call-per-call basis in plain text format. Records containing fixed length fields like calling info, caller info, start of call, end of call, trunk, etc. are output to the call accounting unit in a defined format. Such a record is called *call detail record (CDR)* or *standard message detail record (SMDR)*. One example for a call detail recording format is the TELESEER format.

In relation to public networks, call accounting systems are also referred to as *billing applications*, as they provide a more complete cost management platform. Billing applications often provide feedback of billing data to the switch to enable the generation of charging tickets, e.g. *advice of*

charge (AOC) in ISDN. When leaving the typical telecommunication scenario, the billing application becomes more and more an integral part of the network. Especially in the VOIP world, information has to be retrieved from a vast number of heterogenous network elements, which provide and provision the services. Even though VOIP calls are billed similar to *plain ordinary telephone systems (POTS)* calls - prices are related to the distance - the underlying cost structure differs widely. Transmission cost are often determined on mean value calculations and cost estimations rather than being determined from exact network management records. In order to fill the gap between currently used billing applications, network management and customer databases, so called *IP mediation systems* have been developed. IP mediation systems collect data from various data sources and convert them to CDR records, a standard billing application is able to process. In classic telecommunications an office switch provides authentication, authorization and accounting. In an IP network, authentication and authorization is carried out by *proxy servers*, *firewalls* and RADIUS servers. *RADIUS* stands for *remote authentication dial-in user service* and is mainly used for remote access purposes. Even from that example it can be seen, that different network entities support different applications. The typical data flow in IP mediation systems can be described as follows:

1. A network event triggers a record flow in a router. The record contains low level information such as source and destination IP address, protocol type (usually TCP/UDP), the number of sent and received packets and start/end timestamps.
2. By interfacing to a network management application or to the router directly, the IP mediation system captures the record. In order to identify all involved network entities along the transmission path, the routing table is consulted.
3. The IP mediation system sends a request to the *quality of service (QoS) policy server* to determine the QoS type of the application, which triggered the network event in the router. The QoS policy server is typically a *RSVP policy manager* delivering information like source/destination address, start/end timestamps and requested QoS.
4. Peer network names and information about the internet geography is retrieved from the network management system.

5. In case of dynamic address allocation, e.g. remote access, a request to the RADIUS server is issued. By consulting its logs, the RADIUS server can deliver user name, login and logout times. The service contract associated to a specific user can be retrieved from a directory server (e.g. by using protocols like the *lightweight directory access protocol LDAP*).
6. Finally a CDR like record is created. Such a record is sometimes called *service data record SDR*. It contains information about source/target, start/end timestamps, user name, contract type and QoS data.

Such IP mediation systems can be implemented using a mixed approach - network information should be gathered in a distributed manner, whereas consolidation has to occur centrally. The latter is very important due to the fact, that duplicate source records might occur during operation, e.g. network events triggered by a router and a proxy server.

In order to cope with the requirement of online charging in VOIP and next generation wireless networks, the *Diameter* protocol has been developed by the Internet Engineering Taskforce (IETF) in RFC 3588, 4005 and 4006. In this context online charging is better described as credit control, which is commonly found in prepaid applications. Diameter adheres to the client-server paradigm and follows a request-response scheme to transmit time and event based offline charging information as well as credit control messages.

The description of IP mediation systems is based on a white paper by L. Schweitzer [184]. Information about accounting systems and billing applications is very sparse as they are often treated as little add-on to the overall network design. With more and more telephony like services carried over data networks, billing applications will grow from a simple adjunct to an essential part of the network. With services provided on an international base, e.g. long distance calls, intercarrier settlements have to be served. More information about charging of services in the classic telecommunications environment can be found in [104].

2.5.2 Reader Boards

Reader boards or *wall boards*, are liquid crystal display (LCD) or light emitting diode (LED) panels mounted on walls or hung from ceilings to provide agents and supervisors with easy-to-read information from the call centre.

Although the built in display on the agent's or supervisor's telephone might be an alternative, wall boards are still used, as they are more impressive and informative. Typical information provided by such panels include the number of calls in the queue, the oldest call waiting, and the number of agents who are available. Special messages regarding weather, pricing updates, crisis situations or birthday celebrations can also be displayed. Readerboards keep agents and supervisors aware of call center statistics thus helping to manage resources effectively.

Readerboards are usually connected to the call centre control using a *RS232/V.24 serial interface* [47] or a TCP/IP connection over Ethernet. The protocol used for the connection usually is a proprietary one. If the interface to the ACD is not supported by the readerboard, converter boxes are installed between the readerboard and the corresponding device.

Chapter 3

Queueing Theory

Queueing Theory plays a vital part in almost all investigations of service facilities. This chapter aims to provide an introduction to the highlights of queueing theory. Special attention is paid to multiserver systems, matrix analytic techniques and effects of customer impatience and retrial behaviour. This selection reflects the author's preference, which is biased in the direction of telephony and call centre applications. It has been tried to achieve a certain balance between theoretical rigor and practical value by providing applicable results for a wide range of queueing problems. Whenever no workable exact solutions are available, reasonable approximations are supplied instead.

3.1 Introduction to Queueing Theory

3.1.1 History

Queueing theory as part of probability theory has evolved from classic teletraffic engineering in the last decades. In 1909 A.K. Erlang, a Danish teletraffic engineer published a paper called *The Theory of Probabilities and Telephone Conversations*. In the early 1920s he developed the famous *Erlang model* to evaluate loss probabilities of multi-channel point-to-point conversations. The Erlang model was extended to allow for calculation in finite source input situations by T.O. Engset several years later leading to the *Engset model*. In 1951 D.G. Kendall published his work about *embedded Markov chains*, which is the base for the calculation of queueing systems under fairly general input conditions. He also defined a naming convention for queueing

systems which is still used. Nearly at the same time D.V. Lindley developed an equation allowing for results of a queueing system under fairly general input and service conditions. In 1957 J.R. Jackson started the investigation of networked queues thus leading to so called queueing network models. With the appearance of computers and computer networks, queueing systems and queueing networks have been identified as a powerful analysis and design tool for various applications.

3.1.2 Applications

As mentioned above, queueing theory allows for calculation of a broad spectrum of applications. These include

- In *manufacturing systems*, raw materials are transported from station to station using a conveyor belt. With each station having performed its task, the item is allowed to proceed to the next station. If processing times at all stations are equal and the conveyor belt is filled in the same frequency as items proceed from one station to the other, no waiting can occur, as the assembly line works in *synchronous* mode. In *asynchronous* mode, queueing for stations might occur and clearly has an impact on overall performance.
- *Computer systems* to perform real time or high speed operations are often subject to bad performance due to a single bottleneck device such as CPU, disk drive, graphics card, communication ports or bus system. By the use of analytical models the bottleneck device may be detected and as a consequence upgraded.
- By nature of the protocols used in *computer networks*, delays occur due to congestion of the transport network. These delays may be seen as waiting time until the media becomes free again thus allowing for calculation of throughput, overall delay and other performance values.
- *Teletraffic engineering* deals with the availability of stations, trunks and interconnection lines. Although these systems are characterized by *blocking* more than by delay, they still belong to the world of queueing systems. With the introduction of new media in teletraffic engineering, the delay paradigm becomes more important again. Teletraffic engineering now also has to cover a broad spectrum of new units such as

announcement boards, interactive voice response units, media servers, media and signaling gateways.

- *Workforce management* is concerned about the most efficient allocation of personell. The application of queueing theory in workforce management is most visible in call centres, where agents have to be allocated according to the call load. Relying on other techniques such as forecasting, queueing theory may be seen just as another brick in the wall in a wide range of solution methods to be applied to solve problems appearing in workforce management.

Obviously, the list above is far from being complete and may be extended further to other applications as well. For more information, the reader is referenced to publications such as *IEEE Communications Magazine*, *IEEE Computers*, *Bell Labs Technical System Journal* or similar.

3.1.3 Characterization

A queueing system may be described as a system, where customers arrive according to an *arrival process* to be serviced by a service facility according to a *service process*. Each service facility may contain one or more *servers*. It is generally assumed, that each server can only service one customer at a time. If all servers are busy, the customer has to queue for service. If a server becomes free again, the next customer is picked from the queue according to the rules given by the *queueing discipline*. During service, the customer might run through one or more *stages of service*, before departing from the system. A schematic representation of such a queueing system is given in figure 3.1. Before going into further detail, the most important aspects of queueing systems will be listed and briefly described.

- The *arrival process* is given by a statistical distribution and its parameters. Very often the exponential distribution is assumed resulting in the arrival pattern to be measured as the average number of arrivals per unit of time. When determining the trunk load in a PBX, the arrival pattern is often given in calls per busy hour. More general arrival processes are characterized by other pattern as well. These include batch arrivals and time dependence.

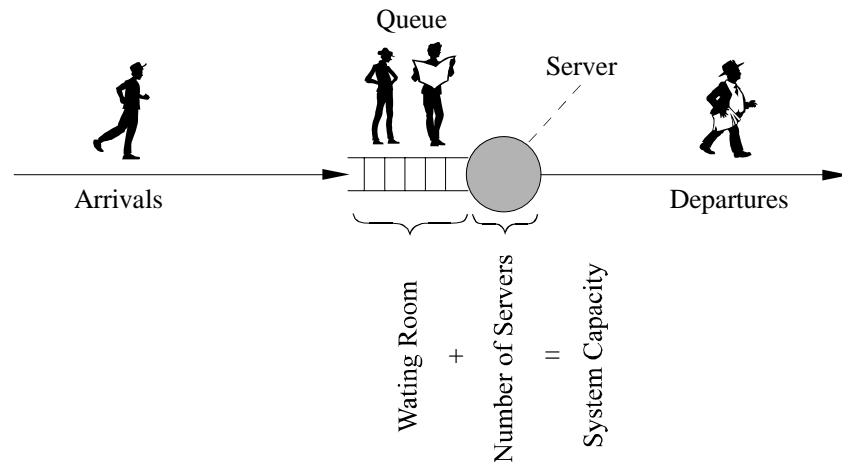


Figure 3.1: Schematic representation of a queueing system

- The *service process* is described similar to the arrival process. Again, exponentiality is often assumed in practice due to intractabilities when releasing these assumptions. In opposite to the arrival process, the service process is highly dependent on the state of the system. In case, the queueing system is empty, the service facility is idle.
- The *queueing discipline* refers to the way, customers are selected for service under queueing conditions. Often used and most common is the *first come, first serve (FCFS)* discipline. Others include *last come, first serve (LCFS)*, random and priority service.
- The *departure process* is seldom used to describe a queueing system, as it can be seen as a result of queueing discipline, arrival and service process. Under certain conditions, arrival and departure process follow the same statistical distribution. This has become a very important fact in queueing network modeling.
- The *system capacity* introduces a natural boundary in queueing systems. In life systems, there are only limited number of resources such as trunks in a PBX, computer memory or network buffers. In queueing networks, nodes with finite system capacities may *block* customers from the previous node, when the node's capacity limit has been reached.

- The *number of servers* refers to the number of parallel nodes, which can service customers simultaneously. In telephone systems servers might describe trunks, tone detectors, tone generators and time slots.
- The number and structure of *service stages*, a customer might have to visit before departing the system. In a computer system, a job might have to visit the CPU twice and the I/O processor once during a single task. In practice, there exist a lot of situations, which can be modeled by complex queueing systems with service stages or simple computer networks.

3.1.4 Use of Statistical Distributions in Queueing Systems

As mentioned above, arrival, service and departure processes are described by means of *statistical distributions*. The most common distributions are the exponential and Poisson distributions. Statistical distributions are adjusted for life situations by customizing their parameters. Clearly, the more parameters are available for a certain distribution, the more flexible it is. On the other hand, estimating a bunch of parameters might become an infeasible task. It also turns out, that more complex distributions result in almost intractable queueing models. So one is concerned with selecting a proper distribution leading to an analytical model which provides a close approximation to the life system under consideration. Sometimes the results are limited to a specific region only. One example are heavy load approximations, which fail to provide proper results for lightly loaded systems.

Exponential Distribution

The *exponential* distribution with density $f(t) = \lambda e^{-\lambda t}$ possesses only one parameter $\lambda > 0$ describing the average rate. For service facilities, very often the average service time $s = \frac{1}{\mu}$ is specified with μ the average service rate. A similar description is available for the arrival and departure processes. In assuming an exponential distribution for the arrival process one addresses the distribution of the times between subsequent arrivals t - the so called *interarrival times* t . This is graphically illustrated in figure 3.2. Although severely limited, the exponential distribution is widely accepted, as queueing models based on the exponential distribution are very easy to handle.

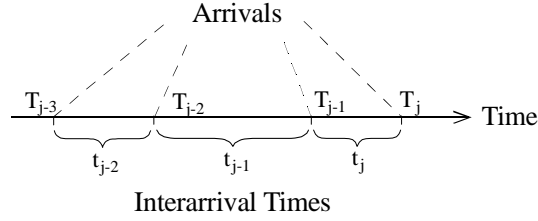


Figure 3.2: Interarrival times in an arrival process

Focusing on the exponential and the Poisson distribution, one arrives at a useful relation between number of arrivals and interarrival times. More formally, consider t_j as the time between two arrivals at T_j and T_{j-1}

$$t_j = T_j - T_{j-1}$$

assuming t_j for all j being exponentially distributed with parameter λ , i.e.

$$\Pr\{t_j \geq t\} = e^{-\lambda t}$$

then the number of arrivals N_t within $[0, t]$ follows a *Poisson* distribution:

$$\Pr\{N_t = j\} = f(j, \lambda) = \frac{(\lambda t)^j}{j!} e^{-\lambda t} \quad (3.1)$$

Without loss of generality, $t = 1$ might be assumed thus allowing for interpretation of λ as the average arrival rate. As an example consider a poisson probability mass function with $\lambda = 4.0$ as shown in figure 3.3. The graph reflects the probability of N customers arriving at a queueing system with average rate of arrivals of four customers per unit time. The probability density and cumulative distribution functions for the corresponding interarrival times are shown in figures 3.4 and 3.5.

One of the most appealing properties arising in queueing systems is the *memoryless or Markov property* of the exponential distribution. The memoryless property states, that the remaining (residual) time of an exponential process does not depend on the past. Consequences for the analysis of queueing systems include

- Given an exponentially distributed service time, a customer in service to be completed at some future time is independent of the time he has

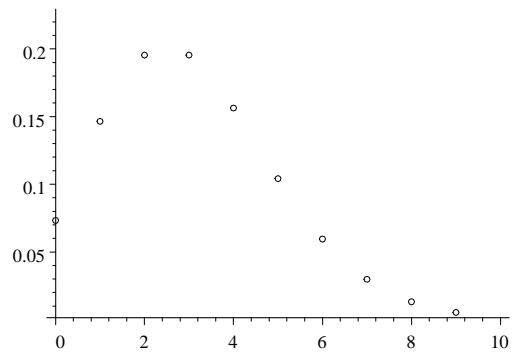


Figure 3.3: Poisson probability mass function with $\lambda = 4$

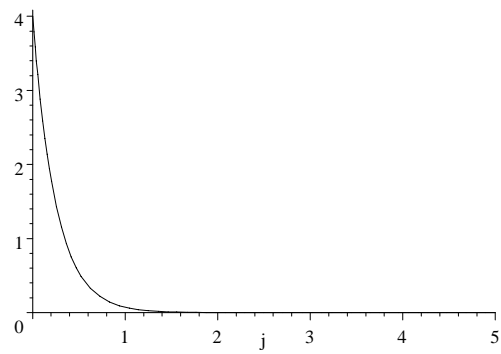


Figure 3.4: Exponential probability density function with $\lambda = 4$

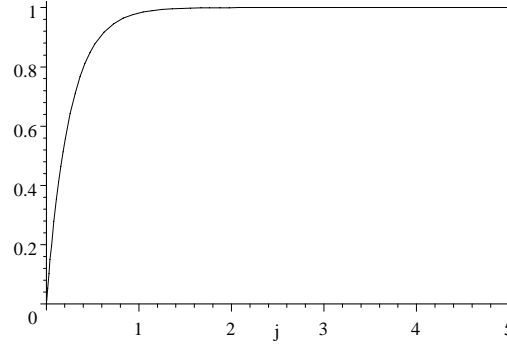


Figure 3.5: Exponential probability distribution function with $\lambda = 4$

been in service so far. The remaining service time is still exponentially distributed. End of work can be seen as a sudden event, not as a result of work progress. The server simply forgets, how long it has been operating.

- Given Poisson distributed arrivals, the time to the next arrival at any point of time is exponentially distributed.

Theorem 4 *The exponential distribution is memoryless.*

Proof. The proof is based on the definition of conditional probability. A random variable T is said to be memoryless, if

$$\Pr\{T > t + t_0 | T > t_0\} = \Pr\{T > t\}$$

Now a random variable T is assumed to be exponentially distributed with parameter λ , i.e. $\Pr\{T \leq t\} = 1 - e^{-\lambda t}$. Hence,

$$\begin{aligned} \Pr\{T > t + t_0 | T > t_0\} &= \frac{\Pr\{T > t + t_0\}}{\Pr\{T > t_0\}} \\ &= \frac{e^{-\lambda(t+t_0)}}{e^{-\lambda t_0}} = e^{-\lambda t} = \Pr\{T > t\} \end{aligned}$$

which completes the proof. ■

Furthermore, it can be shown, that the exponential distribution is the only continuous distribution exhibiting the memoryless property. For more

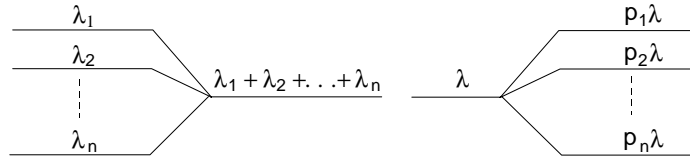


Figure 3.6: Merging and splitting of Poisson streams

information, refer to [92]. The discrete counterpart of the exponential distribution is the *geometric distribution*, which is commonly used to model cell based networks.

Due to the memoryless property of the exponential distribution, additional useful relations for Poisson processes may be derived. Given n independent Poissonian streams, interarrival times are exponentially distributed with parameter λ_i , where $i = 1 \dots n$, i.e. $F_i(t) = 1 - e^{-\lambda_i t}$, these streams may be merged to a single Poissonian stream, where interarrival times are distributed according to the distribution function $F(t) = 1 - e^{-(\lambda_1 + \lambda_2 + \dots + \lambda_n)t}$. Consequently, a single Poisson stream may be splitted up still preserving the Poissonian nature of each substream. The related interarrival times are exponentially distributed with parameter $p_i \lambda$, where p_i denotes the propability, that a single customer joins substream i . For a graphical representation of these relations, refer to figure 3.6. As a consequence, multiple independent Poisson arrival streams may be seen as a single arrival stream. On the other hand, a single Poisson arrival stream presented to multiple servers may be treated like multiple arrival streams. Special care has to be taken, if dependent streams or feedback effects are considered. The Poisson assumption does not necessarily hold under these circumstances.

In order to demonstrate several aspects of statistical distributions and their effect on queueing models, the analysis of a life system has been included as an example. It will be shown, how sampled data can be matched with an exponential service time distribution.

Example 5 Consider a call centre during the busy hour. Using the log of a CTI server, call holding times for each single call have been recorded. In total 1707 calls have been measured. These calls have been arranged in groups with unit time of 15 sec, i.e. the first group includes calls with a holding time of 0-14 sec, the second group includes calls with holding time between 15 and

29 seconds, etc. A call lasts 162 sec on the average, whereas the standard deviation σ of the holding time is 169, i.e. $\sigma = 169$. In order to visualize the distribution of calls, a histogram is shown in figure 3.7. Please note, that the word distribution is not used in the strong statistical sense. Giving a closer

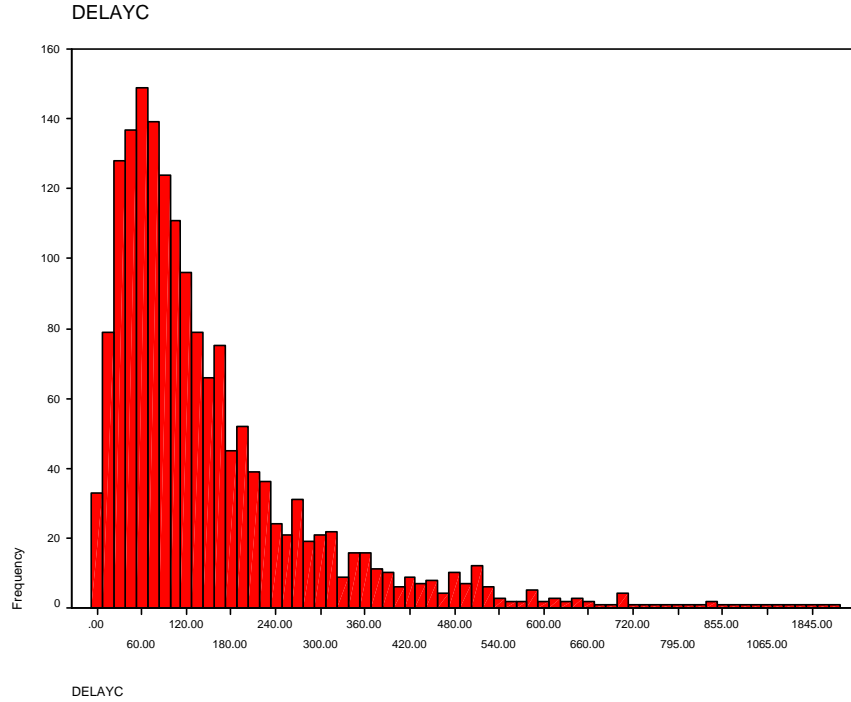


Figure 3.7: Histogram showing the distribution of calls in a call centre

look to figure 3.7, the shape suggests an exponential distribution. Mean and standard deviation of the exponential distribution are both equal to $\frac{1}{\lambda}$, and the measured data also exhibit a similar value for sample mean and standard deviation. We therefore ignore the slight difference and attempt a so called two-moment approximation. As a grouping of data with interval length 15 secs has been introduced, the average holding time will be scaled as well, i.e. $\frac{1}{\lambda} = 162/15 = 10.8$. Plotting the formula for the exponential probability density function (PDF)

$$f(t) = \lambda e^{-\lambda t} \quad (3.2)$$

reveals figure 3.8. The same procedure has been applied to the cumulative dis-

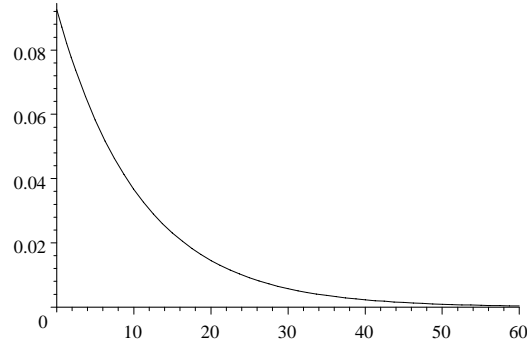


Figure 3.8: Fitted exponential PDF for $\frac{1}{\lambda} = 10.8$

tribution function (CDF)

$$F(t) = 1 - e^{-\lambda t} \quad (3.3)$$

to create figure 3.9. In order to compare the result with the histogram shown in figure 3.7, the probability density function has to be scaled by the number of calls 1707 on the y-axis and the interval length 15 on the x-axis. The resulting plot is shown in figure 3.10. By overlapping the two figures it turns out, that the fitted exponential distribution provides an acceptable approximation to the measured data. Thus we have justified the exponentiality assumption for this set of data. Please note, that usually the match between measured data and the chosen statistical distribution has to be verified by a so called goodness-of-fit test.

Method of Phases

Without the need to handle the residual times, the use of the exponential distribution became very popular. In order to overcome the shortcomings of the exponential distribution in queueing systems, very often mixtures of exponential distributions are used in standard models instead of deriving a new model suitable for the distribution required. In fact, it turns out, that these mixture distributions are highly flexible due to their extensive sets of parameters.

These families include the *Erlangian* (with density $f(t) = \frac{(\mu k)^k}{(k-1)!} t^{k-1} e^{-k\mu t}$) or the *hyperexponential* distribution (with density $f(t) = \sum_{i=1}^k \alpha_i \mu_i e^{-\mu_i t}$,

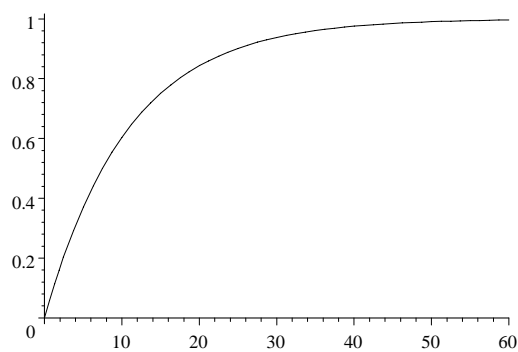


Figure 3.9: Fitted exponential CDF for $\frac{1}{\lambda} = 10.8$

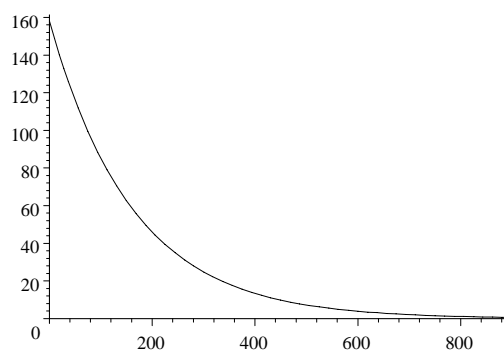


Figure 3.10: Scaled fitted exponential PDF for $\frac{1}{\lambda} = 10.8$

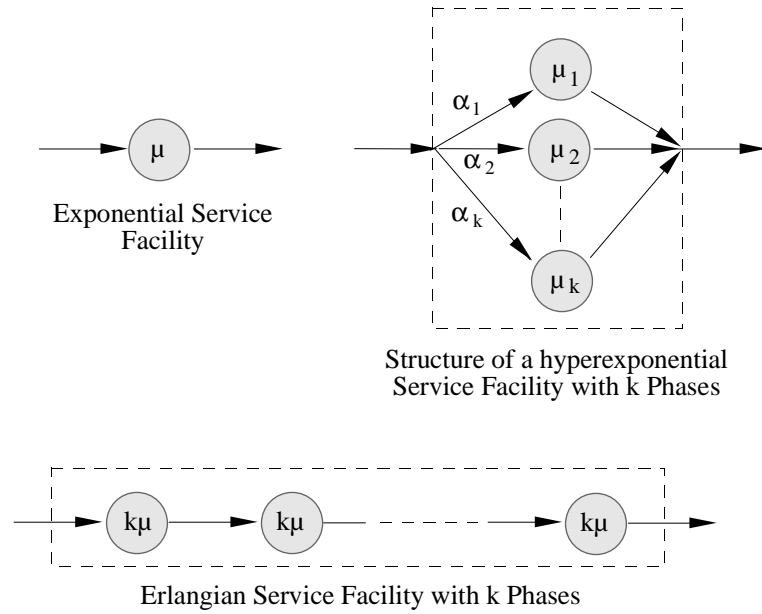


Figure 3.11: Simple and complex service facilities

$\mu_i > 0$). Seen from the perspective of a service facility, one complex service facility is replaced by a certain arrangement of more simple service facilities each having an exponentially distributed service time. For a graphical representation, please refer to figure 3.11. The Erlangian distribution provides a good starting point for systems with *phases* or *stages* such as conveyor belts used in manufacturing systems. The design pattern is purely sequential, whereas the hyperexponential service facility follows a parallel arrangement. For the densities above the number of stages is denoted by k . More general arrangements may be achieved by mixing sequential and parallel configurations resulting in so called *hyper-Erlang distributions*. A detailed treatment may be found in the book by Schassberger [180]. By interconnecting nodes rather arbitrarily, Neuts arrived at what he called *phase type distributions* [153]. State transitions between phases comply to Markovian requirements thus allowing for generalization of analytic methods for memoryless systems. This family of distributions includes all of the aforementioned as special cases and will be discussed in detail in section 3.8.1.

3.1.5 * Approximation of Arbitrary Distributions

In most practical situations one will rarely encounter distributions such as exponential and Erlang ones. This raises the question, which class of distributions might be sufficient to capture almost all situations in practice. Fortunately there is an answer to it. It turns out, that mixtures of *exponential distributions in serial and/or parallel* (also called *generalized Erlang distributions*) are capable of reasonably approximating any non-negative (and absolutely continuous) distribution. But there is more to gain without loss. As a matter of fact, the assumption of equal intensities in each branch does not affect the result[180]. This leads to the family of *hyper-Erlang distributions*, which consists of nothing else than mixtures of Erlang distributions [114]. Before proceeding to the result the concept of *weak convergence* of probability distributions has to be introduced.

Definition 6 *Given a series of distribution functions F_n and a distribution function F with $\lim_{n \rightarrow \infty} F_n(x) = F(x)$ for all continuity points x of F , then F_n is said to converge weakly (or in distribution). This is denoted by $F_n \rightharpoonup F$.*

The F_n above will become a sequence of Erlang mixtures and F denotes the distribution to be approximated. In case of an absolutely continuous distribution F the limit is valid for all x .

Theorem 7 *Choose F to be an arbitrary distribution on the positive reals $(0, \infty)$ with finite k -th moment $\mu_F^{(k)}$. Then for each n there exists a F_n out of the class of hyper-Erlang distributions, which converges weakly to F . Furthermore the moments $\mu_{F_n}^{(l)}$ of F_n converge to $\mu_F^{(l)}$ for all $l \leq k$.*

The proof is omitted here, as it consults concepts such as completeness and denseness in probabilistic metric spaces. For the question raised above, it is interesting to note, that the class of exponential distributions in serial/parallel is equivalent to the family of *Cox distributions*, which is in turn part of the class of phase type distributions. As a consequence each of the stated distribution families sufficiently approximates the desired target distribution. For a mathematical treatment of the subject the reader is referred to [30] and [180].

3.1.6 * Renewal Processes

In the previous sections we've learned that the Poisson process corresponds to exponential interarrival times. By relaxing the exponential assumption, one arrives at the so called *renewal process*. Renewal processes are characterized by independent interarrival times following a common distribution. They may be applied for the arrival as well as service processes, so in the following the event of an arrival will be called a *renewal*. Let T_n now denote the time between the $(n-1)$ st and n th renewal, $S_n = \sum_{k=1}^n T_k$ with $S_0 = 0$ the time of the n th renewal and $N(t) = \sup \{n : S_n \leq t\}$ the total number of renewals in the interval $[0, t]$. Then $N(t)$ for all $t \geq 0$ will formally describe the renewal process. Taking expectation one arrives at the *renewal function* $m(t) = \mathbb{E}N(t)$.

For the Poisson process the T_n were independent identically distributed according to an exponential distribution. Consequently the distribution of S_n results from the n -fold convolution of the exponential distribution, that is an n -stage Erlang distribution. $N(t)$ counts the number of renewals up to the time t , which describes the Poisson process.

By denoting $s = \mathbb{E}T_n = \int_0^\infty t dF(t)$ to be the expected renewal time (e.g. interarrival or service time), where T_n is identically distributed with distribution function F for all $n \geq 1$, one arrives at certain interesting limits

Theorem 8 *Based on the notation above the following limits hold*

$$\lim_{t \rightarrow \infty} \frac{m(t)}{t} = \frac{1}{s}$$

and

$$\Pr \left\{ \lim_{t \rightarrow \infty} \frac{N(t)}{t} = \frac{1}{s} \right\} = 1$$

The proof is based on the strong law of large numbers and is omitted here. The interested reader may consult [175] or [30]. The second limit holds only with probability one. That means, that there are exceptions to the rule, but these exceptions are negligible. In the context of arrival and service processes s simply describes the interarrival or service time. Consequently both limits converge to the arrival and service rates $\frac{1}{s}$. These results confirm our intuition: Observing a process for a very long time and dividing the number of occurrences by the total time, one arrives at the rate of that process.

By utilizing the central limit theorem, we are also able to derive asymptotic results for sufficiently large t . As $t \rightarrow \infty$, $N(t)$ is asymptotically normal distributed with mean $\frac{t}{s}$ and variance $\frac{t\sigma^2}{s^3}$ given the variance $\sigma^2 = \int_0^\infty (t-s)^2 dF(t)$ of the renewal distribution F exists. More details may be found in [30] and [60].

One could ask now, why the Poisson process plays such a prominent rule among the class of renewal processes. The answer lies in the fact of merging and splitting. It will turn out, that this feature is unique in the class of stationary renewal processes. A process $N(t)$ is called *stationary*, if a shift in time does not alter the distribution of the epochs, i.e. $N(t+s) - N(t)$ has the same distribution as $N(s)$.

In the current section we made use of the so called *Riemann-Stieltjes integral* [176]. In case of an absolutely continuous lifetime distribution F , the term $dF(t)$ may be replaced by $f(t)dt$ in the above formulas.

Theorem 9 *Given stationary renewal processes $N_1(t), \dots, N_n(t)$ and $N(t) = N_1(t) + \dots + N_n(t)$ each with common density function continuous on the interval $(0, \infty)$ and right continuous at 0, whereas the $N_1(t), \dots, N_n(t)$ are independent for all $t \geq 0$. Then $N_1(t), \dots, N_n(t)$ are all Poisson processes.*

Again the proof is omitted, because it requires the theory of point processes, which is not central to the current discussion. Also note the exact description of continuity above. It stems from the fact, that distributions defined for the positive reals can not be continuous at 0 from the left. For more information on the superposition of point processes consult [57].

3.1.7 Performance Characteristics of Queueing Systems

So far aspects of queueing models and statistical distributions have been discussed. As the usefulness of a model varies with its results, appropriate models and algorithms have to be selected. Another important factor is the point of view taken. Performance values calculated with respect to an arriving customer are not necessarily the same as those determined from a servers viewpoint. Again, the impact of statistical distributions is not negligible. However, it turns out, that these performance values are the same, when using models with exponentially distributed interarrival and service times. On the other hand, a lot of useful relations have been determined for more general cases as well. Although queueing models vary in application and

complexity, a common set of performance characteristics may be determined as follows.

- The *state probability* p_n is described by the probability of n customers residing in the system, either being served or waiting. Thus,

$$p_n = \Pr\{n \text{ customers in system}\}$$

- The *traffic intensity* ρ is given by the ratio of arrival rate λ and service rate μ , i.e.

$$\rho = \frac{\lambda}{\mu} \quad (3.4)$$

Alternatively, the traffic intensity may also be seen as the ratio of average service time $s = \frac{1}{\mu}$ and average interarrival time $t = \frac{1}{\lambda}$, i.e.

$$\rho = \frac{s}{t} \quad (3.5)$$

The traffic intensity is sometimes expressed in *erlangs* with respect to the Danish teletraffic engineer. In the United States very often *centum call seconds (CCS)* are used instead of erlang, as some manufacturers poll traffic sensitive equipment every 100 seconds [146]. In fact, a server being busy for an hour, carries a load of 36 CCS or equivalently 1 erlang. Expressed in a formula,

$$\rho_{ccs} = 36\rho_{ert}$$

- The proportion of time a server or a group of servers may be busy, is given by the *server utilization*

$$u = \frac{\lambda}{m\mu} = \frac{\rho}{m},$$

whereas m describes the number of servers in a queueing system. Please note, that a system with $u = 1$ is called a fully loaded system. Many common models are based on steady state concepts, which are comparable to the physical concept of equilibrium. As a consequence, they are not applicable to systems in overload, i.e. $u > 1$. Due to statistical effects, they don't provide proper results in fully loaded systems as well. Thus $u < 1$ defines a necessary stability condition for commonly used models.

- The *departure rate* or *throughput* X describes the average number of customers leaving the system. In a stable and work preserving system, the departure rate is usually equal to the arrival rate. The throughput is determined from the state probabilities and the service rate,

$$X = \sum_{n=1}^{\infty} \mu_n p_n \quad (3.6)$$

Please note, that a load dependent service rate has been assumed. In systems with multiple servers, μ_n is different for each state. Take as an example a call centre with 3 agents assuming each agent with the same average call handle time. With one agent being engaged, the effective service rate is μ . The other two agents are still waiting for a call and this can be identified with an idle server. If the second agent receives a call with the first agent still talking, the effective service rate becomes 2μ . When three or more agents are serving an active call, the effective service rate is 3μ . Clearly the fourth call in the system experiences a waiting time as he has to queue for service. Thus a load dependent service rate has to be assumed.

- The *average queueing time* W_q defines the time a customer has to wait, until service begins.
- The *average time in system* W defines the time between arrival and departure of a customer. The average time in system is related to the average waiting time as follows

$$W = W_q + s = W_q + \frac{1}{\mu} \quad (3.7)$$

- The *average queue size* L_q defines the average number of customers in the queue.
- The average system size L defines the average number of customers in the system and may be determined as follows

$$L = \sum_{n=1}^{\infty} n p_n \quad (3.8)$$

Please note, that starting the summation from $n = 1$ delivers the same result as starting from $n = 0$.

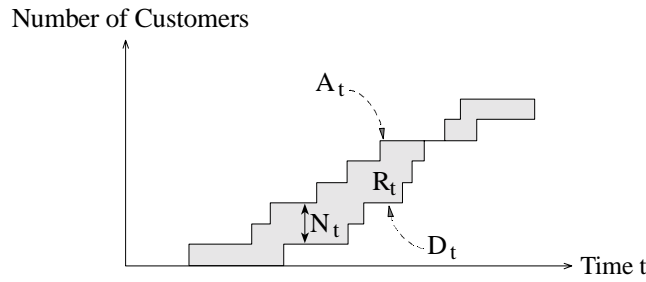


Figure 3.12: Little's Law

A very useful relation between average queueing time and the average number of customers in the system has been determined by J. D. C. Little in the year 1961. He found out, that given the average queueing time, the average queue size may be determined by simply multiplying the former with the arrival rate, i.e.

$$L_q = \lambda W_q \quad (3.9)$$

The same applies to the average system size and the average time in system

$$L = \lambda W \quad (3.10)$$

These relations are called *Little's Law*. Interestingly, Little's Law remains valid under very general assumptions. It does not assume any specific arrival distribution or service process, nor does it depend on the queueing discipline or the number of servers. With limited system capacity, Little's Law does still hold, but the arrival rate λ has to be redefined to exclude the number of customers lost due to blocking.

As shown in figure 3.12, Little's Law may also be derived graphically. By observing the number of customers entering and leaving a queueing system as functions of time in the interval $[0, t]$ denoted by A_t for the arrivals and D_t for the departures, the number of customers N_t in the system is given by

$$N_t = A_t - D_t$$

Defining arrival rate λ_t as

$$\lambda_t = \frac{A_t}{t}$$

Based on the area R_t between A_t and D_t , the average number of customers in the system L_t can be determined as follows

$$L_t = \frac{R_t}{t}$$

Please note, that R_t can be interpreted as the cumulated waiting time in interval $[0, t]$. The average waiting time W_t may now be calculated as the ratio between cumulated waiting time and the number of customers entering the system A_t , i.e.

$$W_t = \frac{R_t}{A_t}$$

Aggregating the last three formulas leads to

$$L_t = \frac{R_t}{t} = \frac{W_t A_t}{t} = W_t \lambda_t$$

Taking the limit as $t \rightarrow \infty$ results in Little's well known formula. Please note, that Little's Law only applies to the average values, but not to the entire distribution. Many proofs have been presented in the literature since 1961, the original text *A Proof of the Queueing Formula $L = \lambda W$* has been published in *Operations Research No. 9* by J. D. C. Little in the year 1961.

3.1.8 * Little's Law for Distributions

It can be said, that Little's law is one of the most important rules in queueing theory. Easy to understand and simple to use it is commonly applied for theoretical and practical purposes. It turns out, that Little's law may be further extended to become a distributional statement. Assume a queueing system, where customers are served one at a time and leave the system (or the queue) in the order of arrival (FCFS). Furthermore, customers entering the system (or the queue) shall remain in the system (or the queue) until served, i.e. there is no form of customer impatience. Now define the random variables \check{L}_q (number of customers in the queue) and \check{W}_q (queueing time). Also note the relations to the performance indicators used in the classic version of Little's law

$$L_q = \mathbb{E}\check{L}_q \quad W_q = \mathbb{E}\check{W}_q$$

Let the arrivals up to time t be described by a stationary renewal process $A(t)$. Then in steady state the average queue size \check{L}_q is distributed with $A(W)$, that is

$$\Pr \left\{ \check{L}_q \geq k \right\} = \Pr \left\{ A(\check{W}_q) \geq k \right\}$$

Given the time in system \check{W} and the number of customers in system \check{L} , a similar relation also holds for single server systems. With multiple servers, overtaking customers would violate the assumption of customers leaving the system in order of their arrival. Our approach is based on [40] and [30], a thorough treatment based on deterministic processes appears in [71].

3.1.9 Notation

Due to the wide range of applications, statistical distributions, parameters and disciplines, the number of queueing system models steadily increases. As a consequence, D. G. Kendall developed a shorthand notation for queueing systems. According to that notation, a queueing system is described by the string $A/B/X/Y/Z$, where A indicates the arrival distribution, B the service pattern, X the number of servers, Y the system capacity and Z the queueing discipline. Standard symbols commonly used in queueing systems are presented in table 3.1.

For example, the shorthand $M/D/3/100/PRI$ describes a queueing system with exponential interarrival times, 3 servers each with deterministic service time, a system capacity of 100 places and a priority service discipline. Clearly the exponential interarrival times directly relate to Poisson arrivals. Also note, that a system capacity of 100 places in a system with 3 servers specify a maximum queue size of 97. Please note, that not all symbols are mandatory, as symbols Y and Z may be omitted thus resulting in an abbreviated string $A/B/X$. In that case, the system capacity is unlimited and the queueing discipline is *first come first served* per default. Thus a queueing system denoted by $M/M/1/\infty/FCFS$ is commonly abbreviated by $M/M/1$.

Kendall's notation has been extended in various ways. One such extension will be adopted to cover the description of impatient customers. Following Bacelli and Hebuterne [34], an impatience distribution I will be added to the standard string, both separated by a plus sign. The distribution itself is defined similar to the first two elements A, B of the standard notation. The extended notation will then appear as $A/B/X/Y/Z + I$ in full length or as $A/B/X + I$ for the abbreviated notation.

characteristics	symbol	description
A - interarrival distribution	D	deterministic
	C_k	Cox (k phases)
	E_k	Erlang (k phases)
	G	general
	GI	general independent
	GEO	geometric (discrete)
	H_k	hyperexponential
	M	exponential (Markov)
	ME	matrix exponential
	MAP	Markov arrival process
	PH	phase type
B - service time distribution	D	deterministic
	C_k	Cox (k phases)
	E_k	Erlang (k phases)
	G	general
	GI	general independent
	GEO	geometric (discrete)
	H_k	hyperexponential
	M	exponential (Markov)
	ME	matrix exponential
	PH	phase type
	SM	semi-Markov
X - number of parallel servers	$1, 2, \dots, \infty$	
Y - system capacity	$1, 2, \dots, \infty$	
Z - queueing discipline	$FCFS$	first come first serve
	$LCFS$	last come first server
	RSS	random selection for service
	PRI	priority service
	RR	round robin
	PS	processor sharing
	GD	general

Table 3.1: Kendall notation for queueing systems

The symbols mentioned in table 3.1 are not exclusive, as some characteristics can be seen as generalizations or specifications of other characteristics. So an exponential distribution may be seen as Erlang distribution with one phase, which in turn is a specialization of the phase type distribution. As a consequence, the capabilities of queueing models may be deduced from this short description. Formulas derived for $M/G/1$ are generalizations of the formulas used in $M/M/1$ systems.

As queueing theory originated from congestion theory in telephone systems, some application specific models survived over the years. The most common is the so called *lost calls cleared (LCC)* system, which can be expressed as $M/M/c/c$ model using Kendall notation. The LCC system does not have any waiting places, calls arriving to a system with all servers busy are *cleared*. As trunks in telephone systems usually do not have a queueing mechanism, the LCC model suits the need of calculating required trunk resources for a given offered load. The counterpart of the LCC system is the *lost calls held (LCH)* system, which relates to $M/M/c/K$ and $M/M/c$ models. Customers, which can not be immediately served on arrival are put in a queue. These models are commonly used to dimension the desired tone detector or tone generator resources in telephone systems given a certain waiting time objective.

3.2 Simple Queueing Models

The most basic model in queueing theory is the $M/M/1$ model. In this section the mathematical derivation will be combined with intuitive insights to prepare the path for more complex models. In a $M/M/1$ model, random arrivals and exponentially distributed service times are assumed. Please note, that *random arrivals* is exactly defined to be Poisson in statistics. Furthermore, there is only a single server serving customers on a first come, first serve base. The population is infinite, so arriving customers are unaffected by the queue size. Parameters given for the $M/M/1$ model are λ , the average arrival rate, μ , the average service rate, which may be calculated from the average service time $\mu = \frac{1}{s}$. Bulk arrivals and group service are not allowed for this type of model.

By giving a closer look on the arrivals to a queueing system, it turns out, that time may be partitioned in slices with length Δt such, that only one arrival per slice is allowed. By following a linear approach, it can be assumed,

that the number of customers entering the system during interval $[t, t + \Delta t]$ are proportional to λ , i.e.

$$\begin{aligned}\Pr\{\text{single arrival in } [t, t + \Delta t]\} &= \lambda \Delta t \\ \Pr\{\text{no arrival in } [t, t + \Delta t]\} &= 1 - \lambda \Delta t \\ \Pr\{\text{more than one arrival in } [t, t + \Delta t]\} &= 0\end{aligned}$$

Please note, that more arrivals during interval $[t, t + \Delta t]$ can be handled by introducing a function $o(\Delta t)$. Due to the fact, that this term will vanish in the subsequent derivation under very general assumptions, the presented results will still hold.

The same arguments may be applied to the service process as well, but it has to be taken into account, that departures can only occur, if the system is not empty, i.e.

$$\Pr\{\text{single departure in } [t, t + \Delta t] | \text{system not empty}\} = \mu \Delta t$$

By defining $p_n(t + \Delta t)$ as the propability of $n > 0$ customers residing in the system at time $t + \Delta t$, an expression may be found in terms of the propabilities at time t , i.e.

$$\begin{aligned}p_n(t + \Delta t) &= p_n(t) \Pr\{\text{no departure}\} \Pr\{\text{no arrival}\} \\ &\quad + p_{n-1}(t) \Pr\{1 \text{ arrival in } [t, t + \Delta t]\} \\ &\quad + p_{n+1}(t) \Pr\{1 \text{ departure in } [t, t + \Delta t]\}\end{aligned}$$

Due to the conditional nature of the departure process, a different equation has to be specified for state 0

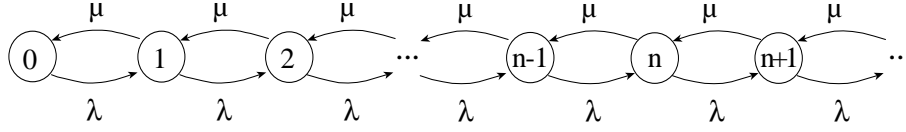
$$\begin{aligned}p_0(t + \Delta t) &= p_0(t) \Pr\{\text{no arrival in } [t, t + \Delta t]\} \\ &\quad + p_1(t) \Pr\{1 \text{ departure in } [t, t + \Delta t]\}\end{aligned}$$

Expressed in formulas

$$\begin{aligned}p_n(t + \Delta t) &= p_n(t)(1 - \lambda \Delta t)(1 - \mu \Delta t) \\ &\quad + p_{n-1}(t)\lambda \Delta t + p_{n+1}(t)\mu \Delta t \quad \text{for } n > 1 \\ p_0(t + \Delta t) &= p_0(t)(1 - \lambda \Delta t) + p_1(t)\mu \Delta t \quad \text{for } n = 0\end{aligned}$$

Rearranging terms and taking the limit $\Delta t \rightarrow 0$ gives

$$\begin{aligned}\frac{dp_n(t)}{dt} &= -(\lambda + \mu)p_n(t) + \lambda p_{n-1}(t) + \mu p_{n+1}(t) \quad \text{for } n > 1 \\ \frac{dp_0(t)}{dt} &= -\lambda p_0(t) + \mu p_1(t) \quad \text{for } n = 0\end{aligned} \tag{3.11}$$

Figure 3.13: State transition diagram for the $M/M/1$ model

At this point it is guaranteed, that only one arrival per time slice Δt can arrive. Please note, that the exception of bulk arrivals and group service has been excluded in the beginning of this section. Formula 3.11 may be used for a time dependent, so called *transient* analysis of the $M/M/1$ model. If someone is interested in the long term behaviour of the system, the so called *steady state equations* have to be determined. These steady state equations are related to the concepts of stochastic balance and furthermore physical gravity, as will be shown below. Assuming a balanced state, changes over time are assumed to be negligible, i.e. $\frac{dp_n(t)}{dt} = 0$ leads to

$$\begin{aligned} 0 &= -(\lambda + \mu)p_n + \lambda p_{n-1} + \mu p_{n+1} & \text{for } n > 1 \\ 0 &= -\lambda p_0 + \mu p_1 & \text{for } n = 0 \end{aligned} \quad (3.12)$$

Queueing models are often analyzed by using so called *state transition diagrams*. Such a diagram is shown in figure 3.13 for the $M/M/1$ model. With a single look, flows into a state and flows out of a state may be determined. For the $M/M/1$ model it turns out, that the flow out of each state $n > 0$ is always $\lambda + \mu$, whereas the flow into a state n are λ from the previous state $n - 1$ and μ from the next state $n + 1$. By following the intuitive concept of balance, i.e. by equating the rates into a state with the rates out of a state, the following equations may be written down immediately

$$\begin{aligned} (\lambda + \mu)p_n &= \lambda p_{n-1} + \mu p_{n+1} & \text{for } n > 1 \\ \lambda p_0 &= \mu p_1 & \text{for } n = 0 \end{aligned}$$

Comparing these equations with formula 3.12 above, they turn out to be the same. Thus analysis of a wide range of queueing models may be carried out based on concepts of balance. As a next step, the steady state probabilities of the $M/M/1$ model are to be derived. Rearranging equation 3.12 results in

$$\begin{aligned} p_{n+1} &= \frac{\lambda + \mu}{\mu} p_n - \frac{\lambda}{\mu} p_{n-1} & \text{for } n > 1 \\ p_1 &= \frac{\lambda}{\mu} p_0 & \text{for } n = 0 \end{aligned}$$

By using the definition of the traffic intensity $\rho = \frac{\lambda}{\mu}$, and continuously substituting, i.e.

$$\begin{aligned} p_1 &= \rho p_0 \\ p_2 &= \rho p_1 = \rho^2 p_0 \\ p_3 &= \rho p_2 = \rho^3 p_0 \\ &\dots \\ p_n &= \rho p_{n-1} = \rho^n p_0 \end{aligned}$$

the state propabilities may be easily determined. As a last step, the propability of state 0 p_0 can be derived by recalling the fact, that propabilities always sum to 1, i.e.

$$1 = \sum_{n=1}^{\infty} p_n = \sum_{n=1}^{\infty} \rho^n p_0$$

Rewriting and utilizing the definition of a geometric series leads to

$$p_0 = \frac{1}{\sum_{n=1}^{\infty} \rho^n} = 1 - \rho$$

and finally to

$$p_n = \rho^n (1 - \rho) \tag{3.13}$$

Please note, that the arrival rate is not allowed to exceed the service rate due to stability of the system. This may be expressed by the stability conditions $\rho < 1$ or $\lambda < \mu$. In order to determine the performance characteristics of the $M/M/1$ system, one statistic has to be derived from the steady state propabilities. Most of the other statistics may be concluded from the relations presented in the previous section. Recalling the fact, that the average system size L is defined by

$$L = \sum_{n=0}^{\infty} n p_n$$

and substituting for p_n leads to

$$\begin{aligned}
 L &= (1 - \rho) \sum_{n=0}^{\infty} n \rho^n \\
 &= (1 - \rho) \rho \sum_{n=1}^{\infty} n \rho^{n-1} \\
 &= (1 - \rho) \rho \frac{d(\sum_{n=0}^{\infty} \rho^n)}{d\rho} \\
 &= (1 - \rho) \rho \frac{d(\frac{1}{1-\rho})}{d\rho} \\
 &= \frac{(1 - \rho) \rho}{(1 - \rho)^2} = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda}
 \end{aligned} \tag{3.14}$$

Please note that $\sum_{n=1}^{\infty} n p_n = \sum_{n=0}^{\infty} n p_n$ and that $\sum_{n=1}^{\infty} n \rho^{n-1}$ is the first order derivative of $\sum_{n=0}^{\infty} \rho^n$. By using Little's Law, the average time in system may be calculated as follows

$$W = \frac{1}{\lambda} L = \frac{1}{\mu(1 - \rho)} = \frac{1}{\mu - \lambda}$$

As a next step the average queueing time may be determined by simply subtracting the service time, i.e.

$$W_q = W - s = W - \frac{1}{\mu} = \frac{\rho}{\mu(1 - \rho)} = \frac{\lambda}{\mu(\mu - \lambda)}$$

Applying Little's Law another time finally leads to the average queue size

$$L_q = \lambda W_q = \frac{\lambda^2}{\mu(\mu - \lambda)} = \frac{\rho^2}{1 - \rho} \tag{3.15}$$

Based on the equations above, another useful relation may be obtained between the system size L and the queue size L_q

$$L = L_q + \rho \tag{3.16}$$

Due to the fact, that Little's Law holds under very general distribution assumptions, equation 3.16 is valid for almost any queueing system with a single server. In case of system capacity limitations, the law still holds, but only with slight modification of the arrival rate.

Example 10 Consider a database system with an average service time of 450 msec. As database requests are initiated by a large number of clients, a random arrival pattern may be assumed. Thus the arrival process is assumed to be Poisson. On the average a new database query arrives every 500 msec. Service times are assumed to be exponentially distributed, the queueing discipline is assumed to follow a first come, first serve pattern. The numbers given for interarrival and service times lead to

$$\begin{aligned}\lambda &= \frac{1}{0.5} \text{ sec} \\ \mu &= \frac{1}{0.45} \text{ sec} \\ u &= \rho = \frac{\lambda}{\mu} = 0.9\end{aligned}$$

With 90% utilization the database server can be deemed heavily loaded. This is also reflected in the average system size and the average answer times, i.e.

$$\begin{aligned}L &= \frac{\rho}{1 - \rho} = \frac{0.9}{0.1} = 9 \\ W &= \frac{1}{\lambda}L = \frac{1}{2}9 = 4.5 \text{ sec}\end{aligned}$$

Due to the high frustration level, it was decided to replace the harddisc by a faster model. As a consequence service times were reduced to 350 msec. As a result, performance characteristics improved, i.e.

$$\begin{aligned}\lambda &= \frac{1}{0.5} \text{ sec} \\ \mu &= \frac{1}{0.35} \text{ sec} \\ u &= \rho = \frac{\lambda}{\mu} = 0.7 \\ L &= \frac{\rho}{1 - \rho} = \frac{0.7}{0.3} = 2.333 \\ W &= \frac{1}{\lambda}L = \frac{1}{2}2.333 = 1.167 \text{ sec}\end{aligned}$$

As can be seen from this example, heavy loaded systems are very sensible to small changes. In this case, the answer times have been reduced to a quarter of the original answer times only by a compareable small increase in service speed.

3.2.1 * Idle and Busy Period

The work process of a queueing system may be splitted into idle and *busy periods*. The latter starts with an arrival at an empty system and ends, when the system becomes idle again. A *busy cycle* is the time between two successive arrivals at an empty system. It is also the sum of a busy and an adjacent idle period [90]. Let the random variable T_{busy} denote the busy time. As shown in [172], the density for a stable $M/M/1$ queue with one customer present in the system at time 0 is given by

$$f_{idle}(t) = \sqrt{\frac{\mu}{\lambda}} \frac{I_1(2\sqrt{\lambda\mu}t)}{t} e^{-(\lambda+\mu)t}, \quad t > 0 \quad (3.17)$$

where I_1 is the modified Bessel function of order 1,

$$I_1(x) = \sum_{k=0}^{\infty} \frac{(x/2)^{2k+1}}{k!(k+1)!}$$

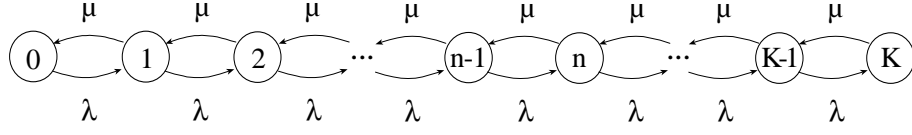
Please note, that this argument is transient, i.e. depends on time t . Based on the density 3.17 or other means [90], one may derive mean and variance of T_{busy} :

$$\begin{aligned} \mathbb{E}T_{busy} &= \frac{1}{\mu - \lambda} \\ Var(T_{busy}) &= \frac{\mu + \lambda}{(\mu - \lambda)^3} \end{aligned}$$

Because of the memoryless property of the arrival process, the idle period T_{idle} is exponentially distributed with mean $\frac{1}{\lambda}$. The distribution of the busy cycle is simply the convolution of the idle time and busy period distribution. The analysis of the busy period is a standard concept in queueing theory, a classic reference is [90].

3.2.2 Capacity Constraints

Consequently, the next step is to extend the $M/M/1$ model to include a system capacity constraint thus becoming a $M/M/1/K$ model. A similar condition as introduced for state 0 before has to be applied to state K as well. This is also reflected in the state transition diagram in figure 3.14. Transition

Figure 3.14: State transition diagram for the $M/M/1/K$ model

rates and state probabilities for states $1 \dots K$ remain the same as given by the $M/M/1$ model, i.e.

$$p_n = \rho^n p_0$$

Only p_0 has to be derived with respect to the limited system capacity. Following the same approach as before leads to

$$\begin{aligned}
 1 &= \sum_{n=0}^K p_n = \sum_{n=0}^K \rho^n p_0 \\
 p_0 &= \frac{1}{\sum_{n=0}^K \rho^n} = \begin{cases} \frac{1-\rho}{1-\rho^{K+1}} & \text{for } \rho \neq 1 \\ \frac{1}{K+1} & \text{for } \rho = 1 \end{cases} \\
 p_n &= \begin{cases} \frac{(1-\rho)\rho^n}{1-\rho^{K+1}} & \text{for } \rho \neq 1 \\ \frac{1}{K+1} & \text{for } \rho = 1 \end{cases}
 \end{aligned}$$

Due to the capacity constraint, the stability condition is no longer required. Customers arriving to a system with busy servers and full queue are lost. In order to determine the performance characteristics of the $M/M/1/K$ model, the same procedure is applied as has been done before, i.e.

$$\begin{aligned}
 L &= \sum_{n=0}^K n p_n = p_0 \rho \sum_{n=1}^K n \rho^{n-1} \\
 &= p_0 \rho \frac{d(\sum_{n=0}^K \rho^n)}{d\rho} = p_0 \rho \frac{d}{d\rho} \left(\frac{1-\rho^{K+1}}{1-\rho} \right) \\
 &= p_0 \rho \frac{1 - (K+1)\rho^K + K\rho^{K+1}}{(1-\rho)^2} \\
 &= \begin{cases} \frac{\rho(1-(K+1)\rho^K + K\rho^{K+1})}{(1-\rho^{K+1})(1-\rho)} & \text{for } \rho \neq 1 \\ \frac{\sum_{n=0}^K n}{K+1} = \frac{K}{2} & \text{for } \rho = 1 \end{cases}
 \end{aligned}$$

Little's Law is still applicable, but based on the fact, that no customers are lost. Therefore, an *effective arrival rate* has to be calculated. The effective arrival rate is dependent on the fraction of calls, which succeeded in entering the system. Giving a closer look on the state transition diagram in figure 3.14, it turns out, that up to state $K - 1$, an arriving customer always can enter the system. But in state K , the call is *blocked* and as a consequence, lost. Thus $\sum_{n=0}^{K-1} p_n$ describes the propability, that customers can enter the system and p_K refers to the so called *blocking propability*. Considering the fact, that both propabilities sum to 1 leads to the following calculation for the effective arrival rate $\bar{\lambda}$

$$\bar{\lambda} = \lambda(1 - p_K)$$

By proceeding as before, all important performance characteristics may be obtained

$$\begin{aligned} W &= \frac{1}{\bar{\lambda}} L \\ W_q &= W - \frac{1}{\mu} \\ L_q &= \bar{\lambda} W_q \end{aligned}$$

Consequently, the relation between average queue size and average system size has to be reviewed. Equation 3.16 has to be modified to include the effective arrival rate $\bar{\lambda}$ instead of the arrival rate λ .

3.2.3 Queueing Disciplines

So far only first come, first serve has been assumed as a queueing discipline. As all performance characteristics mentioned are averages in a statistical sense, they are very insensible to changes of the queueing discipline. All results remain valid for *last come, first serve (LCFS)*, *round robin (RR)* and *processor sharing (PS)* disciplines. In priority systems, customers are grouped in classes and separate characteristics are derived for each class. Although the average values remain the same, the underlying distributions change with the queueing discipline. This is a common feature in queueing theory. One often assumes a system or queueing discipline to be *work conserving*, that is [197]

- the server does not remain idle with customers waiting

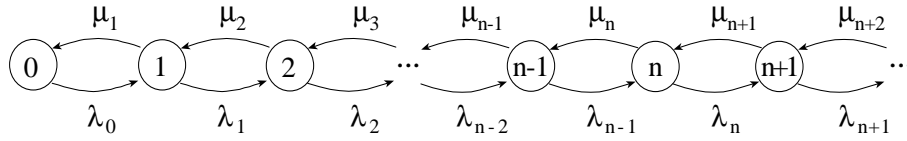


Figure 3.15: State transition diagram for a load dependent system

- the queueing discipline does not affect the arrival time of any message
- the queueing discipline does not affect the amount of service time

If one works within the class of work conserving queueing disciplines, the performance key indicators such as average queueing time and average system size will remain untouched by the choice of a dedicated member. Note, that such an invariance property does not hold for the corresponding distributions. A very good discussion on the topics related to different queueing disciplines is found in [59].

3.3 Birth-Death Process

Both, the $M/M/1$ and the $M/M/1/K$ model considered so far, are special cases of a more general system, which can be modeled by the so called *birth-death process*. The name is related with an application in biology, the birth-death process provides a simple frame to model populations of any sort. In view of technical systems, the birth-death process may be used to model *load dependent systems*. In such a system, arrival and service rates are dependent on the current state of the system. This is also reflected in the state transition diagram in figure 3.15. A wide range of queueing systems can be modeled by customizing the parameters of the birth-death process. Please note, that still exponential interarrival and service time distributions are assumed.

Although more complex, the balance approach still provides easy access

to the solution

$$\begin{aligned}
 p_1 &= \frac{\lambda_0}{\mu_1} p_0 \\
 p_2 &= \frac{\lambda_1}{\mu_2} p_1 = \frac{\lambda_1 \lambda_0}{\mu_2 \mu_1} p_0 \\
 &\dots \\
 p_n &= \frac{\lambda_{n-1}}{\mu_n} p_{n-1} = \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} p_0
 \end{aligned} \tag{3.18}$$

As all propabilities sum to 1, p_0 may be readily obtained as follows

$$\begin{aligned}
 1 &= \sum_{n=0}^{\infty} p_n = p_0 + \sum_{n=1}^{\infty} p_n = p_0 + \sum_{n=1}^{\infty} \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} p_0 \\
 p_0 &= \frac{1}{1 + \sum_{n=1}^{\infty} \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i}}
 \end{aligned} \tag{3.19}$$

The average number of customers in the system L may be determined as usual

$$L = \sum_{n=1}^{\infty} n p_n \tag{3.20}$$

By defining the system throughput X as follows

$$X = \sum_{n=1}^{\infty} \mu_n p_n \tag{3.21}$$

and using Little's Law leads to the average time in system W

$$W = \frac{1}{X} L = \frac{\sum_{n=1}^{\infty} n p_n}{\sum_{n=1}^{\infty} \mu_n p_n} \tag{3.22}$$

The formulas of the birth-death process may be applied to systems with limited capacity as well, but with a slight modification. The upper summation limit of each equation has to be replaced by the system capacity K . Before proceeding with more complex variants, the models considered so far are reviewed in terms of the birth-death process.

Example 11 *The following parametrization of the birth death process leads to the same results as the $M/M/1$ model mentioned above*

$$\begin{aligned}
 \lambda_n &= \lambda \quad \text{for all } n \\
 \mu_n &= \mu \quad \text{for all } n
 \end{aligned}$$

There is also a parametrization available for the $M/M/1/K$ model

$$\begin{aligned}\lambda_n &= \begin{cases} \lambda & \text{for } n = 0 \dots K-1 \\ 0 & \text{for } n > K-1 \end{cases} \\ \mu_n &= \begin{cases} \mu & \text{for } n = 1 \dots K \\ 0 & \text{for } n > K \end{cases}\end{aligned}$$

Please note, by introducing conditions to the parameters, interchanging the upper summation limit of the original formulae becomes obsolete.

3.3.1 Multiserver Systems

Queueing systems with multiple servers may be modeled by a single server system with state dependent service rate. Given n customers are in the system, work is processed n times as fast as a single server would need to do so. Given a limited supply of servers, the load dependent service rate remains the same, if the limit is exceeded. The related model is called $M/M/c$ in the limited case and $M/M/\infty$ in the unlimited case. The latter system is also called *delay server*, as the average answer time is insensible to the number of customers currently in the system. As a single system, the delay server is almost useless, but if combined with other systems to a *queueing network*, it plays an important role. The $M/M/c$ requires the following parametrization

$$\begin{aligned}\lambda_n &= \lambda & \text{for all } n \\ \mu_n &= n\mu & \text{for } 1 \leq n \leq c \\ \mu_n &= c\mu & \text{for } n > c\end{aligned}$$

Substituting these parameters in equations 3.18 and 3.19 leads to

$$\begin{aligned}p_n &= \begin{cases} \prod_{i=0}^n \frac{\lambda}{i\mu} p_0 = \frac{1}{n!} \rho^n p_0 & \text{for } 1 \leq n \leq c \\ p_c \prod_{i=0}^n \frac{\lambda}{c\mu} = \frac{1}{c!c^{n-c}} \rho^n p_0 & \text{for } n > c \end{cases} \\ p_0 &= \left(1 + \sum_{n=1}^{c-1} \frac{1}{n!} \rho^n + \sum_{n=c}^{\infty} \frac{1}{c!c^{n-c}} \rho^n \right)^{-1} \\ &= \left(\sum_{n=0}^{c-1} \frac{1}{n!} \rho^n + \sum_{n=c}^{\infty} \frac{1}{c!c^{n-c}} \rho^n \right)^{-1} \\ &= \left(\sum_{n=0}^{c-1} \frac{1}{n!} \rho^n + \frac{1}{c!} \rho^c \frac{1}{1 - \frac{\rho}{c}} \right)^{-1} \end{aligned} \tag{3.23}$$

As no system capacity constraint has been defined, a stability condition is required to preserve proper analytical results. Based on the intuitive argument, that not more customers should arrive than can be served, a stability condition may be written down immediately

$$\frac{\lambda}{c\mu} = \frac{\rho}{c} = u < 1$$

Please note, that the propability mass function p_n consists of two separate functions. It turns out, that it is easier to determine the average queue size L_q first instead of the average system size L , as only one function is required. A very useful parameter in the derivation of L_q is the propability of delay p_d , which may be obtained as follows

$$\begin{aligned} p_d &= \sum_{n=c}^{\infty} p_n = \sum_{n=c}^{\infty} \frac{p_0 \rho^n}{c! c^{n-c}} \\ &= \frac{p_0 \rho^c}{c!} \sum_{n=c}^{\infty} \left(\frac{\rho}{c}\right)^{n-c} \\ &= \frac{p_0 \rho^c}{c! \left(1 - \frac{\rho}{c}\right)} \end{aligned} \quad (3.24)$$

Expression 3.24 is often referred to as *Erlang C formula* or *Erlang formula of the second kind*. The Erlang C formula has been derived for lost calls held systems (LCH) long before the $M/M/c$ model was developed. Most performance characteristics of interest may be expressed in terms of this expression. Consequently it will be used to derive the average queue size L_q

$$\begin{aligned} L_q &= \sum_{n=c}^{\infty} (n-c) p_n = \sum_{n=0}^{\infty} n p_{n+c} \\ &= \sum_{n=0}^{\infty} n \frac{p_0 \rho^{n+c}}{c! c^n} = \frac{p_0 \rho^c}{c!} \sum_{n=0}^{\infty} n \left(\frac{\rho}{c}\right)^n \\ &= \frac{p_0 \rho^c}{c!} \frac{\frac{\rho}{c}}{\left(1 - \frac{\rho}{c}\right)^2} = \frac{\frac{\rho}{c}}{1 - \frac{\rho}{c}} p_d \\ &= \frac{\lambda}{c\mu - \lambda} p_d \end{aligned} \quad (3.25)$$

Proceeding as before leads to the other measures of effectiveness

$$\begin{aligned}
 W_q &= \frac{L_q}{\lambda} = \frac{1}{c\mu - \lambda} p_d \\
 W &= W_q + \frac{1}{\mu} = \frac{1}{c\mu - \lambda} p_d + \frac{1}{\mu} \\
 L &= \lambda W = \frac{\lambda}{c\mu - \lambda} p_d + \rho
 \end{aligned} \tag{3.26}$$

Example 12 Consider a supermarket with eight cashiers open and a common queue. Customers arrive approximately every 1.25 sec and take on average 5 minutes to be served. Assuming exponential arrival and service processes following a first come, first serve discipline, the model input is given as follows

$$\begin{aligned}
 \lambda &= \frac{1}{1.25} = 0.8 \\
 \mu &= \frac{1}{5} = 0.2 \\
 c &= 8
 \end{aligned}$$

From these parameters the traffic intensity ρ and the server utilization u may be calculated

$$\begin{aligned}
 \rho &= 4 \\
 u &= 0.5
 \end{aligned}$$

thus satisfying the stability condition $u < 1$. Substitution into the formulas for the $M/M/c$ model yield

$$\begin{aligned}
 p_d &= 0.059 \\
 L &= 4.059 \\
 L_q &= 0.059 \\
 W &= 5.0738 \\
 W_q &= 0.0738
 \end{aligned}$$

Thus the propability, that no customer has to wait is given by $1 - p_d = 0.941$. Also note, that the queues are very small in this 50% loaded system, less than one customer has to wait on the average. If three cashiers become ill, a rest

of five employees has to serve the entire customers. By setting $c = 5$ and reapplying the formulas, the following performance measures are obtained

$$\begin{aligned}\rho &= 4 \\ u &= 0.8 \\ p_d &= 0.5541 \\ L &= 6.2165 \\ L_q &= 2.2165 \\ W &= 7.7706 \\ W_q &= 2.7706\end{aligned}$$

Now on the average more than two customers have to wait. The propability, that customers are served without having to wait has decreased to $1 - p_d = 0.4459$. Although this might still be acceptable for a supermarket, this example shows, how sensible queueing systems might become with increased load. If an additonal cashier becomes unavailable or the average service time increases beyond 6.25 minutes, the supermarket can not serve their customers anymore. This can be seen from the stability condition.

3.3.2 Capacity Constraints in Multiserver Systems

As already mentioned above, by customizing the parameters for the load dependent model, capacity constraints may be introduced to a multiserver system $M/M/c/K$, i.e.

$$\begin{aligned}\lambda_n &= \lambda && \text{for } 0 \leq n < K \\ \lambda_n &= 0 && \text{for } n \geq K \\ \mu_n &= n\mu && \text{for } 1 \leq n < c \\ \mu_n &= c\mu && \text{for } c \leq n \leq K \\ \mu_n &= 0 && \text{for } n > K\end{aligned}$$

Having identified the capacity limitations as the only difference between the limited and the unlimited model, the same propabilities for states $1 \dots K$ can be assumed

$$p_n = \begin{cases} \prod_{i=1}^n \frac{\lambda}{i\mu} p_0 = \frac{1}{n!} \rho^n p_0 & \text{for } 1 \leq n \leq c \\ p_c \prod_{i=1}^n \frac{\lambda}{c\mu} = \frac{1}{c!c^{n-c}} \rho^n p_0 & \text{for } c < n \leq K \\ 0 & \text{for } n > K \end{cases} \quad (3.27)$$

Proceeding as before for the $M/M/1/K$ model, the propability for state 0 remains to be determined

$$\begin{aligned}
 1 &= \sum_{n=0}^K p_n = p_0 \sum_{n=0}^c \frac{1}{n!} \rho^n + p_0 \sum_{n=c+1}^K \frac{1}{c! c^{n-c}} \rho^n \\
 p_0 &= \left(1 + \sum_{n=1}^c \frac{1}{n!} \rho^n + \sum_{n=c+1}^K \frac{1}{c! c^{n-c}} \rho^n \right)^{-1} \\
 &= \left(\sum_{n=0}^{c-1} \frac{1}{n!} \rho^n + \sum_{n=c}^K \frac{1}{c! c^{n-c}} \rho^n \right)^{-1} \\
 &= \begin{cases} \left(\sum_{n=0}^{c-1} \frac{1}{n!} \rho^n + \frac{\rho^c}{c!} \frac{1 - (\frac{\rho}{c})^{K-c+1}}{1 - \frac{\rho}{c}} \right)^{-1} & \text{for } \frac{\rho}{c} \neq 1 \\ \left(\sum_{n=0}^{c-1} \frac{1}{n!} c^n + \frac{c^c}{c!} (K - c + 1) \right)^{-1} & \text{for } \frac{\rho}{c} = 1 \end{cases}
 \end{aligned}$$

Following the same procedure as for the $M/M/c$ model, the propability of delay p_d is now derived

$$\begin{aligned}
 p_d &= \sum_{n=c}^{K-1} p_n = \sum_{n=c}^{K-1} \frac{p_0 \rho^n}{c! c^{n-c}} \\
 &= \begin{cases} p_0 \frac{\rho^c}{c!} \frac{1 - (\frac{\rho}{c})^{K-c}}{1 - \frac{\rho}{c}} & \text{for } \frac{\rho}{c} \neq 1 \\ p_0 \frac{c^c}{c!} (K - c) & \text{for } \frac{\rho}{c} = 1 \end{cases}
 \end{aligned}$$

Please note, that no delay can occur, if no waiting room exists, i.e. $K = c$. Then only blocking may occur, whereas the propability of blocking is given by p_K for all values of K . The $M/M/c/K$ model without waiting room also denoted by $M/M/c/c$ directly relates to the lost calls cleared (LCC) system. The blocking propability $p_b = p_c = p_K$ for the $M/M/c/c$ model is often referred to as *Erlang B formula*, *Erlang loss formula* or *Erlang formula of the first kind*. Substituting p_0 into equation 3.27 and simplifying yields

$$p_b = p_c = \frac{\frac{\rho^c}{c!}}{\sum_{n=0}^c \frac{\rho^n}{n!}} \quad (3.28)$$

The most appealing property of the Erlang loss formula lies in the fact, that its validity is not limited to exponential service times. It can be shown, that the Erlang loss formula still holds under very general conditions, i.e. for the

$M/G/c/c$ model. Thus any service time distribution dependency has been reduced to the mean service time only. As the proof is very extensive, it will be omitted here. A proof $M/M/1/1 = M/G/1/1$ for a single server model is presented in [90]. Turning attention back to the more general $M/M/c/K$ model, it remains to determine the performance characteristics. As before L_q provides the most convenient way to receive results

$$\begin{aligned}
L_q &= \sum_{n=c}^K (n-c)p_n = \sum_{n=0}^{K-c} np_{n+c} \\
&= \sum_{n=0}^{K-c} n \frac{p_0 \rho^{n+c}}{c! c^n} = \frac{p_0 \rho^c}{c!} \sum_{n=0}^{K-c} n \left(\frac{\rho}{c}\right)^n \\
&= \frac{p_0 \rho^c}{c!} \frac{\rho}{c} \sum_{n=1}^{K-c} n \left(\frac{\rho}{c}\right)^{n-1} \\
&= \begin{cases} \frac{p_0 \rho^c}{c!} \frac{\rho}{c} \frac{d}{d\rho} \left(\frac{1 - (\frac{\rho}{c})^{K-c+1}}{1 - \frac{\rho}{c}} \right) & \text{for } \frac{\rho}{c} \neq 1 \\ \frac{p_0 \rho^c}{c!} \frac{(K-c)(K-c+1)}{2} & \text{for } \frac{\rho}{c} = 1 \end{cases} \\
&= \begin{cases} \frac{p_0 \rho^{c+1}}{c! c (1-u)^2} (1 - u^{K-c+1} - (1-u)(K-c+1)u^{K-c}) & \text{for } \frac{\rho}{c} \neq 1 \\ \frac{p_0 \rho^c}{c!} \frac{(K-c)(K-c+1)}{2} & \text{for } \frac{\rho}{c} = 1 \end{cases} \quad \text{with } u = \frac{\rho}{c}
\end{aligned}$$

The other measures of effectiveness may be obtained by using Little's Law. Due to the system limitation, the arrival rate has to be modified to exclude lost customers. For telephony applications, one would say the *calls carried* have to be used instead of the *calls offered*. This may be expressed as follows

$$\begin{aligned}
W_q &= \frac{1}{\lambda(1-p_K)} L_q \\
W &= W_q + \frac{1}{\mu} \\
L &= \lambda(1-p_K)W
\end{aligned}$$

3.3.3 * Erlang B revisited

Instead of directly deriving the result for the Erlang B formula 3.28, the following convenient recursion formula may be applied

$$p_c = \frac{\rho p_{c-1}}{c + \rho p_{c-1}}, \quad p_0 = 1$$

Substituting $\varepsilon_c = \frac{1}{p_c}$ provides an equivalent recurrence formula

$$\varepsilon_c = 1 + \frac{c}{\rho} \varepsilon_{c-1}, \quad \varepsilon_0 = 1 \quad (3.29)$$

Due to the waiting room limitation of the $M/M/c/c$ queue, a steady state distribution exists also for the case of *heavy traffic*, that is $u = \frac{\rho}{c} > 1$. Assuming an arrival rate of $c\lambda$, the utilization u exceeds 1. In fact, one can show [172], that the number of empty places converges weakly to a geometric distribution with parameter $\frac{1}{\rho}$. As a consequence the following relation for the blocking probability p_c holds

$$\lim_{c \rightarrow \infty} p_c = 1 - \frac{1}{\rho} = 1 - \frac{\mu}{\lambda} \quad (3.30)$$

With c sufficiently large, formula 3.30 provides a reasonable approximation in heavy traffic situations.

In some cases, it becomes necessary to calculate the blocking probability also for non-integer values of c . By replacing the sum in 3.28 by an appropriate integral, one arrives at the following alternative form [37] of the Erlang loss formula:

$$p_c = \frac{\rho^c e^{-\rho}}{\Gamma(c+1, \rho)} \quad (3.31)$$

where

$$\Gamma(c+1, \rho) = \int_{\rho}^{\infty} t^c e^{-t} dt$$

is the *complement of the incomplete gamma function* [187]. One typical application of expression 3.31 arises in the context of the equivalent random method developed by Wilkinson for alternative routing networks.

3.3.4 Customer Impatience

Modeling customer frustration may be achieved in different ways. In a *balking* scenario, customers are refusing to enter the queue given that it has reached a certain length. At its most extreme, such a system is described by a $M/M/c/K$ model. Alternatively customer discouragement may be modeled by a monotonic decreasing function b_n . By carefully selecting a proper function b_n , one is able to express customer expectations in a nice and accurate way [90].

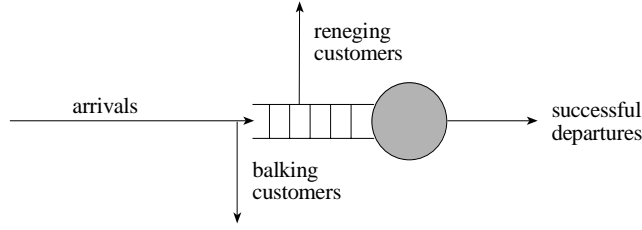


Figure 3.16: A queue with impatient customers

With respect to the birth-death model introduced in equations 3.18 and 3.19, balking affects the arrival rate, i.e.

$$\lambda_n = b_n \lambda \quad (3.32)$$

Please note, that the system arrival rate has been assumed to be constant λ .

Another form of customer impatience is *reneging*. Other than the balking customer, a reneging customer joins the queue waiting for service. If the perceived waiting time exceeds customer expectations, the customer leaves the queue. Proceeding similar as above, a reneging function is introduced

$$r(n) = \lim_{\Delta t \rightarrow 0} \Pr \left\{ \begin{array}{l} \text{customer reneges during } \Delta t \\ \text{given } n \text{ customers in the system} \end{array} \right\}$$

The reneging function clearly affects the service rate, as reneging customers may be seen as virtually serviced customers in addition to regularly service customers. Mathematically expressed

$$\bar{\mu}_n = \mu_n + r(n) \quad (3.33)$$

Both types of impatience may be combined in a single model as shown in figure 3.16. Application of the expressions 3.32 and 3.33 to the general birth-death equations 3.18 and 3.19 yields [90]

$$\begin{aligned} p_n &= \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} p_0 = \lambda^n p_0 \prod_{i=1}^n \frac{b_{i-1}}{\mu_i + r(i)} \\ p_0 &= \left(1 + \sum_{n=1}^{\infty} \lambda^n \prod_{i=1}^n \frac{b_{i-1}}{\mu_i + r(i)} \right)^{-1} \end{aligned}$$

For practical purposes, very often a more specific set of parameters is defined. Assuming c servers with constant service rate μ , i.e.

$$\mu_n = \begin{cases} n\mu & \text{for } 1 \leq n \leq c \\ c\mu & \text{for } c < n \leq K \end{cases}$$

a system capacity of $K > c$, a constant balking rate in queueing situations, i.e.

$$b_n = \begin{cases} 1 & \text{for } n \leq c \\ (1 - \beta) & \text{for } c < n \leq K \\ 0 & \text{for } n > K \end{cases} \quad (3.34)$$

and that customers don't have any knowledge about the system state [214], i.e.

$$r(n) = \begin{cases} 0 & \text{for } n \leq c \\ (n - c)\delta & \text{for } c < n \leq K \end{cases}$$

the model becomes

$$\begin{aligned} p_n &= \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} p_0 = \begin{cases} \frac{1}{n!} \rho^n p_0 & \text{for } 1 \leq n \leq c \\ \frac{\rho^c \lambda^{n-c} (1-\beta)^{n-c}}{c! \prod_{i=c+1}^n c\mu + (i-c)\delta} p_0 & \text{for } c < n \leq K \\ 0 & \text{for } n > K \end{cases} \\ p_0 &= \left(1 + \sum_{n=1}^c \frac{1}{n!} \rho^n + \sum_{n=c+1}^K \frac{\rho^c \lambda^{n-c} (1-\beta)^{n-c}}{c! \prod_{i=c+1}^n c\mu + (i-c)\delta} \right)^{-1} \\ &= \left(\sum_{n=0}^c \frac{1}{n!} \rho^n + \sum_{n=c+1}^K \frac{\rho^c \lambda^{n-c} (1-\beta)^{n-c}}{c! \prod_{i=c+1}^n c\mu + (i-c)\delta} \right)^{-1} \end{aligned} \quad (3.35)$$

with ρ set to $\rho = \frac{\lambda}{\mu}$ as before. This limited capacity system covers balking as well as reneging behaviour. It is best solved by using numerical computation, as no closed form solution is known to the author. The performance characteristics W , L and X are calculated by substituting state probabilities 3.35 in formulas 3.20 to 3.22 for the birth-death model.

By omitting the balking behaviour one arrives at the $M/M/c + M$ model first introduced by C. Palm before 1960 [162]. It has also been given names such as *Erlang A* or *Palm/Erlang A*, because it provides a tradeoff between the Erlang C ($M/M/c$) queueing model and the Erlang B loss ($M/M/c/c$) system. Our treatment will be based on [144]. First note, that by eliminating

the balking definition 3.34, the system becomes infinite. As a consequence the second sum in

$$p_0 = \left(\sum_{n=0}^c \frac{1}{n!} \rho^n + \sum_{n=c+1}^{\infty} \frac{\rho^c \lambda^{n-c}}{c! \prod_{i=c+1}^n c\mu + (i-c)\delta} \right)^{-1} \quad (3.36)$$

must converge to allow for meaningful results of p_n . In fact, convergence can be assured by deriving an upper bound:

$$\begin{aligned} p_0^{-1} &= \sum_{n=0}^c \frac{1}{n!} \rho^n + \sum_{n=c+1}^{\infty} \frac{\rho^c \lambda^{n-c}}{c! \prod_{i=c+1}^n c\mu + (i-c)\delta} \\ &= \sum_{n=0}^c \frac{1}{n!} \frac{\lambda^n}{\mu^n} + \frac{\lambda^c}{\mu^c c!} \sum_{n=c+1}^{\infty} \prod_{i=c+1}^n \frac{\lambda}{(c\mu + (i-c)\delta)} \\ &\leq \sum_{n=0}^{\infty} \frac{(\lambda / \min(\mu, \delta))^n}{n!} = e^{-\frac{\lambda}{\min(\mu, \delta)}} \end{aligned}$$

Hence the $M/M/c + M$ queue always remains stable. Once p_0 is known the entire steady state distribution may be derived from

$$p_n = \begin{cases} \frac{1}{n!} \rho^n p_0 & \text{for } 1 \leq n \leq c \\ \frac{\rho^c \lambda^{n-c}}{c! \prod_{i=c+1}^n c\mu + (i-c)\delta} p_0 & \text{for } n > c \end{cases} \quad (3.37)$$

To avoid numerical difficulties caused by the infinite sum in equation 3.36, Palm presented an ingenious derivation based on the Erlang loss formula and the incomplete gamma function. Rewriting expression 3.36 leads to

$$\begin{aligned} p_0^{-1} &= \sum_{n=0}^c \frac{\rho^n}{n!} + \frac{\rho^c}{c!} \sum_{n=c+1}^{\infty} \prod_{i=c+1}^n \frac{\lambda}{(c\mu + (i-c)\delta)} \\ &= \frac{\rho^c}{c!} \left[\frac{1}{p_b} + \sum_{n=1}^{\infty} \frac{(\lambda/\delta)^n}{\prod_{i=1}^n (c\mu/\delta + i)} \right] \\ &= \frac{\rho^c}{c!} \left[\frac{1}{p_b} + F\left(\frac{c\mu}{\delta}, \frac{\lambda}{\delta}\right) - 1 \right] \end{aligned} \quad (3.38)$$

The auxiliary function F is defined as

$$F(x, y) = \frac{x e^y}{y^x} \gamma(x, y)$$

where $\gamma(x, y) = \int_0^y t^{x-1} e^{-t} dt$ denotes the *incomplete gamma function* [187]. Inverting expression 3.38 results in

$$p_0 = \frac{c!}{\rho^c} \frac{p_b}{1 + [F(c\mu/\delta, \lambda/\delta) - 1] p_b}$$

Inserting p_0 in the system of equation 3.37 and realizing, that

$$p_c = \frac{\rho^c}{c!} p_0$$

finally leads to

$$p_n = \begin{cases} \frac{c!}{n! \rho^{c-n}} p_c & \text{for } 0 \leq n < c \\ \frac{p_b}{1 + [F(c\mu/\delta, \lambda/\delta) - 1] p_b} & \text{for } n = c \\ \frac{(\lambda/\delta)^{n-c}}{\prod_{i=1}^{n-c} c\mu/\delta + i} p_c & \text{for } n > c \end{cases} \quad (3.39)$$

The above set of formulas provides an easy way to calculate the steady state distribution by performing the following steps

1. Calculate the blocking probability for a c server loss system p_b by applying the Erlang B formula 3.28
2. Look up the value of the incomplete gamma function $\gamma(c\mu/\delta, \lambda/\delta)$
3. Insert both results in the expression for p_c
4. Use the remaining formulas to derive p_n , $n \neq c$ from p_c

Having derived a closed form solution for the equilibrium distribution, we are now able to derive various performance characteristics. Let \check{W}_q denote a random variable associated with the current queueing time. Then the probability of delay is given by

$$\begin{aligned} p_d &= \Pr \left\{ \check{W}_q > 0 \right\} = \sum_{n=c}^{\infty} p_n \\ &= \frac{p_b}{1 + [F(c\mu/\delta, \lambda/\delta) - 1] p_b} \left[1 + \sum_{n=c+1}^{\infty} \frac{(\lambda/\delta)^{n-c}}{\prod_{i=1}^{n-c} c\mu/\delta + i} \right] \quad (3.40) \\ &= \frac{F(c\mu/\delta, \lambda/\delta) p_b}{1 + [F(c\mu/\delta, \lambda/\delta) - 1] p_b} \end{aligned}$$

In the $M/M/c + M$ queue a customer decides to leave the queue at an exponential rate. In determining the probability of getting ultimately served, one encounters, what has been called competition of exponentials in [144]:

$$\begin{aligned} p_0^s &= \frac{c\mu}{c\mu + \delta} \\ p_1^s &= \frac{c\mu + \delta}{c\mu + 2\delta} p_0^s = \frac{c\mu}{c\mu + 2\delta} \end{aligned}$$

Proceeding further one arrives at p_n^s the probability of the n -th customer getting served, that is

$$p_n^s = \frac{c\mu}{c\mu + (n+1)\delta}, \quad n \geq 1$$

The probability to abandon service and losing the customer is given by

$$p_n^a = 1 - p_n^s = \frac{(n+1)\delta}{c\mu + (n+1)\delta}, \quad n \geq 0$$

One can now derive the conditional probability that a customer abandons given he does not receive immediate service

$$\begin{aligned} \Pr \left\{ \text{Abandon} | \check{W}_q > 0 \right\} &= \sum_{n=c}^{\infty} \frac{p_n p_{c-n}^a}{p_d} \\ &= \frac{1}{\rho F(c\mu/\delta, \lambda/\delta)} + 1 - \frac{1}{\rho} \end{aligned} \quad (3.41)$$

The calculations have been omitted, because they are rather lengthy. They utilize an identity derived by Palm for the function F based on properties of the incomplete gamma function [187]. A partial derivation is given in [144]. If required, one may consult the original paper [162] by Palm. Due to independence, the probability of an arbitrary customer abandoning the queue is given by the product of the expressions 3.40 and 3.41:

$$\begin{aligned} p_a &= \Pr \{ \text{Abandon} \} = \Pr \left\{ \text{Abandon} | \check{W}_q > 0 \right\} \Pr \left\{ \check{W}_q > 0 \right\} \\ &= \left(\frac{1}{\rho F(c\mu/\delta, \lambda/\delta)} + 1 - \frac{1}{\rho} \right) p_d \end{aligned} \quad (3.42)$$

Based on the loss probability 3.42, the average queueing time W_q may be easily derived by an application of Little's law. First note that in equilibrium,

the rate of customers abandoning the queue and the rate of customers entering the system have to be the same, i.e. $\delta L_q = \lambda p_a$. So the average queue size is given by

$$L_q = \frac{\lambda}{\delta} p_a$$

Applying Little's law $L_q = \lambda W_q$ leads to

$$W_q = \frac{p_a}{\delta}$$

The remaining measures of effectiveness are obtained in much the same way as has been done for the $M/M/c/K$ model, that is

$$\begin{aligned} W &= W_q + \frac{1}{\mu} = \frac{p_a}{\delta} + \frac{1}{\mu} \\ L &= \lambda W = \frac{\lambda p_a}{\delta} + \rho \end{aligned}$$

In comparison to the $M/M/c$ queueing system, performance is superior in terms of average waiting time and mean queue length. Additionally, the $M/M/c + M$ system is immune to any kind of congestion. This is also, what we encounter especially in real life situations concerned with human behaviour. Impatience becomes a mandatory assumption for the analysis of such models. This might be different for technical systems.

Another, although not well known form of customer impatience exists in multiqueue systems and is called *jockeying*. Customer dissatisfaction is expressed by simply joining another queue. Rather simple in description, these models are hard to solve and will not be covered in this text. For general information on customer impatience refer to [90]. A more specific model with limited sources, limited capacity and reneging is described in [24]. Equivalence relations between systems with customer impatience and machine inference problems are derived in [91]. Balking and reneging for birth-death processes has also been considered in [179].

3.3.5 Bounded Holding Times

In certain situations it becomes necessary to bound the time a customer resides in the system. As an example consider a call centre, where customers are rerouted to an IVR system, when a predefined waiting time limit has been reached. By expressing the waiting time limit in terms of an exponential

distribution, the system fits nicely in the framework of birth-death processes. It has been introduced by B.V. Gnedenko and I.N. Kovalenko in their book [86]. Assume arrival rate, service rate, system capacity and the number of servers to be λ , μ , K and c similar to the $M/M/c/K$ model. Further let the waiting time boundary be exponentially distributed with rate δ . Then the corresponding average is given by $\frac{1}{\delta}$. Any customer having reached the waiting time limit will depart from the system. Combination of rates leads to

$$\mu_n = \begin{cases} n\mu + n\delta & \text{for } 1 \leq n < c \\ c\mu + n\delta & \text{for } c \leq n \leq K \\ 0 & \text{for } n > K \end{cases}$$

Substitution of $\lambda_n = \lambda$ and μ_n into the birth-death equation 3.18 leads to expression 3.37 for the steady state distribution. It follows, that the model with exponentially bounded holding times is equivalent to the Erlang A $M/M/c + M$ queueing system.

3.3.6 Finite Population Models

The previous discussion focused on queueing problems with infinite customer population. Although mathematically convenient, such an assumption only serves well as an approximation to situations with a large population. One anticipates, that prediction errors become negligible. If this is not the case, then one has to take care about finiteness. This is best done by modifying the birth rate λ in the standard birth-death model as follows

$$\lambda_n = \begin{cases} (M - n) \lambda & \text{for } 0 \leq n < M \\ 0 & \text{for } n \geq M \end{cases}$$

Here M denotes the size of the population. Assuming a system with $c < M$ service units, i.e.

$$\mu_n = \begin{cases} n\mu & \text{for } 1 \leq n < c \\ c\mu & \text{for } n \geq c \end{cases}$$

and substituting in equation 3.18 leads to

$$p_n = \begin{cases} \binom{M}{n} \rho^n p_0 & \text{for } 0 \leq n < c \\ \binom{M}{n} \frac{n!}{c^{n-c} c!} \rho^n p_0 & \text{for } c \leq n \leq M \end{cases} \quad (3.43)$$

with $\binom{M}{n} = \frac{M!}{(M-n)!n!}$ denoting the *binomial coefficient*. Applying $\sum_{n=0}^M p_n = 1$ and solving for p_0 gives

$$p_0 = \left[\sum_{n=0}^{c-1} \binom{M}{n} \rho^n + \sum_{n=c}^M \binom{M}{n} \frac{n!}{c^{n-c}c!} \rho^n \right]^{-1}$$

For efficient calculation of the steady state probabilities, D. Gross and C.M. Harris [90] suggest the following recursion based on the properties of the binomial coefficient:

$$f_n = \frac{p_{n+1}}{p_n} = \begin{cases} \frac{M-n}{n+1} & \text{for } 0 \leq n < c \\ \frac{M-n}{c} & \text{for } c \leq n \leq M \end{cases}, \quad p_n = \prod_{i=0}^{n-1} f_i p_0$$

Using the definition of the expected value, one is now able to derive the average system size

$$L = \sum_{n=0}^M n p_n = \left[\sum_{n=0}^{c-1} n \binom{M}{n} \rho^n + \sum_{n=c}^M n \binom{M}{n} \frac{n!}{c^{n-c}c!} \rho^n \right] p_0$$

Following [90], the average queue size L_q may be derived from L as follows

$$\begin{aligned} L_q &= \sum_{n=c}^M (n-c) p_n = \sum_{n=c}^M n p_n - c \sum_{n=c}^M p_n \\ &= L - \sum_{n=c}^{c-1} n p_n - c \left(1 - \sum_{n=0}^{c-1} n p_n \right) \\ &= L - c + \sum_{n=0}^{c-1} (c-n) p_n \\ &= L - c + p_0 \sum_{n=0}^{c-1} (c-n) \binom{M}{n} \rho^n \end{aligned}$$

In order to derive the waiting time indicators using Little's law, one first has to determine the mean arrival rate $\bar{\lambda}$. With n customers already in the system, a maximum of $M-n$ customers remain outside waiting for arrival. This results in a mean arrival rate of $(M-n)\lambda$. Averaging yields

$$\bar{\lambda} = \sum_{n=0}^M (M-n) \lambda p_n = \lambda \left(M \sum_{n=0}^M p_n - \sum_{n=0}^M n p_n \right) = \lambda (M - L)$$

Using Little's law with the just derived mean arrival rate $\bar{\lambda}$ leads to

$$W = \frac{L}{\lambda(M-L)}, \quad W_q = \frac{L_q}{\lambda(M-L)}$$

Assuming the size of the waiting room to be 0 results in a finite-source variation of the classic $M/M/c/c$ Erlang Loss system. This model is often used in telecommunication applications and is called the *Engset model*. The steady state distribution 3.43 simplifies to

$$p_n = \frac{\binom{M}{n} \rho^n}{\sum_{i=0}^M \binom{M}{i} \rho^i}, \quad 0 \leq n \leq M$$

This is also known as the *Engset distribution*. The probability of a customer being blocked and getting lost due to call congestion is determined by a full system, that is $p_b = p_c$. Similar to the general finite population model, a recurrence relation may be deducted for easy calculation. In telephony applications, the recursion is usually defined for the blocking probability:

$$p_c = \frac{(M-c) \rho p_{c-1}}{c + (M-c) \rho p_{c-1}}$$

With M getting very large, the Engset distribution approaches the probabilities p_n given by the $M/M/c/c$ Erlang loss system. As a reference related to queueing theory consider any standard text book such as [90]. For telephony applications we refer to [37].

3.3.7 Relation to Markov Chains

Now an attempt will be made to relate birth-death processes to continuous time Markov chains. For a short introduction please refer to appendix A.3. A birth-death process may be understood as a *skip-free* Markov chain, meaning that the process can only move to a neighbouring state in a single step. Combining birth and death rates

$$\begin{aligned} q_{n,n+1} &= \lambda_n \\ q_{n,n-1} &= \mu_n \\ q_{nn} &= -(\lambda_n + \mu_n) \\ q_{mn} &= 0 \text{ for } |m-n| > 1 \end{aligned}$$

leads to the infinitesimal generator

$$\mathbf{Q} = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & \cdots \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & 0 & \cdots \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & \cdots \\ \vdots & & & \ddots & \end{pmatrix} \quad (3.44)$$

Alternatively one may address the discrete Markov chain embedded into the birth-death process. By choosing the occurrences of the state transitions as regeneration points, the corresponding transition matrix becomes

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots \\ \frac{\mu_1}{\lambda_1 + \mu_1} & 0 & \frac{\lambda_1}{\lambda_1 + \mu_1} & 0 & \cdots \\ 0 & \frac{\mu_2}{\lambda_2 + \mu_2} & 0 & \frac{\lambda_2}{\lambda_2 + \mu_2} & \cdots \\ \vdots & & & \ddots & \end{pmatrix}$$

Apparently both matrices suggest irreducibility. From the conservation equation 3.19 it follows, that an equilibrium can only be assumed, if

$$\sum_{n=1}^{\infty} \prod_{i=1}^n \frac{\lambda_{i-1}}{\mu_i} < \infty \quad (3.45)$$

Due to the fact, that stationarity implies positive recurrence, equation 3.45 may be used as a criterion for positive recurrence. Focusing on the embedded chain, it can be shown [217], that a birth-death process is recurrent, if

$$\frac{1}{\lambda_0} + \sum_{n=1}^{\infty} \frac{\mu_1 \mu_2 \cdots \mu_n}{\lambda_0 \lambda_1 \cdots \lambda_n} = \infty \quad (3.46)$$

holds and vice versa. Multiplying the left hand side by λ_0 and omitting the first term simplifies formula 3.46 to

$$\sum_{n=1}^{\infty} \prod_{i=1}^n \frac{\mu_i}{\lambda_i} = \infty$$

The stationary distribution may also be calculated using Markov chain methods. One can either chose to solve the Chapman Kolmogorov equations for the embedded chain or apply Kolmogorov's differential systems directly. Either case leads to the same results. For further information please consult [30] and [217]. Especially the latter reference provides a rigorous treatment on the topic.

3.4 Extended Markovian Models

Because of the memoryless property of the exponential distribution, Markovian models such as the birth-death process become analytically tractable. In most cases it is possible to derive a closed form solution as well. In the current section we enhance the models treated so far by features, which can not be classified as typically Markovian.

3.4.1 Some Useful Relations

Before getting hands on some rarities in queueing we will derive some useful tools. The first deals with an interesting property of Poisson arrivals. A Poisson stream is sometimes called purely random. Provided the state of the system changes at most by one, a customer arriving in the stream finds the same state distribution as an outside observer. One can say, that *Poisson arrivals see time averages* (PASTA). It turns out, that PASTA also applies to the transient case, which obviously includes the steady state version as special case.

Theorem 13 (PASTA) *Define $a_n(t)$ as the probability of n customers in the system seen by an arrival just after entering the system. Let $p_n(t)$ denote the distribution of n customers in the system at an arbitrary point in time. Then for Poisson arrivals*

$$a_n(t) = p_n(t) \quad \text{for all } n \geq 0, t \geq 0$$

Proof. Define $N(t)$ as the number of customers in the system at time t . Now consider the number of arrivals $A(t, t+h)$ in an infinitesimal interval $(t, t+h)$. Then $a_n(t)$ is defined as the limit $h \rightarrow 0$ of the probability, that the number of customers in the system is n given an arrival has occurred just

after t . In mathematical terms

$$\begin{aligned}
 a_n(t) &= \lim_{h \rightarrow 0} \Pr \{N(t) = n | A(t, t+h) = 1\} \\
 &= \lim_{h \rightarrow 0} \frac{\Pr \{N(t) = n, A(t, t+h) = 1\}}{\Pr \{A(t, t+h) = 1\}} \\
 &= \lim_{h \rightarrow 0} \frac{\Pr \{A(t, t+h) = 1 | N(t) = n\} \Pr \{N(t) = n\}}{\Pr \{A(t, t+h) = 1\}} \\
 &= \lim_{h \rightarrow 0} \frac{\Pr \{A(t, t+h) = 1\} \Pr \{N(t) = n\}}{\Pr \{A(t, t+h) = 1\}} \\
 &= \lim_{h \rightarrow 0} \Pr \{N(t) = n\} = p_n(t)
 \end{aligned}$$

Please note, that $\Pr \{A(t, t+h) = 1 | N(t) = n\} = \Pr \{A(t, t+h) = 1\}$ follows from the fact, that the number of arrivals occurring in two disjoint time intervals are independent. ■

Another proof based on the assumption, that future increments are independent of the past has been given by R.W. Wolff in [215]. A proof tailored to the requirements of the $M/G/1$ queue may be found in [90].

A similar result also holds for exponential service times. Assuming equilibrium, let p_n be the probability that n customers are in the system. The probability that n customers are in the system just prior to an arrival is denoted by \tilde{p}_n .

Theorem 14 (Rate Conservation Law) *Consider a queueing system with general arrivals, exponential service times, $c \leq \infty$ servers and system limit $K \leq \infty$. Furthermore assume a work conserving queueing discipline and no interruption of service. Then the following relation holds*

$$\min(c, n) p_n = \rho \tilde{p}_{n-1}$$

Rewriting the above equation to $\min(c, n) \mu p_n = \lambda \tilde{p}_{n-1}$ one may intuitively explain the result as follows. The left term represents a state transition from state n to state $n-1$, whereas the right term is just the opposite. Given a work conserving queueing discipline in accordance with the local balance principle, the rate downwards must equal the rate upwards [24]. For the proof we refer to theorem 6.4.3 of [49] or to page 154 of [196].

We will turn attention now to a theorem from complex analysis often employed in queueing theory. Its main use lies in assuring the existence of

roots within a closed contour such as the unit circle $|z| = 1$. Usually a given function F is split into two parts, i.e. $F(z) = f(z) + g(z)$, where $f(z)$ has a known number of zeros inside a given domain.

Theorem 15 (Rouche) *If $f(z)$ and $g(z)$ are functions analytic inside and on a closed contour C and if $|g(z)| < |f(z)|$ on C , then both $f(z)$ and $f(z) + g(z)$ possess the same number of zeros inside C .*

A proof may be found in almost any standard textbook on complex analysis, for example see [189]. In some cases, one wants to extend the assumptions of Rouche's theorem to the boundary of C without invalidating the conclusion. This is indeed possible, as shown in a recent paper by V. Klimenok [125].

Theorem 16 (Extended Rouche) *Let $f(z)$ and $g(z)$ be analytic inside the open unit disc $|z| < 1$, continuous on the boundary $|z| = 1$ and differentiable at the point $z = 1$. Assume, that the following relations are satisfied*

$$\begin{aligned} |g(z)|_{|z|=1, z \neq 1} &< |f(z)|_{|z|=1, z \neq 1} \\ f(1) &= -g(1) \neq 0 \\ \frac{\frac{d}{dz}f(z)|_{z=1} + \frac{d}{dz}g(z)|_{z=1}}{f(1)} &> 0 \end{aligned}$$

Then the number of zeros N_f and N_{f+g} of the functions $f(z)$ and $f(z) + g(z)$ in the unit disc $|z| < 1$ are related as follows:

$$N_{f+g} = N_f - 1$$

For the proof we refer to the paper of Klimenok [125]. As pointed out there, theorem 16 is in particular very useful for matrix geometric models of the $M/G/1$ type.

3.4.2 General Impatience Distribution

One possible generalization to Palm's $M/M/c + M$ model is to allow for a general impatience distribution. In doing so, one arrives at the $M/M/c + G$ model. A variant thereof has first been introduced by F. Bacelli and G. Hebuterne in 1981. Our treatment will be based on their paper [34] and the paper by S. Zeltyn and A. Mandelbaum [216]. Although the model

assumes, that arriving customers are fully aware of the offered wait V and abandon service immediately, if their patience time is exceeded, the model coincides with the $M/M/c + G$ model in terms of all relevant stationary performance characteristics. The patience time is assumed to be distributed according to $G(\cdot)$, in the latter more often referenced to by the *survival function* $\bar{G}(\cdot) = 1 - G(\cdot)$. Although the exponentiality assumption is violated, the model may be described by a Markov process $\{N(t), \eta(t) : t \geq 0\}$, where $N(t)$ describes the number of customers in the system at time t and $\eta(t)$ denotes the virtual offered waiting time of a customer arriving at time t . As long as there are $c - 1$ customers in the system, $\eta(t) = 0$, whereas for c or more customers $\eta(t)$ becomes positive. In the latter case, it is only relevant to know, that there are c or more customers in the system, the exact number is irrelevant. Therefore we choose $N(t) = c$ for $\eta(t) > 0$ and $\eta(t) = 0$ for $0 \leq N(t) \leq c - 1$. With the system described that way, one preserves the Markov property. Let $v(x)$ denote the density of the virtual offered waiting time and define

$$\begin{aligned} v(x) &= \lim_{t \rightarrow \infty} \lim_{h \rightarrow 0} \frac{\Pr\{N(t)=c, x < \eta(t) \leq x+h\}}{h} & x \geq 0 \\ p_n &= \lim_{t \rightarrow \infty} \Pr\{N(t) = n, \eta(t) = 0\} & 0 \leq n \leq c - 1 \end{aligned}$$

Assign to λ , μ and $\rho = \frac{\lambda}{\mu}$ the usual meanings. Obviously the system behaves like a classical $M/M/c$ queue for $0 \leq n \leq c - 1$, i.e.

$$p_n = \frac{\rho^n}{n!} p_0 \quad \text{for } 1 \leq n \leq c - 1 \quad (3.47)$$

By realizing that state $c - 1$ may only be entered from above, if an arriving customer would not have to wait for service, one arrives at

$$\lambda p_{c-1} = v(0) \quad (3.48)$$

The case of positive virtual wait may be obtained from

$$\begin{aligned} \Pr\{\eta(t+h)\} &= \Pr\{\eta(t) > x+h\} + \Pr\{\eta(t+h) > x, \eta(t) = 0\} \\ &\quad + \Pr\{\eta(t+h) > x, 0 < \eta(t) \leq x+h\} \end{aligned}$$

The second term on the right side describes the increase of the virtual wait caused by an arrival occupying the last free server. This increase will exceed x with probability $e^{-c\mu x}$, as the intervals between service departures are distributed according to an exponential distribution with rate $c\mu$. The third

term corresponds to an arrival at a full system with positive virtual wait. The probability of receiving service is determined by the survival function $\bar{G}(x)$. Thus in steady state one can expect

$$\int_x^\infty v(y)dy = \int_{x+h}^\infty v(y)dy + \lambda h p_{n-1} e^{-c\mu x} + \lambda h \int_0^x e^{-c\mu(x-y)} v(y) \bar{G}(y) dy + o(h)$$

Differentiating with respect to h and letting $h \rightarrow 0$ yields

$$v(x) = \lambda p_{n-1} e^{-c\mu x} + \lambda e^{-c\mu x} \int_0^x e^{c\mu y} v(y) \bar{G}(y) dy \quad (3.49)$$

By observing, that $H(x) = e^{c\mu x} v(x)$ is a solution to the integral equation $H(x) = \lambda p_{n-1} + \lambda \int_0^x H(y) \bar{G}(y) dy$ and solving that integral equation directly one obtains $H(x) = \lambda p_{n-1} \exp \left\{ \lambda \int_0^x \bar{G}(y) dy \right\}$ and

$$v(x) = \lambda p_{n-1} \exp \left\{ \lambda \int_0^x \bar{G}(y) dy - c\mu x \right\} \quad (3.50)$$

Normalizing probabilities

$$\begin{aligned} 1 &= \sum_{n=0}^{c-1} p_n + \Pr \{V > 0\} = \sum_{n=0}^{c-1} p_n + \int_0^\infty v(x) dx \\ &= p_0 \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \lambda \frac{\rho^{c-1}}{(c-1)!} \int_0^\infty \exp \left\{ \lambda \int_0^x \bar{G}(y) dy - c\mu x \right\} dx \end{aligned} \quad (3.51)$$

results in

$$p_0 = \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^{c-1} \lambda J}{(c-1)!} \right]^{-1}, \quad J := \int_0^\infty \exp \left\{ \lambda \int_0^x \bar{G}(y) dy - c\mu x \right\} dx \quad (3.52)$$

Please note, that we are now able to describe the steady state behaviour of the $M/M/c + G$ model in terms of equations 3.47, 3.48, 3.49 and 3.52. Furthermore, there is a straightforward generalization to include balking behaviour into the model. As before for the birth-death equations define a state dependent arrival rate

$$\lambda_n = \begin{cases} b_n \lambda & 0 \leq n \leq c-1 \\ b_{c-1} \lambda & \eta(t) > 0 \end{cases}$$

One may immediately write down the steady state equations

$$\begin{aligned}
p_n &= p_0 \prod_{i=1}^n \frac{\lambda_{i-1}}{i\mu} = \lambda^n p_0 \prod_{i=1}^n \frac{b_{i-1}}{i\mu} \quad \text{for } 1 \leq n \leq c-1 \\
\lambda_{c-1} p_{c-1} &= \lambda b_{c-1} = v(0) \\
v(x) &= \lambda_{c-1} \left(p_{n-1} e^{-c\mu x} + e^{-c\mu x} \int_0^x e^{c\mu y} v(y) \bar{G}(y) dy \right) \\
&= \lambda b_{c-1} \left(p_{n-1} e^{-c\mu x} + e^{-c\mu x} \int_0^x e^{c\mu y} v(y) \bar{G}(y) dy \right) \\
p_0 &= \left[\sum_{n=0}^{c-1} \prod_{i=1}^n \frac{\lambda_{i-1}}{i\mu} + \lambda_{c-1} J \prod_{i=1}^{c-1} \frac{\lambda_{i-1}}{i\mu} \right]^{-1} \\
&= \left[\sum_{n=0}^{c-1} \lambda^n \prod_{i=1}^n \frac{b_{i-1}}{i\mu} + \lambda^c b_{c-1} J \prod_{i=1}^{c-1} \frac{b_{i-1}}{i\mu} \right]^{-1}
\end{aligned} \tag{3.53}$$

For both models, the system may be assumed to be stable, if the integral in the expression for J in equation 3.52 converges. This in turn is equivalent to the condition $\lambda \bar{G}(\infty) < c\mu$ or $u \bar{G}(\infty) < 1$ with u the utilization. For a proper probability distribution $G(\cdot)$, i.e. $\lim_{x \rightarrow \infty} \bar{G}(x) = 0$, the system will not become unstable and show behaviour similar to the $M/M/c/K$ queueing system. Otherwise $G(\cdot)$ is called *defective* and the above mentioned condition has to be considered.

Considering the model without balking, let p_a denote the probability, that an arriving customer refrains from being serviced because of excessive wait. Then one may express the server occupancy as $\bar{u} = u(1 - p_a) = \frac{\rho}{c}(1 - p_a)$. Alternatively it may be calculated from $\bar{u} = \frac{1}{c} \sum_{n=0}^{c-1} n p_n + \Pr\{V > 0\} = \frac{1}{c} \sum_{n=0}^{c-1} n p_n + (1 - \sum_{n=0}^{c-1} p_n)$. The last substitution is evident from the normalization condition 3.51. Combining expressions leads to

$$\begin{aligned}
p_a &= 1 - \frac{c}{\rho} \bar{u} = 1 - \frac{c}{\rho} \left(\frac{1}{c} \sum_{n=0}^{c-1} n p_n + 1 - \sum_{n=0}^{c-1} p_n \right) \\
&= 1 - \frac{c}{\rho} \left(\sum_{n=1}^{c-1} \frac{n p_n}{c} + 1 - \sum_{n=0}^{c-1} p_n \right) = 1 - \sum_{n=1}^{c-1} \frac{\rho^{n-1} p_0}{(n-1)!} + \frac{c}{\rho} \left(1 - \sum_{n=0}^{c-1} p_n \right) \\
&= 1 - \sum_{n=0}^{c-2} p_n + \frac{c}{\rho} \left(1 - \sum_{n=0}^{c-1} p_n \right) = \left(1 - \frac{c}{\rho} \right) \left(1 - \sum_{n=0}^{c-1} p_n \right) + p_{c-1}
\end{aligned}$$

For the average waiting time one starts with equation 3.50 to derive

$$\Pr \{V > t\} = \lambda p_{n-1} \int_t^\infty \exp \left\{ \lambda \int_0^x \bar{G}(y) dy - c\mu x \right\} dx$$

which is just the survival function of V . With V defined only for non-negative values, the average queueing time may be calculated as follows

$$\begin{aligned} W_q &= \int_0^\infty \bar{G}(t) \Pr \{V > t\} dt \\ &= \lambda p_{c-1} \int_0^\infty \bar{G}(t) \exp \left\{ \lambda \int_0^x \bar{G}(y) dy - c\mu x \right\} dx dt \\ &= \lambda p_{c-1} \int_0^\infty \int_0^t \bar{G}(y) dy \exp \left\{ \lambda \int_0^t \bar{G}(y) dy - c\mu t \right\} dt \quad (3.54) \end{aligned}$$

whereas the last expression has been derived integrating by parts. Note, that by combining expression 3.47 and 3.52 and using Erlang's loss formula 3.28 for $c - 1$ servers, one derives p_{c-1} as follows

$$p_{c-1} = \frac{\frac{\rho^{c-1}}{(c-1)!}}{\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \rho^{c-1} \frac{\lambda J}{(c-1)!}} = \left[\frac{1}{p_b} + \lambda J \right]^{-1}$$

The remaining performance characteristics may be determined by an application of Little's Law

$$L_q = \lambda W_q, \quad W = W_q + \frac{1}{\mu}, \quad L = \lambda W$$

With balking included into the model, the average queueing time 3.54 adapts well to the modification, one has only to replace λ by λb_{c-1} and calculate p_{c-1} according to the first equation of 3.53. Giving a closer look to the effective arrival rate and the expression for p_a , the results become quite cumbersome. It turns out, that an explicit inclusion of states above c provides a feasible solution. This will be one of the main ideas of the model by A. Brandt and M. Brandt introduced next.

The $M/M/c+G$ model is a rather general one, as it includes the $M/M/c$ queueing system and Palm's $M/M/c+M$ model as special cases. For the latter let the patience times follow an exponential distribution with parameter δ , whereas for the former assume infinite patience, i.e. $\bar{G}(x) = 1$. Note, that

$\bar{G}(x) = 1$ is indeed a defective distribution putting the stability condition $1 > u\bar{G}(\infty) = u$ in effect. Obviously we arrived at the stability condition for the $M/M/c$ model. Another important special case is the $M/M/c + D$ queueing system as introduced by Gnedenko and Kovalenko in their book [86], which is based on D.Y. Barrer's derivations [179]. In practice, it applies well to computer networks with deterministic timeouts.

As mentioned above, Brandt and Brandt derived a generalization to the $M/M/c + G$ queueing system by allowing for state dependent arrival and service rates [50]. We will only provide the results here, as the derivations are rather lengthy but in concept similar to the classic model. The main difference lies in the fact, that steady state probabilities are now defined for c or more customers and that the residual patience time has been taken into account. As before, as long as there are servers available, the system follows the well-known birth-death approach. It assumes a bounded sequence of arrival rates λ_n , i.e. there is only a finite number of $\lambda_n > 0$. This leads to the steady-state distribution

$$p_n = \begin{cases} p_0 \left(\prod_{i=0}^{n-1} \lambda_i \right) \left(\prod_{i=n+1}^c \mu_i \right) & 0 < n \leq c \\ p_0 \left(\prod_{i=0}^{n-1} \lambda_i \right) \frac{\mu_c}{(n-c)!} \int_0^\infty \left(\int_0^y \bar{G}(z) dz \right)^{n-c} e^{-\mu_c y} dy & n > c + 1 \\ \left[\sum_{j=0}^{c-1} p_j + \sum_{j=0}^\infty \left(\prod_{i=0}^{c+j-1} \lambda_i \right) \frac{\mu_c}{j!} \int_0^\infty \left(\int_0^y \bar{G}(z) dz \right)^j e^{-\mu_c y} dy \right]^{-1} & n = 0 \end{cases}$$

The system can be considered stable, if one is able to calculate a non-trivial p_0 . Isolating the relevant part yields the stability condition

$$\sum_{j=0}^\infty \left(\prod_{i=0}^{c+j-1} \lambda_i \right) \frac{1}{j!} \int_0^\infty \left(\int_0^y \bar{G}(z) dz \right)^j e^{-\mu_c y} dy < \infty$$

Based on the effective arrival rate $\bar{\lambda} = \sum_{n=0}^\infty \lambda_n p_n$, the probability, that a customer has to wait, i.e.

$$p_d = 1 - \frac{1}{\bar{\lambda}} \sum_{n=0}^c \lambda_n p_n$$

and the probability, that an arriving customer will leave the system later due to impatience

$$p_a = 1 - \frac{1}{\bar{\lambda}} \left(\mu_c + \sum_{n=0}^{c-1} (\mu_n - \mu_c) p_n \right)$$

The mean queueing time may be derived by using Little's law

$$W_q = \frac{1}{\bar{\lambda}} \sum_{n=c+1}^{\infty} (n - c) p_n$$

One may split the queueing time into two parts, representing the wait an arriving customer is exposed to in case of being served or lost due to impatience,

$$\begin{aligned} W_q^s &= \frac{p_s}{(1 - p_a)\bar{\lambda}} \sum_{j=1}^{\infty} \left(\prod_{i=0}^{c+j-1} \lambda_i \right) \frac{\mu_c}{j!} \int_0^{\infty} \left(\int_0^y \bar{G}(z) dz \right)^j (\mu_c y - 1) e^{-\mu_c y} dy \\ W_q^a &= \frac{p_s}{p_a \bar{\lambda}} \sum_{j=1}^{\infty} \left(\prod_{i=0}^{c+j-1} \lambda_i \right) \frac{\mu_c}{j!} \int_0^{\infty} \left(\int_0^y \bar{G}(z) dz \right)^j (j + 1 - \mu_c y) e^{-\mu_c y} dy \end{aligned}$$

For the proofs we refer to the paper of Brandt and Brandt [50]. Some of them rely on Palm distributions and stationary point processes, especially those, which are concerned with the relation between distributions at arrival epochs and their general counterpart. For more information on these topics, please consult [49]. In their paper, Brandt and Brandt also consider the special case of an impatience time defined as the minimum of a constant and an exponentially distributed random variable. In the extreme, one arrives either at Palm's $M/M/c + M$ model or Gnedenko's $M/M/c + D$ queueing system.

3.4.3 Retrials and the Orbit Model

Up to now it has been assumed for systems with limited capacity, that blocked customers are lost. In the following we will consider these customers to retry for service after some period of time. Obviously there is some dependency introduced in the model, which violates the memoryless property of the arrival stream. By describing the system as a two dimensional Markov process $\{C(t), N(t) : t \geq 0\}$, one restores the desired features. Here $C(t)$ denotes the number of busy servers at time t and $N(t)$ describes the number of retrying sources. One can think of blocked customers beeing redirected to an *orbit* instead of getting lost. For clarification, this situation is shown in figure 3.17. From a different viewpoint, a retrial system forms a queueing network consisting of loss and infinite server nodes. Retrial systems have become important in telephony applications, as a typical caller retries after some time,

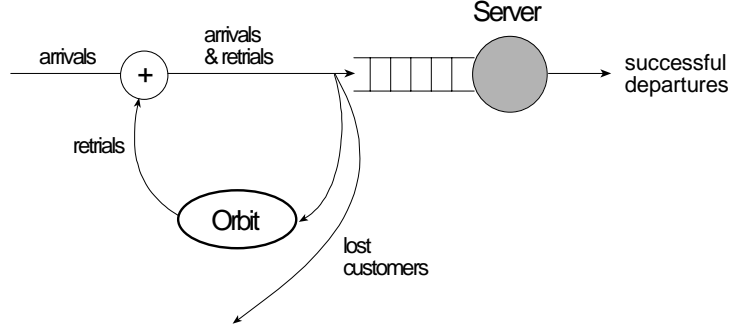


Figure 3.17: Orbit Model

if he does not reach the desired target. For this reason, the service facility is often modeled as loss system, not as queueing system with limited capacity. We will follow this convention and introduce a $M/M/c/c$ queue as service facility. If the service distribution is not exponential, we'll lose the Markov property of the above mentioned process again. Then a supplementary variable describing the elapsed service time has to be introduced to preserve it.

Adhering to the usual notation we'll turn attention to the single server case now, i.e. assume a $M/M/1/1$ service facility. Consequently $C(t)$ can only take the values 0 and 1. Assume that the time lengths between the retrials are independent and follow an exponential distribution with parameter η . Thus on the average every $\frac{1}{\eta}$ seconds (or any other preferred time unit) a retrial occurs. Introducing $p_{m,n} := \Pr \{C(t) = m, N(t) = n\}$, we may proceed as usual and equate the flow in with the flow out. Hence,

$$\begin{aligned} (\lambda + n\eta) p_{0,n} &= \mu p_{1,n} \\ (n+1)\eta p_{0,n+1} &= \lambda p_{1,n} \end{aligned}$$

Following the treatment of [93], both expressions may be combined to

$$p_{1,n+1} = \rho \left(1 + \frac{\lambda}{\eta(n+1)} \right) p_{1,n}$$

This leads to

$$p_{1,n} = \begin{cases} \rho^n \prod_{i=1}^n \left(1 + \frac{\lambda}{\eta i} \right) p_{1,0} & n \geq 1 \\ p_{1,0} & n = 0 \end{cases} \quad (3.55)$$

For a single server system the server utilization $u = \rho$ may be written alternatively as

$$\begin{aligned}
 \rho &= \sum_{n=0}^{\infty} p_{1,n} = p_{1,0} \left[1 + \sum_{n=1}^{\infty} \rho^n \prod_{i=1}^n \left(1 + \frac{\lambda}{\eta i} \right) \right] \\
 &= p_{1,0} \left[1 + \sum_{n=1}^{\infty} \rho^n \prod_{i=1}^n \left(\frac{i + \lambda/\eta}{i} \right) \right] \\
 &= p_{1,0} \left[1 + \sum_{n=1}^{\infty} \rho^n \frac{(1 + \lambda/\eta) \cdots (n + \lambda/\eta)}{n!} \right] \\
 &= p_{1,0} \left[\sum_{n=0}^{\infty} \rho^n \binom{n + \lambda/\eta}{n} \right] \\
 &= p_{1,0} \left[(1 - \rho)^{-1 - \lambda/\eta} \right]
 \end{aligned}$$

where for the last step the binomial theorem has been applied. A simple multiplication yields

$$p_{1,0} = \left[(1 - \rho)^{-1 - \lambda/\eta} \right]^{-1} = \rho (1 - \rho)^{1 + \lambda/\eta} \quad (3.56)$$

So far the steady state distribution for the single server retrial model has been derived. The computation of the expected number of customers in orbit will be performed in two steps. By calculating

$$\begin{aligned}
 \sum_{n=1}^{\infty} n p_{1,n} &= p_{1,0} \sum_{n=1}^{\infty} n \rho^n \prod_{i=1}^n \left(1 + \frac{\lambda}{\eta i} \right) \\
 &= p_{1,0} \left(1 + \frac{\lambda}{\eta} \right) \rho \frac{d}{d\rho} \sum_{n=1}^{\infty} \rho^n \prod_{i=2}^n \left(1 + \frac{\lambda}{\eta i} \right) \\
 &= p_{1,0} \left(1 + \frac{\lambda}{\eta} \right) \rho \frac{d}{d\rho} \sum_{n=1}^{\infty} \rho^{n-1} \prod_{i=1}^{n-1} \left(1 + \frac{\lambda}{\eta i} \right) \\
 &= p_{1,0} \left(1 + \frac{\lambda}{\eta} \right) \rho \frac{d}{d\rho} (1 - \rho)^{-1 - \lambda/\eta} \\
 &= p_{1,0} \left(1 + \frac{\lambda}{\eta} \right) \rho (1 - \rho)^{-2 - \lambda/\eta} \\
 &= \frac{\rho^2}{1 - \rho} \left(1 + \frac{\lambda}{\eta} \right)
 \end{aligned}$$

and

$$\begin{aligned}
\sum_{n=1}^{\infty} n p_{0,n} &= \sum_{n=1}^{\infty} \frac{n\mu}{(\lambda + n\eta)} p_{1,n} = \sum_{n=1}^{\infty} \frac{\mu}{\eta} \left(\frac{\lambda + n\eta}{n\eta} \right)^{-1} p_{1,n} \\
&= \frac{\mu}{\eta} p_{1,0} \sum_{n=1}^{\infty} \left(1 + \frac{\lambda}{n\eta} \right)^{-1} \rho^n \prod_{i=1}^n \left(1 + \frac{\lambda}{\eta i} \right) \\
&= p_{1,0} \frac{\mu}{\eta} \rho \left(1 + \sum_{n=2}^{\infty} \rho^{n-1} \prod_{i=1}^{n-1} \left(1 + \frac{\lambda}{\eta i} \right) \right) \\
&= p_{1,0} \frac{\lambda}{\eta} \left(1 + \sum_{n=1}^{\infty} \rho^n \prod_{i=1}^n \left(1 + \frac{\lambda}{\eta i} \right) \right) \\
&= p_{1,0} \frac{\lambda}{\eta} (1 - \rho)^{-1-\lambda/\eta} = \frac{\lambda}{\eta} \rho
\end{aligned}$$

one easily derives the expected number of customers in orbit

$$\begin{aligned}
L_q &= \sum_{n=1}^{\infty} n (p_{0,n} + p_{1,n}) = \frac{\rho^2}{1 - \rho} \left(1 + \frac{\lambda}{\eta} \right) + \frac{\lambda}{\eta} \rho \\
&= \frac{\rho(\rho\eta + \lambda)}{\eta(1 - \rho)} = \frac{\rho^2}{1 - \rho} + \frac{\lambda\rho}{\eta(1 - \rho)} = \frac{\lambda\rho}{\eta(1 - \rho)} + L_q^{M/M/1} \quad (3.57)
\end{aligned}$$

By comparing the result with expression 3.15 for the $M/M/1$ model, one identifies the second term as some kind of *expected excess* in the number of customers [93]. Letting η approach infinity, the excess vanishes and we arrive at an ordinary $M/M/1$ queueing model. There is no delay between subsequent retries and so the orbit attaches as queue to the $M/M/1/1$ loss model. The remaining performance characteristics are determined by an application of Little's law

$$\begin{aligned}
W_q &= \frac{1}{\lambda} L_q = \frac{\lambda(\rho\eta + \lambda)}{\eta(1 - \rho)} \\
L &= L_q + \rho = \frac{\rho(\rho\eta + \lambda) + \rho\eta(1 - \rho)}{\eta(1 - \rho)} = \frac{\rho(\eta + \lambda)}{\eta(1 - \rho)} \\
W &= \frac{1}{\lambda} L = \frac{\eta + \lambda}{\eta\mu(1 - \rho)}
\end{aligned}$$

It turns out, that the single server retrial system is stable for $\rho < 1$ [74]. One may also derive the conditional average waiting time in orbit for an arriving

customer given a busy server. By realizing, that the arrival rate to the orbit is $\lambda\rho$ and using Little's law yields

$$\mathbb{E} \left\{ \check{W}_q | \check{W}_q > 0 \right\} = \frac{1}{\lambda\rho} L_q = \frac{1}{1-\rho} \left(\frac{1}{\mu} + \frac{1}{\eta} \right)$$

Again we detect an excess to the average queueing time of the $M/M/1$ model. A slightly different approach to the one presented here is given in [74] by using a generating function approach to derive the main performance characteristics. G.I. Falin and J.G.C. Templeton also present results for the variance of the average number of customers in orbit and in system. Their results are stated here for completeness without proof

$$\begin{aligned} \sigma_L^2 &= \frac{\rho(\eta + \lambda)}{\eta(1-\rho)^2} \\ \sigma_{L_q}^2 &= \frac{\rho(\rho\eta + \rho^2\eta - \rho^3\eta + \lambda)}{\eta(1-\rho)^2} \end{aligned}$$

The calculation of the variance of the average waiting time is not straightforward, as customers may overtake each other randomly in orbit. For a detailed analysis on the waiting time distribution we refer to their book [74].

The multiserver case may be approached by letting $C(t)$ assume values between 0 and c . Proceeding as usual leads to the following system of equations for the steady state probabilities

$$\begin{aligned} \left(\rho + m + \frac{\eta}{\mu} \right) p_{m,n} &= \rho p_{m-1,n} + (m+1) p_{m+1,n} + \frac{\eta}{\mu} (n+1) p_{m-1,n+1} \\ (\rho + c) p_{c,n} &= \rho p_{c-1,n} + \rho p_{c,n-1} + \frac{\eta}{\mu} (n+1) p_{c-1,n+1} \end{aligned}$$

Please note, that we have divided the equations by μ for convenience. Introducing the partial generating function $P_m(z) = \sum_{n=0}^{\infty} z^n p_{m,n}$ for $0 \leq m \leq c$ and $|z| < 1$, the above system of equation may be written as [83]

$$\begin{aligned} (\rho + m) P_m(z) + \frac{\eta}{\mu} z \frac{d}{dz} P_m(z) &= \rho P_{m-1}(z) + (m+1) P_{m+1}(z) + \frac{\eta}{\mu} z \frac{d}{dz} P_{m-1}(z) \\ (\rho + c) P_c(z) &= \rho P_{c-1}(z) + \rho z P_c(z) + \frac{\eta}{\mu} \frac{d}{dz} P_{c-1}(z) \end{aligned}$$

Introducing the bivariate generating function $P(y, z) = \sum_{m=0}^c y^m P_m(z)$ and repeating the step above yields

$$\begin{aligned} \rho(1-y)P(y, z) + \frac{\eta}{\mu}(z-y)\frac{\partial}{\partial z}P(y, z) + (y-1)\frac{\partial}{\partial y}P(y, z) \\ + \rho y^c(y-z)P_c(z) + \frac{\eta}{\mu}(y-z)\frac{d}{dz}P_{c-1}(z) = 0 \end{aligned}$$

Differentiating with respect to z , y , yy , yz , zz at the point $y = 1$, $z = 1$ leads to a system of equations, which can be solved in terms of $L_q = \frac{d}{dz}P(1, 1)$, the blocking probability $p_b = P_c(1)$, the utilization $u = \mathbb{E}C(t) = \frac{d}{dy}P(1, 1)$ and other variables. Simplification yields

$$\begin{aligned} u &= \rho \\ L_q &= \left(1 + \frac{\eta}{\mu}\right) \frac{\rho - \sigma_C^2}{c - \rho} \end{aligned} \quad (3.58)$$

The detailed calculations have been omitted for sake of readability, they do not provide any further insight into the problem. The interested reader is referred to the book of Falin and Templeton [74]. It can be shown, that the multiserver retrial system is stable for $u < 1$. As will be shown next, the model is bounded from above by the classic $M/M/c$ queueing system and so the stability condition adheres to intuition.

This is as far as one can get with exact techniques. Closed form solutions only exist in the case of one or two servers [74], for $c \geq 3$ the average number of customers in orbit depends on the variance of the number of busy servers σ_C^2 . In the extreme for $\eta \rightarrow \infty$ the retrial model approaches the classic $M/M/c$ queueing system, whereas for $\eta = 0$ it reduces to an Erlang loss system. This allows for an approximation for high and low retrial rates η . In the former case the blocking probability p_b is approximated by the probability of delay $p_d^{(M/M/c)}$ for the $M/M/c$ queue given by expression 3.24. The same applies to the average queue length, i.e. $L_q \approx L_q^{(M/M/c)}$.

For η small the Erlang loss formula 3.28 with traffic intensity $\bar{\rho} = \frac{\lambda+r}{\mu}$ and c servers provides a starting point for an approximation. Hereby we assume, that the unknown retrial arrival rate r does not depend on the number of busy servers. It is easy to verify, that the Erlang loss formula constitutes a distribution allowing us to calculate mean and variance. For the purposes of the current section we will denote it by $E(\bar{\rho}, c)$, where $\bar{\rho}$ and c are the parameters. Keeping in mind, that this distribution describes the random variable *busy servers*, its expectation must equal the utilization of the retrial system, i.e. $u = \rho = \bar{\rho}(1 - E(\bar{\rho}, c))$ leading to $E(\bar{\rho}, c) = 1 - \rho/\bar{\rho} = \frac{r}{\lambda+r}$.

Similar considerations yield $\sigma_b^2 = \rho - (c - \rho)(\bar{\rho} - \rho)$, the variance of the random variable busy servers. Returning to the high rate approximation, the same idea may be applied to the Erlang delay formula leading to an approximation of the variance $\sigma_d^2 = \rho \left(1 - p_d^{(M/M/c)}\right)$. Although both variances are related to the number of busy servers, we kept the suffixes to show the origin of the formulas.

For intermediate values of η , the most straightforward way to provide an approximation is via interpolation, i.e.

$$\begin{aligned} p_b &\approx \frac{1}{1 + \frac{\eta}{\mu}} E(\bar{\rho}, c) + \frac{\frac{\eta}{\mu}}{1 + \frac{\eta}{\mu}} p_d^{(M/M/c)} \\ \sigma_C^2 &\approx \frac{1}{1 + \frac{\eta}{\mu}} \sigma_b^2 + \frac{\frac{\eta}{\mu}}{1 + \frac{\eta}{\mu}} \sigma_d^2 \end{aligned}$$

Inserting the expression for σ_C^2 into formula 3.58 yields

$$\begin{aligned} L_q &= \left(1 + \frac{\eta}{\mu}\right) \frac{\rho - \sigma_C^2}{c - \rho} = \left(1 + \frac{\eta}{\mu}\right) \frac{\left(1 + \frac{\eta}{\mu}\right) \rho - \sigma_b^2 - \frac{\eta}{\mu} \sigma_d^2}{(c - \rho) \left(1 + \frac{\eta}{\mu}\right)} \\ &\approx \frac{\left(1 + \frac{\eta}{\mu}\right) \rho - \rho + (c - \rho)(\bar{\rho} - \rho) - \frac{\eta}{\mu} \rho \left(1 - p_d^{(M/M/c)}\right)}{c - \rho} \quad (3.59) \\ &= \frac{(c - \rho) \frac{\eta}{\mu} + \frac{\eta}{\mu} \rho p_d^{(M/M/c)}}{c - \rho} = \frac{r}{\mu} + \frac{\eta}{\mu} \frac{\rho}{c - \rho} p_d^{(M/M/c)} \\ &= \frac{r}{\mu} + \frac{\eta}{\mu} L_q^{(M/M/c)} \end{aligned}$$

where the unknown quantity r is calculated from $E(\frac{\lambda+r}{\mu}, c) = \frac{r}{\lambda+r}$ for given values of λ , μ and c . Although there is an appealing relation to the $M/M/c$ queue, it is in general not additive as one would expect from the single server case. The remaining performance characteristics may be determined from an application of Little's law, i.e.

$$W_q = \frac{1}{\lambda} L_q, \quad W = W_q + \frac{1}{\mu}, \quad L = \lambda W = L_q + \rho$$

Some of the ideas presented here have to be attributed to R.I. Wilkinson [171], but the most complete reference in the field is the book by Falin and

Templeton [74]. A survey on retrial queues is provided by V.G. Kulkarni and H.M. Liang in [134]. G. Fayolle and M.A. Brun have treated a model with customer impatience and repeated calls in their paper [75]. Their model is rather cumbersome and difficult to analyze.

3.5 Arbitrary Service Processes

In certain cases the Poisson assumption is not appropriate. One has to consider non-Poisson arrival and service processes. The current section will provide some results on queueing systems with rather arbitrary distributions. If either one of the distributions is Markovian, the corresponding queue may be analyzed by embedding a discrete time Markov chain (see appendix A.3). This method introduced by Kendall suggests the system to be modeled by viewing it only at times where the Markov property holds. By careful selection of the regeneration points, one is able to nullify the impact of residual service and residual interarrival time.

3.5.1 Single Server Systems

Consider a single server queue with Poissonian arrivals at rate λ and arbitrary (absolute continuous) service distribution $B(\cdot)$ with average service time $\frac{1}{\mu}$ and finite variance. As regeneration points choose the instance at which customers complete service and depart from the system. At that time either a waiting customer commences service or the system becomes idle. More exact, the residual life time is zero, but the customer has not left the system yet. Define $\bar{b}(s) = \int_0^\infty b(x)e^{-sx}dx$ as the Laplace transform of the service density $b(\cdot)$ corresponding to $B(\cdot)$. Due to the Poissonian nature of the arrival process, the probability of n arrivals between two successive departures within an interval of length t is given as

$$q_{n|t} = \frac{e^{-\lambda t} (\lambda t)^n}{n!} \quad (3.60)$$

Averaging yields the probability of n arrivals within an interval of length t , that is

$$q_n = \int_0^\infty q_{n|t} b(t) dt$$

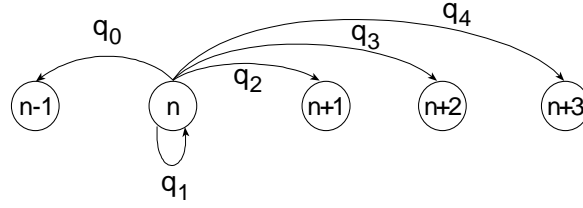


Figure 3.18: Embedded Markov chain for the M/G/1 queue

We are now able to assemble the matrix of transition probabilities

$$\mathbf{Q} := \begin{pmatrix} q_0 & q_1 & q_2 & \cdots \\ q_0 & q_1 & q_2 & \cdots \\ 0 & q_0 & q_1 & \cdots \\ 0 & 0 & q_0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The number of customers in the queue is decreased by one, if no customer arrives in the observed interval. If only one customer arrives, the system remains in state n . Otherwise the process can reach any state $k > n$. We say, the embedded Markov chain is *skip-free to the left*. Also refer to figure 3.18 for a graphical representation of the possible state transitions out of state n . As shown in appendix A.3, the equilibrium distribution \mathbf{p} is found by solving $\mathbf{pQ} = \mathbf{p}$ and normalizing. Reverting to classic notation leads to

$$p_n = p_0 q_n + \sum_{i=0}^{n+1} p_i q_{n-i+1} = p_0 q_n + \sum_{i=0}^n p_{i+1} q_{n-i}, \quad n \geq 0 \quad (3.61)$$

By the use of generating functions

$$\begin{aligned}
P(z) &= \sum_{i=0}^{\infty} p_i z^i, \quad |z| < 1 \\
P_+(z) &= \sum_{i=0}^{\infty} p_{i+1} z^i = z^{-1} \sum_{i=0}^{\infty} p_{i+1} z^{i+1} \\
&= -z^{-1} p_0 + z^{-1} \sum_{i=-1}^{\infty} p_{i+1} z^{i+1} \\
&= -z^{-1} p_0 + z^{-1} \sum_{i=0}^{\infty} p_i z^i = z^{-1} (P(z) - p_0) \\
Q(z) &= \sum_{i=0}^{\infty} q_i z^i = \sum_{i=0}^{\infty} z^i \int_0^{\infty} q_i b(t) dt \\
&= \sum_{i=0}^{\infty} z^i \int_0^{\infty} \frac{e^{-\lambda t} (\lambda t)^i}{i!} b(t) dt = \bar{b}(\lambda(1-z)) \quad (3.62)
\end{aligned}$$

the system of equations 3.61 may be written as

$$P(z) = p_0 Q(z) + P_+(z) Q(z) = p_0 Q(z) + z^{-1} (P(z) - p_0) Q(z)$$

Rearranging

$$P(z) (1 - z^{-1} Q(z)) = p_0 (1 - z^{-1}) Q(z)$$

finally leads to

$$P(z) = \frac{p_0 (1 - z^{-1}) Q(z)}{1 - z^{-1} Q(z)} = \frac{p_0 (1 - z) Q(z)}{Q(z) - z} \quad (3.63)$$

To determine p_0 one needs to apply the properties of moment generating functions [83], that is

$$P(1) = 1, \quad Q(1) = 1, \quad \frac{d}{dz} Q(z)|_{z=1} = \frac{\lambda}{\mu} = \rho$$

The last equation stems from the fact, that ρ is the expected value for a transition to occur. Rearranging expression 3.63 and letting z approach 1 yields

$$p_0 = \lim_{z \rightarrow 1} \frac{P(z) (Q(z) - z)}{(1 - z) Q(z)}$$

Applying L'Hospital's rule provides us with the desired result

$$p_0 = 1 - \rho \quad (3.64)$$

Hence equation 3.63 now becomes

$$P(z) = \frac{(1 - \rho)(1 - z)Q(z)}{Q(z) - z} \quad (3.65)$$

To obtain a solution to equation 3.65 and derive the steady state distribution one needs to invert the transforms involved. This is not possible without assuming a specific service distribution $B(\cdot)$. Even then, it is questionable, if the desired inversion can be carried out. From a theoretical standpoint, one has only to isolate the coefficients of z^i in the series $P(z)$. Strictly speaking, we have only determined the system size distribution as seen by arrivals. Note, that in a stable system, the system size distribution seen by departures equals the one seen by arrivals. We are now able to apply theorem 13 (PASTA) to see, that the latter are equal to the system size distribution [215]. Following that daisy chain, we conclude, that expression 3.65 sufficiently describes the distribution of the number of customers in system. To determine the average system size L , one once again makes use of the properties of generating functions [83]:

$$L = \frac{d}{dz}P(z)|_{z=1}$$

It turns out, that this operation also might become rather cumbersome. Fortunately a more direct approach exists and will be presented next.

Let A_k and D_k denote the number of customers entering and leaving the system immediately after the k -th departure has occurred. Then the following recurrence relation for the number of customers in the system N_k may be observed

$$N_{k+1} = \begin{cases} N_k - 1 + A_{k+1} & \text{for } N_k > 0 \\ A_{k+1} & \text{for } N_k = 0 \end{cases} \quad (3.66)$$

By introducing an auxillary function

$$U(N_k) = \begin{cases} 1 & \text{for } N_k > 0 \\ 0 & \text{for } N_k = 0 \end{cases}$$

relation 3.66 may be simplified to

$$N_{k+1} = N_k - U(N_k) + A_{k+1} \quad (3.67)$$

Taking expectation gives

$$\mathbb{E}N_{k+1} = \mathbb{E}N_k - \mathbb{E}U(N_k) + \mathbb{E}A_{k+1}$$

By realizing, that the number of customers after the k -th departure N_k does not depend on k , both terms may be eliminated from the above equation. This leads to

$$\mathbb{E}U(N_k) = \mathbb{E}A_{k+1}$$

Another expression for $U(N_k)$ may be derived by averaging over all possible values, i.e.

$$\mathbb{E}U(N_k) = \sum_{n=0}^{\infty} U(N_k) \Pr\{N_k = n\} = \sum_{n=1}^{\infty} \Pr\{N_k = n\}$$

The last term on the right is simply the probability of the server being busy. Recall from equation 3.64, that the probability of an idle system is given by $p_0 = 1 - \rho$. Putting it all together results in

$$\mathbb{E}U(N_k) = \mathbb{E}A_{k+1} = 1 - p_0 = \rho \quad (3.68)$$

As a next step, equation 3.67 has to be squared

$$N_{k+1}^2 = N_k^2 + U^2(N_k) + A_{k+1}^2 - 2N_k U(N_k) - 2A_{k+1} U(N_k) + 2N_k A_{k+1}$$

Taking expectation and eliminating terms $\mathbb{E}N_{k+1}^2 = \mathbb{E}N_k^2$ yields

$$0 = \mathbb{E}U^2(N_k) + \mathbb{E}A_{k+1}^2 - 2\mathbb{E}(N_k U(N_k)) - 2\mathbb{E}(A_{k+1} U(N_k)) + 2\mathbb{E}(N_k A_{k+1}) \quad (3.69)$$

Notice the two obvious relations $U^2(N_k) = U(N_k)$ and $N_k U(N_k) = N_k$, which follow immediately from the definition. By independence of A_{k+1} and N_k , one may write $\mathbb{E}(N_k A_{k+1}) = \mathbb{E}N_k \mathbb{E}A_{k+1} = \rho^2$. Substitution of these identities in expression 3.69 gives

$$0 = \rho + \mathbb{E}A_{k+1}^2 - 2L - 2\rho^2 + 2\rho L \quad (3.70)$$

As the arrivals follow a Poisson process, the number of arrivals occurring in the interval between two subsequent departures depends only on the length of the interval, not on the interval itself. Consequently the index may be omitted, i.e. $\mathbb{E}A^2 := \mathbb{E}A_{k+1}^2$. Next, the second moment is expressed in terms of expectation and variance, that is $\mathbb{E}A^2 = \text{Var}(A) + (\mathbb{E}A)^2 = \text{Var}(A) +$

ρ^2 . With S denoting the service time (with $\mathbb{E}S = \frac{1}{\mu}$), one may split the variance to $Var(A) = \mathbb{E}(Var(A|S)) + Var(\mathbb{E}(A|S)) = \mathbb{E}(\lambda S) + Var(\lambda S)$. By denoting the variance of S as σ_S^2 and recalling $\mathbb{E}(\lambda S) = \lambda \mathbb{E}(S) = \frac{\lambda}{\mu} = \rho$ one arrives at the expression $Var(A) = \rho + \lambda^2 Var(S) = \rho + \lambda^2 \sigma_S^2$ leading to

$$\mathbb{E}A^2 = \rho + \lambda^2 \sigma_S^2 + \rho^2$$

Substituting in equation 3.70 yields

$$0 = 2\rho - \rho^2 + \lambda^2 \sigma_S^2 + 2(1 - \rho)L$$

Rearranging terms finally results in

$$L = \frac{2\rho - \rho^2 + \lambda^2 \sigma_S^2}{2(1 - \rho)} = \rho + \frac{\rho^2 + \lambda^2 \sigma_S^2}{2(1 - \rho)} \quad (3.71)$$

The above result 3.71 is often referred to as *Pollaczek-Khintchine Formula* and enables us to derive the remaining performance characteristics by applying Little's law. This leads to

$$\begin{aligned} W &= \frac{1}{\lambda} L = \frac{1}{\mu} + \frac{\rho^2 + \lambda^2 \sigma_S^2}{2\lambda(1 - \rho)} \\ W_q &= W - \frac{1}{\mu} = \frac{\rho^2 + \lambda^2 \sigma_S^2}{2\lambda(1 - \rho)} \\ L_q &= \lambda W_q = \frac{\rho^2 + \lambda^2 \sigma_S^2}{2(1 - \rho)} \end{aligned}$$

For further reading, we again refer to classic textbooks on queueing theory. We were mainly led by [90]. Similar derivations may be found also in [86] and [215]. For more advanced approaches consider [30] and [172]. Due to the vast amount of results on the $M/G/1$ model, we had to omit some of them. These include for example waiting time distributions, batch arrivals and priorities. Much of the material not covered may be found in the given references.

Example 17 Assume an exponential distribution for the service times, i.e. we specialize on the $M/M/1$ model. Applying the Laplace transform of the

exponential density $\bar{b}(s) = \frac{\mu}{\mu+s}$ to expression 3.62 and 3.65 results in

$$\begin{aligned} P(z) &= \frac{(1-\rho)(1-z) \frac{\mu}{\mu+\lambda(1-z)}}{\frac{\mu}{\mu+\lambda(1-z)} - z} \\ &= \frac{(1-\rho)(1-z)\mu}{\mu - z\mu - z\lambda(1-z)} \\ &= \frac{(1-\rho)\mu}{\mu - z\lambda} = \frac{(1-\rho)}{1-z\rho} \end{aligned}$$

Expanding the result in a geometric series, i.e.

$$(1-\rho) \frac{1}{1-z\rho} = (1-\rho) \sum_{n=0}^{\infty} z^n \rho^n$$

and isolating the coefficients of z^n leads to the steady state distribution

$$p_n = (1-\rho) \rho^n$$

which equals expression 3.13. We have shown, that the M/M/1 model fits perfectly in the framework of the more general M/G/1 queueing system. To derive a result for the average system size we first note, that exponential distributed service times S with rate μ have mean $\mathbb{E}S = \frac{1}{\mu}$ and variance $\sigma_S^2 = \frac{1}{\mu^2}$. Substituting these values in the Pollaczek-Khintchine formula 3.71 leads to

$$L = \frac{2\rho - \rho^2 + \lambda^2 \frac{1}{\mu^2}}{2(1-\rho)} = \frac{\rho}{1-\rho}$$

Comparing the result to expression 3.14 shows the expected result.

Example 18 Now consider deterministic service times, i.e. we specialize to the M/D/1 model. To gain results one usually has to employ integro-differential equations. By applying the results for the M/G/1 model we are able to significantly reduce the mathematical effort necessary. The deterministic distribution is in some sense malformed, as there is only a single point with mass 1, i.e.

$$b(x) = \delta\left(x - \frac{1}{\mu}\right)$$

Here the function $\delta(z)$ describes the Kronecker function

$$\delta(z) = \begin{cases} 0 & z \neq 0 \\ 1 & z = 0 \end{cases} \quad (3.72)$$

The Laplace transform of the density is given by

$$b(s) = e^{-\frac{1}{\mu}}$$

Applying to expression 3.62 and 3.65 leads to

$$\begin{aligned} P(z) &= \frac{(1-\rho)(1-z)e^{-\lambda(1-z)/\mu}}{e^{-\lambda(1-z)/\mu} - z} \\ &= \frac{(1-\rho)(1-z)}{1 - ze^{\rho(1-z)}} \end{aligned}$$

Expanding in a geometric series

$$P(z) = (1-\rho)(1-z) \sum_{n=0}^{\infty} z^n e^{n\rho(1-z)}$$

and expressing the exponential function in an exponential series allows one to isolate the coefficients of z^n :

$$\begin{aligned} p_0 &= 1 - \rho \\ p_1 &= (1 - \rho)(e^\rho - 1) \\ p_n &= (1 - \rho) \sum_{i=0}^n \frac{(-i\rho)^{n-i} e^{i\rho}}{(n-i)!} - \sum_{i=0}^{n-1} \frac{(-i\rho)^{n-i-1} e^{i\rho}}{(n-i-1)!} \end{aligned} \quad (3.73)$$

Compared to the calculation so far, the derivation of the average system size gets even simpler than for the M/M/1 model. As there is no variation in the model, i.e. $\sigma_S^2 = 0$, the Pollaczek-Khintchine formula 3.71 immediately becomes

$$L = \rho + \frac{\rho^2}{2(1-\rho)} \quad (3.74)$$

Rewriting expression 3.74 reveals an interesting relation between the M/D/1 and the M/M/1 model:

$$L = \frac{\rho}{1-\rho} - \frac{\rho^2}{2(1-\rho)} = L^{(M/M/1)} - \frac{\rho^2}{2(1-\rho)}$$

It turns out, that given the same parameters the number of customers is always smaller for systems with deterministic service times. In case of heavy

traffic, i.e. $\rho \rightarrow 1$, the system size of the $M/M/1$ model is twice the size of the $M/D/1$ queueing system:

$$\lim_{\rho \rightarrow 1} L = \frac{1}{2} \lim_{\rho \rightarrow 1} L^{(M/M/1)}$$

For sake of readability, some calculations have been omitted. The detailed calculations may be found in [179].

By introducing the coefficient of variation $c_S = \frac{\sqrt{\text{Var}(S)}}{\mathbb{E}S} = \mu\sigma_S$, one may think of using it as a control parameter to interpolate between deterministic ($c_S = 0$) and exponential service times ($c_S = 1$). In fact, this is possible by reinterpreting the Pollaczek-Khintchine formula 3.71 as follows

$$\begin{aligned} L &= \frac{2\rho - \rho^2 + \rho^2 c_S^2}{2(1 - \rho)} = \frac{2\rho - \rho^2 - \rho^2 c_S^2 + 2\rho c_S^2 - 2\rho c_S^2}{2(1 - \rho)} \\ &= c_S^2 \frac{2\rho}{2(1 - \rho)} + (1 - c_S^2) \frac{2\rho - \rho^2}{2(1 - \rho)} \\ &= c_S^2 \frac{\rho}{1 - \rho} + (1 - c_S^2) \frac{2\rho - \rho^2}{2(1 - \rho)} \\ &= c_S^2 L^{(M/M/1)} + (1 - c_S^2) L^{(M/D/1)} \end{aligned} \tag{3.75}$$

By Little's law the same convex combination may also be applied to the other performance characteristics. Although this interpretation is not very appealing in its own sense, it becomes of great interest for the approximation of multiserver limited capacity systems. In fact, it will turn out, that the idea extends to the most general models.

3.5.2 Finite Single Server Systems

In this section we will get in touch with finite source and limited capacity systems. We will provide some ideas and summarize the major results. For sake of readability lengthy discussions will be omitted, the details missing may be found in the given literature. First consider the $M/G/1/K$ model. As before for the exponential version a limit of K customers is allowed and customers arriving at a full system are turned away. The service time S follows an arbitrary distribution $B(\cdot)$ with expectation $\frac{1}{\mu}$ and finite variance. No stable system needs to be assumed, as the finite waiting room provides

an upper limit for the number of customers in the system. It can be shown [86][198], that the steady state distribution of $M/G/1/K$ system is proportional to the stationary solution of a stable $M/G/1$ queue given the same parameter. The latter is sufficiently described by expression 3.65 and will be denoted as $p_n^{(M/G/1)}$. Following Gnedenko and Kovalenko [86], the first K probabilities are given by

$$\begin{aligned} p_n &= \kappa p_n^{(M/G/1)}, \quad 0 \leq n < K \\ \kappa &= \left[1 - \rho + \rho \sum_{n=0}^{K-1} p_n^{(M/G/1)} \right]^{-1} \end{aligned} \quad (3.76)$$

Applying the usual normalization condition $\sum_{n=0}^{K-1} p_n$ leads to the expression for p_K , which is also the probability of being blocked

$$p_b = p_K = \kappa \left[1 - \rho + (\rho - 1) \sum_{n=0}^{K-1} p_n^{(M/G/1)} \right] \quad (3.77)$$

It turns out, that a similar relation also exists, when the infinite system becomes unstable. More details may be found in [86]. Having calculated the blocking probability we are now in the position to derive an expression for the effective arrival rate,

$$\bar{\lambda} = \lambda (1 - p_K)$$

From the steady state distribution, the average number of customers in the system may be determined by

$$L = \sum_{n=0}^K n p_n$$

Application of Little's law yields the remaining performance characteristics

$$\begin{aligned} W &= \frac{L}{\bar{\lambda}} = \frac{L}{\lambda (1 - p_K)} \\ W_q &= W - \frac{1}{\mu} \\ L_q &= \bar{\lambda} W_q = \lambda (1 - p_K) W_q \end{aligned}$$

Alternatively one may follow the embedded Markov chain approach as has been done before for the $M/G/1$ queueing system. This leads to a finite state

Markov chain in discrete time. Each of the results given above may then be determined in terms of the corresponding stationary solution. For further details we refer to one of the most complete references on the $M/G/1/K$ model available, that is [198].

We now proceed to the finite population $M/G/1$ queueing system, also referred to as *machine-repairman system*. The working machines are associated with the source and the broken machines form a queue waiting for repair. In our case a single repairman is available to perform the job. The average time in system then becomes the expected machine outage time. Based on such key indicators the cost of operating a production business may be inferred. We will now present some results derived by H. Takagi in [198] without proof. Consider a system with population size N and the other parameters defined as in the $M/G/1$ model. Given the Laplace transform $\bar{b}(s)$ of the service time density, the mean arrival rate is given by

$$\bar{\lambda} = \frac{N\lambda \left[1 + \sum_{n=1}^{N-1} \binom{N-1}{n} \prod_{i=1}^n (\bar{b}^{-1}(i\lambda) - 1) \right]}{1 + N\rho \left[1 + \sum_{n=1}^{N-1} \binom{N-1}{n} \prod_{i=1}^n (\bar{b}^{-1}(i\lambda) - 1) \right]}$$

From the expression for the average time in system

$$W = \left[1 + \sum_{n=1}^{N-1} \binom{N-1}{n} \prod_{i=1}^n (\bar{b}^{-1}(i\lambda) - 1) \right]^{-1} + \frac{N}{\mu} - \frac{1}{\lambda}$$

the remaining performance characteristics may be derived by applying Little's law

$$\begin{aligned} L &= \bar{\lambda}W = N - \frac{\bar{\lambda}}{\lambda} \\ W_q &= W - \frac{1}{\mu} \\ L_q &= \bar{\lambda}W_q \end{aligned}$$

From the results of the two above models one can imagine, that the corresponding calculations quickly become cumbersome. Both models occur relatively rare in queueing literature and are completely omitted in standard queueing theory textbooks. An exception to the rule is [198], where all the necessary details are to be found.

3.5.3 Multiserver Systems

In generalizing to more servers, we loose all the powerful tools used so far. In fact, the $M/G/c$ queueing system does not permit a simple analytical solution. We can not apply the method of embedded Markov chains the usual way and there is no such relation as the Pollaczek-Khintchine formula. The only item left in our toolbox is Little's law. Before obtaining some approximations for the $M/G/c$ model, we note, that rather simple solutions exist for two special cases. The first is the $M/G/c/c$ model already discussed on page 3.3.2. The second is the so called infinite server queue $M/G/\infty$, which derives from the $M/G/c/c$ model by allowing c to become infinite. This immediately leads to the stationary distribution

$$p_n = \frac{e^{-\rho} \rho^n}{n!}$$

The remaining results may be determined in the same fashion [90].

One of the simplest approximations to be obtained is based on a generalization of the idea, which led us to expression 3.75. By considering the result to be valid for multiple servers as well, one arrives at

$$L \approx c_S^2 L^{(M/M/c)} + (1 - c_S^2) L^{(M/D/c)} \quad (3.78)$$

The expression for $L^{(M/M/c)}$ may be determined from formula 3.26. As shown in [179] and [168], the generating function for system size distribution of the $M/D/c$ model is given by

$$P(z) = \frac{\sum_{n=0}^c p_n (z^n - z^c)}{1 - z^c e^{\rho(1-z)}}, \quad |z| < 1 \quad (3.79)$$

By realizing, that the numerator is a polynomial of degree c and rewriting expression 3.79 after some manipulations yields [179]

$$P(z) = -\frac{c - \rho}{(1 - z_1) \cdots (1 - z_{c-1})} \frac{(z - 1)(z - z_1) \cdots (z - z_{c-1})}{1 - z^c e^{\rho(1-z)}}$$

The z_1, \dots, z_c are the zeros of the numerator within the unit circle, that is $z_n = \{z : |z| < 1 \text{ and } P(z) = 0\}$, $0 < n \leq c$. The last zero is always given by $z_c = 1$. Note, that Rouché's theorem 15 assures, that z_1, \dots, z_c are found within the unit circle. The probabilities p_n may now be obtained as the coefficient of z^n in the power series expansion of $P(z)$ for a given number of

servers c . The average system size may be determined by applying the well known property of generating functions $L = \frac{d}{dz}P(z)|_{z=1}$ to expression 3.79. Some further algebra finally leads to

$$L^{(M/D/c)} = \frac{\rho^2 - c(c-1) + \sum_{n=0}^{c-1} [c(c-1) - n(n-1)] p_n^{(M/D/c)}}{2c(1 - \rho/c)} + \rho$$

Due to the fact, that the $p_n^{(M/D/c)}$ have not vanished from the expression above, the derivation of an exact solution still remains a rather tedious task. Fortunately an approximation developed by G.P. Cosmetatos and W. Whitt has been suggested in [204][122]:

$$L^{(M/D/c)} \approx \frac{L_q^{(M/M/c)}}{2} [1 + \varsigma(\rho, c)] + \rho \quad (3.80)$$

$$\varsigma(\rho, c) = \min \left(\frac{1 - 10^{-6}}{4}, \left(1 - \frac{\rho}{c}\right) (c-1) \frac{\sqrt{4 + 5c} - 2}{16\rho} \right) \quad (3.81)$$

By applying a regenerative approach and including information about the elapsed service time into the model, M. van Hoorn was able to deduce another approximation. Consider the following assumptions

- The residual service times are independent random variables each with residual life distribution

$$B_r(t) = \mu \int_0^t (1 - B(x)) dx$$

- Given a full system, the time until the next departure has distribution function $B(ct)$. Thus the $M/G/c$ queue is treated as $M/G/1$ queue with rate $c\mu$.

By applying the following recursion scheme, one arrives at an approximation for the steady state distribution

$$\begin{aligned} p_n &\approx \begin{cases} \left[\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!(1-\rho/c)} \right]^{-1} & n = 0 \\ \frac{\rho^n}{n!} p_0 & n < 0 < c \\ \lambda (\alpha_{n-c} p_{c-1} + \sum_{i=c}^n \beta_{n-i} p_i) & n \geq c \end{cases} \quad (3.82) \\ \alpha_n &= \int_0^\infty (1 - B_r(x))^{c-1} (1 - B(x)) e^{-\lambda x} \frac{(\lambda x)^n}{n!} dx, \quad n \geq 0 \\ \beta_n &= \int_0^\infty (1 - B(cx)) e^{-\lambda x} \frac{(\lambda x)^n}{n!} dx, \quad n \geq 0 \end{aligned}$$

Proceeding further, van Hoorn was able to determine the expected queue length

$$L_q = \left[(c\mu - \lambda) \int_0^\infty (1 - B_r(x))^c dx + \frac{\lambda\mu}{c} \mathbb{E}S^2 \right] L_q^{(M/M/c)} \quad (3.83)$$

For all approximation presented above, the remaining performance indicators may be determined by the help of Little's law and the usual relations

$$L = L_q + \rho, \quad W_q = \lambda L_q = L + \frac{1}{\mu}, \quad W = \lambda L$$

Now consider a system with multiple servers and waiting room limitation K . By combining the results for the $M/G/1/K$ model with the approximation above, van Hoorn was able to derive reasonable approximations for the $M/G/c/K$ queueing system. Denoting the probabilities for the infinite server system given by 3.82 with $p_n^{(M/G/c)}$, the corresponding probabilities for the limited capacity system are given by

$$p_n \approx \kappa p_n^{(M/G/c)}, \quad 0 \leq n < K$$

$$\kappa = \left[1 - \rho + \rho \sum_{n=0}^{K-1} p_n^{(M/G/c)} \right]^{-1}$$

Please note the similarity to expression 3.76 obtained for the single server system. The probability for an arriving customer being blocked from entering the system and getting lost is

$$p_d = p_K \approx \rho p_{c-1} - (1 - \rho) \sum_{n=0}^{K-1} p_n$$

By noting, that the effective arrival rate $\bar{\lambda} = (1 - p_K) \lambda$, one may now determine the performance characteristics the same way as has been several times before,

$$L = \sum_{n=0}^K n p_n, \quad W = \frac{L}{\lambda(1 - p_K)}$$

$$W_q = W - \frac{1}{\mu}, \quad L_q = \lambda(1 - p_K) W_q$$

Classic queueing literature provides a wealth of information on bounds and approximation, although much of it is devoted to the more general $G/G/1$ and $G/G/c$ queues. For example, see [90]. A very simple relation between the $M/M/c$ model and the more general $M/G/c$ queueing system with processor sharing discipline has been derived by R.W. Wolff in [215]. Some exact results in terms of generating functions may be found in [179].

3.5.4 Customer Impatience

Balking may be introduced to the $M/G/1$ model in a straightforward manner by prescribing a probability b that a customer enters the system on arrival. In modifying equation 3.60 accordingly, i.e. writing instead

$$q_{n|t} = \frac{e^{-b\lambda t} (b\lambda t)^n}{n!}$$

one is able to carry out the entire set of calculations as given for the classic $M/G/1$ model [90]. Obviously it is much more difficult to attach more than a constant balking rate to the mode. Even for single server systems, the inclusion of customer impatience effects into the model becomes a tedious task. There exist some solutions in the literature. Most of them are based on the refinement of a $G/G/1$ queueing system with impatient customers to the case of Poissonian arrivals. F. Bacelli and G. Hebuterne [34] have shown, that the distribution for the virtual offered waiting time and the distribution for the waiting time coincide for the extended $M/G/1 + G$ model with impatient customers. Consider a single server queue with Poissonian arrivals and arbitrary service distribution $B(\cdot)$ as before for the classic $M/G/1$ queueing system. Let $V(\cdot)$ denote the (absolutely continuous) distribution of the virtual offered waiting time. Define the survival function $\bar{G}(\cdot) = 1 - G(\cdot)$ to the impatience distribution $G(\cdot)$. Note, that $V(\cdot)$ can be interpreted as a mixed distribution, as there is a positive probability for an arriving customer to join service immediately. It splits in a discrete part $V(0)$ and a (absolutely) continuous part with density $v(\cdot)$. Bacelli and Hebuterne have shown, that $v(\cdot)$ is the solution of the following system of integral equations

$$\begin{aligned} v(t) &= \lambda V(0)(1 - B(t)) + \int_0^t v(s)G(s)(1 - B(t - s))ds \\ 1 &= V(0) + \int_0^\infty v(s)ds \end{aligned} \tag{3.84}$$

By substitution, they were able to identify expression 3.84 as Fredholm integral equation of the second kind. The solution allows $v(\cdot)$ to be represented as integral series. For details we refer to the paper by Bacelli and Hebuterne [34]. Another approach is to generalize the equations derived by L. Takacs [196] for the classic $M/G/1$ model to consider forms of customer impatience. This has been carried out by Gnedenko and is shown in [179]. Transient solutions to $M/G/1$ queueing systems with balking and reneging have been investigated by S. Subba Rao in his papers [194] and [195]. It should be noted, that Subba Rao assumes the service distribution to belong to a certain class of distributions following an exponential pattern. He does not consider arbitrary service distributions. Gnedenko's system with limited waiting time has been extended to a multiserver system with balking and reneging in the recent paper by L. Liu and V.G. Kulkarni [135].

3.5.5 Retrials

Under the usual assumptions for a $M/G/1$ queue, we will now attempt to analyze such a system with retrying customers. Time periods between retrials are assumed to follow an exponential distribution with mean $\frac{1}{\eta}$. The system state will be described by a Markov process $\{C(t), \xi(t) < x, N(t) : t \geq 0\}$, where $C(t)$ denotes the number of busy servers, $N(t)$ represents the number of retrials and $\xi(t)$ describes the elapsed service time. To introduce $\xi(t)$ into the model preserves the Markov property and is called the technique of *supplementary variables*. Please note, that for $C(t) = 0$ there is no need to define an elapsed service time, as no customer is present in the system. The relevant states are collapsed into a single simpler state. Putting it together the equilibrium probabilities are defined as

$$\begin{aligned} p_{0,n} &= \Pr \{C(t), N(t)\} \\ p_{1,n}(x) &= \Pr \{C(t), \xi(t) < x, N(t)\} \end{aligned}$$

Following the approach by Falin and Templeton [74], the steady state equations are given by

$$\begin{aligned} (\lambda + n\eta) p_{0,n} &= \int_0^\infty p_{1,n}(x) b(x) dx \\ \frac{d}{dx} p_{1,n}(x) &= -(\lambda + b(x)) p_{1,n}(x) + \lambda p_{1,n-1}(x) \\ p_{1,n}(0) &= \lambda p_{0,n}(x) + (n+1) \eta p_{1,n+1}(x) \end{aligned}$$

recalling that $b(x)$ describes the density of service times. Introducing the generating functions $P_0(z) = \sum_{n=0}^{\infty} z^n p_{0,n}$ and $P_1(z, x) = \sum_{n=0}^{\infty} z^n p_{1,n}(x)$ the above system of equations may be written as

$$\lambda P_0(z) + \eta z \frac{d}{dz} P_0(z) = \int_0^{\infty} P_1(z, x) b(x) dx \quad (3.85)$$

$$\frac{\partial}{\partial x} P_1(z, x) = -(\lambda - \lambda z + b(x)) P_1(z, x) \quad (3.86)$$

$$P_1(z, 0) = \lambda P_0(z) + \eta \frac{d}{dz} P_0(z) \quad (3.87)$$

Solving the ordinary differential equation 3.86 yields

$$P_1(z, x) = P_1(z, 0) (1 - b(x)) e^{-(\lambda - \lambda z)x} \quad (3.88)$$

which allows one to rewrite 3.85 as

$$\lambda P_0(z) + \eta z \frac{d}{dz} P_0(z) = \bar{b}(\lambda - \lambda z) P_1(z, 0) \quad (3.89)$$

Expressing in terms $\eta \frac{d}{dz} P_0(z)$ and inserting in expression 3.87 leads to

$$P_1(z, 0) = \lambda \frac{1 - z}{\bar{b}(\lambda - \lambda z) - z} P_0(z)$$

Substituting in equation 3.88 gives

$$P_1(z, x) = \lambda \frac{1 - z}{\bar{b}(\lambda - \lambda z) - z} P_0(z) (1 - b(x)) e^{-(\lambda - \lambda z)x} \quad (3.90)$$

In order to derive an expression for $P_0(z)$ one inserts $P_1(z, 0)$ into equation 3.87 and 3.89, i.e.

$$\eta [\bar{b}(\lambda - \lambda z) - z] \frac{d}{dz} P_0(z) = \lambda [1 - \bar{b}(\lambda - \lambda z)] P_0(z) \quad (3.91)$$

By careful inspection of $\bar{b}(\lambda - \lambda z) - z$ leads to $z < \bar{b}(\lambda - \lambda z) \leq 1$. By considering the limit of $\frac{1 - \bar{b}(\lambda - \lambda z)}{\bar{b}(\lambda - \lambda z) - z}$ from below, one can show, that the convergence radius determined by the generating function may be extended to the boundary $0 \leq z \leq 1$. This in turn enables one to state expression 3.91 as ordinary differential equation

$$\frac{d}{dz} P_0(z) = \frac{\lambda}{\eta} \frac{1 - \bar{b}(\lambda - \lambda z)}{\bar{b}(\lambda - \lambda z) - z} P_0(z)$$

with solution

$$P_0(z) = P_0(1) \exp \left\{ \frac{\lambda}{\eta} \int_0^z \frac{1 - \bar{b}(\lambda - \lambda y)}{\bar{b}(\lambda - \lambda y) - y} dy \right\}$$

By noting, that $1 - P_0(1) = u = \rho := -\lambda \frac{d}{dz} \bar{b}(z) \Big|_{z=0}$, one immediately arrives at

$$P_0(z) = (1 - \rho) \exp \left\{ \frac{\lambda}{\eta} \int_0^z \frac{1 - \bar{b}(\lambda - \lambda y)}{\bar{b}(\lambda - \lambda y) - y} dy \right\} \quad (3.92)$$

Thus the steady state solution for the single server retrial system with arbitrary service times is completely characterized in terms of generating functions by expression 3.90 and 3.92. It can be shown, that the system remains stable for $u = \rho < 1$, for details refer to [74]. The distribution of the number of repeating customers is given by

$$\begin{aligned} P(z) &= P_0(z) + \int_0^\infty P_1(z, x) dx \\ &= (1 - \rho) \frac{1 - \bar{b}(\lambda - \lambda z)}{\bar{b}(\lambda - \lambda z) - z} \exp \left\{ \frac{\lambda}{\eta} \int_0^z \frac{1 - \bar{b}(\lambda - \lambda y)}{\bar{b}(\lambda - \lambda y) - y} dy \right\} \end{aligned}$$

By using the properties of the Laplace transform one arrives at the average number of customers in orbit, i.e.

$$L_q = \frac{d}{dz} P(z) \Big|_{z=1} = \frac{2\lambda\rho/\eta + \rho^2 + \lambda^2\sigma_S^2}{2(1 - \rho)} = \frac{\lambda\rho}{\eta(1 - \rho)} + L_q^{(M/G/1)}$$

As highlighted above, the average queue length exceeds the one from the classic $M/G/1$ model by an additive factor. Considering, that the retrial system is not work conserving, i.e. there is a positive probability, that waiting customers do not immediately receive service, this factor becomes intuitively clear. Furthermore by letting η approach infinity, one arrives at the classic $M/G/1$ model. Please note, that we encountered the same effect for the average queue length of the exponential retrial system given in expression 3.57. The remaining performance characteristics may be determined from an application of Little's law, i.e.

$$W_q = \frac{1}{\lambda} L_q, \quad W = W_q + \frac{1}{\mu}, \quad L = \lambda W = L_q + \rho$$

We have already seen for the classic $M/G/c$ queue, that no simple analytical solution is available for more than one server. The same is true for

the multiserver system with retrials. Fortunately the approach taken for the classic multiserver queue also works for the retrial version. Recall from equation 3.78 the approximation for the average system size:

$$L \approx c_s^2 L^{(M/M/c)} + (1 - c_s^2) L^{(M/D/c)}$$

Considering equation 3.80, the average number of customers in system for the $M/D/c$ queue may roughly approximated by $L^{(M/D/c)} \approx \frac{1}{2} L^{(M/M/c)}$ leading to

$$L \approx \frac{1 + c_s^2}{2} L^{(M/M/c)}$$

Replacing the $M/M/c$ model with its retrial counterpart results in

$$L \approx \frac{1 + c_s^2}{2} (L_q^{(rM/M/c)} + \rho)$$

where $L_q^{(rM/M/c)}$ is given by equation 3.59 for the multiserver retrial system with exponential service times. A slightly different argument for its derivation called the *processor sharing method* is given in [215]. Wolff also discusses an approximation deploying the *retrials see time averages* property for retrial models. He makes a difference between a customers initial entry and retry and assigns different probabilities to each event. With finite probability on retries the case of a finite (geometric) orbit is also covered by the model. Up to now we have only considered retrial models with exponential retry times. Note, that in relaxing this assumption, the system under consideration may become instable even with $\rho < 1$. For more details please refer to [28] or [134].

3.6 Arbitrary Arrival Processes

In the the preceding section the $M/G/1$ queue was modeled as an embedded Markov chain with regeneration points taken at the instances of service departure. Using the same technique, we are able to analyze systems with arbitrary arrivals and exponential service distribution even for multiple servers. For this non-Poisson system, the regeneration points occur at the epochs of arrival. As there is only a slight difference in derivation of the $G/M/1$ and the $G/M/c$ model, we will focus on the latter. The interarrival times are assumed to follow a general distribution $A(\cdot)$ with expectation $\frac{1}{\lambda}$.

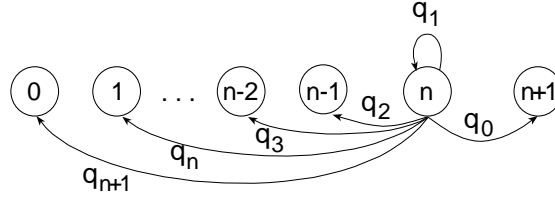


Figure 3.19: Some state transition for the G/M/c queue

3.6.1 Multiserver Systems

To obtain the matrix of transition probabilities $\mathbf{Q} = (q_{ij})$ for the embedded chain, one has to partition the (i, j) plane into four parts and consider each case separately.

- Customers do not arrive in batches, so there is obviously no transition from state $i + 1$ to state j for $i + 1 < j$, i.e.

$$q_{ij} = 0, \quad i + 1 < j$$

- Let $a(t)$ denote the density of the interarrival times. With all servers being busy, the system serves at rate $c\mu$. Defining q_n as the probability of n customers getting served within two subsequent arrivals, i.e.

$$q_n = \int_0^\infty \frac{e^{-c\mu t} (c\mu t)^n}{n!} a(t) dt$$

yields

$$q_{ij} = q_{i+1-j}, \quad c \leq j \leq i + 1$$

For a graphical representation refer to figure 3.19.

- Now consider the case, when no customers are waiting. During an interarrival period, $i + 1 - j$ service completions must occur at time t . Because of the exponentially distributed service time, the probability of a customer to depart is given by $1 - e^{-\mu t}$. Thus the probability, that a customer will remain in service is $e^{-\mu t}$. Employing the binomial distribution leads to

$$q_{ij} = \int_0^\infty \binom{i+1}{i+1-j} (1 - e^{-\mu t})^{i+1-j} e^{-\mu t j} a(t) dt, \quad c \leq i + 1 \leq j$$

By noting, that $\binom{i+1}{i+1-j} = \binom{i+1}{j}$, one may also write [123]

$$q_{ij} = \int_0^\infty \binom{i+1}{j} (1 - e^{-\mu t})^{i+1-j} e^{-\mu t j} a(t) dt, \quad c \leq i+1 \leq j$$

- It remains to discuss the case $j < c < i+1$. Following [90], assume, that arrival goes into service after some time V , when all prior customers have left. Then V is the time until $i - c + 1$ customers have been served by a full system at rate $c\mu$. This V is distributed according to a $(i - c + 1)$ -stage Erlang distribution. To get from state i to j between two subsequent arrivals, $c - j$ service departures must occur in the interval from V to the end of the interarrival period. Again utilizing the binomial distribution and using the memoryless property of the service distribution, one arrives at

$$q_{ij} = \binom{c}{c-j} \frac{(c\mu)^{i-c+1}}{(i-c)!} \int_0^\infty \int_0^t (1 - e^{-\mu(t-v)})^{c-j} e^{-\mu(t-v)j} v^{j-c} e^{-c\mu v} dv a(t) dt$$

for $c < j < i+1$.

Assuming a stable system, the stationary solution for the embedded Markov chain may be obtained by solving the linear system of equations

$$\tilde{\mathbf{p}}\mathbf{Q} = \tilde{\mathbf{p}}, \quad \sum_{i=0}^\infty \tilde{p}_i = 1 \quad (3.93)$$

Please note, that we have denoted the stationary solution by $\tilde{\mathbf{p}}$ to reflect the dependence on the arrival epochs. For $j \geq c$ this leads to

$$\tilde{p}_j = \sum_{i=0}^{j-2} 0\tilde{p}_i + \sum_{i=j-1}^\infty q_{i+1-j}\tilde{p}_i = \sum_{i=j-1}^\infty q_{i+1-j}\tilde{p}_i$$

Being familiar with the solution of the $M/M/1$ queueing model, this suggests a solution of the form

$$\tilde{p}_i = C\omega^i, \quad i \geq c \quad (3.94)$$

where ω is the root of the equation

$$z = \sum_{n=0}^\infty q_n z^n = \bar{a}(c\mu - c\mu z) \quad (3.95)$$

with \bar{a} denoting the Laplace transform of the interarrival density. Due to the assumption of a stable system, we only accept solutions from inside the unit disc, that is $|\omega| < 1$. In fact, it can be shown [196], that a unique solution exists for the case $\rho < 1$. Please note, that for Poissonian arrivals ω equals the utilization $u = \frac{\rho}{c}$ and that $u = \omega$ does not hold in general. For this reason, ω is often referred to as *generalized server occupancy* [59]. It remains to determine the constant C from the normalization condition and the first $c-1$ equations of the system 3.93. Although possible, this may not be an easy task due to the fact, that $c+1$ equations in $c+1$ unknowns containing infinite sums have to be solved. However, Gross & Harris [90] suggest to modify the term to depend on $\tilde{p}_0, \dots, \tilde{p}_c$ and ω only by the following procedure. By noting, that $1 = \sum_{i=0}^{\infty} \tilde{p}_i = \sum_{i=0}^{c-1} \tilde{p}_i + C \sum_{i=c}^{\infty} \omega^i$ one arrives at

$$C = \frac{1 - \sum_{i=0}^{c-1} \tilde{p}_i}{\sum_{i=c}^{\infty} \omega^i} = \frac{1 - \sum_{i=0}^{c-1} \tilde{p}_i}{\omega^c (1 - \omega)^{-1}}$$

Based on the above calculations one may easily obtain the average queueing time as follows. As queueing occurs only in a full system, customers are served at a constant rate $c\mu$. An arriving customer $j \geq c$ has to wait for the departure of the $j - c + 1$ preceding customers before entering service. Averaging over all possible cases yields [59][123]

$$\begin{aligned} W_q &= \frac{1}{c\mu} \sum_{i=c}^{\infty} (i - c + 1) \tilde{p}_i = \frac{C}{c\mu} \sum_{i=c}^{\infty} (i - c + 1) \omega^i \\ &= \frac{C\omega^c}{c\mu(1 - \omega)^2} \end{aligned} \quad (3.96)$$

Please note, that neither the result for W_q nor the queueing time distribution itself do depend on the arrival epochs [123]. Let the random variable \check{W}_q denote the queueing time and define A, A^c as the events {arrival queues} and {arrival does not queue}. The corresponding distribution may be written as

$$\begin{aligned} \Pr \left\{ \check{W}_q \leq w \right\} &= 1 - \Pr \left\{ \check{W}_q > w \right\} \\ &= 1 - \Pr \left\{ \check{W}_q > w | A \right\} - \Pr \left\{ \check{W}_q > w | A^c \right\} \\ &= 1 - \Pr \left\{ \check{W}_q > w | A \right\} = \Pr \left\{ \check{W}_q \leq w | A \right\} \end{aligned}$$

By summing the relevant terms, we obtain the probability of an arriving customer being delayed

$$\tilde{p}_d = \sum_{i=c}^{\infty} \tilde{p}_i = C \sum_{i=c}^{\infty} \omega^i = \frac{C\omega^c}{1-\omega}$$

From Little's law we may readily compute the average queue size from expression 3.96:

$$L_q = \lambda W_q = \frac{C\rho\omega^c}{c(1-\omega)^2}$$

Please note, that a similar result would have been obtained by applying the rate conservation law 14 to the formula $L_q = \sum_{i=c}^{\infty} (i-s) p_j$ [24]. Adding the average service time $\frac{1}{\mu}$ yields

$$W = W_q + \frac{1}{\mu}, \quad L = L_q + \rho$$

Next, consider the queue length distribution given that all servers are busy, i.e. that an arrival is delayed. Let the random variable \check{L}_q denote the queue size. Following [123],

$$\begin{aligned} \Pr \left\{ \check{L}_q = n | \text{arrival delayed} \right\} &= \frac{\tilde{p}_{c+n}}{\tilde{p}_d} = \frac{C\omega^{c+n}}{C\omega^c / (1-\omega)} \\ &= (1-\omega)\omega^n \end{aligned} \quad (3.97)$$

one has to conclude, that the conditional queue length distribution is geometric. Proceeding further, one may also determine the distribution of the queueing time [90] and the distribution of the queueing time given an arrival is delayed [123]. As shown by Kleinrock, the latter is an exponential distribution. Loosely speaking, we encountered $M/M/1$ behaviour in the conditional distributions of the $G/M/c$ queueing system. In [59] some results on the waiting time of queues with disciplines other than FCFS are presented.

3.6.2 Single Server Systems

We may now readily apply the preceding results to the single server case $c = 1$. Similar to the calculations for the $M/M/1$ queueing model one may readily obtain from expression 3.94 the probabilities

$$\tilde{p}_n = (1-\omega)\omega^n \quad (3.98)$$

The root ω is calculated from equation 3.95 given $c = 1$. As stated in [123], the number of customers found in the system by an arriving customer is geometric. Thus we find resemblance with the system size distribution of the corresponding system with exponential service times. Furthermore it can be shown [123], that the queueing time distribution has the same form as for the $M/M/1$ model. The performance characteristics are now easily calculated by substituting $C = 1 - \omega$ and $c = 1$ to the relevant expressions:

$$\begin{aligned} L_q &= \frac{\rho\omega}{(1-\omega)}, & W_q &= \frac{\omega}{\mu(1-\omega)} \\ L &= \frac{\rho}{(1-\omega)}, & W &= \frac{1}{\mu(1-\omega)} \end{aligned}$$

Please note, that the average time in system and the queueing time are the same for a random observer and an arriving customer. This is not true for the average system size and the average queue size, as can be seen by applying the rate conservation law. Rewriting the expression in theorem 14 yields $\min(c, n)p_n = \omega p_{n-1} = \rho \tilde{p}_{n-1}$. By generalizing to a random observer, we have to scale each probability by a factor $\frac{\omega}{\rho}$. Fortunately this factor carries over to the expressions for L and L_q by linearity. So a simple multiplication with $\frac{\omega}{\rho}$ yields the desired result

$$\tilde{L}_q = \frac{\omega}{\rho} L_q = \frac{\omega^2}{(1-\omega)}, \quad \tilde{L} = \frac{\omega}{\rho} L = \frac{\omega}{(1-\omega)}$$

For a detailed derivation of the $G/M/1$ queue please consult [90]. Although the models for single and multiple servers are similar in analysis, only some queueing theory textbooks cover the more general case. For further interest we refer to the literature stated in the text.

Example 19 *First consider the case of exponential service times. Then the Laplace transform of the density is given by*

$$\bar{a}(s) = \frac{\lambda}{\lambda + s}$$

Substituting $\bar{a}(s)$ and $c = 1$ in equation 3.95 leads to

$$\omega = \bar{a}(\mu - \mu\omega) = \frac{\lambda}{\lambda + \mu - \mu\omega}$$

Rearranging

$$\mu\omega^2 - (\lambda + \mu)\omega + \lambda = 0$$

and factoring

$$(\omega - 1)(\mu\omega - \lambda) = 0$$

suggests two solutions $\omega = 1$ and $\omega = \frac{\lambda}{\mu} = \rho$. We omit the first solution as it does not lie inside the unit disc. Applying the result to equation 3.98 yields the well known steady state distribution

$$\tilde{p}_n = (1 - \rho) \rho^n$$

By either applying PASTA or the rate conservation law, one immediately proves the identity of \tilde{p}_n and p_n .

Example 20 *In assuming deterministic arrivals the corresponding Laplace transform of the arrival process is given by*

$$\bar{a}(s) = e^{-s/\lambda}$$

Substituting $\bar{a}(s)$ and $c = 1$ in equation 3.95 leads to

$$\omega = \bar{a}(\mu - \mu\omega) = e^{-\frac{\mu(1-\omega)}{\lambda}} = e^{-(1-\omega)/\rho}$$

which can be numerically solved for predetermined values of ρ .

As the second example shows, even for such a intuitively simple model as the $D/M/1$ model an analytic solution becomes intractable. In most cases one resorts to the use of numeric procedures to obtain the desired result. Fortunately some approximations devoted to more general models are available and indeed suitable to approximate queues like $G/M/c$ and $D/M/1$. We shall discuss these topics below in the context of arbitrary arrivals and departures.

3.6.3 Capacity Constraints

We will now state some results on the multiserver queue with limited capacity first derived by Takacs as presented in [85]. Denoting the Laplace transform

of the interarrival density with $\bar{a}(s)$ as before the steady state probabilities seen by an arriving customer are given by

$$\begin{aligned}
\tilde{p}_n &= \begin{cases} \sum_{i=0}^{c-1} (-1)^{i-n} \binom{i}{n} v a_*(i) \sum_{j=i+1}^{\infty} \frac{w_j a_*(j)}{1 - \bar{a}(j\mu)} & 0 \leq n \leq c \\ v g_{K-n} & c < n \leq K \end{cases} \\
v &:= \left[\sum_{j=0}^{K-c} g_j + \sum_{j=1}^c \frac{w_j a_*(j)}{1 - \bar{a}(j\mu)} \right]^{-1} \\
g_n &:= \frac{1}{n!} \frac{d^n}{dz^n} \left(\frac{(1-z) \bar{a}(c\mu - c\mu z)}{\bar{a}(c\mu - c\mu z) - z} \right) \Big|_{z=0} \\
w_n &:= \begin{cases} \binom{c}{n} \left[\sum_{i=1}^{K-c+1} g_i a_{K-c+1-i} \right. \\ \quad \left. - \bar{a}(n\mu) \sum_{i=1}^{K-c} g_i \left(\frac{c}{c-n} \right)^{K-c+1-i} \right. \\ \quad \left. + a_n + \sum_{i=1}^{K-c} g_i \sum_{k=1}^{K-c} a_k \left(\frac{c}{c-n} \right)^{K-c+1-i-k} \right. \\ \quad \left. + \sum_{k=1}^{K-c} a_k \left(\frac{c}{c-n} \right)^{K-c-k} - \bar{a}(n\mu) \left(\frac{c}{c-n} \right)^{K-c} \right] & 0 \leq n < c \\ \bar{a}(c\mu) g_{K-c+1} & n = c \end{cases} \\
a_*(n) &:= \prod_{k=1}^n \frac{\bar{a}(k\mu)}{1 - \bar{a}(k\mu)}, \quad a_n := \int e^{-c\mu x} \frac{(c\mu x)^n}{n!} a(x) dx
\end{aligned}$$

From these the unconditional equilibrium distribution is easily determined by applying the rate conservation law. Using some of the functions introduced above, the average queueing time may be calculated from

$$W_q = \frac{v}{c\mu(1-v)} \sum_{n=1}^{K-c} (K-c+1-n) g_n$$

An arriving customer is turned away from the system and gets lost with probability $p_b = \tilde{p}_K$. As a consequence the effective arrival rate is readily obtained, i.e.

$$\bar{\lambda} = \lambda(1 - \tilde{p}_K)$$

Through the application of Little's law we arrive at the remaining performance key indicators in terms of W_q and $\bar{\lambda}$:

$$W = W_q + \frac{1}{\mu}, \quad L = \bar{\lambda}W, \quad L_q = \bar{\lambda}W_q$$

The special case of $K = c$ has already been treated by C. Palm in 1943 in the context of telephony. He referred to the $G/M/c/c$ queueing model as *loss*

system with recurrent input and derived a rather simple formula for the call congestion probability [85][24], i.e.

$$p_b = \hat{p}_c = \left[1 + \sum_{n=1}^{c-1} \binom{c}{n} \prod_{k=1}^n \frac{1 - \bar{a}(k\mu)}{\bar{a}(k\mu)} \right]^{-1} \quad (3.99)$$

A useful reference for the $G/M/m/K$ model is the research paper [97] by P. Hokstad. It also covers the connection between the time-continuous and the embedded process, which is just another name for the rate conservation law given by theorem 14. Hokstad managed to reduce the derivation of the steady state solution to a linear system of equations. Finally he determines performance key indicators and presents some examples.

3.7 Arbitrary Arrival and Service Processes

Arriving at the most general queueing model, it is remarkable, that still some results may be found. In the following discussion we will first focus on the single server model and then provide some approximations for the multiserver case.

3.7.1 Single Server Systems

The single server system has first been analyzed by D. Lindley in 1952. Our approach will follow the presentation in [90]. Let the random variables $\check{W}_q^{(n)}$, $\check{S}^{(n)}$, $\check{T}^{(n)}$ denote the queueing time, the service time and the interarrival time of the n -th customer. Assume, that $\check{S}^{(n)}$ and $\check{T}^{(n)}$ are mutually independent and independently indentially distributed according to $B(\cdot)$ and $A(\cdot)$, respectively. Furthermore both distribution functions shall be non negative and absolutely continuous. Consequently the densities exist and we may denote their Laplace transforms with $\bar{b}(s)$ and $\bar{a}(s)$. Introducing the random variable $\check{U}^{(n)} = \check{S}^{(n)} - \check{T}^{(n)}$ as the difference between service time and interarrival time, the Laplace transform of the density of $\check{U}^{(n)}$ may be determined as the convolution

$$\bar{u}(s) = \bar{a}(-s)\bar{b}(s) \quad (3.100)$$

Following the given notation, we may describe the queueing time for the $(n + 1)$ -th customer as

$$\check{W}_q^{(n+1)} = \begin{cases} \check{W}_q^{(n)} + \check{U}^{(n)} & \check{W}_q^{(n)} + \check{U}^{(n)} > 0 \\ 0 & \check{W}_q^{(n)} + \check{U}^{(n)} \leq 0 \end{cases} \quad (3.101)$$

Let $W_q^{(n)}(t)$ describe the distribution of the queueing time of the n -th customer. Then the waiting time distribution for the subsequent customer may be readily obtained from relation 3.101, i.e.

$$W_q^{(n+1)}(t) = \begin{cases} \int_{-\infty}^t W_q^{(n)}(t-x)u(x)dx & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (3.102)$$

Here $u(\cdot)$ denotes the density function of the random variable $\check{U}^{(n)}$. Due to the fact, that arrivals and services are independently and indentially distributed, we may write down the average arrival and service rates as $\lambda = 1/\mathbb{E}\check{T}^{(1)}$ and $\mu = 1/\check{S}^{(1)}$. Assuming $\rho = \lambda/\mu < 1$, it may be shown [49], that a steady state solution $\lim_{n \rightarrow \infty} W_q^{(n)}(t) = W_q(t)$ for the queueing time exists. Consequently, the two waiting time distributions in equation 3.102 must be identical, i.e.

$$W_q(t) = \begin{cases} \int_{-\infty}^t W_q(t-x)u(x)dx & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (3.103)$$

The above equation is often referred to as *Lindley Integral Equation* and belongs to the class of Wiener-Hopf Integral Equations. Introducing

$$W_q^-(t) = \begin{cases} 0 & t \geq 0 \\ \int_{-\infty}^t W_q(t-x)u(x)dx & t < 0 \end{cases}$$

equation 3.103 may be expressed as follows

$$W_q^-(t) + W_q(t) = \int_{-\infty}^t W_q(t-x)u(x)dx$$

Taking the Laplace transform of both sides and substituting equation 3.100 results in

$$\bar{W}_q^-(s) + \bar{W}_q(s) = \bar{W}_q(s)\bar{u}(s) = \bar{W}_q(s)\bar{a}(-s)\bar{b}(s) \quad (3.104)$$

By applying the properties of Laplace transforms [83], the Laplace transform of the unknown queueing time density $\bar{w}_q(s)$ may be expressed in terms of its distribution functions as $\bar{w}_q(s) = s\bar{W}_q(s)$. Rewriting equation 3.104

$$\bar{W}_q^-(s) + \frac{1}{s}\bar{w}_q(s) = \frac{1}{s}\bar{w}_q(s)\bar{a}(-s)\bar{b}(s)$$

finally leads to

$$\bar{w}_q(s) = \frac{s\bar{W}_q^-(s)}{\bar{a}(-s)\bar{b}(s) - 1} \quad (3.105)$$

The result stated above can not provide complete satisfaction, as it still remains to determine $\bar{W}_q^-(s)$. There exist various approaches to the solution of the $G/G/1$ queueing system, but no exact closed form solution is provided. For example, refer to the textbooks [123], [59], [30] and [215]. If a closed form solution is desired, one has to consider approximative methods. As discussed in section 3.1.4, the arrival and service distribution may be arbitrarily well approximated by the class of exponential distributions in serial and parallel, which is in turn part of the family of phase type distributions. This approach has been studied in detail by R. Schassberger in [180].

3.7.2 Multiserver Systems

The analysis of the multiserver case poses some technical difficulties, e.g. in certain situations, a stable system does not become empty. Additional assumptions are required to deal with questions like which server will serve an arriving customer in an underload situation. Main results have already been provided by J. Kiefer and J. Wolfowitz in 1955, but we will turn attention to approximations for the $G/G/c$ queueing system. One such approximation for the average queueing time is the Allen-Cunneen formula

$$W_q \approx \frac{p_d^{(M/M/c)}}{\mu(c - \rho)} \left(\frac{c_T^2 + c_S^2}{2} \right) \quad (3.106)$$

Here c_T^2 and c_S^2 describe the squared coefficient of variation of interarrival and service time. The probability $p_d^{(M/M/c)}$ is given by the Erlang Delay formula 3.24. As shown in [26], formula 3.106 is exact for the queueing systems $M/G/1$ and $M/M/c$. Another reasonable approximation may be obtained by extending the concept introduced in section 3.5.1 for the $M/G/c$ queueing system. Provided the squared coefficient of variation c_T^2 does not exceed 1, T. Kimura [121][122] suggested the expression

$$W_q \approx (c_T^2 + c_S^2) \left[\frac{1 - c_T^2}{W_q^{(D/M/c)}} + \frac{1 - c_S^2}{W_q^{(M/D/c)}} + \frac{2(c_T^2 + c_S^2 - 1)}{W_q^{(M/M/c)}} \right]^{-1} \quad (3.107)$$

where $W_q^{(M/D/c)}$ may be derived from expression 3.80 through an application of Little's law (after subtracting ρ). For $W_q^{(M/M/c)}$ one may substitute

the exact result provided by formula 3.26. The remaining term is best approximated by a relation due to Cosmetatos, Krämer and Langenbach-Belz [213]

$$W_q^{(D/M/c)} \approx \left(\frac{1}{2} - 2\varsigma(\rho, c) \right) e^{-2(c-\rho)/3\rho} W_q^{(M/M/c)}$$

where $\varsigma(\rho, c)$ is given by expression 3.81. For $c_T^2 > 1$ an approximation attributed to E. Page [161] turns out to be a suitable choice. Assembled from the same building blocks, the corresponding formula for the mean waiting time is given by

$$W_q \approx c_S^2 (1 - c_T^2) W_q^{(D/M/c)} + c_T^2 (1 - c_S^2) W_q^{(M/D/c)} + c_T^2 c_S^2 W_q^{(M/M/c)} \quad (3.108)$$

Although his considerations were limited to the analysis of $E_k/E_l/c$ queueing systems, his findings may be applied in a wider context. It turns out, that the Kimura and the Page approximation are special cases of a more general framework. For details and further improvements refer to the paper [122]. By an application of Little's law, the performance characteristics L , L_q and W are readily derived from expression 3.107 or 3.108. For some notes on the $G/G/c$ model, refer to the book [215]. The phase-type approach has been investigated by R. Schassberger in [180]. An advanced mathematical discussion covering relations between queueing processes and concepts of convergence may be found in [49].

3.7.3 Customer Impatience

The most straightforward way to build customer impatience into the $G/G/1$ model is to adapt expression 3.101 and derive a generalization of Lindley's integral equation. Without consideration of balking behaviour this leads to the so called $G/G/1 + G$ model. As before, let the random variables $\check{W}_q^{(n)}$, $\check{S}^{(n)}$, $\check{T}^{(n)}$ denote the queueing time, the service time and the interarrival time of the n -th customer. Also define $\check{U}^{(n)} = \check{S}^{(n)} - \check{T}^{(n)}$ with the usual meaning. A new random variable $\check{G}^{(n)}$ shall describe the patience time of the n -th customer. In order to model impatience effects, a potential customer refuses to join the queue, if $\check{G}^{(n)} \leq \check{W}_q^{(n)}$. On the other hand, for $\check{G}^{(n)} > \check{W}_q^{(n)}$

he enters the system to get served. Taking this into account, one arrives at

$$\check{W}_q^{(n+1)} = \begin{cases} \check{W}_q^{(n)} + \check{U}^{(n)} & \check{W}_q^{(n)} + \check{U}^{(n)} > 0, \check{G}^{(n)} > \check{W}_q^{(n)} \\ 0 & \check{W}_q^{(n)} + \check{U}^{(n)} \leq 0, \check{G}^{(n)} > \check{W}_q^{(n)} \\ \check{W}_q^{(n)} - \check{T}^{(n)} & \check{W}_q^{(n)} - \check{T}^{(n)} > 0, \check{G}^{(n)} \leq \check{W}_q^{(n)} \\ 0 & \check{W}_q^{(n)} - \check{T}^{(n)} \leq 0, \check{G}^{(n)} \leq \check{W}_q^{(n)} \end{cases}$$

Introducing the survival function $\bar{G}(\cdot) = 1 - G(\cdot)$ for the distribution $G(\cdot)$ of the impatience time and proceeding as before leads to

$$W_q(t) = \begin{cases} \int_{-\infty}^t W_q(t-x) [\bar{G}(x)u(x) + (1 - \bar{G}(x))a(-x)] dx & t \geq 0 \\ 0 & t < 0 \end{cases}$$

Here $u(\cdot)$ and $a(\cdot)$ denote the density functions of the random variables $\check{T}^{(n)}$ and $\check{U}^{(n)}$, respectively. If the distribution of impatience times $G(\cdot)$ is non-defective, that is for $\lim_{x \rightarrow \infty} G(x) = 1$ the system remains stable for $\rho < 1$. For details on a similar derivation we refer to the paper of F. Bacelli and G. Hebuterne [34]. An approximative solution to a $G/G/1$ queueing system with balking and reneging is given in [211].

3.8 Matrix Analytic Solutions

The matrix analytic approach is rooted in the work of M. Neuts [153]. He was mainly concerned with finding some sort of structure in Markov chains to allow for optimized algorithms. As an example, consider the infinitesimal generator 3.44 in section 3.3.7 and think of each scalar being replaced by a matrix. This immediately leads to what is called a *quasi birth-death process*. Credits for this special process go to V. Wallace, who found solutions even 12 years before Neuts has published his book and managed to provide solutions for a broader class of models. Furthermore he developed the necessary mathematical theory and defined the family of phase type distributions. Recently, these distributions have been generalized to *batch markov arrival processes (BMAP)* and *matrix exponential distributions*. The former approach allows for incorporation of dependency effects and batch arrivals whereas the latter allows for representation of distributions with rational Laplace-Stieltjes transformation (Think of the Laplace-Stieltjes transformation of an absolutely continuous non-negative distribution function as

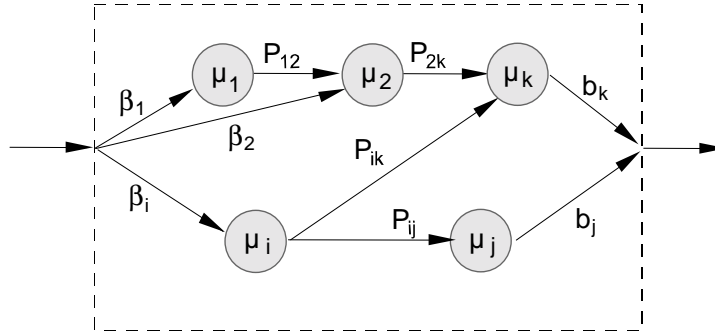


Figure 3.20: Typical phase type service facility

the Laplace transform of its density function). It seems that researchers focusing on BMAPs are rooted deeply in Markov chain theory and aim to provide optimal numerical procedures for the solution of their models. The works of G. Latouche, V. Ramaswami [138] and D. Bini, G. Latouche, B. Meini [44] are falling into this category. The other school is more rooted in queueing theory and linear algebra calling their approach *linear algebraic queueing theory (LAQT)*. Centered around researchers such as L. Lipsky [142] and A. van de Liefvoort [150], focus lies on the solution of models adhering to matrix exponential arrival and service patterns. Although not as widespread as the aforementioned school, we will adopt the LAQT approach here, as for simpler models it is more accessible to practitioners. It has to be noted, that none of these schools works isolated and that best progress is made only in considering both approaches. Following that tradition, we'll also include some Markovian results.

3.8.1 Distribution Theory

Like other staged distributions such as the Erlang or the hyperexponential distribution we may characterize matrix exponential and phase type distributions in terms of density, distribution function and Laplace transform of the former. However, the more appealing approach is to attempt a probabilistic interpretation of the corresponding parameters. Although this perspective is absolutely valid in the context of phase type distributions it is too limited for the family of matrix exponential distributions. This requires a more abstract viewpoint and will be elaborated on later.

For the moment let us consider a service facility consisting of k arbitrarily interconnected nodes similar to the setup of a Markov chain. Let \mathbf{M} denote the *matrix of service rates*, i.e. $\mathbf{M} := \text{diag}(\mu_1, \mu_2, \dots, \mu_k)$. Furthermore assume an *entry vector* $\boldsymbol{\beta} := (\beta_1, \beta_2, \dots, \beta_k)$, an *exit vector* $\mathbf{b} := (b_1, b_2, \dots, b_k)^T$ and a (substochastic) transition matrix $\mathbf{P} := (P_{ij})$. In putting all the pieces together one arrives at a phase type service facility with a configuration similar to the one shown in figure 3.20. Obviously the network of phases defines a continuous time Markov chain, so we may utilize the relation between probability transition matrix and infinitesimal generator as explained in appendix A.3.2. Incorporating service rates as well, this leads to the *service rate matrix*

$$\mathbf{B} := \mathbf{M}(\mathbf{P} - \mathbf{I}) \quad (3.109)$$

Note, that this is only a part of the infinitesimal generator, as we are still missing the exit vector. Completion leads to a transient Markov chain with absorbing state described by

$$\bar{\mathbf{Q}} = \begin{pmatrix} \mathbf{B} & \mathbf{b} \\ \mathbf{0} & 0 \end{pmatrix}$$

where \mathbf{b} may be expressed as $\mathbf{b} := -\mathbf{B}\mathbf{1}$ due to the stochastic nature of $\bar{\mathbf{Q}}$ [32]. Adopting conventional notation, the term $\mathbf{1}$ is used to describe a column vector of ones. Whenever service is requested from the facility, the transient Markov chain is entered. Each stage in this chain may be envisaged as a task of the current work cycle. In reaching the absorbing state, service is completed and the facility becomes ready to accept a new request. In mathematical terms this is called a *renewal process*.

As suggested by the name, matrix exponential distributions are defined in terms of *matrix exponential (function)*. Simply speaking, the matrix exponential is a matrix function defined by the series expansion $\exp(x) := \sum_{n=0}^{\infty} \frac{1}{n!} x^n$ of its scalar counterpart. In other words, the scalar argument of an ordinary function are replaced by a matrix. Obviously these functions have to be handled with care when approaching concepts such as convergence and differentiation. We will provide only a rough overview here, for more details refer to the book by F.R. Gantmacher [81][82].

Definition 21 *The matrix exponential function is defined as*

$$e^{\mathbf{A}} := \exp \{ \mathbf{A} \} := \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{A}^n$$

where \mathbf{A} is a finite square matrix of complex values.

With such powerful tools in hands, we are ready to specify density and distribution function of a phase type distribution

Definition 22 *The distribution and density function of a phase type distribution are given by*

$$F(x) := \begin{cases} \beta_{k+1} & x = 0 \\ 1 - \boldsymbol{\beta} \exp \{ \mathbf{B}x \} \mathbf{1} & x > 0 \end{cases} \quad (3.110)$$

and

$$f(x) := \boldsymbol{\beta} \exp \{ \mathbf{B}x \} \mathbf{b} \quad x > 0 \quad (3.111)$$

with $\mathbf{b} := -\mathbf{B}\mathbf{1}$ provided $(\mathbf{P} - \mathbf{I})^{-1}$ exists and

$$\mathbf{M}_{ii} = \mu_i > 0, \quad P_{ij} \geq 0, \quad \sum_{j=1}^k P_{ij} \leq 1$$

are satisfied for all i, j . Furthermore it is assumed, that $\sum_{i=1}^{k+1} \beta_i = \boldsymbol{\beta}\mathbf{1} + \beta_{k+1} = 1$. The corresponding family of distributions will be denoted as $PH(\boldsymbol{\beta}, \mathbf{B})$.

Please note, that Neuts has given a slightly more general definition by allowing for non-transient states. If a non-singular matrix \mathbf{B} is chosen, it can be shown [153], that all states $1, \dots, k$ are transient. This in turn follows from the existence of $(\mathbf{P} - \mathbf{I})^{-1}$.

As a matter of fact, the probabilistic interpretation previously given is also reflected by the conditions of this definition. In relaxing these conditions we arrive at the more general class of matrix exponential distributions. One simply accepts a parameter tuple $(\boldsymbol{\beta}, \mathbf{B})$ as a matrix exponential representation, if one is able to construct a valid density or distribution function from it. To be more specific we first need to define what is understood by the term distribution function, i.e.

Definition 23 *We call a function F distribution function of a non-negative random variable, if $F(x)$ is non-negative and non-decreasing for all $x \geq 0$, F is right-continuous (i.e. the condition $\lim_{h \rightarrow 0^+} F(x+h) = F(x)$ is satisfied for all $x \geq 0$) and $\lim_{x \rightarrow \infty} F(x) = 1$.*

Now we are ready to give a characterization of the family of matrix exponential distributions, i.e.

Definition 24 *If an arbitrary parameter tuple $(\boldsymbol{\beta}, \mathbf{B})$ admits a probability distribution of the form 3.110, this tuple generates a matrix exponential distribution with representation $ME(\boldsymbol{\beta}, \mathbf{B})$.*

Please note, that we also could have checked, if $\boldsymbol{\beta}$ and \mathbf{B} substituted in expression 3.111 yield a proper density function. In fact, there exist several equivalent methods to verify, if a certain tuple $(\boldsymbol{\beta}, \mathbf{B})$ generates a valid matrix exponential distribution [142][73]. Obviously, each phase type distribution is a member of the class of matrix exponential distributions but not vice versa. The idea of using negative or complex values instead of proper probabilities in the context of staged distributions dates back to 1955 and has first been considered by D.R. Cox [61].

In the same context D.R. Cox showed, that distributions, where the related densities possess a *rational Laplace transformation (RLT)* may be represented in terms of a staged distribution. The same concept also carries over to the family of matrix exponential distributions [142][73]. First consider the Laplace transformation of the matrix exponential density function 3.111

$$\begin{aligned}\bar{f}(u) &= \int_0^\infty e^{-ux} f(x) dx = \boldsymbol{\beta} (u\mathbf{I} - \mathbf{B})^{-1} \mathbf{b} + \beta_{k+1} \\ &= -\boldsymbol{\beta} (u\mathbf{I} - \mathbf{B})^{-1} \mathbf{B} \mathbf{1} + \beta_{k+1} \\ &= -\boldsymbol{\beta} (\mathbf{uB}^{-1} - \mathbf{I})^{-1} \mathbf{1} + \beta_{k+1}\end{aligned}\tag{3.112}$$

Assume an arbitrary density function $g(\cdot)$ with rational Laplace transform, i.e.

$$\bar{g}(u) = \frac{\sum_{j=1}^k \beta_j u^{j-1}}{u^m + \sum_{j=1}^k \alpha_j u^{j-1}} + \beta_{k+1}$$

where $k \geq 1$, $0 \leq \beta_{k+1} \leq 1$ and the coefficients β_j and α_j are real values for $1 \leq j \leq k$. Then the corresponding matrix exponential distribution has a

representation $ME(\boldsymbol{\beta}, \mathbf{B})$, where

$$\begin{aligned} \boldsymbol{\beta} &= (\beta_1, \beta_2, \dots, \beta_k) \\ \mathbf{B} &= \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -\alpha_1 & -\alpha_2 & -\alpha_3 & \dots & -\alpha_{k-1} & -\alpha_k \end{pmatrix} \end{aligned} \quad (3.113)$$

This special representation is also known as *companion form representation*. We now state some details about phase type and matrix exponential distributions without proof. For details please refer to the thesis [73] by M.W. Fackrell.

- Phase type and matrix exponential representations are not unique. It is possible to construct representations of different structure and dimension, which lead to the same distribution. Please note, that the dimension or order of a representation directly relates to the number of stages.
- Every matrix exponential distribution has a unique *minimal representation* $ME(\boldsymbol{\beta}, \mathbf{B})$, that is a representation of minimal order, where $\boldsymbol{\beta}$ and \mathbf{B} are given by expression 3.113. However, this problem still remains unsolved for phase type distributions.
- Every phase type / matrix exponential distribution has a representation $ME(\boldsymbol{\beta}, \mathbf{B})$, where $\beta_{k+1} = 0$. This representation is not necessarily of minimal order.
- There do exist matrix exponential distributions, which do not belong to the family of phase type distributions. One such example has first been constructed by C.A. O'Cinneide.

Returning to our view of a service facility, let us now approach the question of how long a potential customer has to wait for service completion. Following [142], a customer served by node i resides for a period of $\frac{1}{\mu_i} = (\mathbf{M}^{-1}\mathbf{1})$. Then he either leaves the facility with probability b_i or proceeds to phase j

with probability P_{ij} . This leads to a recursion for the vector of average service times conditioned on the starting phase of the customer, i.e.

$$\mathbf{s} := (s_1, s_2, \dots, s_k) = \mathbf{M}^{-1}\mathbf{1} + \mathbf{P}\mathbf{s}$$

Rearranging yields

$$\mathbf{s} = (\mathbf{I} - \mathbf{P})\mathbf{M}^{-1}\mathbf{1} = [\mathbf{M}(\mathbf{I} - \mathbf{P})]^{-1}\mathbf{1} = -\mathbf{B}^{-1}\mathbf{1}$$

To determine the average time spent in the service facility, one has to incorporate the chance β_i of a customer starting at node i , i.e.

$$s = \sum_{j=1}^k \beta_j s_j = \boldsymbol{\beta}\mathbf{s} = -\boldsymbol{\beta}\mathbf{B}^{-1}\mathbf{1} \quad (3.114)$$

Because of its importance we further proceed similar to Lipsky in [142] and define the operator

$$\Psi[\mathbf{X}] := \boldsymbol{\beta}\mathbf{X}\mathbf{1}$$

for any square matrix \mathbf{X} . It transforms a square matrix into a scalar value of interest. The meaning of this operator is obvious from expression 3.114 and it will be of great help in deriving key indicators for the entire service facility. In a similar fashion we may use the operator Ψ to provide an alternative formula for the Laplace transform 3.112

$$\bar{f}(u) = -\Psi \left[(u\mathbf{B}^{-1} - \mathbf{I})^{-1} \right] + \beta_{k+1} \quad (3.115)$$

Note, that similar representations do also exist for the description of the arrival process.

In determining the time a potential customer has to wait until service completes we have done nothing else than to derive the first moment of the matrix exponential distribution with representation $ME(\boldsymbol{\beta}, \mathbf{B})$. In order to arrive at a more general expression for all moments, it is more convenient to apply methods of transform theory. Differentiating expression 3.112 n times with respect to u and letting u approach 0 leads to the n -th moment of a matrix exponentially distributed random variable \check{T}

$$\mathbb{E}\check{T}^n = (-1)^{n+1} n! \boldsymbol{\beta} \mathbf{B}^{-(n+1)} \mathbf{b} = n! \Psi \left[(-\mathbf{B}^{-1})^n \right] \quad (3.116)$$

From these moments, the expression for mean time and variance are easily calculated. For the former compare expressions 3.114 and 3.116 to get an

impression of the conclusiveness of the above results. For the proofs and more information we refer to the books by G. Latouche and V. Ramaswami [138], M. Neuts [153], L. Lipsky [142] and W. Fackrell [73].

From a philosophical viewpoint we may distinguish between physical and fictitious nodes. Physical nodes resemble real entities such as the stations along a conveyor belt. Fictitious nodes are only included in the model to reach a predetermined analytic goal such as the exactness in fitting an arbitrary distribution. We have already learned in section 3.1.5, that such general distributions may be well approximated by the family of generalized Erlang distributions, which is in turn a subclass of the family of phase type distributions. Leaving the context of service facilities, phase type distributions also prove to be useful in the description of arrival processes. Usually there is no physical resemblance of nodes here, so one has to rely on purely fictitious nodes. Some of the methods available for approximation of arbitrary distribution functions will be investigated later in chapter 4.

As explained in section 3.1.4 all types of staged distributions belong to the class of phase type / matrix exponential distributions. We will now again turn attention to the Erlang and hyperexponential distribution from the just presented viewpoint.

Example 25 *Considering the Erlang distribution (also refer to figure 3.11 for a graphical representation) as a sequential service facility, one immediately arrives at the transition matrix*

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

Each node in the service facility operates at rate $k\mu$, so $\mathbf{M} = k\mu\mathbf{I}$. Using expression 3.109, this leads to

$$\mathbf{B} = \mathbf{M}(\mathbf{P} - \mathbf{I}) = \begin{pmatrix} -k\mu & k\mu & 0 & \cdots & 0 & 0 \\ 0 & -k\mu & k\mu & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -k\mu & k\mu \\ 0 & 0 & 0 & \cdots & 0 & -k\mu \end{pmatrix}$$

Each arriving customer starts service at the first node, i.e. $\beta = (1, 0, \dots, 0)$. Inserting all values in expression 3.111 yields

$$\begin{aligned} f(x) &= \beta \exp \{ \mathbf{B}x \} \mathbf{b} = \sum_{n=0}^{\infty} \frac{x^n}{n!} \beta \mathbf{B}^n \mathbf{b} \\ &= \sum_{n=k-1}^{\infty} \binom{n}{k-1} (-1)^{n+k+1} (k\mu)^{n+1} \frac{x^n}{n!} \\ &= \frac{(\mu k)^k}{(k-1)!} x^{k-1} e^{-k\mu x}, \quad x > 0 \end{aligned}$$

which is the density function of an Erlang distribution (compare with section 3.1.4). Note, that in the above calculation we made use of the fact, that the elements of \mathbf{B}^n different from zero are given by $B_{i,i+j}^n = \binom{n}{j} (-1)^{n+j} (k\mu)^n$ for all $1 \leq i \leq i+j \leq k$, $n \geq 0$ [138].

Example 26 Now we turn attention to the hyperexponential distribution with density function $f(x) = \sum_{i=1}^k \beta_i \mu_i e^{-\mu_i x}$. Again we treat the system under consideration as service facility. The i -th stage will be entered by an arriving customer with probability β_i providing service at rate μ_i . This is a purely parallel arrangement described by the parameters $\beta = (\beta_1, \beta_2, \dots, \beta_k)$ and $\mathbf{M} = \text{diag}(\mu_1, \mu_2, \dots, \mu_k)$. The transition matrix $\mathbf{P} = \mathbf{0}$, because there are no transitions between internal stages. Proceeding as for the Erlang distribution one arrives at

$$\mathbf{B} = \mathbf{M}(\mathbf{P} - \mathbf{I}) = -\mathbf{M} = \text{diag}(-\mu_1, -\mu_2, \dots, -\mu_k)$$

Using the series expansion of the (matrix) exponential function

$$\begin{aligned} e^{\mathbf{B}x} &= \sum_{n=0}^{\infty} \frac{x^n}{n!} \mathbf{B}^n \\ &= \text{diag} \left(\sum_{n=0}^{\infty} \frac{(-\mu_1 x)^n}{n!}, \sum_{n=0}^{\infty} \frac{(-\mu_2 x)^n}{n!}, \dots, \sum_{n=0}^{\infty} \frac{(-\mu_k x)^n}{n!} \right) \\ &= \text{diag} (e^{-\mu_1 x}, e^{-\mu_2 x}, \dots, e^{-\mu_k x}) \end{aligned}$$

immediately proves the phase type representation of the hyperexponential dis-

tribution, i.e.

$$\begin{aligned}
 f(x) &= \boldsymbol{\beta} \exp \{ \mathbf{B}x \} \mathbf{b} \\
 &= \boldsymbol{\beta} \exp \{ \mathbf{B}x \} \text{diag}(\mu_1, \mu_2, \dots, \mu_k) \\
 &= \sum_{i=1}^k \beta_i \mu_i e^{-\mu_i x}
 \end{aligned}$$

for $x > 0$. Also note, that $\beta_{k+1} = 0$ for the hyperexponential distribution.

One way of constructing a more general phase type distribution is by arranging exponential phases in serial and parallel fashion. Assuming equal service times for the phases in each branch, this results in the class of hyper-Erlang distributions. Think of each exponential node in a hyperexponential service facility being replaced by an Erlang facility. By reusing the matrices derived in the examples above and concatenating them in appropriate order, one easily derives the service rate matrix \mathbf{B} and the entry vector $\boldsymbol{\beta}$ as follows

$$\begin{aligned}
 \boldsymbol{\beta} &= (\beta_1 \boldsymbol{\beta}^{(1)}, \beta_2 \boldsymbol{\beta}^{(2)}, \dots, \beta_k \boldsymbol{\beta}^{(k)}) \\
 \mathbf{B} &= \begin{pmatrix} \mathbf{B}^{(1)} & 0 & \dots & 0 \\ 0 & \mathbf{B}^{(2)} & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & \mathbf{B}^{(k)} \end{pmatrix}
 \end{aligned}$$

Indeed, these parameters representing the macro level of a hierarchical configuration are very similar to the ones presented in example 26 for the hyperexponential distribution. Each scalar has been replaced by a matrix, which describes the corresponding Erlangian sublevel. The parameters $\boldsymbol{\beta}^{(i)}$ and $\mathbf{B}^{(i)}$ are of similar structure as the ones given in example 25 for the Erlang distribution. The main difference between sublevels is their dimension, i.e. $\mathbf{B}^{(i)}$ is a $n_i \times n_i$ matrix and $\boldsymbol{\beta}^{(i)}$ holds exactly n_i elements where $1 \leq i \leq k$. The corresponding distribution function is given by

$$f(x) = \sum_{i=1}^k \beta_i \lambda^{n_i} \frac{x^{n_i-1}}{(n_i-1)!} e^{-\lambda x}$$

for $x > 0$, $n_i > 0$ and $\sum_{i=1}^k \beta_i = 1$. Please note the alternative description of the Erlang distribution, as each exponential phase now operates with rate

$\lambda = k\mu$. The class of hyper-Erlang distributions has been extensively studied by R. Schassberger in [180]. By relaxing the condition of having equal service rates for the entire model, one arrives at the generalized Erlang distribution. Both distributions are very special cases of an embedding of one matrix exponential distribution into another.

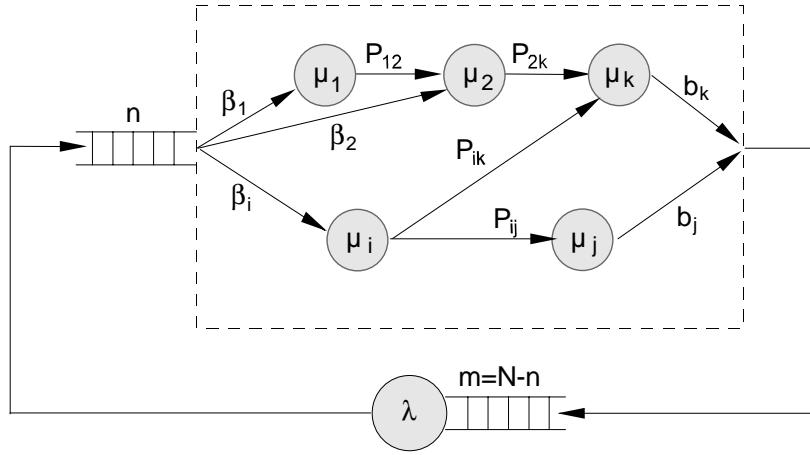
3.8.2 Single Server Systems

We will now turn attention to queueing models with phase type / matrix exponential distribution and discuss the single server queues $M/ME/1$ and $ME/M/1$ in some detail. Note, that there is a relation between the former model and queueing networks, so one might ask, why not use them instead. In fact, we will also discuss systems with limited source supply, where queueing networks do not provide exact results. This is different for linear algebraic queueing theory (LAQT) as introduced by Lipsky.

Poisson Arrivals

As the analysis of the $M/ME/1$ queueing system relies heavily on matrix theory, one better starts analyzing the system under source constraints, that is the $M/ME/1//N$ model. Then we are ensured only to encounter finite matrices during our calculations. Systems with a limited number of resources may also be viewed as being closed. This is also a common strategy in the analysis of queueing networks. Customers in a closed system circulate through a service *loop* as shown in figure 3.21. As an example, consider terminal usage. The time of no user input called *think time* is represented by an exponential server with rate λ , whereas mainframe operation is modeled by a matrix exponential service node. To stress the example further, each stage may represent a certain component such as CPU, harddisc, communication controller and I/O.

The derivation of the equilibrium solution is approached the usual way by solving the balance equations under the assumption of a steady state. The main difference is, that the steady state probabilities are now represented by vectors $\boldsymbol{\pi}_n := (\pi_{n,1}, \pi_{n,2}, \dots, \pi_{n,k})$ to reflect the phased configuration. But we will also define a scalar version, i.e. the probability of n customers in the service facility and denote it by $p_n := \boldsymbol{\pi}_n \mathbf{1} = \sum_{i=1}^k \pi_{n,i}$. For analytical purposes we also define the vector $\boldsymbol{\pi}_0 := p_0 \boldsymbol{\beta}$. We are now ready to write down the balance equations for the $M/ME/1//N$ queueing model. Note,

Figure 3.21: The $M/ME/1/N$ service loop

that we could have chosen the $M/ME/1/N$ system with capacity limitation as well to arrive at the same set of equations. Let the tuple (i, n) denote the state of the service facility where i is the phase occupied by the customer in service and n is the number of customers residing in the service facility, i.e. the customers waiting plus 1. Obviously the phase has no meaning when $n = 0$. The probability of entering a state must equal the probability of leaving it, so

$$\lambda p_0 = \sum_{i=1}^k \pi_{1,i} b_i = \boldsymbol{\pi}_1 \mathbf{b} = -\boldsymbol{\pi}_1 \mathbf{B} \mathbf{1}$$

Multiplying the above term with $\boldsymbol{\beta}$ from the right results in

$$\lambda \boldsymbol{\pi}_0 = -\boldsymbol{\pi}_1 \mathbf{B} \mathbf{1} \boldsymbol{\beta} \quad (3.117)$$

Given N customers already inhabit the service facility, no arrivals can occur and only an internal transition would lead to the same state. With one customer less, an arrival would raise the number of customers to N . Adding

both contributions yields

$$\begin{aligned}
\pi_{N,i}\mu_i &= \lambda\pi_{N-1,i} + \sum_{j=1}^k \pi_{N,j}\mu_j P_{ji} \\
\pi_N \mathbf{M} &= \pi_{N-1} \lambda \mathbf{I} + \pi_N \mathbf{M} \mathbf{P} \\
\pi_N \mathbf{M} (\mathbf{I} - \mathbf{P}) &= \pi_{N-1} \lambda \mathbf{I} \\
\pi_N &= -\lambda \pi_{N-1} \mathbf{B}^{-1}
\end{aligned} \tag{3.118}$$

The balance equations for $0 < n < N$ combine all the contributions derived for the boundary equations,

$$\begin{aligned}
\pi_{n,i}(\mu_i + \lambda) &= \sum_{j=1}^k \pi_{n,j}\mu_j P_{ji} + \sum_{j=1}^k \pi_{n+1,j}b_j\beta_i + \pi_{n-1,i}\lambda\mathbf{I} \\
\pi_n(\mathbf{M} + \lambda\mathbf{I}) &= \pi_n \mathbf{M} \mathbf{P} + \pi_{n+1} \mathbf{b} \boldsymbol{\beta} + \pi_{n-1} \lambda \mathbf{I}
\end{aligned}$$

Rearranging as before leads to

$$\pi_n(\lambda\mathbf{I} - \mathbf{B}) = \pi_{n-1}\lambda\mathbf{I} - \pi_{n+1}\mathbf{B}\mathbf{1}\boldsymbol{\beta} \tag{3.119}$$

for $0 < n < N$. It turns out, that this system of equations is best solved from the back. Setting $n = N - 1$ in equation 3.119 and substituting expression 3.119 for π_N gives

$$\begin{aligned}
\pi_{N-1}(\lambda\mathbf{I} - \mathbf{B}) &= \pi_{N-2}\lambda\mathbf{I} + \lambda\pi_{N-1}\mathbf{B}^{-1}\mathbf{B}\mathbf{1}\boldsymbol{\beta} \\
&= \pi_{N-2}\lambda\mathbf{I} + \lambda\pi_{N-1}\mathbf{1}\boldsymbol{\beta} \\
\pi_{N-1}(\lambda\mathbf{I} - \mathbf{B} - \lambda\mathbf{1}\boldsymbol{\beta}) &= \pi_{N-2}\lambda\mathbf{I} \\
\pi_{N-1}\left(\mathbf{I} - \frac{1}{\lambda}\mathbf{B} - \mathbf{1}\boldsymbol{\beta}\right) &= \pi_{N-2}
\end{aligned}$$

By defining an auxiliary matrix $\mathbf{R} := (\mathbf{I} - \frac{1}{\lambda}\mathbf{B} - \mathbf{1}\boldsymbol{\beta})^{-1}$, the above expression may be simplified to

$$\pi_{N-1} = \pi_{N-2}\mathbf{R}$$

Indeed, the same is true for arbitrary n . For a rigorous proof based on the induction principle, we refer to [142]. Summarizing the above results, the steady state probabilities are given as follows

$$\begin{aligned}
\pi_n &= \pi_0 \mathbf{R}^n = p_0 \boldsymbol{\beta} \mathbf{R}^n \\
\pi_N &= -\lambda \pi_{N-1} \mathbf{B}^{-1} = -\lambda p_0 \boldsymbol{\beta} \mathbf{R}^{N-1} \mathbf{B}^{-1}
\end{aligned}$$

By careful inspection of the results one has to note the similarity to the solution of the $M/M/1$ and $G/M/1$ queueing systems. In fact, all of them have the same geometric structure (geometric as in the *geometric series* $1 + x + x^2 + \dots$), but now with the matrix \mathbf{R} instead of the scalars ρ or ω . This is why phase type models are said to possess a *matrix geometric solution*. Also note, that the matrix \mathbf{R} is an instance of what is called *rate matrix* or *Neuts' matrix*. But the similarities do not end at this point, even the probability of no customers in the service facility p_0 may be calculated the same way, that is by normalization. Proceeding leads to

$$\begin{aligned} 1 &= \sum_{n=0}^N p_n = \sum_{n=0}^N \pi_n \mathbf{1} \\ &= -\lambda p_0 \beta \mathbf{R}^{N-1} \mathbf{B}^{-1} \mathbf{1} + \sum_{n=0}^{N-1} p_0 \beta \mathbf{R}^n \mathbf{1} \\ &= p_0 \beta \left(-\lambda \mathbf{R}^{N-1} \mathbf{B}^{-1} + \sum_{n=0}^{N-1} \mathbf{R}^n \right) \mathbf{1} \end{aligned}$$

and

$$p_0 = \left[\beta \left(-\lambda \mathbf{R}^{N-1} \mathbf{B}^{-1} + \sum_{n=0}^{N-1} \mathbf{R}^n \right) \mathbf{1} \right]^{-1}$$

This result may be further simplified by applying the geometric series expansion

$$\sum_{n=0}^{N-1} \mathbf{R}^n = (\mathbf{I} - \mathbf{R}^N) (\mathbf{I} - \mathbf{R})^{-1}$$

to the *normalization constant*

$$\begin{aligned} \mathbf{K} &= -\lambda \mathbf{R}^{N-1} \mathbf{B}^{-1} + \sum_{n=0}^{N-1} \mathbf{R}^n \\ &= -\lambda \mathbf{R}^{N-1} \mathbf{B}^{-1} + (\mathbf{I} - \mathbf{R}^N) (\mathbf{I} - \mathbf{R})^{-1} \end{aligned} \tag{3.120}$$

provided that all eigenvalues of \mathbf{R} are strictly less than 1. As mentioned in [153], this must not always be the case. Alternatively one may derive a recursion formula for \mathbf{K} as shown in [142]. Now denote $\mathbf{K}(N) := \mathbf{K}$ with an emphasis on the number of sources. By expanding the sum in expression

3.120 one immediately arrives at

$$\begin{aligned}
 \mathbf{K}(N) &= -\lambda \mathbf{R}^{N-1} \mathbf{B}^{-1} + \mathbf{I} + \mathbf{R} + \mathbf{R}^2 + \dots + \mathbf{R}^{N-1} \\
 &= \mathbf{I} + \mathbf{R} (-\lambda \mathbf{R}^{N-1} \mathbf{B}^{-1} + \mathbf{I} + \mathbf{R} + \mathbf{R}^2 + \dots + \mathbf{R}^{N-2}) \\
 &= \mathbf{I} + \mathbf{R} \mathbf{K}(N-1)
 \end{aligned} \tag{3.121}$$

and

$$\mathbf{K}(1) = \mathbf{I} - \lambda \mathbf{B}^{-1}$$

In any case the probability of no customers at the service facility is then given by

$$p_0 = [\beta \mathbf{K} \mathbf{1}]^{-1} = \frac{1}{\Psi[\mathbf{K}]}$$

We may now apply some well known formulas to find the relevant performance characteristics

$$\begin{aligned}
 L &= \sum_{n=1}^N n p_n = \frac{1}{\Psi[\mathbf{K}]} \sum_{n=1}^N n \beta \mathbf{R}^n \mathbf{1} \\
 W &= \frac{1}{\lambda} L \\
 W_q &= W - s = W + \Psi[\mathbf{B}^{-1}] \\
 L_q &= \lambda W_q
 \end{aligned}$$

where s denotes the average service time as given by equation 3.114. For more information on the $M/ME/1//N$ queueing system, please refer to the books by Lipsky [142] and Neuts [153]. Especially the former author gives a very detailed treatment of almost all aspects of the system.

By letting the number of sources N approach infinity, i.e. $N \rightarrow \infty$, our closed system loop becomes an open $M/ME/1$ model. Careful inspection of the steady state equations shows, that only minor modifications are necessary. In fact, only the normalization constant \mathbf{K} has to be considered. Instead of performing rather tedious matrix manipulations, we recall a result for the $M/G/1$ queueing system. Expression 3.64 states, that the probability of no customer present at the service facility is given by

$$p_0 = 1 - \rho = 1 - \lambda s = 1 + \lambda \Psi[\mathbf{B}^{-1}]$$

with s the average service time as determined by equation 3.114. This leads to the steady state distribution for the $M/ME/1$ queueing model,

$$\begin{aligned}
 \pi_n &= p_0 \beta \mathbf{R}^n = (1 - \rho) \beta \mathbf{R}^n = (1 + \lambda \Psi[\mathbf{B}^{-1}]) \beta \mathbf{R}^n \\
 p_n &= p_0 \beta \mathbf{R}^n \mathbf{1} = (1 - \rho) \Psi[\mathbf{R}^n] = (1 + \lambda \Psi[\mathbf{B}^{-1}]) \Psi[\mathbf{R}^n]
 \end{aligned}$$

Note, that we have assumed $\beta_{k+1} = 0$ to simplify notation. If a customer has the chance to bypass the service facility, rescaling may be necessary. For a rigorous treatment in the matrix geometric sense, we refer to [142] and [153]. In a similar way we may utilize the Pollaczek-Khintchine formula 3.71 of section 3.5.1,

$$L = \rho + \frac{\rho^2 + \lambda^2 \sigma_S^2}{2(1 - \rho)} = \rho + \frac{\lambda^2 \mathbb{E} \check{S}^2}{2(1 - \rho)}$$

where \check{S} denotes the random variable service time with second moment given by expression 3.116 for $n = 2$,

$$\begin{aligned} L &= \lambda s + \frac{\lambda^2 \Psi \left[(\mathbf{B}^{-1})^2 \right]}{1 - \lambda s} \\ &= \frac{\lambda^2 \Psi \left[(\mathbf{B}^{-1})^2 \right]}{1 + \lambda \Psi \left[\mathbf{B}^{-1} \right]} - \lambda \Psi \left[\mathbf{B}^{-1} \right] \end{aligned}$$

The remaining performance characteristics may be easily derived using Little's Law in the usual way, i.e.

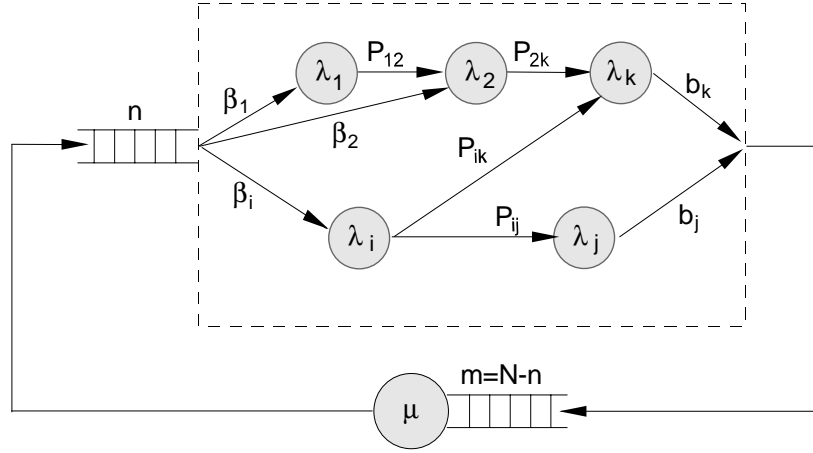
$$W = \frac{1}{\lambda} L, \quad W_q = W - s, \quad L_q = \lambda W_q$$

Most of the results above are stated in terms of matrix inverses, which are not warranted to exist. However, it can be shown, that no problems occur for a stable system, that is, when $\rho < 1$ [142]. At this point we have to note, that the waiting time distribution is also of phase type, that is $PH \left((1 - \rho) (\beta \mathbf{B}^{-1} \mathbf{1})^{-1} \beta \mathbf{B}^{-1}, \mathbf{B} \right)$. This result may be derived by the use of phase type renewal theory as given in [138], [142], [32] and [153].

With such powerful formulas in hands, one can immediately analyze systems with Erlangian or hyperexponential service facilities. We will omit these examples and refer to [45] instead. Alternatively one may look up more classical texts such as [90] for a conventional derivation. Note, that we did not dwell on arrival and departure dependencies, as some details have been touched already for models with general service pattern. For some detailed results in matrix geometric fashion, the reader should consult [142].

Exponential Service

So far we presented a closed loop allowing for analysis of the $M/ME/1//N$ queueing model. By closer inspection it turns out, that both nodes under

Figure 3.22: The $ME/M/1//N$ service loop

consideration do not adhere to a prescribed role. They only have been named arrival and service node. By simply interchanging their meaning we arrive at the $ME/M/1//N$ closed loop as shown in figure 3.22. For comparison you may also consult figure 3.21. With μ the service rate and λ the vector of arrival rates define $\mathbf{M} := \text{diag}(\lambda)$ to allow for reuse of our former calculations. Any derived parameters have to be changed accordingly, e.g. the traffic intensity becomes $\rho := \mu t := \mu \Psi [\mathbf{B}^{-1}]$. For stability the exponential node representing the service node has to be the faster one leading to the well known constraint $\rho < 1$. Repeating the analysis for the system under resource constraints yields the desired solution. But note, that the $ME/M/1//N$ system does not provide the same results as the $ME/M/1/K$ model, as the corresponding arrival node is not memoryless anymore. Taking the limit for N to infinity and proceeding as above would finally result in the equations for the open $ME/M/1$ queue. Instead we will make use of some results of section 3.6.2 for the more general $G/M/1$ queueing system.

First recall expression 3.98 for the description of the steady state probabilities prior to an arrival

$$\tilde{p}_n = (1 - \omega) \omega^n, \quad n \geq 0$$

where ω is the root of the equation 3.95 for $c = 1$, that is

$$z = \bar{a}(\mu - \mu z)$$

As we have assumed a matrix exponential arrival pattern, we may now substitute its Laplace transform as stated in expression 3.112, i.e.

$$z = -\beta ((\mu - \mu z) \mathbf{I} - \mathbf{B})^{-1} \mathbf{B} \mathbf{1}$$

Here we have chosen a matrix exponential representation $ME(\beta, \mathbf{B})$ with $\beta \mathbf{1} = 1$. Introducing an auxiliary variable $h := \mu - \mu z$ and rewriting the above equation leads to

$$1 - h\mu^{-1} = -\beta (h\mathbf{I} - \mathbf{B})^{-1} \mathbf{B} \mathbf{1}$$

and

$$\begin{aligned} \mu^{-1} &= h^{-1} [1 + \beta (h\mathbf{I} - \mathbf{B})^{-1} \mathbf{B} \mathbf{1}] \\ &= \int_0^\infty (1 - F(x)) e^{-hx} dx \\ &= \int_0^\infty e^{-hx} \beta \exp\{\mathbf{B}x\} \mathbf{1} dx \\ &= \beta (h\mathbf{I} - \mathbf{B})^{-1} \mathbf{1} \end{aligned}$$

Summarizing, ω is the root of the equation

$$\mu \beta ((\mu - \mu \omega) \mathbf{I} - \mathbf{B})^{-1} \mathbf{1} = 1 \quad (3.122)$$

As for the $G/M/1$ queueing system, we may now use the rate conservation law 14 to determine the steady state probabilities

$$p_n = \begin{cases} 1 - \omega & n = 0 \\ (1 - \omega) \rho \omega^{n-1} & n > 0 \end{cases}$$

Any performance characteristics may now be calculated with the help of the relevant formulas for the $G/M/1$ model. One might ask now for the advantage of the matrix exponential approach as opposed to its classic counterpart. The answer lies in expression 3.122, as now we are only concerned with the solution of a *linear* equation. Moreover, it can be shown, that ω is an eigenvalue of the matrix $\mathbf{R}^{-1} = \mathbf{I} - \frac{1}{\mu} \mathbf{B} - \mathbf{1} \beta$. Introducing the normalized eigenvector \mathbf{v} of \mathbf{R}^{-1} , that is the solution of $\mathbf{v} \mathbf{R}^{-1} = \mathbf{v} \omega$ and $\mathbf{v} \mathbf{1} = 1$, we are able to state the vector steady state probabilities

$$\pi_n = \begin{cases} \frac{(1-\omega)}{\mathbf{v} \mathbf{B}^{-1} \mathbf{1}} \mathbf{v} \mathbf{B}^{-1} & n = 0 \\ (1 - \omega) \rho \omega^{n-1} \mathbf{v} & n > 0 \end{cases}$$

Note, that $\mathbf{vB}^{-1}\mathbf{1}$ is a scalar value and even for state 0 the phase does matter. In other words, the arrival process exhibits some sort of memory about the last arriving customer. We are certainly not facing a memoryless system. For the detailed derivation of the steady state probabilities, we refer to the book of Lipsky [142].

Relation to Markov chains

In section 3.3.7 we have seen, that Markovian queues are easy to express in terms of the infinitesimal generator of a Markov chain. We will attempt to repeat our procedure for the more versatile phase type queues introduced so far. With a little abuse of notation let

$$\begin{aligned} q_{01} &= \lambda\beta \\ q_{n,n+1} &= \lambda\mathbf{I} \text{ for } n > 0 \\ q_{10} &= \mathbf{b} \\ q_{n,n-1} &= \mathbf{b}\beta \text{ for } n > 0 \\ q_{00} &= -\lambda \\ q_{nn} &= -(\lambda\mathbf{I} - \mathbf{B}) \text{ for } n > 0 \\ q_{mn} &= 0 \text{ for } |m - n| > 1 \end{aligned}$$

Proceeding yields the infinitesimal generator of the $M/PH/1$ queue

$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda\beta & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{b} & -(\lambda\mathbf{I} - \mathbf{B}) & \lambda\mathbf{I} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{b}\beta & -(\lambda\mathbf{I} - \mathbf{B}) & \lambda\mathbf{I} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{b}\beta & -(\lambda\mathbf{I} - \mathbf{B}) & \lambda\mathbf{I} & \dots \\ \vdots & & & & \ddots & \end{pmatrix}$$

which shows a similar structure as the one generated by a simple Markovian queueing system. As another example, consider the infinitesimal generator of the $PH/M/1$ queue, i.e.

$$\mathbf{Q} = \begin{pmatrix} \mathbf{B} & \mathbf{b}\beta & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \mu\mathbf{I} & -(\mu\mathbf{I} - \mathbf{B}) & \mathbf{b}\beta & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mu\mathbf{I} & -(\mu\mathbf{I} - \mathbf{B}) & \mathbf{b}\beta & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mu\mathbf{I} & -(\mu\mathbf{I} - \mathbf{B}) & \mathbf{b}\beta & \dots \\ \vdots & & & & \ddots & \end{pmatrix}$$

Obviously this way of matrix partitioning leads to a rather general class of queueing systems. Replacing our specific matrices with general ones leads to what is called a (homogenous) *quasi birth-death process*, first introduced by Wallace and extended by Neuts [153]. Please note, that \mathbf{Q} is not the same as the generator matrix for a phase type distribution $\bar{\mathbf{Q}}$.

3.8.3 Multiserver Systems

The power of matrix analytic calculus is best shown in the analysis of the $M/ME/c$ and $M/ME/c/N$ queueing systems. Whereas the classic $M/G/c$ queue resists any of the simpler solution methods, its matrix exponential counterpart allows for a linear description. Here the real power is revealed, as only some matrix algebra is required keeping the mathematical burden at a reasonable level. As before we will trace the footsteps left by Lipsky [142]. Please note the equivalence of the $M/ME/c/N$ and the $M/ME/c/K$ queueing system for $K = N$. This again is a result of the memoryless property of the interarrival distribution, because the residual interarrival time follows an exponential distribution.

Poisson Arrivals

For the case of simple Markovian queueing systems, we were able to analyse the multiserver system by introducing a load dependent service node. As shown in section 3.3.1 we did not distinguish between a server working at rate $c\mu$ and c servers each operating at rate μ . This is only possible for memoryless systems. For the following calculations we have to maintain the relation between customer and server to know if overtaking occurs. In other words, the departures must not necessarily follow the order of the arrival. This is called the *resequencing problem*.

We will now proceed very similar to the single server case and also reuse some of the results presented so far. Starting with a closed system under resource constraints one of the most difficult tasks is to set up a proper state description and define the necessary parameters. Consider a compound service node consisting of c phase type servers and denote its \mathbf{i} -th state as $\mathbf{i} := \{i_1, i_2, \dots, i_n\}$ for $1 \leq i_1 \leq i_2 \leq \dots \leq i_n \leq k$ and $0 < n \leq c$. Note, that state \mathbf{i} is defined as vector, because it simultaneously describes the phases occupied by the active customers. One customer is at phase i_1 , another one is at phase i_2 , and so on, but each of them occupies a different server. Due

to a maximum capacity of c servers, only c customers are allowed to enter, so they never collide within the compound service facility. So our state space consists of

$$d_n := \binom{k+n-1}{n}$$

different elements for n customers active in a service facility of c servers each with k phases. As suggested by such a description, arrivals and departures will trigger a change in dimension of the state space. So we will introduce some strange notation and define $\mathbf{1}_n$ a column vector of ones with dimension d_n and \mathbf{I}_n the $d_n \times d_n$ unity matrix. Obviously the subscript n describes the number of active customers.

In a similar way we also generalize the matrix of service rates by introducing load dependency, that is $\mathbf{M}_n := (M_{n,i,j})$ with elements

$$M_{n,i,j} := \begin{cases} \sum_{m=1}^n \mu_{i_m} & \mathbf{i} = \mathbf{j} \\ 0 & \mathbf{i} \neq \mathbf{j} \end{cases}$$

Again this is only a diagonal matrix and although cumbersome in notation, its meaning is analogous to the single server case. Next we define the probability $E_{n,i,j}$ that a customer entering the service facility currently in state \mathbf{i} will put it right to state \mathbf{j} . Note that the set \mathbf{j} has exactly one member more than the set \mathbf{i} . Due to the assumption, that simultaneous events occur with negligible probability, this is the standard situation we have already encountered for classic queueing systems. Consequently we do not expect an additional internal transition here. Denoting the additional element by v , that is $v := \mathbf{j} \setminus \mathbf{i}$, we formally write $\mathbf{E}_n := (E_{n,i,j})$ and

$$E_{n,i,j} := \begin{cases} \beta_v & \mathbf{i} \cap \mathbf{j} = \mathbf{i} \\ 0 & \text{else} \end{cases}$$

As we are concerned with the arrival of a customer, this has to be a $d_{n-1} \times d_n$ matrix, as the state space is enriched by one dimension. Still assuming, that $\beta_{k+1} = 0$, we have the following relation

$$\mathbf{E}_n \mathbf{1}_n = \mathbf{1}_{n-1} \tag{3.123}$$

In a similar fashion we may describe $L_{n,i,j}$ as the probability, that a leaving customer puts the service facility from state \mathbf{i} to state \mathbf{j} . Denoting the left-over element by $v := \mathbf{i} \setminus \mathbf{j}$, we have to include the case, that v occurs more

than once, as any of these customers could complete service. Introducing the multiplicity α_v , that is the number of occurrences of v in state \mathbf{i} , we are now able to pin down a formal description of $\mathbf{L}_n := (L_{n,\mathbf{i},\mathbf{j}})$,

$$L_{n,\mathbf{i},\mathbf{j}} := \begin{cases} \frac{\alpha_v b_v}{M_{n,\mathbf{i},\mathbf{i}}} & \mathbf{i} \cap \mathbf{j} = \mathbf{j} \\ 0 & \text{else} \end{cases}$$

Related to a departure, this matrix has to be of dimension $d_n \times d_{n-1}$. Please note, that we have used \mathbf{b} in a purely formal way, because the vector $\mathbf{b} = -\mathbf{B}\mathbf{1} = \mathbf{M}(\mathbf{I} - \mathbf{P})\mathbf{1}$ consists of elements $\mu_v q_v$. Here q_v denotes the exit probability from the v -th state. Finally we have to generalize the description of the internal transitions. Define $\mathbf{P}_n := (P_{n,\mathbf{i},\mathbf{j}})$ as transition rate matrix provided n customers are active in the service facility. As before, only one transition occurs at a time, so assume a customer moves from phase u to phase v . More formally, $u := \mathbf{i} \setminus \mathbf{j}$ and $v := \mathbf{j} \setminus \mathbf{i}$. Denoting multiplicity of u by α_u , we can define

$$P_{n,\mathbf{i},\mathbf{j}} := \begin{cases} P_{uv} \frac{\alpha_u \mu_u}{M_{n,\mathbf{i},\mathbf{i}}} & |\mathbf{i} \cap \mathbf{j}| = n - 1 \\ 0 & \text{else} \end{cases}$$

where $\mathbf{P} := (P_{ij})$ is the transition rate matrix of the corresponding service distribution. The notation $|\mathbf{i} \cap \mathbf{j}| = n - 1$ expresses the fact, that $P_{n,\mathbf{i},\mathbf{j}} = 0$ if the set $\mathbf{i} \cap \mathbf{j}$ does not count $n - 1$ elements. As we are concerned with internal state transitions only, the matrix \mathbf{P}_n is square and of dimension $d_n \times d_n$.

As we are still operating in a closed system, we can expect some equilibrium between an internal transition and the combination of departure and subsequent arrival. Taking into account the memoryless arrival node, an observer taking a short break cannot make out a difference when only reviewing state information. Such a reasoning leads to

$$(\mathbf{P}_n + \mathbf{L}_n \mathbf{E}_n) \mathbf{1}_n = \mathbf{1}_n$$

Substituting expression 3.123

$$\begin{aligned} (\mathbf{P}_n + \mathbf{L}_n \mathbf{E}_n) \mathbf{1}_n &= \mathbf{P}_n \mathbf{1}_n + \mathbf{L}_n \mathbf{1}_{n-1} = \mathbf{1}_n \\ \mathbf{L}_n \mathbf{1}_{n-1} &= (\mathbf{I}_n - \mathbf{P}_n) \mathbf{1}_n \end{aligned}$$

and using the common definition $\mathbf{B}_n := \mathbf{M}_n (\mathbf{P}_n - \mathbf{I}_n)$ results in

$$\mathbf{B}_n \mathbf{1}_n = \mathbf{M}_n (\mathbf{P}_n - \mathbf{I}_n) \mathbf{1}_n = -\mathbf{M}_n \mathbf{L}_n \mathbf{1}_{n-1} \quad (3.124)$$

These results will prove to be useful in the derivation of the balance equations.

But before we need to generalize the notion of the steady state probability vector. Define $\pi_{n,m}$ the probability of m customers residing at the service facility with $n = \min(m, c)$ of them being serviced. Still considering a system with limited source supply, only $N - m$ customers inhabit the arrival node. For notational convenience let $\pi_{1,0} := p_0\beta$. The corresponding scalar probabilities follow the same relation as in the single server case, that is $p_m := \pi_{n,m}\mathbf{1}_n$. We are now ready to write down the balance equations for the $M/ME/c//N$ queueing model. First note, that the boundary equations 3.117 and 3.118 may be carried over from the single server case with some notational modifications

$$\lambda\pi_{1,0} = -\pi_{1,1}\mathbf{B}_1\mathbf{1}\beta \quad (3.125)$$

$$\pi_{c,N} = -\lambda\pi_{c,N-1}\mathbf{B}_c^{-1} \quad (3.126)$$

As for the simpler Markovian models, the form of the balance equations changes at boundary states and not somewhere inbetween. Common boundary states for multiserver models are 0, c , N and are applicable here as well. So we may derive

$$\pi_{c,c}(\mathbf{M}_c + \lambda\mathbf{I}_c) = \pi_{c,c+1}\mathbf{M}_c\mathbf{L}_c\mathbf{E}_c + \pi_{c,c}\mathbf{M}_c\mathbf{P}_c + \lambda\pi_{c-1,c-1}\mathbf{E}_c \quad (3.127)$$

whereas the terms on the right correspond to a departure with subsequent arrival, an internal state transition and an arrival occupying an idle resource. Multiplication with \mathbf{E}_c from the right is necessary to restore the appropriate phase configuration. Rearranging and substituting equation 3.124 leads to

$$\pi_{c,c}(\lambda\mathbf{I}_c - \mathbf{B}_c) = \pi_{c,c+1}\mathbf{M}_c\mathbf{L}_c\mathbf{E}_c + \lambda\pi_{c-1,c-1}\mathbf{E}_c \quad (3.128)$$

For $1 < n < c - 1$ a similar reasoning yields

$$\pi_{n,n}(\mathbf{M}_n + \lambda\mathbf{I}_n) = \pi_{n+1,n+1}\mathbf{M}_{n+1}\mathbf{L}_{n+1} + \pi_{n,n}\mathbf{M}_n\mathbf{P}_n + \lambda\pi_{n-1,n-1}\mathbf{E}_n$$

and

$$\pi_{n,n}(\lambda\mathbf{I}_n - \mathbf{B}_n) = \pi_{n+1,n+1}\mathbf{M}_{n+1}\mathbf{L}_{n+1} + \lambda\pi_{n-1,n-1}\mathbf{E}_n \quad (3.129)$$

Here changes in state space occur more naturally, arrivals increase dimension and departures decrease it. We are not concerned with some special behaviour at state c . With a little modification we can also cover the case $n = 1$, that is

$$\pi_{1,1}(\lambda\mathbf{I}_1 - \mathbf{B}_1) = \pi_{2,2}\mathbf{M}_2\mathbf{L}_2 + \lambda\pi_{1,0} \quad (3.130)$$

The next set of equations will cover behaviour of a fully loaded service facility, i.e. for $c < m < N$

$$\pi_{c,m}(\mathbf{M}_c + \lambda \mathbf{I}_c) = \pi_{c,m+1} \mathbf{M}_c \mathbf{L}_c \mathbf{E}_c + \pi_{c,m} \mathbf{M}_c \mathbf{P}_c + \lambda \pi_{c,m-1}$$

Compared with equation 3.127 the main difference lies in the missing factor \mathbf{E}_c . As there are no spare servers available, an arrival can not change the internal state of the service facility. Proceeding as above leads to

$$\pi_{c,m}(\lambda \mathbf{I}_c - \mathbf{B}_c) = \pi_{c,m+1} \mathbf{M}_c \mathbf{L}_c \mathbf{E}_c + \lambda \pi_{c,m-1} \quad (3.131)$$

So finally we arrived at a complete system of balance equations 3.125, 3.126, 3.127, 3.128, 3.129, 3.130 and 3.131. Although this is an impressive set of expressions, a solution algorithm may be found. Due to the structural properties of LAQT, their number of relevant equations is far less than the set of equations derived when applying Markov chain methods in a brute force way.

We now attempt to find a recursion for the calculation of the equilibrium distribution similar to expression 3.121 for the single server case. Due to the dependency on the number of active customers we expect an algorithm of higher complexity. Starting with something familiar we assume the steady state to follow

$$\pi_{c,m} = \pi_{c,m-1} \mathbf{R}_{c,N-m} \quad (3.132)$$

for $c < m \leq N$ starting with

$$\mathbf{R}_{c,0} = -\lambda \mathbf{B}_c^{-1} \quad (3.133)$$

Expressing $\pi_{c,m+1}$ and $\pi_{c,m-1}$ in terms of the recursion 3.132 and substituting the result into equation 3.131 leads to

$$\pi_{c,m}(\lambda \mathbf{I}_c - \mathbf{B}_c) = \pi_{c,m} \mathbf{R}_{c,N-m-1} \mathbf{M}_c \mathbf{L}_c \mathbf{E}_c + \lambda \pi_{c,m} \mathbf{R}_{c,N-m}^{-1}$$

Rearranging and applying an index change from $N - m$ to m yields

$$\mathbf{R}_{c,m} = \lambda (\lambda \mathbf{I}_c - \mathbf{B}_c - \mathbf{R}_{c,m-1} \mathbf{M}_c \mathbf{L}_c \mathbf{E}_c)^{-1} \quad (3.134)$$

So far we are able to calculate the values for all $\mathbf{R}_{c,m}$ with m ranging from 0 to N . In a similar fashion we substitute $\pi_{c,m} \mathbf{R}_{c,N-m-1}$ for $\pi_{c,m+1}$ in equation 3.128 to find

$$\pi_{c,c}(\lambda \mathbf{I}_c - \mathbf{B}_c - \mathbf{R}_{c,N-c-1} \mathbf{M}_c \mathbf{L}_c \mathbf{E}_c) = \lambda \pi_{c-1,c-1} \mathbf{E}_c$$

which simplifies to

$$\boldsymbol{\pi}_{c,c} = \boldsymbol{\pi}_{c-1,c-1} \mathbf{E}_c \mathbf{R}_{c,N-c} \quad (3.135)$$

by the use of expression 3.134 for $m = N - c$. This in turn suggests the definition of

$$\mathbf{R}_{n,N-c} := \lambda (\lambda \mathbf{I}_n - \mathbf{B}_n - \mathbf{E}_{n+1} \mathbf{R}_{n+1,N-c} \mathbf{M}_{n+1} \mathbf{L}_{n+1})^{-1} \quad (3.136)$$

implying the recursion

$$\boldsymbol{\pi}_{n,n} = \boldsymbol{\pi}_{n-1,n-1} \mathbf{E}_n \mathbf{R}_{n,N-c}$$

which allows us to calculate the remaining values $\mathbf{R}_{n,N-c}$ with n ranging to $c - 1$ to 1. Now construct an auxiliary vector

$$\mathbf{r}_{n,N-m} := \begin{cases} \beta & n = 0, m = c \\ \beta \mathbf{R}_{1,N-c} & n = 1, m = c \\ \mathbf{r}_{n-1,N-c} \mathbf{E}_n \mathbf{R}_{n,N-c} & 1 < n \leq c, m = c \\ \mathbf{r}_{c,N-m+1} \mathbf{R}_{c,N-m} & n = c, c < m \leq N \end{cases} \quad (3.137)$$

which is the same as $\mathbf{r}_{n,N-c} = \beta \mathbf{E}_1 \mathbf{R}_{1,N-c} \mathbf{E}_2 \mathbf{R}_{2,N-c} \cdots \mathbf{E}_n \mathbf{R}_{n,N-c}$ for $0 < n \leq c$ and $\mathbf{r}_{c,N-m} = \mathbf{r}_{c,N-c} \mathbf{R}_{c,N-c-1} \mathbf{R}_{c,N-c-2} \cdots \mathbf{R}_{c,N-c-m}$ for $c < m \leq N$. Although hidden by cumbersome notation, we only require the matrices calculated so far by equation 3.134 and 3.136. So the set of vectors is completely specified. This finally leads to the vector steady state distribution

$$\begin{aligned} \boldsymbol{\pi}_{n,n} &= p_0 \mathbf{r}_{n,N-c} & 0 \leq n \leq c \\ \boldsymbol{\pi}_{c,m} &= p_0 \mathbf{r}_{c,N-m} & c \leq m \leq N \end{aligned} \quad (3.138)$$

and its scalar counterpart

$$p_n = \begin{cases} \boldsymbol{\pi}_{n,n} \mathbf{1}_n & 0 \leq n \leq c \\ \boldsymbol{\pi}_{c,m} \mathbf{1}_c & c \leq m \leq N \end{cases}$$

Applying the normalization criterion yields the value for p_0 , that is

$$\begin{aligned} p_0 &= \left(\sum_{n=0}^N p_n \right)^{-1} \\ &= \left(\sum_{n=0}^{c-1} \mathbf{r}_{n,N-c} \mathbf{1}_n + \sum_{m=c}^N \mathbf{r}_{c,N-m} \mathbf{1}_c \right)^{-1} \end{aligned} \quad (3.139)$$

For clarification, we will now summarize the steps necessary to calculate the equilibrium distribution for the $M/ME/c/N$ queueing system.

1. From the system description assemble the parameters \mathbf{M}_n , \mathbf{L}_n , \mathbf{E}_n , \mathbf{P}_n and \mathbf{B}_n as described in the beginning of this section.
2. Apply equation 3.134 and 3.133 recursively to find the values for $\mathbf{R}_{c,m}$, $0 \leq m \leq N - c$.
3. Calculate $\mathbf{R}_{n,N-c}$, $0 < n < c$ starting from top by the use of expression 3.136.
4. Insert the results from the former two steps in definition 3.137 to assemble the set of auxiliary vectors.
5. Express the steady state probabilities in terms of p_0 to receive the final result after normalization 3.139.

As the algorithm suggests, an explicit result is only available for the case $c = 1$. Although no closed form solution exists, the solution is exact in the mathematical sense. Compared with any other methods suited for the class of $M/G/c$ models, the presented procedure remains the simplest. Furthermore it is straightforward to implement as a computer program. From the steady state probabilities the corresponding performance indicators of the $M/ME/c//N$ system are found the common way, that is

$$\begin{aligned}
 L &= \sum_{n=1}^N np_n, & W &= \frac{1}{\lambda}L \\
 W_q &= W - s = W + \Psi [\mathbf{B}^{-1}] \\
 L_q &= \lambda W_q = L + \Psi [\mathbf{B}^{-1}]
 \end{aligned} \tag{3.140}$$

Please note, that our state description provided a good choice for the model under consideration but does not generalize very well. The common approach is to count the number of customers occupying a certain phase. In order to adjust for this alternative, minor modifications have to be carried out. This has been done by Lipsky and is called the *generalized $M/ME/c//N$ system*. For further information we refer to his book [142].

In the process of opening the $M/ME/c//N$ loop we do exactly the same as for the single server version, that is by letting N approach infinity. To allow for meaningful results, we have to assume a stable system. So we ask for a *maximum utilization* $u_{\max} < 1$, which in turn may be expressed as $u_{\max} = \lambda s_{\min}$ with s_{\min} the *minimal service time* of the service facility. To

derive the latter we need to introduce the probability $Y_{n,i,j}$, that the service facility is in state \mathbf{j} immediately after a departure provided it was in state \mathbf{i} before, and no customers have entered. The corresponding $d_n \times d_{n-1}$ matrix $\mathbf{Y}_n := (Y_{n,i,j})$ may be written as

$$\mathbf{Y}_n = \mathbf{L}_n + \mathbf{P}_n \mathbf{Y}_n$$

where the first term relates to a customer departing and the second is a combination of an internal transition with subsequent departure. Rearranging leads to

$$\mathbf{Y}_n = (\mathbf{I}_n - \mathbf{P}_n)^{-1} \mathbf{L}_n = -\mathbf{B}_n^{-1} \mathbf{M}_n \mathbf{L}_n$$

Now define \mathbf{v} as the left normalized eigenvector of $\mathbf{Y}_c \mathbf{E}_c$ to the unit eigenvalue, i.e. \mathbf{v} satisfies

$$\mathbf{v} \mathbf{Y}_c \mathbf{E}_c = \mathbf{v}, \quad \mathbf{v} \mathbf{1}_c = 1$$

Then the maximum utilization is given by

$$u_{\max} := -\lambda \mathbf{v} \mathbf{B}_c^{-1} \mathbf{1}_c$$

So for a stable system we have to assume $\lambda \mathbf{v} \mathbf{B}_c^{-1} \mathbf{1}_c < 1$ instead of the usual condition $\lambda s < c$. The reason is, that $\frac{s}{c}$ is only an average value. But we have to deal with minimal service time to assure proper results. We have omitted the derivation as it relies on results about the departure process and certain transient properties. For a complete treatment we refer to [142].

As shown above, an iterated procedure has been applied to the system with limited supply. Only concerned with a finite number of customers we could have been certainly sure to reach an end. This is not true for the open system. Although we can follow the same steps, additional criteria for ending the process have to be found. In our case we can monitor the difference between iterates and stop the algorithm, if it becomes insignificant. Formally we are concerned with a limiting process for $N \rightarrow \infty$ as in

$$\mathbf{R}_{n,\infty} := \lim_{N \rightarrow \infty} \begin{cases} \mathbf{R}_{c,N} & n = c \\ \mathbf{R}_{n,N-c} & 0 < n < c \end{cases}$$

Note, that the $\mathbf{R}_{c,\infty}$ are still required to calculate the $\mathbf{R}_{n,\infty}$ for $0 < n < c$. There is no shortcut in the steps presented above, although a little simplification is gained for the limiting auxiliary vectors

$$\mathbf{r}_{n,\infty} := \begin{cases} \beta & n = 0 \\ \beta \mathbf{R}_{1,\infty} & n = 1 \\ \mathbf{r}_{n-1,\infty} \mathbf{E}_n \mathbf{R}_{n,\infty} & 1 < n \leq c \\ \mathbf{r}_{c,\infty} \mathbf{R}_{c,\infty}^{n-c} & c < n \end{cases}$$

and the steady state probabilities

$$\boldsymbol{\pi}_{n,n} = p_0 \mathbf{r}_{n,\infty}$$

The value for p_0 may be calculated by the use of expression 3.139 for $N = \infty$, which simplifies to

$$p_0 = \left[\sum_{n=0}^{c-1} \mathbf{r}_{n,\infty} \mathbf{1}_n + \mathbf{r}_{c,\infty} (\mathbf{I}_c - \mathbf{R}_{c,\infty})^{-1} \mathbf{1}_c \right]^{-1}$$

by identifying the term $\sum_{n=c}^{\infty} \mathbf{R}_{c,\infty}^{n-c} = \sum_{m=0}^{\infty} \mathbf{R}_{c,\infty}^m$ as geometric sum equivalent to $(\mathbf{I}_c - \mathbf{R}_{c,\infty})^{-1}$. Comparing p_0 with expression 3.23 for the $M/M/c$ queueing system shows a close resemblance. This once more justifies the solution to be called *matrix geometric*. The relevant performance characteristics are calculated the same way as for the system with limited number of customers, so refer to expressions 3.140 with N replaced by ∞ .

The iterative procedure presented above may not be the best from a numerical viewpoint and indeed, there exist better algorithms. Much of them have been adapted from numerical procedures for Markov chains to be applied to the class of quasi birth-death processes introduced later in section 3.8.4. On the other hand, the current approach resembles the sequence of systems with limited source supply and describes its development for an increased number of customers. A closed form solution may only be found for the case $c = 1$ even for such a simple matrix exponential model such as the $E_l/E_k/c$ queue [90].

Exponential Service

For the solution of the single server model we have provided a shortcut solution based on the framework derived for the $G/M/c$ queueing system. Paying attention to the methods used, there is certainly an emphasis on Laplace transforms. Based on our results in section 3.6.1 we can say, that for $c > 1$ servers the calculation of the normalization constant becomes a rather tedious task. It would be desirable to find a solution which is better adapted for implementation as a computer program. Focusing on linear algebraic manipulations rather than on transform theory seems an appropriate path as exemplified by the analysis of the $M/ME/c$ queueing system. In fact such a path exists and we will adopt it here for the analysis of the $ME/M/c$ queue.

Furthermore, some results from the single server system with Poisson arrivals can be reused.

As might have been expected, a system with limited supply is considered first. Employing the memoryless feature of the service facility we do not need to trace each customer. Instead we can imagine it as a single load dependent node. This is exactly the same approach as has been used for the analysis of the $M/M/c$ queue. Usually we would now introduce a service rate of $n\mu$ for $n < c$ and $c\mu$ for $n \geq c$. However, for the following analysis this would be too restrictive and without gain. Instead we will work with the more general load dependent rates and only assume $\mu_n := \mu_c$ for $n \geq c$. Noting the similarities to the closed $M/ME/1//N$ loop, all balance equations remain valid and may be reused. Applying some notational modification, that is changing n to $N - n$ and λ to μ_n , expressions 3.117, 3.119 and 3.118 may be rewritten as

$$\mu_n \pi_N = -\pi_{N-1} \mathbf{B} \mathbf{1} \beta \quad (3.141)$$

$$\pi_n (\mu_n \mathbf{I} - \mathbf{B}) = \pi_{n+1} \mu_{n+1} \mathbf{I} - \pi_{n-1} \mathbf{B} \mathbf{1} \beta \quad (3.142)$$

$$\pi_0 = -\mu_1 \pi_1 \mathbf{B}^{-1} \quad (3.143)$$

But similarities do not end here. In fact, the derivation of Neuts' \mathbf{R} follows the same way and will not be repeated here. Due to load dependency we have to introduce an index and write

$$\mathbf{R}_n := \frac{\mu_{n+1}}{\mu_n} \left(\mathbf{I} - \frac{1}{\mu_n} \mathbf{B} - \mathbf{1} \beta \right)^{-1} \quad (3.144)$$

for $n > 0$. This allows for specification of the vector steady state distribution

$$\pi_n = \pi_N \prod_{j=n}^{N-1} \mathbf{R}_j = p_N \beta \prod_{j=n}^{N-1} \mathbf{R}_j \quad (3.145)$$

and its scalar counterpart $p_n = \pi_n \mathbf{1}$. Applying the normalization criterion leads to the remaining probability

$$\begin{aligned} p_N &= \left(\sum_{n=0}^N \pi_n \mathbf{1} \right)^{-1} \\ &= \left[\beta \left(\mathbf{I} + \sum_{n=0}^{N-1} \prod_{j=n}^{N-1} \mathbf{R}_j \right) \mathbf{1} \right]^{-1} \\ &= [\beta \mathbf{K}(\mathbf{N}) \mathbf{1}]^{-1} = \Psi^{-1} [\mathbf{K}(\mathbf{N})] \end{aligned}$$

whereas $\mathbf{K}(\mathbf{N})$ denotes the normalization constant. Up to now we did not employ the special structure of μ_n for $n \geq c$. By doing so now sheds some light on expression 3.144. In fact, $\mathbf{R}_n = \mathbf{R}_c = \left(\mathbf{I} - \frac{1}{\mu_c} \mathbf{B} - \mathbf{1}\beta\right)^{-1}$ remains constant for $n \geq c$. This in turn allows for a rather accelerated calculation of the normalization constant, i.e.

$$\begin{aligned} \mathbf{K}(N) &= \mathbf{I} + \sum_{n=0}^{N-1} \prod_{j=n}^{N-1} \mathbf{R}_j \\ &= \mathbf{I} + \mathbf{R}_c^{N-c} \left(\mathbf{I} + \sum_{n=0}^c \prod_{j=n}^c \mathbf{R}_j \right) + \sum_{n=c+1}^{N-1} \mathbf{R}_c^{N-n} \\ &= \mathbf{R}_c^{N-c} \mathbf{K}(c) + (\mathbf{I} - \mathbf{R}_c^{N-c}) (\mathbf{I} - \mathbf{R}_c)^{-1} \end{aligned}$$

Note, that once again we encountered a geometric series. With the entire steady state distribution at hands, one is only left with the calculation of the relevant performance indicators. Again we decide not to repeat the standard approach and instead refer to the set of expressions 3.140.

The corresponding open system may be analyzed either by the common procedure of letting N go to infinity or by making use of the special structure of μ_n for $n \geq c$. The former approach may be found in Lipsky's book [142], whereas the latter is due to Neuts [153]. In analogy to the single server model with Poisson arrivals we assume a stable system, i.e.

$$1 < \rho^{-1} := \Psi [\mathbf{B}^{-1}] \max \{ \mu_1, \mu_2, \dots, \mu_{c-1}, \mu_c \}$$

assuring the existence of \mathbf{R}_n^{-1} . Note, that in case of a classic multiserver queue, this condition reduces to $1 < \Psi [\mathbf{B}^{-1}] c\mu$. By employing the local balance principle, which is equating the flows between two adjacent states one arrives at

$$\begin{aligned} \mu_n \pi_n \mathbf{1} &= \pi_{n-1} \mathbf{b} \\ &= -\pi_{n-1} \mathbf{B} \mathbf{1} \end{aligned}$$

Substituting the above term and $\pi_n \mathbf{R}_n^{-1}$ for π_{n+1} into expression 3.142 yields

$$\begin{aligned} \pi_n (\mu_n \mathbf{I} - \mathbf{B}) &= \pi_{n+1} \mu_{n+1} \mathbf{I} - \pi_{n-1} \mathbf{B} \mathbf{1} \beta \\ &= \pi_n \mathbf{R}_n^{-1} \mu_{n+1} + \mu_n \pi_n \mathbf{1} \beta \end{aligned}$$

which becomes

$$\pi_{c-1} (\mu_{c-1} \mathbf{I} - \mathbf{B}) = \mu_c \pi_n \mathbf{R}_c^{-1} + \mu_{c-1} \pi_n \mathbf{1} \beta$$

for $n = c - 1$. By rearranging this leads to

$$\begin{aligned} \pi_{c-1} &= [\mu_{c-1} \mathbf{I} - \mathbf{B} - \mu_{c-1} \mathbf{1} \beta - \mu_c \mathbf{R}_c^{-1}]^{-1} \\ &= \left[\mu_{c-1} \mathbf{I} - \mathbf{B} - \mu_{c-1} \mathbf{1} \beta - \mu_c \left(\mathbf{I} - \frac{1}{\mu_c} \mathbf{B} - \mathbf{1} \beta \right) \right]^{-1} \\ &= [(\mu_{c-1} - \mu_c) (\mathbf{I} - \mathbf{1} \beta)]^{-1} \end{aligned} \quad (3.146)$$

which can be explicitly solved. The remaining vector probabilities are easily found using an adapted version of equation 3.145, that is

$$\pi_n = \begin{cases} -\mu_1 \pi_1 \mathbf{B}^{-1} & n = 0 \\ \pi_{c-1} \prod_{j=n}^{c-2} \mathbf{R}_j & 0 < n < c - 1 \\ \pi_{c-1} R_c^{n-c+1} & n \geq c - 1 \end{cases}$$

Please note, that the above set of equations determines the vector probabilities only up to a multiplicative constant C^{-1} . Normalizing the results will provide the proper steady state distribution. The corresponding normalization constant is given by

$$\begin{aligned} C &= \sum_{n=0}^{c-2} \pi_n \mathbf{1} + \pi_{c-1} \left(\sum_{n=c-1}^{\infty} R_c^{n-c+1} \right) \mathbf{1} \\ &= \sum_{n=0}^{c-2} \pi_n \mathbf{1} + \pi_{c-1} (\mathbf{I} - \mathbf{R}_c)^{-1} \mathbf{1} \end{aligned}$$

As for the system with limited number of customers, the relevant performance indicators are found by the set of expressions 3.140 for $N = \infty$. As a matter of fact we have found another instance of a quasi birth-death process, but this time it is not homogenous. This is mainly due to the load dependency introduced for multiserver queues with exponential service. The infinitesimal generator of the $PH/M/c$ queueing system is given by

$$\mathbf{Q} = \begin{pmatrix} \mathbf{B} & \mathbf{b}\beta & 0 & 0 & 0 & 0 & \cdots \\ \mu_1 \mathbf{I} & -(\mu_1 \mathbf{I} - \mathbf{B}) & \mathbf{b}\beta & 0 & 0 & 0 & \cdots \\ 0 & \mu_2 \mathbf{I} & -(\mu_2 \mathbf{I} - \mathbf{B}) & \mathbf{b}\beta & 0 & 0 & \cdots \\ \vdots & & & \ddots & & & \\ 0 & 0 & & \mu_c \mathbf{I} & -(\mu_c \mathbf{I} - \mathbf{B}) & \mathbf{b}\beta & \cdots \\ \vdots & & & & & \ddots & \end{pmatrix}$$

Note the similarity with the single server case especially for the submatrix starting at row c and column c . If full capacity has been reached, the multiserver queue resembles a single server queue operating at rate μ_c . As we have already seen in our calculations, this property constitutes the matrix geometric nature of the solution. It appears, that without this feature a closed form solution becomes very unlikely. In fact, it turns out, that for more complex systems an iterative procedure provides a reasonable path to the solution. We will shed some light on these topics in the next section.

3.8.4 Quasi Birth-Death Process

As has been demonstrated above, the matrix geometric approach provides a rather flexible framework while retaining some important features well known from the theory of Markovian queues. It even allows for a generalization of the so commonly used birth-death process introduced in section 3.3. These so called *quasi birth-death processes (QBD)* rely on matrix partitioning and possess a similar structure as their scalar counterparts. As before, we will focus on continuous processes and omit the discrete case. As common for quasi birth-death processes, one starts with the analysis of the infinitesimal generator of the corresponding Markov chain. Due to this choice, the arrival and the service process have to be specified as phase-type distributions. We already know, this class is rich enough to approximate almost any (absolutely continuous) distribution, but not every distribution with rational Laplace transform may be included right away. In fact, matrix exponential representations allow for complex entries which violates some basic assumptions of Markov chain theory.

Recalling the structure of the infinitesimal generator for certain memoryless queueing models, there is a significant difference in the related birth-death process for the classic single and multiserver systems. For the former each level of the generator matrix is the same whereas for the latter some level dependent factor is involved. This factor usually models effects such as load dependency and customer impatience. Although this separation has not been spelled out explicitly for birth-death processes it is relevant for quasi birth-death processes. Furthermore, the interpretation just given also carries over. So for the analysis of the $PH/PH/1$ queue the *homogenous quasi birth-death process* seems most appropriate, whereas for multiserver and retrial phase queues the *inhomogenous* version is preferred.

Homogenous Quasi Birth-Death Process

The homogenous quasi birth-death process has been introduced by V. Wallace in 1969 in his dissertation [207]. Although superseded by other works in the field, it is one of the most readable accounts on quasi birth-death processes. It is the authors impression, that some common texts focusing on more general models lack dedication to this topic. So this approach will follow the works by Wallace [207] and Latouche and Ramaswami [138], the latter also being an exceptional introduction to subject matter.

As a formal definition of a homogenous quasi birth-death process, consider a continous time Markov chain with infinitesimal generator

$$\mathbf{Q} = \begin{pmatrix} \mathbf{E} & \bar{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \bar{\mathbf{C}} & \mathbf{D} & \mathbf{A} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{C} & \mathbf{D} & \mathbf{A} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{C} & \mathbf{D} & \mathbf{A} & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix} \quad (3.147)$$

subject to $\mathbf{E}\mathbf{1} + \bar{\mathbf{A}}\mathbf{1} = \mathbf{0}$, $\bar{\mathbf{C}}\mathbf{1} + (\mathbf{D} + \mathbf{A})\mathbf{1} = \mathbf{0}$ and $(\mathbf{C} + \mathbf{D} + \mathbf{A})\mathbf{1} = \mathbf{0}$. If not otherwise stated all matrices are square matrices and of dimension $k \times k$. Exceptions are the $k_0 \times k$ and $k \times k_0$ boundary matrices given by

$$\bar{\mathbf{A}} = \begin{pmatrix} \mathbf{A} \\ \mathbf{0} \end{pmatrix}, \quad \bar{\mathbf{C}} = \begin{pmatrix} \mathbf{C} & \mathbf{0} \end{pmatrix}$$

and the $k_0 \times k_0$ matrix \mathbf{E} , $k_0 \geq k$. In the literature very often the same dimension is assumed for all matrices, but here we want to emphasize the resemblance of the generators introduced so far in section 3.8.2. For the matrix exponential models above we were able to derive Neuts' matrix \mathbf{R} explicetely from the balance equations. We now assert, that the same can be done here and provided a steady state solution exists, it will be given by

$$\pi_n = \pi_0 \bar{\mathbf{R}} \mathbf{R}^{n-1} \quad (3.148)$$

Note the change in dimension of the solution vector which requires

$$\bar{\mathbf{R}} = \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$$

to be used instead of \mathbf{R} . As a matter of fact, a stable system is warranted by the condition $\rho(\mathbf{R}) < 1$. Here the common symbol $\rho(\cdot)$ denotes the *spectral*

radius, that is the largest eigenvalue of its argument [81]. For more details on the stability issue, we refer to [207], [138] and [153].

Being aware from the theory of Markov chains, that an equilibrium distribution satisfies $\pi \mathbf{Q} = \mathbf{0}$, we may take advantage of the special structure of the infinitesimal generator 3.147 to arrive at

$$\begin{aligned}\pi_0 \mathbf{E} + \pi_1 \bar{\mathbf{C}} &= \mathbf{0} \\ \pi_0 \bar{\mathbf{A}} + \pi_1 \mathbf{D} + \pi_2 \mathbf{C} &= \mathbf{0} \\ \pi_{n-1} \mathbf{A} + \pi_n \mathbf{D} + \pi_{n+1} \mathbf{C} &= \mathbf{0}\end{aligned}$$

for all $n > 1$. Substituting expression 3.148 leads to

$$\pi_0 (\mathbf{E} + \bar{\mathbf{R}} \bar{\mathbf{C}}) = \mathbf{0} \quad (3.149)$$

$$\pi_0 (\bar{\mathbf{A}} + \bar{\mathbf{R}} \mathbf{D} + \bar{\mathbf{R}} \mathbf{R} \mathbf{C}) = \mathbf{0} \quad (3.150)$$

$$\pi_0 \bar{\mathbf{R}} \mathbf{R}^{n-2} (\mathbf{A} + \mathbf{R} \mathbf{D} + \mathbf{R}^2 \mathbf{C}) = \mathbf{0} \quad (3.151)$$

We do need one additional equation to justify a unique solution and this remaining expression is provided by the normalization criterion, that is

$$\begin{aligned}1 &= \pi_0 \left(\mathbf{I} + \bar{\mathbf{R}} \sum_{n=1}^{\infty} \mathbf{R}^{n-1} \right) \mathbf{1} \\ &= \pi_0 [\mathbf{I} + \bar{\mathbf{R}} (\mathbf{I} - \mathbf{R})^{-1}] \mathbf{1}\end{aligned} \quad (3.152)$$

Close inspection of the above equations reveals the following two conditions for the existence of a steady state solution

1. A unique positive solution for π_0 may be found from expression 3.149 and 3.152
2. From expression 3.151 the quadratic matrix equation $\mathbf{A} + \mathbf{R} \mathbf{D} + \mathbf{R}^2 \mathbf{C}$ has a minimal non-negative solution

The main problem to this strategy is to solve the *quadratic matrix equation*, because it does not behave like its scalar counterpart. In fact, more than two solutions may exist and are not easy to find. One approach is to treat a matrix equation as polynomial matrix function and apply an iterative procedure [82][99]. Another resorts to the analysis of the related quadratic matrix equation $\mathbf{A} \mathbf{G}^2 + \mathbf{D} \mathbf{G} + \mathbf{C}$ due to its more suitable properties. Neuts'

matrix is then derived from the relation $\mathbf{R} = -\mathbf{A}(\mathbf{A}\mathbf{G} + \mathbf{D})^{-1}$. It turns out, that the matrix \mathbf{G} still admits a probabilistic interpretation [138][190], but we will treat it as pure mathematical item in our brief introduction. Now several methods exist to find a solution for this new equation. To name a few of them, *linearization* and *canonical Wiener-Hopf factorization* provide an appropriate background. For the former the quadratic matrix equation $\mathbf{A}\mathbf{G}^2 + \mathbf{D}\mathbf{G} + \mathbf{C}$ is transformed into the system of linear equations

$$\begin{pmatrix} \mathbf{D} & \mathbf{A} & & & \\ \mathbf{C} & \mathbf{D} & \mathbf{A} & & \\ & \mathbf{C} & \mathbf{D} & \mathbf{A} & \\ & & & \ddots & \end{pmatrix} \begin{pmatrix} \mathbf{G} \\ \mathbf{G}^2 \\ \mathbf{G}^3 \\ \vdots \end{pmatrix} = \begin{pmatrix} -\mathbf{C} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{pmatrix}$$

whereas for the latter one works with the corresponding *matrix Laurent polynomial* $\mathbf{S}(\mathbf{z}) = \mathbf{C}\mathbf{z}^{-1} + \mathbf{D} + \mathbf{A}\mathbf{z}$. If there exists a factorization

$$\mathbf{S}(\mathbf{z}) = (\mathbf{U}_0 + \mathbf{U}_1\mathbf{z}) (\mathbf{L}_0 + \mathbf{L}_{-1}\mathbf{z}^{-1})$$

where \mathbf{U}_0 , \mathbf{U}_1 , \mathbf{L}_0 and \mathbf{L}_{-1} are $k \times k$ square matrices, then $\mathbf{G} = -\mathbf{L}_0^{-1}\mathbf{L}_{-1}$ is a unique solution to the quadratic matrix equation under consideration. As suggested by the notation, this method is related to the UL decomposition of (Toeplitz) matrices. For more information on these specific topics we refer to the book by Bini, Latouche and Meini [44]. Although concerned with the solution of Markov chains, Stewart provides some insight into the solution of quasi birth-death processes in [190] making it a valuable reference. Some useful relations with respect to the analysis of telecommunication systems may be found in [62]. As you might expect, there exists a wealth of powerful iterative procedures waiting for application. One of them will be described in the context of inhomogenous quasi birth-death processes, as we do not gain much from the homogeneity property.

Inhomogenous Quasi Birth-Death Process

For the inhomogenous quasi birth-death process the infinitesimal generator is of similar structure, that is

$$\mathbf{Q} = \begin{pmatrix} \mathbf{D}_0 & \mathbf{A}_0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{C}_1 & \mathbf{D}_1 & \mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{C}_2 & \mathbf{D}_2 & \mathbf{A}_2 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_3 & \mathbf{D}_3 & \mathbf{A}_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix}$$

Although the boundary matrices have been renamed we can still adhere to our convention, that \mathbf{D}_0 , \mathbf{A}_0 and \mathbf{C}_1 are of dimension $k_0 \times k_0$, $k_0 \times k$ and $k \times k_0$, respectively. In assuming a limited number of sources N , the matrix \mathbf{Q} becomes finite and we may apply an iterative procedure to find the steady state vectors $\boldsymbol{\pi}_n$, $0 \leq n \leq N$. This algorithm is called *linear level reduction* and was first introduced by D.P. Gaver, P.A. Jacobs and G. Latouche [138][29]. It proceeds as follows

1. Compute the auxiliary matrices $\mathbf{B}_n = \mathbf{D}_n + \mathbf{C}_n (-\mathbf{B}_{n-1}^{-1}) \mathbf{A}_{n-1}$ for $0 < n \leq N$, where $\mathbf{B}_0 := \mathbf{D}_0$
2. Solve the system of linear equations $\boldsymbol{\pi}_n \mathbf{B}_n = \mathbf{0}$, $\boldsymbol{\pi}_n \mathbf{1} = 1$
3. Let $K := 1$
4. Recursively calculate $\boldsymbol{\pi}_n = \boldsymbol{\pi}_{n+1} \mathbf{C}_{n+1} (-\mathbf{B}_n^{-1})$ and $K = K \boldsymbol{\pi}_n \mathbf{1}$ for $0 \leq n < N$
5. Normalize the probability vector, i.e. $\boldsymbol{\pi}_n = K^{-1} \boldsymbol{\pi}_n$

In case an overflow problem occurs during calculation, additional normalization steps might be included. Note, that no use has been made of the rate matrix \mathbf{R}_n . For an infinite number of sources matters become more complicated. It can be shown, that the steady state solution vector is still of the form $\boldsymbol{\pi}_n = \boldsymbol{\pi}_{n-1} \mathbf{R}_n$, but this time Neuts' matrix depends on the level too. It is a common practice to truncate the infinite system and apply algorithms such as the linear level reduction to find an approximation for the equilibrium distribution. For more details we refer to the books [138], [190], [44] and [29].

3.8.5 More General Models

The concept of the quasi birth-death process may be further extended similar to the transition from a classic birth-death process to semimarkovian models. By allowing a triagonal structure of the infinitesimal generator, one arrives at the phase type counterpart of the classic $M/G/1$ and $G/M/1$ queueing systems. In the literature these models are known under different names often underlining the special dedication of their authors. For Latouche et. al. the corresponding processes are called *skipfree in one direction*, whereas

Neuts raises the class of $M/G/1$ or $G/M/1$ type models. One has to be very careful, as sometimes there are little differences in the assumptions. Most of the relevant theory was developed by Neuts in his two books [153] and [154]. A good introduction has been provided by G. Latouche and V. Ramaswami in the later chapters of [138]. An interesting account relying on the combination of matrix analysis and transform theory has been given by J.N. Daigle in [63], because his methods came right from the technicians toolbox. Solution aspects are emphasized in the work of D.A. Bini, G. Latouche and B. Meini [44], which might be regarded as the standard reference in the field. In the current context it is important to note, that some of these more complex models may be represented as quasi birth-death process underlining the importance of the latter.

So far we have been concerned with phase type or matrix exponential arrival processes of the renewal type [32]. The corresponding assumption of independent arrivals is also reflected by the constant entry vector β . By relaxing this restriction and allowing β to depend on the last arrival, that is introducing correlated interarrivals, one arrives at the so called *Markovian arrival process (MAP)*. It turns out, that the block structure of the infinitesimal generator remains unaltered. As a consequence, many concepts and ideas carry over from linear algebraic queueing theory. Allowing for a triangular structure of the generator, one immediately arrives at the so called *batch Markovian arrival process (BMAP)*. The resulting class of models has been successfully applied to problems of communication systems. At the time of writing, the analysis of single server systems with Markovian arrival processes was very popular and some parts in the analysis of multiserver queues remained an open challenge. Despite this fact, some authors of more recent textbooks on queueing theory decided to focus on MAPs and treat phase type renewal processes as a special case, e.g. [52] and [200]. Furthermore one has to mention the book by S. Asmussen [30], which provides a more theoretical introduction to the subject.

Although there are many different approaches to choose from, one should stick to the principle of Ockham's razor and avoid any unnecessary complexities. The simpler the model, the simpler its solution. In practice this often leads to more stable results.

3.8.6 Retrials

The remainder of this chapter is dedicated to an extension of the multiserver retrial model presented in section 3.4.3, that is the $M/PH/c/c$ queueing system with phase type retrial times. Although there has been some research in the past, only few papers capturing the retrial effect for phase type queues have appeared [29]. A high effort has been placed to analyze single server systems with complex arrival pattern, so the multiserver queue presented now has been a natural choice to be included here by the author.

Recall, that the memoryless version of section 3.4.3 has been described by a two state Markov process $\{C(t), N(t) : t \geq 0\}$, where $C(t)$ and $N(t)$ denote the number of busy servers and the orbit size. We will adopt the same concept for the macro level, but there is some work to do at the micro level. For the model under consideration service times are assumed to follow a phase type distribution of order k with representation $PH(\beta, \mathbf{B})$. As before the exit vector is denoted by $\mathbf{b} := -\mathbf{B}\mathbf{1}_k$, where the subscript k is used to emphasize the dimension of the corresponding vector. The same applies to the $k \times k$ identity matrix \mathbf{I}_k . For the underlying service node we assume a $M/PH_k/c/c$ loss system in agreement with its memoryless counterpart. The period between retrials shall follow a phase type distribution of order l with representation $PH(\tau, \mathbf{T})$ and exit vector $\mathbf{t} := -\mathbf{T}\mathbf{1}_l$. At the macro level we still have a state description (m, n) corresponding to the number of active servers and the orbit size. To incorporate the phase information into the model, we have to extend the state space and assign an appropriate sublevel to each m and n . An exception occurs, when either m or n is zero. Then no customer is present at the service facility or in orbit and the affected sublevel collapses. This leads to a Markov process $\{C(t), N(t), \mathbf{m}(t), \mathbf{n}(t) : t \geq 0\}$, where $\mathbf{m}(t)$ and $\mathbf{n}(t)$ represent the phase configuration of the service facility or the orbit. Its infinitesimal generator may be described as

$$\mathbf{Q} = \begin{pmatrix} \mathbf{D}_0 & \mathbf{A}_0 & & & & \\ \mathbf{C}_1 & \mathbf{D}_1 & \mathbf{A}_1 & & & \\ & \mathbf{C}_2 & \mathbf{D}_2 & \mathbf{A}_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{C}_{N-c-1} & \mathbf{D}_{N-c-1} & \mathbf{A}_{N-c-1} \\ & & & & \mathbf{C}_{N-c} & \mathbf{D}_{N-c} \end{pmatrix}$$

where we have assumed a system with limited source supply. Obviously this is the generator for a finite inhomogenous quasi birth-death process and we

may apply the methods presented in section 3.8.4 to find a solution. This also gives rise to the approximation of the infinite supply case by simply choosing a sufficiently large value for N in the current model. It remains to assign some more detail to the submatrices \mathbf{A}_n , \mathbf{C}_n and \mathbf{D}_n given above.

The matrices \mathbf{A}_n describe the case where the orbit size is increased by one due to an arrival to a full system [29], i.e.

$$\mathbf{A}_n = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (N - c - n) \lambda \mathbf{I}_{k^n} \otimes \boldsymbol{\tau} \otimes \mathbf{I}_{l^c} \end{pmatrix}$$

for $0 \leq n < N - c$. Here the symbol \otimes denotes the *Kronecker product*, which is defined as

$$\mathbf{A} \otimes \mathbf{B} := \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & & \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & & \\ & & \ddots & \end{pmatrix}$$

for two matrices $\mathbf{A} := (a_{ij})$ and \mathbf{B} . The Kronecker product arises naturally in the theory of linear matrix equations and has become a handsome tool in matrix analysis [99][183]. The next matrix to be defined captures the case, where the orbit size decreases by one due to a successful repeated attempt, that is

$$\mathbf{C}_n = \begin{pmatrix} \mathbf{0} & \mathbf{C}_n^{(0,1)} & & & \\ & \mathbf{0} & \mathbf{C}_n^{(1,2)} & & \\ & & \ddots & \ddots & \\ & & & \mathbf{0} & \mathbf{C}_n^{(c-1,c)} \\ & & & & \mathbf{0} \end{pmatrix}$$

for $0 < n \leq N - c$. The matrices $\mathbf{C}_n^{(m,m+1)}$ describe the transition between two adjacent sublevels (m, n) and $(m + 1, n - 1)$. Formally this is expressed as

$$\mathbf{C}_n^{(m,m+1)} = (\oplus_{i=1}^n \mathbf{t}) \otimes \mathbf{I}_{l^m} \otimes \boldsymbol{\beta}$$

provided $0 \leq m < c$. The first term

$$\oplus_{i=1}^n \mathbf{t} = (\mathbf{t} \otimes \mathbf{I}_k \otimes \dots \otimes \mathbf{I}_k) + \dots + (\mathbf{I}_k \otimes \dots \otimes \mathbf{I}_k \otimes \mathbf{t})$$

is a *generalized direct sum* [183] and allows us to identify the retrial unit performing a successful attempt [29]. The last term $\boldsymbol{\beta}$ relates to the phase being occupied by the retrying customer now becoming active. The middle term \mathbf{I}_{l^m} corresponds to the fact, that no change in the service phase configuration

has occurred. Turning attention to the diagonal elements \mathbf{D}_n we are left with those transitions, that have no impact on the orbit size, e.g. arrivals to an empty server, internal transitions and blocked retrials followed by an internal transition. Thus the matrices for $0 \leq n \leq N - c$ are given by

$$\mathbf{D}_n = \begin{pmatrix} \mathbf{D}_n^{(0,0)} & \mathbf{D}_n^{(0,1)} & & & \\ \mathbf{D}_n^{(1,0)} & \mathbf{D}_n^{(1,1)} & \mathbf{D}_n^{(1,2)} & & \\ & & \ddots & & \\ & & & \mathbf{D}_n^{(c-1,c-2)} & \mathbf{D}_n^{(c-1,c-1)} & \mathbf{D}_n^{(c-1,c)} \\ & & & & \mathbf{D}_n^{(c,c-1)} & \mathbf{D}_n^{(c,c)} \end{pmatrix}$$

where

$$\begin{aligned} \mathbf{D}_n^{(m,m-1)} &= \mathbf{I}_{k^n} \otimes (\oplus_{i=1}^m \mathbf{b}) & 0 < m \leq c \\ \mathbf{D}_n^{(m,m)} &= -(N - m - n) \lambda \mathbf{I}_{k^n} \otimes \mathbf{I}_l^m \\ &+ (\oplus_{i=1}^n \mathbf{T} + \delta(c - m) (\oplus_{i=1}^n \mathbf{t}) \otimes \boldsymbol{\tau}) \otimes \mathbf{I}_l^m & 0 \leq m \leq c \\ &\quad + \mathbf{I}_{k^n} \otimes (\oplus_{i=1}^m \mathbf{B}) \\ \mathbf{D}_n^{(m,m+1)} &= (N - m - n) \lambda \mathbf{I}_{k^n} \otimes \mathbf{I}_l^m \otimes \boldsymbol{\beta} & 0 \leq m < c \end{aligned}$$

Here $\delta(\cdot)$ denotes the Kronecker function 3.72. This completes the specification of our retrial model with phase type service and retrial times. The main advantage is, that we only made use of matrices and matrix operators both well suited for implementation in a computer program. The same applies to the model itself. By recalling that phase type distributions may be used to approximate any absolutely continuous distribution arbitrarily close, this is indeed a powerful result. For a discussion on this and other retrial models we refer to the recent book by J.R. Artalejo and A. Gómez-Corral [29]. A model similar to the one presented here is given in the paper [25] by A.S. Alfa and K.P.S. Isotupa. The main difference between the two lies in the assumption of exponential retrial times.

Chapter 4

Model Parameter Estimation

In the former chapter we attempted to provide a mathematical description suited well enough to capture certain properties of real world situations. Usually there is a bias towards the application due to an implicit usage of domain specific knowledge during the modeling process. This certainly introduces some restrictions but also provides a more realistic model. In some cases additional degrees of freedom are gained by adding some screws and turnwheels for adjustment. In a mathematical model we use parameters for the same purpose. Obeying to the principle of Ockham's razor, we have to prevent the model from becoming overparametrized. Although more parameters provide a higher level of flexibility, they also require a larger amount of *prior knowledge* for proper adjustment. Part of this prior knowledge originates from measurements of the system under consideration. By collecting data and applying quantitative methods one is able to estimate the parameters of the model. It is the aim of this chapter to provide some insight into the methods available.

4.1 Data Analysis and Modeling Life Cycle

Model parametrization is part of a larger process called data analysis. Connecting the model to the life system under consideration, data analysis has become a vital part in performance engineering. It can be seen as a set of methods related to the following procedures

1. Data collection

2. Distribution identification
3. Parameter estimation
4. Distribution validation

Provided the key indicators have been specified and a suitable model has been chosen, one usually proceeds with the *data collection*. These data often provide a guess about the underlying distribution. One method is to visually compare the relative frequency plot with the graph of well known probability or density functions. Sometimes there is no choice, e.g. for purely memoryless models. In that case one can immediately proceed with the parameter estimation. Some methods for determining the numerical values of the parameters will be presented in the subsequent sections. Once these have been assigned, one usually has to validate the distribution. Although the true distribution is unknown, performing a *goodness-of-fit test* based on the collected data is a widely accepted procedure [27]. Examples are the *chi-square test* and the *Kolmogorov-Smirnov test*, whereas the former exhibits some sensibility to deviations from a normal distribution. The main idea is to compare the test statistic based on some deviation measure to a predefined threshold value. If exceeded, one can expect the true distribution not being properly represented by the approximation. There are several resolutions to this problem. Either one collects more data and attempts another estimate or another model is considered. Even if the approximation is accepted, the model might be inaccurate or invalid. Therefore it is advisable to perform an additional *model verification* step. If possible the model output should be compared to the corresponding measurements of the system under consideration. Otherwise a simulation model may be set up for verification. If there is a pattern in the deviations, the drift might become part of the model. This is called *model adaption*.

4.2 Concepts in Parameter Estimation

The estimation of parameters has become an important part of statistical inference. There are basically two distinct approaches available, *parametric* and *nonparametric estimation*. Whereas the former attempts to determine the parameters used in characterizing families of distributions from observations, the latter targets distributionfree methods. Almost all queueing models

incorporate some type of distribution assumption, our main interest lies in parametric methods. Parametric inferences may be further divided. Finding a suitable estimate for an unknown parameter is called *point estimation*. In *interval estimation* one attempts to find an interval containing the parameter of interest to some degree of confidence. If one wants to choose between two alternative hypotheses regarding a parameter, the methods applied belong to the domain of *hypothesis testing*. As the chapter name suggests we are mainly concerned with point estimation.

In the last twenty or thirty years, statistical inference has been related to decision theory. Researchers aimed to find the common ground of hypothesis testing, point and interval estimation. In fact they succeeded and created a theory based on the maximization of utility functions. Alternatively one may decide to minimize the corresponding loss function. Classic statistical decisions are often related to the minimization of the so called quadratic loss function, but it turned out, that results are not always reliable. Using an absolute loss function instead leads to the usage of medians rather than means, which often proves to be the more robust result. Robustness is often related to the deviation of the true distribution from the corresponding model assumption. We will not dive any further here and instead refer to the book of E.L. Lehman and G. Casella [140].

Another important topic in statistics is the idea of information, especially the one of prior information. When designing a model, one often aims to put as much prior knowledge as possible into the model to reduce the level of uncertainty. Being one approach, there is another called *Bayesian statistics*. Here the parameter of interest itself is considered as a random variable and any prior knowledge is assumed to be represented by its distribution often called the *prior distribution*. By an application of Bayesian methods, one arrives at the so called *posterior distribution*, which combines the features of the model with prior information. According to some loss functions, an appropriate estimator called the *Bayes estimator* is chosen. For quadratic loss functions, this is simply the mean of the posterior distribution. Although there is some type of infinite loop in the Bayesian idea, i.e. estimating the parameters of parameters etc., the related methods are extremely powerful.

Considering the estimate itself, there are a number of properties often mentioned in statistical texts. We will summarize some of the most important here. Assuming we have derived an estimator $\hat{\theta}$ for the unknown parameter

θ , the principle of *unbiasedness* says, that

$$\mathbb{E}\hat{\theta} = \theta$$

There is a weaker version stating that for an increasing number of data we want θ to approach the expected value in the limit. This is called *asymptotic unbiasedness*. Another important concept is the *consistency* of an estimator. A sequence of estimators is called consistent, if the probability for a difference between θ and $\hat{\theta}$ tends to zero provided the number of observations tends to infinity. Quadratic loss functions are related to the variance of a sequence of estimators. If an unbiased estimator minimizing variance everywhere is available, it is called *uniform minimum variance unbiased (UMVU)* estimator. Their existence is warranted under rather general conditions. The corresponding results are mainly due to C.R. Rao, D. Blackwell, E.L. Lehman and H. Scheffe [140][77]. Sometimes one is interested to represent a set of observations in compact form with any loss of information. This leads to the concept of sufficiency. A statistic T is *sufficient* for a family of distributions indexed by θ , if the conditional data distribution given $T = t$ is independent of θ , i.e.

$$\Pr \{Data | \theta, T = t\} = \Pr \{Data | T = t\}$$

The distribution restriction is indeed necessary, as without any structure the feature of compactness without loss cannot be preserved. As an example consider the arithmetic mean, which is sufficient for the class of exponential distributions. If a statistic cannot be compressed any further, we are talking about a *minimal sufficient* statistic [140][77].

4.3 Moment Estimation

4.3.1 Classic Approach

The *method of moments* is a technique for the construction of estimators by matching sample moments with the corresponding theoretical moments [77][141]. Assuming a random variable \check{X} , the r -th theoretical moment is given by $\mathbb{E}\check{X}^r$. If a random sample x_1, x_2, \dots, x_n is drawn for the random variable \check{X} , the related sample moments may be calculated from

$$m_r := \frac{1}{n} \sum_{i=1}^n x_i^r$$

Suppose, that \check{X} is distributed according to $F(\theta_1, \theta_2, \dots, \theta_p)$ and that we need to estimate the parameters $\theta_1, \theta_2, \dots, \theta_p$. By interpreting m_r as the estimate of the theoretical moments, i.e.

$$\mathbb{E}\check{X}^r = m_r$$

and knowing, that $\mathbb{E}\check{X}^r$ is well represented as a function of the unknown parameters

$$\mathbb{E}\check{X}^r = g_r(\theta_1, \theta_2, \dots, \theta_p)$$

approximate values for $\theta_1, \theta_2, \dots, \theta_p$ are found from solving the system of equations

$$\begin{aligned} m_1 &= g_1(\theta_1, \theta_2, \dots, \theta_p) \\ m_2 &= g_2(\theta_1, \theta_2, \dots, \theta_p) \\ &\vdots \\ m_P &= g_P(\theta_1, \theta_2, \dots, \theta_p) \end{aligned}$$

where $P \geq p$ is chosen to assure the uniqueness of the solution. Provided a solution exists, it is usually denoted by $\hat{\boldsymbol{\theta}} := (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_p)^T$. It is possible to apply a similar procedure to the *central moments*, e.g. for $p = 2$ this results in equating mean and variance.

Example 27 *Considering the exponential distribution, there is only one parameter λ (or μ) to be estimated. Assuming an exponentially distributed variable \check{T} we know, that*

$$\mathbb{E}\check{T} = \frac{1}{\lambda}$$

By equating the mean m_1 and $\mathbb{E}\check{T}$, one immediately arrives at the moment estimator $\hat{\lambda} = m_1^{-1}$. So an appropriate estimator for the exponential distribution is given by the reciprocal of the sample mean. This is another reason for the wide application of memoryless queueing systems - parameter estimates are almost immediately accessible from standard reports and spreadsheets.

Example 28 *Proceeding in a similar way we may calculate the moment estimators of the Erlang distribution with parameters k and λ . Since there are*

two parameters, also two equations are required, i.e.

$$\begin{aligned} m_1 &= \mathbb{E}\check{T} := \frac{k}{\lambda} \\ m_2 &= \mathbb{E}\check{T}^2 := \frac{k}{\lambda^2} \end{aligned}$$

Solving the above system of equations yields the moment estimators

$$\hat{\lambda} = \frac{m_1}{m_2}, \quad \hat{k} = \frac{m_1^2}{m_2}$$

Obviously the second estimator \hat{k} needs to be rounded after calculation, as the Erlang distribution assumes an integer number of phases. By allowing fractions to be used as well, one arrives at the rather general family of gamma distributions.

As can be seen from the examples, the method of moments is only applicable for a fixed number of unknown parameters. This is obviously not the case for the majority of phase type and matrix exponential distributions. One of the major features of the Erlang distribution is, while providing an arbitrary number of phases, that only two parameters are required. This also explains the appealing nature of the early method of stages as suggested by A.K. Erlang. By considering the extremes of the Erlang distribution, that is the deterministic and the exponential distribution, the coefficient of variation c is seen to range from 0 to 1. Therefore it may be well applied to approximate distributions with a coefficient of variation in the same range. A similar result holds for the hyperexponential distribution, but now for $c > 1$. Obviously some type of combination would result in a higher degree of flexibility. But the method of moments fails even for so simple distributions such as the hyperexponential and the Coxian unless a certain structure is assumed. More general, we have to insist on order k of a phase type / matrix exponential distribution to apply the above procedure to

$$\mathbb{E}\check{T}^n = (-1)^{n+1} n! \beta \mathbf{B}^{-(n+1)} \mathbf{b}$$

which has been derived before as expression 3.116 in the context of phase type queues. So without assuming a fixed number of phases, we are in desperate need for more advanced methods.

4.3.2 Iterative Methods for Phase Type Distributions

L. Schmickler's *mixed Erlang distributions for approximation (MEDA)* is one of two moment matching techniques for the approximation of the first three moments of a hyper-Erlang distribution. As stated in section 3.1.5, the family of hyper-Erlang distributions provides sufficient approximations to non-negative and continuous distributions in a certain mathematical sense. Starting with a two branch hyper-Erlang distribution described by

$$f(x) = \alpha_1 \frac{\lambda_1}{(k_1 - 1)!} (\lambda_1 x)^{k_1 - 1} e^{-\lambda_1 x} + (1 - \alpha_1) \frac{\lambda_2}{(k_2 - 1)!} (\lambda_2 x)^{k_2 - 1} e^{-\lambda_2 x}$$

the algorithm determines the best approximation by searching through all possible combinations of k_1 and k_2 [181]. The calculation of moments for a hyper-Erlang distribution with more branches is based on the results for one branch less. This results in a recursive procedure as described in [182]. The optimization algorithm is based on the *flexible polyhedron search*, which tracks down the solution of a nonlinear programming problem without the need for differential calculus. As for simple moment matching, continuous values are required for all parameters. This led Schmickler to suggest a mixture of gamma distributions instead of the hyper-Erlang distribution. Due to the similarities in structure transforming one into another is a simple task. As noted in [137], the MEDA algorithm performs quite well and reliable, as approximations are obtained in most cases. It exhibits a certain weakness in the reproduction of uniform, lognormal and multimodal matrix exponential distributions. It is highly advisable to carry out a visual inspection of the results. In fact, human interaction is necessary, although it is kept to a minimum level from a user's perspective.

Similar in performance and reliability is the *mixture of Erlang distributions fitting (MEFIT)* method. There are some improvements compared to MEDA, but at the cost of an increased user interaction. An expert is required for the specification of the number of Erlang branches and the corresponding orders [137]. Compared to the classic method of moments it appears, that the flexibility of estimating a variable number of parameters comes at a very high price. This is not necessarily true, as for simple distributions the narrow scope of the model increases the chance of a modeling error. Anyway, the need for processor resources prevents MEDA and MEFIT from being implemented in an automated environment. Choosing an appropriate method of estimation once again depends on the application domain and again we face

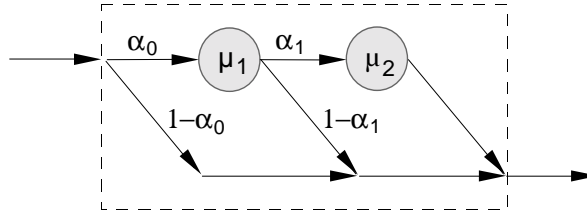


Figure 4.1: Cox distribution with two phases

the principle of Ockham's razor.

4.3.3 The EC_2 Method

The iterative methods described above have shown how to reduce the complexity of the parameter estimation problem by assigning more structure to the set of phase type distributions. In fact, the smallest class possible, i.e. the class of hyper-Erlang distributions has been chosen for that purpose. What might happen, if that class is further reduced ? Such a reasoning led T. Osogami and M. Harchol-Balter to what is called the EC_2 method [160]. By introducing a very specific family of phase type distributions, they managed to provide a numerically stable closed form approximation to a very large class of distributions. To be more specific, any distribution not subject to an approximation by an appropriate phase type distribution with coincidence of the first three moments is not required to possess a proper representation. However, the EC_2 method provides a good quality of results in terms of the number of moments matched.

Cox and Erlang-Cox distributions

The classic *Cox distribution* is in some way a generalization of the Erlang distribution. Imagined as conveyor belt it assigns different service rates to each work station. The second distinctive feature is the possibility to bypass all subsequent stations. By allowing the entire conveyor belt to be bypassed with a certain probability, one arrives at the *generalized Cox distribution*. As an example consider the two phase Cox distribution depicted in figure 4.1. Please note, that some authors follow a different classification for Cox distribution. As noted in section 3.1.5, the family of Cox distributions is

equivalent to the family of exponential distributions in serial/parallel. As such its members are indeed phase type distributions represented by

$$\mathbf{B} = \begin{pmatrix} -\mu_1 & \alpha_1\mu_1 & & & \\ & -\mu_2 & \alpha_2\mu_2 & & \\ & & \ddots & \ddots & \\ & & & -\mu_{k-1} & \alpha_{k-1}\mu_{k-1} \\ & & & & -\mu_k \end{pmatrix}$$

$$\boldsymbol{\beta} = (\alpha_0, 0, 0, \dots, 0)$$

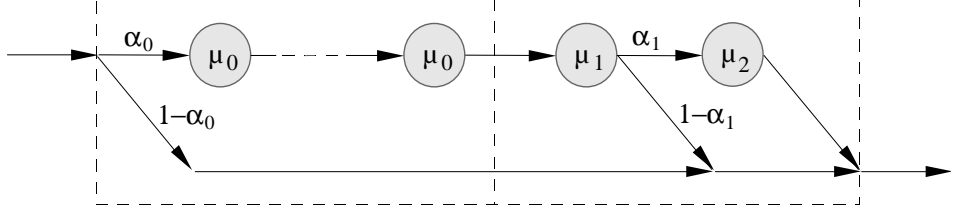
where k describes the number of phases. For the two phase Cox distribution shown in figure 4.1 we have

$$\mathbf{B} = \begin{pmatrix} -\mu_1 & \alpha_1\mu_1 \\ & -\mu_2 \end{pmatrix}, \quad \boldsymbol{\beta} = (\alpha_0, 0)$$

Obviously $\beta_{k+1} = 1 - \alpha_0$, that is the probability of immediate absorption. Please check out section 3.8.1 for the details on phase type distributions. Also note, that most queueing models assume $\beta_{k+1} = 0$, which is best dealt with by using the normalized vector $(1 - \beta_{k+1})^{-1}\boldsymbol{\beta}$ instead of $\boldsymbol{\beta}$ [153]. It is well known, that the Cox distribution is related to the hyperexponential distribution with respect to its capability to approximate distributions of high variability. In addition to that it can also assume an Erlang distribution, which is the proper selection in case of low variability. Furthermore the Erlang distribution has a fixed number of parameters. Combining features leads to a specific Cox distribution called the EC_2 distribution, which is well suited for approximation of non-negative (absolutely continuous) distribution functions. For a graphical representation refer to figure 4.2. As a matter of fact, the EC_2 distribution is fully determined by the set of parameters $(k, \alpha_0, \mu_0, \alpha_1, \mu_1, \mu_2)$ making it a good candidate for classic moment matching techniques.

Normalized Moments

Osogami and Harchol-Balter have chosen to represent their formulas in terms of *normalized moments*. They are dimensionless and allow for a representation of the distribution independent of any linear change in scale. In the

Figure 4.2: EC_2 distribution

present context we only need the second and third normalized moment, that is

$$m_F^{(2)} = \frac{\mu_F^{(2)}}{(\mu_F^{(1)})^2}, \quad m_F^{(3)} = \frac{\mu_F^{(3)}}{\mu_F^{(1)} \mu_F^{(2)}}$$

where $\mu_F^{(n)}$ denotes the n -th moment of the distribution function F . The second normalized moment is related to the squared coefficient of variation c_F^2 by

$$m_F^{(2)} = c_F^2 + 1$$

Also note the correspondence to more common concepts such as the skewness of a distribution, which may be revealed by simple algebraic manipulations. For more details we refer to the papers [159] and [160] by Osogami and Harchol-Balter.

Parameter Estimation

With respect to the quality of approximation the necessary conditions are easily expressed in terms of normalized moments. From above we know, that a given non-negative distribution F is well represented by an EC_2 distribution, if we can find a phase type distribution, which matches the first three moments. It can be shown, that this is the case for $m_F^{(3)} > m_F^{(2)} > 1$. Since any non-negative distribution F satisfies the condition $m_F^{(3)} \geq m_F^{(2)} \geq 1$, almost all non-negative distributions can be properly matched by an EC_2 distribution [160]. Now consider a Cox distribution with $k > 1$ phases, which provides a satisfactory approximation for an arbitrary non-negative (and absolutely continuous) distribution F with normalized moments $m_F^{(2)} > \frac{k}{k-1}$ and

$m_F^{(3)} \geq \frac{k+2}{k+1} m_F^{(2)}$ [159]. Note the coincidence of the limiting process with the above condition for the quality of approximation of an EC_2 distribution.

Reviewing the phase configuration of the EC_2 distribution in figure 4.2 we can identify two parts. The Erlangian part captures effects of low variability, whereas the C_2 distribution covers the opposite. Approaching the extreme the major contribution is provided by one part only. In order to achieve a minimal number of phases the best choice is to omit the insignificant part. From the conditions above we know, that F is well represented by a C_2 distribution for $m_F^{(2)} > 2$ and $m_F^{(3)} \geq \frac{4}{3} m_F^{(2)}$.

In considering the Coxian part alone and inspecting the effect of successively adding exponential phases, the first moment of the subsequent node always assumes the same value

$$\mu_0 = \frac{1}{\left(m_C^{(2)} - 1\right) \mu_C^{(1)}} \quad (4.1)$$

when constrained to a minimal second moment. The subscript C is used to denote the moments from the Coxian part of the EC_2 distribution. With the same value assigned to the intensity of each additional phase, we have encountered the reasoning for specifying an Erlang distribution rather than a serial arrangement with arbitrary rates as part of the EC_2 distribution. Furthermore a predefined value has been assigned to one of the parameters, which allows us to narrow down the set of eligible distributions to the one given by the parameter set $(k, \alpha_0, \mu_0, \alpha_1, \mu_1, \mu_2)$ with μ_0 as defined in expression 4.1. In other words, there is one parameter less to estimate. The possibility for this kind of reduction becomes evident from the fact, that multiple representations exist for the same phase type distributions.

The main achievement of Osogami and Harchol-Balter was the derivation of a closed form solution for the approximation of an arbitrary non-negative (and absolutely continuous) distribution F . By identifying the regions of approximation in terms of normalized moments and the corresponding candidate distribution, they proceeded to calculate the moment estimates for each candidate. Enhanced by some numerical stability considerations, the following procedure has been worked out in [160]

1. Calculate the first three moments $\mu_F^{(1)}$, $\mu_F^{(2)}$ and $\mu_F^{(3)}$ from a known distribution F or from the observations of an unknown distribution F .
2. Derive the normalized moments from $m_F^{(2)} = \frac{\mu_F^{(2)}}{(\mu_F^{(1)})^2}$ and $m_F^{(3)} = \frac{\mu_F^{(3)}}{\mu_F^{(1)} \mu_F^{(2)}}$.

3. If $m_F^{(3)} \leq 2m_F^{(2)} - 1$ proceed with step 6.

4. The number of phases is given by $k = \left\lceil \frac{3m_F^{(2)} - 2 + \sqrt{(m_F^{(2)})^2 - 2m_F^{(2)} + 2}}{2(m_F^{(2)} - 1)} \right\rceil$,

where $\lceil x \rceil$ denotes the ceiling of x .

5. Set $\alpha_0 = \frac{1}{2m_F^{(2)}} \left(\frac{k-1}{l-2} + \frac{k}{k-1} \right)$, $\mu_W^{(1)} = \frac{\mu_F^{(1)}}{\alpha_0}$, $m_W^{(2)} = \alpha_0 m_F^{(2)}$ and $m_W^{(3)} = \alpha_0 m_F^{(3)}$ and proceed with step 9.

6. Set $\alpha_0 = \begin{cases} \frac{(m_F^{(2)})^2 + 2m_F^{(2)} - 1}{2(m_F^{(2)})^2} & m_F^{(3)} > 2m_F^{(2)} - 1, \\ & (m_F^{(2)} - 1)^{-1} \text{ is integer} \\ (2m_F^{(2)} - m_F^{(3)})^{-1} & m_F^{(3)} < 2m_F^{(2)} - 1 \\ 1 & \text{otherwise} \end{cases}$.

7. Set $\mu_W^{(1)} = \frac{\mu_F^{(1)}}{\alpha_0}$, $m_W^{(2)} = \alpha_0 m_F^{(2)}$ and $m_W^{(3)} = \alpha_0 m_F^{(3)}$.

8. The number of phases for the Erlang part of the EC_2 distribution is

given by $k = \begin{cases} \left\lceil \frac{m_W^{(2)}}{m_W^{(2)} - 1} \right\rceil & m_W^{(2)} < 2, \quad m_W^{(3)} = 2m_W^{(2)} - 1 \\ \left\lceil \frac{m_W^{(2)}}{m_W^{(2)} - 1} + 1 \right\rceil & \text{otherwise} \end{cases}$, where $\lfloor x \rfloor$ denotes the floor of x .

9. Derive the Coxian second moment $m_C^{(2)} = \frac{(k-3)m_W^{(2)} - (k-2)}{(k-2)m_W^{(2)} - (k-1)}$ and the Coxian first moment $\mu_C^{(1)} = \frac{\mu_W^{(1)}}{(k-2)m_C^{(2)} - (k-3)}$.

10. Calculate the auxiliary variables

$$\begin{aligned} \nu &= (k-2) (m_C^{(2)} - 1) \\ \gamma &= \nu \left[k(k-1) (m_C^{(2)})^2 - k(2k-5) m_C^{(2)} + (k-1)(k-3) \right] \\ \delta &= \left[(k-1) m_C^{(2)} - (k-2) \right] \left[(k-2) m_C^{(2)} - (k-3) \right]^2 \end{aligned}$$

and the third Coxian moment $m_C^{(3)} = \frac{\delta m_W^{(3)} - \gamma}{m_C^{(2)}}$.

11. Calculate the auxiliary variables $u = \begin{cases} 1 & 3m_C^{(2)} = 2m_C^{(3)} \\ \frac{6-2m_C^{(3)}}{3m_C^{(2)}-2m_C^{(3)}} & \text{otherwise} \end{cases}$
 and $v = \begin{cases} 0 & 3m_C^{(2)} = 2m_C^{(3)} \\ \frac{12-6m_C^{(2)}}{m_C^{(2)}(3m_C^{(2)}-2m_C^{(3)})} & \text{otherwise} \end{cases}$.
12. Derive the Coxian parameters $\mu_1 = \frac{u+\sqrt{u^2-4v}}{2\mu_C^{(1)}}$, $\mu_2 = \frac{u-\sqrt{u^2-4v}}{2\mu_C^{(1)}}$, $\alpha_1 = \frac{\mu_2\mu_C^{(1)}(\mu_1\mu_C^{(1)}-1)}{\mu_1\mu_C^{(1)}}$ and the rate $\mu_0 = \frac{1}{(m_C^{(2)}-1)\mu_C^{(1)}}$ for the Erlang part of the EC_2 distribution.
13. The complete parameter set for the matching EC_2 distribution is now available as $(k, \alpha_0, \mu_0, \alpha_1, \mu_1, \mu_2)$.

As mentioned above, this representation is valid for $m_F^{(3)} > m_F^{(2)} > 1$. This is indeed a powerful procedure, which is also well suited for an application in real time environments. Due to the closed form of the solution, computer run time is easy to predict. But there is a word of warning. If one is concerned with the shape of the underlying distribution F , maximum likelihood methods might provide a better choice. In other words, moment based methods ignore the information carried by higher moments not considered in the estimation process. We have attempted to present the main ideas, which led to the above procedure and refer to the paper [160] for a rigorous derivation.

4.3.4 Fixed Node Approximation

The EC_2 Method by Osogami and Harchol-Balter suggested a powerful approximation for rather arbitrary non-negative distributions by means of exponential mixtures featuring a fixed number of parameters. In order to apply this approach in a phase type context, one has to accept node limits for the Erlang part of the corresponding Erlang-Cox distribution. This raises the demand for approximations with a fixed number of nodes and a fixed number of parameters.

Two Nodes

We already know, for a coefficient of variation greater than one an arbitrary non-negative distribution is well represented by a H_2 distribution in terms of the first two moments. Without the requirement of balanced means, i.e.

$$\alpha_1 \mu_1^{-1} = \alpha_2 \mu_2^{-1}$$

three moments would have been necessary to provide estimates for the three parameters of the hyperexponential distribution. The corresponding moment estimators are given by [212]

$$\begin{aligned}\hat{\alpha}_1 &= \frac{1}{2} \left(1 + \sqrt{\frac{c^2 - 1}{c^2 + 1}} \right) \\ \hat{\alpha}_2 &= \frac{1}{2} \left(1 - \sqrt{\frac{c^2 - 1}{c^2 + 1}} \right) \\ \hat{\mu}_1 &= 2\hat{\alpha}_1 m_1, \quad \hat{\mu}_2 = 2\hat{\alpha}_2 m_1\end{aligned}$$

where $c^2 = m_2 m_1^{-2} - 1$ is the squared coefficient of variation and m_1, m_2 are the empirical moments of the underlying data.

Even more can be achieved by considering a classic Cox distribution of order two, that is we assume $\alpha_0 = 1$. For a graphical representation please refer to figure 4.1 of the previous section. Due to an ingenious idea of D.R. Cox it became possible to represent any distribution of a non-negative random variable with $c^2 \geq \frac{1}{2}$ by a C_2 distribution in terms of the leading moments. By ignoring the physical interpretation and considering the mathematical limits only, he extended the range of the branching probability α_1 such that complex values are included as well. This concept also generalizes to Coxian distributions of higher order. Compared to the H_2 distribution and the EC_2 method, this type of C_2 distribution does not belong to the class of phase type distributions, but it is still of matrix exponential type. Accordingly one has to be careful in applying any algorithms designed for phase type configurations, when representing a part of it by a Coxian distribution with negative branching probabilities. However, any solution methods designated to matrix exponential models are not affected. Proceeding as before and selecting a specific C_2 distribution, i.e. the C_2 distribution with gamma normalization, the three corresponding parameters may be estimated using the

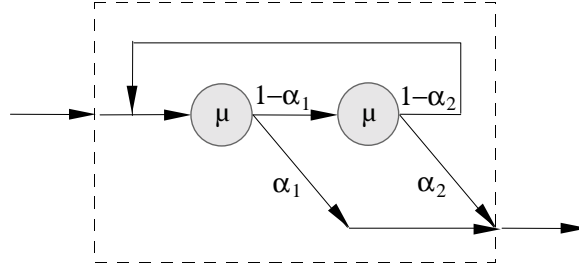


Figure 4.3: Two stage feedback node

first two moments only [201]. This yields

$$\begin{aligned}\hat{\mu}_1 &= \frac{2}{m_1} \left(1 + \sqrt{\frac{c^2 - \frac{1}{2}}{c^2 + 1}} \right) \\ \hat{\mu}_2 &= \frac{4}{m_1} - \hat{\mu}_1 \\ \hat{\alpha}_1 &= \hat{\mu}_2 m_1 - \frac{\hat{\mu}_2}{\hat{\mu}_1}\end{aligned}$$

where $c^2 = m_2 m_1^{-2} - 1$ is the squared coefficient of variation and m_1, m_2 are defined as before.

It remains to find a fixed node approximation for distributions with a squared coefficient of variation less than $\frac{1}{2}$. The Erlang distribution is insufficient for our purposes, because the more deterministic the distribution under consideration the more phases we need. By considering a feedback system of two exponential nodes S. Nojo and H. Watanabe managed to find a representation, which is also capable to replace the deterministic distribution in queueing models [156]. The corresponding configuration is shown in figure 4.3. Please note, that this again is only a conceptual model and that negative branching probabilities are allowed and that the same words of caution as for Coxian approach apply.

According to [156], the Laplace transformation of the lifetime a customer transits through the two node feedback queue is given by

$$\bar{f}(s) = \frac{\mu \alpha_1 s + K}{s^2 + 2\mu s + K} \quad (4.2)$$

where

$$K = \mu^2 (\alpha_1 + \alpha_2 - \alpha_1 \alpha_2)$$

Given the first three moments m_1 , m_2 and m_3 almost any non-negative distribution can be matched, if the condition

$$m_1 m_3 < \frac{3}{2} m_2^2 \quad (4.3)$$

is satisfied [201]. The moments of the Laplace transformation 4.2 are derived as follows [83]

$$\begin{aligned} \mu_F^{(1)} &= -\lim_{s \rightarrow 0} \frac{d\bar{f}(s)}{ds} = \frac{\mu(2 - \alpha_1)}{K} \\ \mu_F^{(2)} &= \lim_{s \rightarrow 0} \frac{d^2 \bar{f}(s)}{ds^2} = \frac{4\mu^2(2 - \alpha_1)}{K^2} - \frac{2}{K} \\ \mu_F^{(3)} &= -\lim_{s \rightarrow 0} \frac{d^3 \bar{f}(s)}{ds^3} = \frac{24\mu^3(2 - \alpha_1)}{K^3} - \frac{6\mu(4 - \alpha_1)}{K^2} \end{aligned} \quad (4.4)$$

Equating moments $m_r = \mu_F^{(r)}$, $r = 1, 2, 3$ and expressing the result in terms of the parameters yields the moment estimators

$$\begin{aligned} \hat{\alpha}_1 &= \frac{2(6m_1 m_2 - 6m_1^3 - m_3)}{3m_1 m_2 - m_3} \\ \hat{\alpha}_2 &= \frac{2(18m_1^4 m_2 + 15m_1 m_2 m_3 - 9m_2^3 - 18m_1^3 m_3 - m_3^2)}{(12m_1^3 - 9m_1 m_2 + m_3)(3m_1 m_2 - m_3)} \\ \hat{\mu} &= \frac{m_3 - 3m_1 m_2}{2m_1 m_3 - 3m_2^2} \end{aligned} \quad (4.5)$$

Please note, that $\hat{\mu} > 0$ by the above condition, while $\hat{\alpha}_1$ and $\hat{\alpha}_2$ might assume negative values for a small coefficient of variation. The latter is by no way an indication of a wrong result and so the resulting approximation may be readily applied as part of a matrix exponential configuration. To validate the former argument, consider expression 4.5. For a positive value of $\hat{\mu}$ we require $m_3 < 3m_1 m_2$. As we only deal with non-negative distributions, we may assume $m_1 \geq 0$ and write $m_1 m_3 < 3m_1^2 m_2$, where the trivial case has been excluded. Noting, that the variance $m_2 - m_1^2$ is always non-negative, this leads to $m_1 m_3 < 3m_2^2$, which is satisfied in a non-exhaustive way by condition 4.3. This also ensures a finite value for $\hat{\alpha}_1$. However, the current

reasoning does not extend to the first term in the denominator of $\hat{\alpha}_2$ and so we need to introduce another condition

$$3m_1 (4m_1^2 - 3m_2) + m_3 \neq 0$$

to warrant a finite value. For more details and an application to the $G/M/c/c$ queueing system please refer to the paper [156] by S. Nojo and H. Watanabe. Their approach differs from the one presented above, because the authors express the parameters $\hat{\mu}$, $\hat{\alpha}_1$ and $\hat{\alpha}_2$ in terms of mean, coefficient of variation and skewness.

In the literature one can find several fixed node approaches under various names. One example is the so called general exponential distribution, which is a variant of a generalized Cox distribution. However, in other fields the same term identifies a totally different class of distribution. Unfortunately there is no common understanding of these entities and so it remains to carefully inspect the corresponding definitions.

General Case

It is a well known fact, that any non-negative and absolute continuous distribution with coefficient of variation $c^2 \geq \frac{1}{k}$ is well approximated by a Cox distribution with k phases [159]. As shown in section 4.3.3, it is relatively easy to find the corresponding matrix exponential representation. The two node approximation already raises two important points, which should be considered carefully.

- Similar to the family of matrix exponential distributions, Cox representations are not unique. In other words, different sets of parameters may lead to the same distribution. By introducing a balance argument such as the gamma normalization used above, one may reduce the number of parameters. However, one has to be careful not to omit important members of the distribution family.
- The sharp borderline at $c^2 = \frac{1}{2}$ allowed us to perfectly separate feedback node approximation and Coxian approximation. This is not true for more than 2 fictitious phases, as there will be an overlap increasing with the number of phases k . For any distribution lying within that range it might be useful to consider additional matching criteria beyond moments. For example, one may focus on the shape of the distribution

and apply standard statistical tests such as the *Kolmogorov-Smirnov* or the *chi-square test*.

We will now approach the feedback node approximation from a more general perspective by deriving an expression for the rational Laplace transform (RLT) approximation, which is then represented by a matrix exponential distribution. In doing so, we gain two advantages. First, we are not limited to two nodes and second, we directly arrive at a matrix exponential representation. Let $g(\cdot)$ denote the density of the original non-trivial lifetime (non-negative and absolute continuous) distribution and $\bar{g}(s)$ the corresponding Laplace transform. Following the approach in [157], we expand the latter in a Taylor series

$$\bar{g}(s) = \gamma_0 + \gamma_1 s + \gamma_2 s^2 + \gamma_3 s^3 + \dots \quad (4.6)$$

where

$$\gamma_n = \frac{(-1)^n}{n!} \mu_n = \frac{1}{n!} \frac{d^n}{ds^n} \bar{g}(s) \Big|_{s=0}$$

for $n \geq 1$ and μ_n denotes the n -th moment. As $g(\cdot)$ is a lifetime distribution, we may safely assume, that $\gamma_0 = 1$. Before proceeding we need to introduce some new mathematical terms. Consider a fraction of the form

$$\frac{d_0}{e_0 + \frac{d_1}{e_1 + \frac{d_2}{e_2 + \dots}}}$$

and denote it by

$$\frac{d_0|}{|e_0|} + \frac{d_1|}{|e_1|} + \frac{d_2|}{|e_2|} + \dots$$

Such an expression is called *continued fraction*. Being an item of close investigation for centuries, continued fractions are rarely used today. However, they exhibit certain properties useful for approximation, which might be considered superior to classic series approximations. Certain types of continued fractions exist, but we will only need one of them, i.e.

$$\frac{d_0|}{|1|} + \frac{d_1 s|}{|1|} + \frac{d_2 s^2|}{|1|} + \dots \quad (4.7)$$

These so-called *C-fractions* [109] or *RITZ fractions* [95] are associated to the family of Moebius transformations (rotation, inversion, translation) and

are well suited for our purposes. Accordingly we want to express $\bar{g}(s)$ given by equation 4.6 as a RITZ fraction. This is best approached by the use of *Hankel's determinant*, i.e.

$$H_k^{(n)} = \begin{vmatrix} \gamma_n & \gamma_{n+1} & \cdots & \gamma_{n+k-1} \\ \gamma_{n+1} & \gamma_{n+2} & & \gamma_{n+k} \\ \vdots & & \ddots & \vdots \\ \gamma_{n+k-1} & \gamma_{n+k} & \cdots & \gamma_{n+2k-2} \end{vmatrix}, \quad H_0^{(n)} = 1$$

Please note the arrangement of the Taylor coefficients giving the determinant its special name. Based on the theory of continued fractions [95], we may now calculate the coefficients of expression 4.7 as follows

$$\begin{aligned} d_{2k-1} &= -\frac{H_k^{(1)} H_{k-1}^{(0)}}{H_{k-1}^{(1)} H_k^{(0)}} \\ d_{2k} &= -\frac{H_{k+1}^{(0)} H_{k-1}^{(1)}}{H_k^{(0)} H_k^{(1)}} \\ d_0 &= H_0^{(0)} = 1 \end{aligned} \tag{4.8}$$

where $k = 1, 2, \dots$. This leads to a continued fraction representation of the Laplace transform

$$\bar{g}(s) = \frac{d_0|}{|1} + \frac{d_1 s|}{|1} + \frac{d_2 s^2|}{|1} + \cdots$$

which is still exact. In order to arrive at an approximation, one simply truncates the continued fraction. This practice is well known from series approximations. We will terminate the continued fraction after $2k$ terms to arrive at the k -stage approximation

$$\bar{f}_k(s) = \frac{d_0|}{|1} + \frac{d_1 s|}{|1} + \frac{d_2 s^2|}{|1} + \cdots + \frac{d_{2k-1} s^{2k-1}|}{|1} \tag{4.9}$$

Now the RITZ fraction will be transformed into the ratio of two polynomials. As we are dealing with a Laplace transform this leads to a *rational Laplace transform* (RLT)

$$\bar{f}_k(s) = \frac{p_{2k}(s)}{q_{2k}(s)}$$

The algorithm is relatively simple and derived from the one given in [95] or [109] and works as follows

$$\begin{aligned}
 p_m(s) &= s d_{m-1} p_{m-2}(s) + p_{m-1}(s) \\
 q_m(s) &= s d_{m-1} q_{m-2}(s) + q_{m-1}(s) \\
 p_0 &= 0, \quad p_1 = d_0 = 1 \\
 q_0 &= 1, \quad q_1 = 1
 \end{aligned} \tag{4.10}$$

where $0 \leq m \leq 2k$. As a result we get an expression of the form

$$\bar{f}_k(s) = \frac{b_k s^{k-1} + b_{k-1} s^{k-2} + \dots + b_1}{s^k + a_k s^{k-1} + a_{k-1} s^{k-2} + \dots + a_1} \tag{4.11}$$

Before proceeding we note some properties of our intermediate result

- That $2k$ terms are required for an k -stage approximation is a consequence of the polynomial degrees of p_k and q_k , i.e. we have $\text{grad}(p_k) = \left\lfloor \frac{k-1}{2} \right\rfloor$ and $\text{grad}(q_k) = \left\lfloor \frac{k}{2} \right\rfloor$.
- From $d_0 = 1$ it follows, that $a_1 = b_1$ per construction.
- The n -th moment of $g(\cdot)$ equals the n -th moment of $f_k(\cdot)$ for $n = 1, \dots, 2k - 1$. So all our efforts resulted in finding an approximation, which coincides with the original functions in terms of the leading moments.

As a next step we aim to find a matrix exponential representation for this rational Laplace transform. At this point, a useful theorem found in the work of S. Asmussen and M. Bladt [32] comes in handy. We restate it here for convenience.

Theorem 29 *The Laplace transform of a general matrix exponential distribution can be written as*

$$\bar{f}(s) = \frac{b_k s^{k-1} + b_{k-1} s^{k-2} + \dots + b_1}{s^k + a_k s^{k-1} + a_{k-1} s^{k-2} + \dots + a_1}$$

for some $k \geq 1$ and some constants $a_1, \dots, a_k, b_1, \dots, b_k$. Then it has the following representation

$$f(x) = \boldsymbol{\alpha} \exp \{ \mathbf{A}x \} \mathbf{a}$$

where

$$\begin{aligned}
 \boldsymbol{\alpha} &= (b_1, \dots, b_k) \\
 \mathbf{A} &= \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_1 & -a_2 & -a_3 & \cdots & -a_{k-1} & -a_k \end{pmatrix} \\
 \mathbf{a} &= (0, 0, 0, \dots, 0, 1)^T
 \end{aligned} \tag{4.12}$$

Please note, that $a_1 = b_1$ and that $\mathbf{a} \neq -\mathbf{A}\mathbf{1}$. The latter fact is the reason for calling this representation $ME(\boldsymbol{\alpha}, \mathbf{A}, \mathbf{a})$ a general matrix exponential representation.

To prove the above result, an extensive set of algebraic manipulations has to be worked through. So we decided to omit the proof and instead refer to the original paper [32] for further details. In our introduction to matrix exponential distribution functions 3.8.1 we have already noted the correspondence between distributions with rational Laplace transforms and matrix exponential distributions. This fact is also underlined by the above theorem. To safely apply the result, we have to be certain, that $f(\cdot)$ determines a valid distribution. As a next step we need to derive the common representation $ME(\boldsymbol{\beta}, \mathbf{B})$ from $ME(\boldsymbol{\alpha}, \mathbf{A}, \mathbf{a})$. First noting, that matrix exponential representations are not unique, we may write the density given in the above theorem as

$$\begin{aligned}
 f(x) &= \boldsymbol{\alpha} \exp\{\mathbf{A}x\} \mathbf{a} \\
 &= \boldsymbol{\alpha} \mathbf{T} \exp\{\mathbf{T}^{-1} \mathbf{A} \mathbf{T} x\} \mathbf{T}^{-1} \mathbf{a}
 \end{aligned}$$

Introducing the familiar condition $\mathbf{b} = -\mathbf{B}\mathbf{1}$ and substituting leads to

$$\mathbf{T}^{-1} \mathbf{a} = -\mathbf{T}^{-1} \mathbf{A} \mathbf{T} \mathbf{1}$$

which is the same as

$$\mathbf{A}^{-1} \mathbf{a} = -\mathbf{T} \mathbf{1}$$

The left side exhibits the following structure

$$\begin{aligned}\mathbf{A}^{-1}\mathbf{a} &= \begin{pmatrix} -\frac{a_2}{a_1} & -\frac{a_3}{a_1} & -\frac{a_4}{a_1} & \cdots & -\frac{a_k}{a_1} & -\frac{1}{a_1} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \mathbf{a} \\ &= \left(\frac{1}{a_1}, 0, 0, \dots, 0 \right)^T\end{aligned}$$

which gives rise to an educated guess of the transformation matrix

$$\mathbf{T} = \begin{pmatrix} \frac{1}{a_1} & 0 & 0 & \cdots & 0 & 0 \\ 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{pmatrix} \quad (4.13)$$

Assuming for the moment, that $a_1 > 0$, we may readily calculate the parameters β and \mathbf{B} for the matrix exponential representation $ME(\beta, \mathbf{B})$ as follows

$$\begin{aligned}\mathbf{B} &= \mathbf{T}^{-1}\mathbf{A}\mathbf{T} \\ &= \begin{pmatrix} a_1 & -a_1 & 0 & \cdots & 0 & 0 \\ a_1 & -1 - a_1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ a_1 & -1 - a_1 & 0 & \cdots & 1 & 0 \\ a_1 & -1 - a_1 & 0 & \cdots & 0 & 1 \\ 1 + a_1 - a_2 & 1 - a_1 - a_2 + a_3 & a_4 - a_3 & \cdots & a_k - a_{k-1} & 1 - a_k \end{pmatrix} \\ \beta &= \alpha\mathbf{T} \\ &= \left(\frac{b_1}{a_1} + b_2, -b_2 + b_3, \dots, -b_{k-1} + b_k, b_k \right)\end{aligned} \quad (4.14)$$

It is easily verified, that the new exit vector $\mathbf{b} = \mathbf{T}^{-1}\mathbf{a}$ obeys the condition $\mathbf{b} = -\mathbf{B}\mathbf{1}$. Furthermore, the immediate exit rate is given by $b_{k+1} = 1 - \beta\mathbf{1} = 1 - \frac{b_1}{a_1} = 1 - d_0 = 1 - \bar{g}(0) = 0$. In other words, the above representation does not admit an immediate exit from the system. From $d_0 = 1$ we can also see, that $a_1 \neq 0$.

We now summarize the most important steps in deriving a matrix exponential distribution, which approximates a non-trivial lifetime distribution $g(\cdot)$. These are

1. Expand $g(\cdot)$ into a Taylor series 4.6.
2. Convert this series to a continued fraction 4.7 by iterating expressions 4.8.
3. Truncate the fraction after $2k$ terms to arrive at expression 4.9.
4. Convert the fraction to a ratio of polynomials 4.11 using algorithm 4.10.
5. Arrange the coefficients in matrix form to construct the parameter set 4.12 of a general matrix exponential representation $ME(\boldsymbol{\alpha}, \mathbf{A}, \mathbf{a})$.
6. Transform $ME(\boldsymbol{\alpha}, \mathbf{A}, \mathbf{a})$ to a standard representation $ME(\boldsymbol{\beta}, \mathbf{B})$ using expression 4.14 and transformation matrix \mathbf{T} given by 4.13.

Alternatively one may join the last two steps by inserting the coefficients of the rational Laplace transform into expression 4.14. For an algorithm featuring a fixed number of stages k , this might be optimal. When designing a computer program allowing for initial specification of an arbitrary k , the proposed steps are certainly a better choice, as \mathbf{T} is much simpler to construct than the final result.

Example 30 *This example follows the one given in the paper [157] by S. Nojo and G. Kimura. Although we have excluded the trivial case by assumption, we now assume a deterministic distribution $g(t) = \delta(t - a)$ with Laplace transform*

$$\bar{g}(s) = e^{-as}$$

the corresponding moments are given by

$$\mu_n = \left. \frac{d^n}{ds^n} \bar{g}(s) \right|_{s=0} = (-a)^n$$

Setting $a = 1$ leads to $\mu_n = 1$ for all $n \geq 1$ and to the Taylor series

$$\bar{g}(s) = 1 - s + \frac{1}{2}s^2 - \frac{1}{6}s^3 + \frac{1}{24}s^4 - \frac{1}{120}s^5 + \frac{1}{720}s^6 - \dots$$

We aim to find a 3-stage approximation, so the explicitly shown coefficients are sufficient for our purposes. Iterating the algorithm 4.8 results in a RITZ fraction of the form

$$\bar{f}_3(s) = \frac{1|}{|1} + \frac{s|}{|1} - \frac{\frac{1}{2}s^2|}{|1} + \frac{\frac{1}{6}s^3|}{|1} - \frac{\frac{1}{6}s^4|}{|1} + \frac{\frac{1}{10}s^5|}{|1}$$

Applying algorithm 4.10 leads to a rational Laplace transform

$$\bar{f}_3(s) = \frac{3s^2 - 24s + 60}{s^3 + 9s^2 + 36s + 60}$$

which in turn may be represented by the parameter set

$$\begin{aligned}\boldsymbol{\alpha} &= (60, -24, 3) \\ \mathbf{A} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -60 & -36 & -9 \end{pmatrix} \\ \mathbf{a} &= (0, 0, 1)^T\end{aligned}$$

Using the set of equations 4.14, this representation may be transformed to

$$\begin{aligned}\boldsymbol{\beta} &= (-23, 27, -3) \\ \mathbf{B} &= \begin{pmatrix} 60 & -60 & 0 \\ 60 & -61 & 1 \\ 97 & -88 & 8 \end{pmatrix}\end{aligned}$$

Using a computer program to carry out the numerical calculations we arrive at the following moments

$$\begin{aligned}\mu_1 &= 1, & \mu_2 &= 1, & \mu_3 &= 1 \\ \mu_4 &= 1, & \mu_5 &= 1, & \mu_6 &= 1.1\end{aligned}$$

As expected the first five moments coincide with their theoretical counterpart, while μ_6 already shows some deviation. Finally we give a look to the "density function" of the matrix exponential approximation presented by figure 4.4 (The solid line marks the location of the unit pulse). Obviously this is not a valid density as there do exist negative values. By truncating the continued fraction we have introduced an error too large to cope with. This underlines the necessity of a final check for a valid distribution.

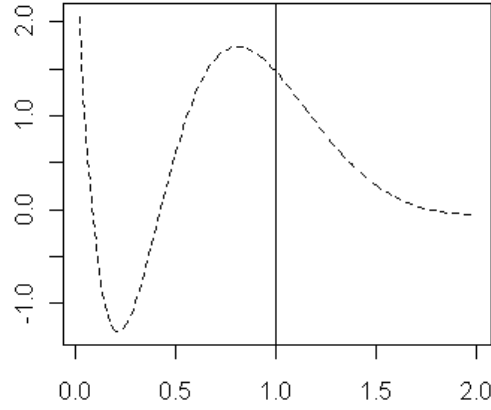


Figure 4.4: ME approximation to the deterministic distribution

Example 31 Consider a two-parameter gamma distribution with density function

$$g(t) = \frac{\lambda^\alpha t^{\alpha-1}}{\Gamma(\alpha)} e^{-\lambda t}$$

and Laplace transformation

$$\bar{g}(s) = \left(\frac{\lambda}{\lambda + s} \right)^\alpha$$

where α determines the shape of the distribution and λ is the rate parameter. Applying common calculus for Laplace transforms [83] yields the moments

$$\mu_n = \left. \frac{d^n}{ds^n} \bar{g}(s) \right|_{s=0} = \frac{\alpha(\alpha-1) \cdots (\alpha+n-1)}{\lambda^n}$$

Proceeding as in the previous example allows us to calculate matrix exponential approximations for various parameter combinations. We have used a numerical computer package to derive the matrix exponential representations for $\alpha = 0.5, \lambda = 2$ and $\alpha = 2, \lambda = 0.4$. For the former we have considered 4

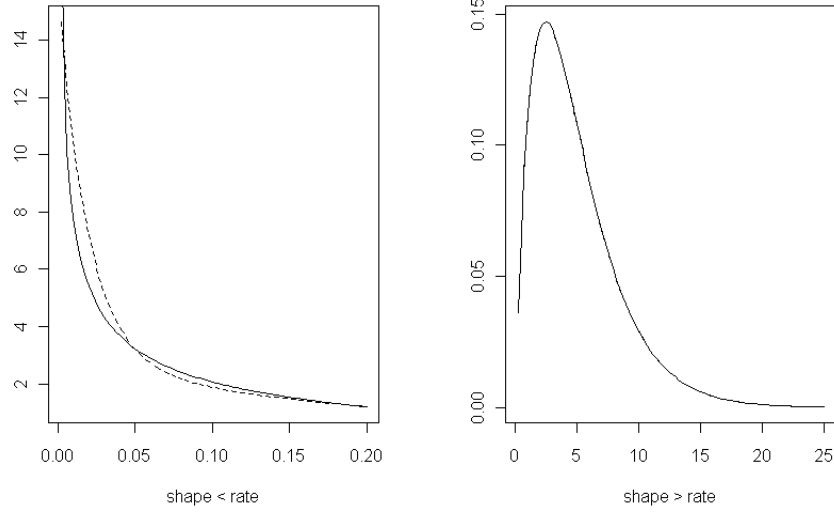


Figure 4.5: ME approximation to the gamma distribution

stages to be sufficient, whereas the latter has been based on 5 stages. Drawing the approximation against the original density yields a picture similar to figure 4.5. The quality of approximation is very good, especially for the second parameter set. In a wider sense we may view the first parameter set as being related to the hyperexponential distribution, while the second is easily associated to the Erlang distribution, both belonging to the class of Matrix exponential distributions. As opposed to these two, the shape parameter for the gamma distribution is allowed to assume real values as well. In exchange for this flexibility we cannot find a staged representation and so the gamma distribution becomes unsuitable for linear models and matrix calculus. However, by finding a closely related matrix exponential representation both advantages prevail for the little price of an approximation error.

Our approach differs radically from the one presented in [157], as the structure of \mathbf{B} in the result 4.14 does not resemble its counterpart in the paper. In fact, forcing a specific structure for \mathbf{B} leads to a superior performance in case of approximations to the deterministic distribution. An application of the results stated in the paper [157] by S. Nojo and G. Kimura allows one to

find a matrix exponential representation (called a *Markov process with negative branching probabilities* by the authors), which admits a valid probability distribution. Although both approaches share the same base of assumptions, they end up with different approximations. Except for some special cases, the corresponding representations do not belong to the same distribution. What both have in common, is the introduction of feedback nodes. This underlines its major role and confirms our intuition from the Erlang distribution. Recalling, that an accurate approximation to the deterministic distribution requires an excessive number of nodes, one might deduce a similar result by an application of a feedback system. Leaving the realm of the deterministic distribution, the procedure described above allows for a higher grade of flexibility. In practice one might want to leverage between our approach and the one developed by S. Nojo, H. Watanabe and G. Kimura in their papers. However, before using the results in stochastic models such as queueing systems one has to be aware to deal with valid probability distributions.

4.4 Maximum Likelihood Estimation

4.4.1 Classic Approach

One of the most commonly applied procedures in parameter estimation is the method of *maximum likelihood*. Although it is quite intuitive, the logic of *maximum likelihood estimation (MLE)* contains the seeds of a very flexible modeling strategy. Moreover, the resulting estimators possess certain good qualities such as consistency under rather general conditions [77][141]. The basic concept in maximum likelihood estimation is the so called *likelihood function*

$$l(\boldsymbol{\theta}, \mathbf{x}) \propto f(\mathbf{x} | \boldsymbol{\theta})$$

which is proportional to the joint density function $f(\mathbf{x} | \boldsymbol{\theta})$ of a random sample $\mathbf{X} = (X_1, X_2, \dots, X_n)$. To emphasize the dependence on a certain parameter set $\boldsymbol{\theta}$, it has been included in the notation. In case of identically and independent distributed data one often writes

$$l(\boldsymbol{\theta}, \mathbf{x}) \propto f(x_1 | \boldsymbol{\theta}) f(x_2 | \boldsymbol{\theta}) \cdots f(x_n | \boldsymbol{\theta})$$

Given a specific set $\mathbf{x} = (x_1, x_2, \dots, x_n)$ of n observations, we could ask for the most appropriate choice of $\boldsymbol{\theta}$. Provided such a value $\hat{\boldsymbol{\theta}}$ exists, it has to

assign the largest likelihood to the observed data set. Therefore we may find the *maximum likelihood estimator* $\hat{\boldsymbol{\theta}}$ by maximizing the likelihood function. Recapitulating, our prior knowledge is enhanced by the information provided through observations. This shows, why the likelihood function has also become an important concept of Bayesian statistics. Sometimes a monotone function is applied to the likelihood function before optimization. This often simplifies the calculations without changing the result. Due to the fact, that many distributions in statistics are given in terms of the exponential function, the logarithmic function is an appropriate choice. Therefore it has become a common practice to maximize the so called *log-likelihood function* $\ln l(\boldsymbol{\theta}, \mathbf{x})$.

Example 32 Consider the exponential distribution with density function $f(t) = \lambda e^{-\lambda t}, t \geq 0$. Assuming an identical and independent random sample t_1, t_2, \dots, t_n of size n , the likelihood function is given by

$$\begin{aligned} l(\boldsymbol{\theta}, t_1, t_2, \dots, t_n) &= \prod_{i=1}^n \lambda \exp\{-\lambda t_i\} \\ &= \lambda^n \exp\left\{-\lambda \sum_{i=1}^n t_i\right\} \end{aligned}$$

Maximization of the log-likelihood function yields

$$\begin{aligned} 0 &= \frac{d}{d\lambda} \left(n \ln \lambda - \lambda \sum_{i=1}^n t_i \right) \\ &= \frac{n}{\lambda} - \sum_{i=1}^n t_i \end{aligned}$$

leading to the maximum likelihood estimator

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^n t_i}$$

which is the reciprocal of the sample mean. By considering the second derivative $-\frac{n}{\lambda^2} < 0$ we can be assured of having achieved a maximum. Note the coincidence of $\hat{\lambda}$ with the corresponding moment estimator.

Although the above example suggests a certain coincidence with the estimators derived by other methods, they may differ under certain circumstances. Furthermore one has to be very careful in assigning properties such as unbiasedness to the maximum likelihood estimator. The best way for doing so is to classify the distribution function according to the assumptions of theorems common to mathematical statistics [140][77]. The principle of maximum likelihood has also been successfully applied to the theory of Markov processes by P. Billingsley in his book [43]. Although this treatment is rather theoretical in nature, the main ideas are well suited for an application in queueing theory.

4.4.2 EM Algorithm

If certain observations are missing or have been censored, the EM algorithm provides an iterative procedure for the calculation of the maximum likelihood estimator. Rather than performing the complex optimization task at once, the problem is reduced to a series of much simpler maximization steps (*M-steps*). During an additional expectation step (*E-step*) the complete data set for the subsequent M-step is computed from the observed data and the parameter values derived in the previous M-step. This results in a sequence of E- and M-steps, which explains the name of the algorithm. The objective function is usually given by the likelihood function assuming that all data are completely specified. If possible, the complexity can be further reduced by the consideration of sufficient statistics and transformations of the likelihood function. To prevent the iterations from running forever, an appropriate stopping criterion has to be selected. It is a common practice to alternate the E- and M- steps until the difference between subsequent likelihood values remains below an acceptable threshold [148].

As a matter of fact, the EM algorithm may also be applied to so called complete data problems providing an alternative to the classic maximum likelihood estimation. Considering some of its properties, the EM algorithm improves the result with every iteration. In case multiple stationary points such as saddle points and local maxima do exist, the convergence to a local maximum depends on the choice of the initial parameters. This is a problem common to optimization theory and not unique to the EM algorithm. Being a modern method, it is usually easy to implement and does not use much system memory. The classic EM algorithm converges at a linear rate, which is rather slow especially for a large amount of censored observations. Several

approaches have been proposed to increase the rate of convergence, for details we refer to [148].

Considering the problem of fitting a phase type distribution to an arbitrary non-negative distribution, the EM algorithm provides a suitable choice. With respect to the data set we assume, that n separate realizations \mathbf{y} of the underlying phase process have been recorded in terms of the number of processes M_i starting in state i , the total time Z_i spent in state i and the number of jumps N_{ij} from state i to state j . It can be shown, that the set $\{M_j, Z_j, N_{ij}\}$ for $1 \leq i \leq k$ and $0 \leq j \leq k$ constitutes a sufficient statistic for the continuous time Markov chain corresponding to the phase type distribution under consideration [36]. To denote the contribution of the m -th realization y_m the subscript $m \leq n$ will be added as in $\{M_{j,m}, Z_{j,m}, N_{ij,m}\}$. The familiar superscript notation emphasizes the iteration of the algorithm. Following the description of [31], the v -th E-step is given by

$$\begin{aligned} M_i^{(v)} &= \sum_{m=1}^n \mathbb{E} \left[M_{i,m} \mid y_m, \boldsymbol{\beta}^{(v-1)}, \mathbf{B}^{(v-1)} \right] = \sum_{m=1}^n \frac{\beta_i^{(v-1)} g_i(y_m \mid \mathbf{B}^{(v-1)})}{\boldsymbol{\beta}^{(v-1)} \mathbf{g}(y_m \mid \mathbf{B}^{(v-1)})} \\ Z_i^{(v)} &= \sum_{m=1}^n \mathbb{E} \left[Z_{i,m} \mid y_m, \boldsymbol{\beta}^{(v-1)}, \mathbf{B}^{(v-1)} \right] = \sum_{m=1}^n \frac{h_i(y_m; i \mid \boldsymbol{\beta}^{(v-1)}, \mathbf{B}^{(v-1)})}{\boldsymbol{\beta}^{(v-1)} \mathbf{g}(y_m \mid \mathbf{B}^{(v-1)})} \\ N_{ij}^{(v)} &= \sum_{m=1}^n \mathbb{E} \left[N_{ij,m} \mid y_m, \boldsymbol{\beta}^{(v-1)}, \mathbf{B}^{(v-1)} \right] \\ &= \begin{cases} \sum_{m=1}^n \frac{B_{i0}^{(v-1)} f_i(y_m \mid \boldsymbol{\beta}^{(v-1)}, \mathbf{B}^{(v-1)})}{\boldsymbol{\beta}^{(v-1)} \mathbf{g}(y_m \mid \mathbf{B}^{(v-1)})} & j = 0 \\ \sum_{m=1}^n \frac{B_{ij}^{(v-1)} h_j(y_m; i \mid \boldsymbol{\beta}^{(v-1)}, \mathbf{B}^{(v-1)})}{\boldsymbol{\beta}^{(v-1)} \mathbf{g}(y_m \mid \mathbf{B}^{(v-1)})} & j > 0 \end{cases} \end{aligned}$$

where $f_i(\cdot)$, $g_i(\cdot)$ and $h_i(\cdot)$ are elements of the k -dimensional vector functions $\mathbf{f}(\cdot)$, $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$ defined by

$$\begin{aligned} \mathbf{f}(y \mid \boldsymbol{\beta}, \mathbf{B}) &= \boldsymbol{\beta} \exp \{ \mathbf{B} y \} \\ \mathbf{g}(y \mid \mathbf{B}) &= \exp \{ \mathbf{B} y \} \mathbf{b} \\ \mathbf{h}(y; i \mid \boldsymbol{\beta}, \mathbf{B}) &= \int_0^y \boldsymbol{\beta} \exp \{ \mathbf{B} u \} \mathbf{e}_i \exp \{ \mathbf{B} (y - u) \} \mathbf{b} du \end{aligned}$$

for $1 \leq i \leq k$ and $0 \leq j \leq k$, $i \neq j$ and \mathbf{e}_i describing the i -th unit vector. As the values of $\boldsymbol{\beta}$ and $\mathbf{B} = (B_{ij})$ are known for each iteration, these functions

can be shown to satisfy the linear system of homogenous differential equations [31]

$$\begin{aligned}\mathbf{f}^T(y|\boldsymbol{\beta}, \mathbf{B}) &= \mathbf{f}(y|\boldsymbol{\beta}, \mathbf{B}) \mathbf{B} \\ \mathbf{g}^T(y|\mathbf{B}) &= \mathbf{B} \mathbf{g}(y|\mathbf{B}) \\ \mathbf{h}^T(y; i|\boldsymbol{\beta}, \mathbf{B}) &= \mathbf{B} \mathbf{h}(y; i|\boldsymbol{\beta}, \mathbf{B}) + f_i(y|\boldsymbol{\beta}, \mathbf{B}) \mathbf{b}\end{aligned}$$

for $1 \leq i \leq k$. The related initial conditions are given by

$$\begin{aligned}\mathbf{f}(0|\boldsymbol{\beta}, \mathbf{B}) &= \boldsymbol{\beta} \\ \mathbf{g}(0|\mathbf{B}) &= \mathbf{b} \\ \mathbf{h}(0; i|\boldsymbol{\beta}, \mathbf{B}) &= \mathbf{0}\end{aligned}$$

The above system of equations can be solved numerically by an appropriate algorithm such as the Runge-Kutta method. One has to be careful in using certain eigenvalue based methods, as this might result in numerical instabilities due to almost equal eigenvalues. Such problems are known to occur even for the Erlang distribution. As might have been expected by the reader, the E-step constitutes the most difficult part of our phase type fitting problem. The corresponding M-step works out relatively simple as

$$\begin{aligned}\boldsymbol{\beta}^{(v)} &= \frac{M_i^{(v-1)}}{n} \\ B_{ij}^{(v)} &= \begin{cases} \frac{N_{ij}^{(v)}}{Z_i^{(v)}} & i \neq j \\ -\sum_{j=0, i \neq j}^k B_{ij}^{(v-1)} & i = j \end{cases}\end{aligned}$$

where the above expressions are nothing else than the maximum likelihood estimator for the underlying Markov process [36][43]. If the distance measures $\|\mathbf{B}^{(v)} - \mathbf{B}^{(v-1)}\|$ and $\|\boldsymbol{\beta}^{(v)} - \boldsymbol{\beta}^{(v-1)}\|$ have become sufficiently small, the algorithm ends providing an approximation for the maximum likelihood estimators $\hat{\mathbf{B}} \approx \mathbf{B}^{(v)}$ and $\hat{\boldsymbol{\beta}} \approx \boldsymbol{\beta}^{(v)}$.

The above procedure has been implemented as a computer program known as *EMPHT* [158]. Although the number of phases have to be defined by the user in advance, EMPHT is not as limited as other methods. In fact, it is one of very few methods providing enough flexibility to deal with general phase configurations. When compared to moment based techniques such as MEDA and MEFIT, EMPHT is concerned with the approximation of the shape of a

distribution rather than with the matching of moments. One has to accept a certain relative error in the second and third moments when working with EMPHT. Especially for long tailed distributions large deviations have been recorded [137].

4.5 Bayesian Analysis

So far we have learned about the posterior distribution as a combination of prior knowledge and the information carried by observation. The former is specified in terms of a prior distribution, whereas the latter is represented by the likelihood function, which we have already encountered in section 4.4 as the central element of maximum likelihood estimation. Following the description there we will use the symbol $l(\boldsymbol{\theta}, \mathbf{x})$ for the likelihood function of an unknown parameter $\boldsymbol{\theta}$ provided a data set \mathbf{x} has been observed. Denoting the prior density with $\pi(\boldsymbol{\theta})$, we may apply the Bayes theorem to derive the posterior density

$$\pi(\boldsymbol{\theta} | \mathbf{x}) = \frac{\pi(\boldsymbol{\theta}) l(\boldsymbol{\theta}, \mathbf{x})}{\int \pi(\boldsymbol{\theta}) l(\boldsymbol{\theta}, \mathbf{x}) d\boldsymbol{\theta}} \quad (4.15)$$

As the marginal density $m(\mathbf{x}) = \int \pi(\boldsymbol{\theta}) l(\boldsymbol{\theta}, \mathbf{x}) d\boldsymbol{\theta}$ plays the role of a normalization constant, we may also write

$$\pi(\boldsymbol{\theta} | \mathbf{x}) = \frac{\pi(\boldsymbol{\theta}) l(\boldsymbol{\theta}, \mathbf{x})}{m(\mathbf{x})} \propto \pi(\boldsymbol{\theta}) l(\boldsymbol{\theta}, \mathbf{x}) \quad (4.16)$$

In other words, the posterior density is proportional to the product of prior density and likelihood function. In some way expression 4.16 can be seen as the heart of Bayesian statistics. It provides a good starting point for the analysis of many challenging problems. For example, one could ask for the structure of $\pi(\boldsymbol{\theta})$, if no prior information is available. We will not discuss this issue here and refer to the book [173] for some answers. We are more interested in how to efficiently build the posterior distribution assuming that we have prior knowledge. Obviously there is a brute-force method, namely selecting an appropriate prior distribution and applying expression 4.15 right away. Considering the wealth of available distributions this immediately raises the demand for a suitable choice. This leads to the concept of a *conjugate prior distribution*. Using expression 4.16 with a conjugate prior density yields a posterior distribution of the same type. In other words, only

the parameters change. These parameters are often referred to as *hyperparameters* to emphasize their static role. One might consider hyperparameters as parameters in a separate analysis, so the true Bayesian is only aware of random variables and does not know anything like a parameter. Denoting the set of hyperparameters by ω , expression 4.16 becomes

$$\pi(\theta | \omega^*) \propto \pi(\theta | \omega) l(\theta, \mathbf{x})$$

where ω^* is determined from ω and the data set \mathbf{x} . In some cases the entire data information is carried by a suitable statistic, which may replace the entire set of observations without loss. This results in a reduction of dimension and often leads to simpler calculations. For an illustration of the above concepts consider the following example

Example 33 *Given n independent realizations $\mathbf{x} = (x_1, x_2, \dots, x_n)$ of an underlying exponential distribution with parameter λ , the corresponding likelihood function becomes*

$$l(\lambda, \mathbf{x}) = \prod_{i=1}^n \lambda \exp\{-\lambda x_i\} = \lambda^n \exp\left\{-\lambda \sum_{i=1}^n x_i\right\}$$

As the average value $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ constitutes a sufficient statistic for the family of exponential distributions, we may also write

$$l(\lambda, \bar{x}) = \lambda^n \exp\{-\lambda n \bar{x}\}$$

Considering a variant of the Erlang distribution as a prior distribution, i.e.

$$\pi(\lambda | k, \mu) = \frac{\mu^k}{(k-1)!} \lambda^{k-1} \exp\{-\lambda \mu\}$$

the posterior density is given by

$$\begin{aligned} \pi(\lambda | \bar{x}) &\propto \pi(\lambda | k, \mu) l(\lambda, \bar{x}) \\ &= \frac{\mu^k}{(k-1)!} \lambda^{k-1} \exp\{-\lambda \mu\} \lambda^n \exp\{-\lambda n \bar{x}\} \\ &\propto \lambda^{k+n-1} \exp\{-\lambda(\mu + n \bar{x})\} \end{aligned}$$

where we have omitted the constant $\frac{\mu^k}{(k-1)!}$. This result is of the same structure as the prior density and so we have

$$\pi(\lambda | \bar{x}) = \pi(\lambda | k + n, \mu + n \bar{x})$$

That is an Erlang density function with parameters $k + n$ and $\mu + n \bar{x}$.

Please note, that there may be more than one choice for the conjugate prior distribution. In the above example the gamma distribution also constitutes an appropriate choice. In some cases the search for a suitable conjugate prior distribution may become a tricky problem, as common notation prevents some hidden structures to be revealed.

At the time of writing there has been no evidence in the literature, that Bayesian methods have been applied in the context of general phase type distributions. In case of discrete phase type distributions the theory for discrete time Markov chains developed by Martin in [145] provides a solid foundation. Consider a discrete phase type distribution described by the two parameters β and \mathbf{B} . Martin managed to identify the conjugate prior distribution for \mathbf{B} as *matrix beta distribution* provided the likelihood function has been assembled from data stated in terms of transition counts. In other words, each row of the transition probability matrix follows a *Dirichlet* (aka multivariate beta) distribution. Note, that the matrix of transition counts forms a sufficient statistic, which conveys all the information of the sample. Assuming an unknown initial state as well, β also becomes a random variable and subject to Bayesian reasoning. The transition count matrix is now only marginally sufficient and the conjugate prior density needs to be enhanced to reflect the inclusion of β . The result has been assigned the name *matrix beta-1 distribution*.

If only state counts have been observed, the global balance principle may be employed to uncover a relation between equilibrium and transition probabilities. Replacing the former with the corresponding observed proportions and adding an error component leads to a linear statistical model, which can be solved by conventional methods such as least squares or maximum likelihood estimation. However, neither method necessarily provides probabilities as a result. Obviously some type of restriction needs to be imposed.

A similar approach for a different problem has been exercised by T.S. Ferguson in his paper [78]. His concern was to find conjugate prior distributions for probability measures. For the current section, imagine a probability measure as a generalization of probability function and density function. By introducing the notion of a tailfree process he has been able to include the continuous case as well. It is conjectured by the author, that a similar reasoning also leads to results for the Bayesian analysis of continuous phase type distributions.

With respect to the unavailability of observations we have to note, that the EM algorithm is also a common choice for the iterative computation of

Bayes estimators. Provided the necessary theory for the Bayesian analysis of phase type distributions has been laid out, it is indeed a potential candidate for a practical implementation.

4.6 Concluding Remarks

So far we have presented some mainstream methods providing estimates for more or less structured phase type distributions. Our aim was to present some ideas rather than compiling an exhaustive list of techniques. Therefore we have omitted some important contributions such as [133] and [186]. In both cases, the authors attempted to fit a Cox distribution to the realizations of an unknown distribution. In [133] the Cox distribution function is represented in terms of divided differences to avoid the problems caused by the extreme sensitivity of exponential sums. The quality of approximation is measured by the mean squared difference between the empirical and the approximating Cox distribution. The importance of the first few moments has been emphasized by adding a penalty term to the objective function. This results in a nonlinear programming problem, which is solved by a variant of gradient minimization. In [186] an evolutionary algorithm has been developed to fit the empirical distribution function to a Cox distribution with a predefined number of phases.

Another interesting contribution called *PhFit* has been proposed by A. Horváth and M. Telek. As a combined algorithm it separately approximates body and tail of the distribution under consideration. Whereas the body is fitted according to some distance measure provided by the user, the approximation of the tail part is based on a heuristic method developed by Z. Feldman and W. Whitt. The latter captures heavy tail behaviour by a mixture of exponential distributions and has been adapted to suit the needs of the PhFit framework. PhFit is not limited in the choice of phase configuration and accepts either a discrete or a continuous phase type distribution as an approximating distribution. For more information, we refer to the paper [100] and the references stated therein.

Although the exceptional paper [137] by A. Lang and J.L. Arthur provides a qualitative survey of almost all important methods, we have also learned, that the optimal method has not been found yet. In fact, we have to keep the balance between matching the moments or capturing the shape of the target distribution. So the choice of an appropriate algorithm still depends

on the application. With respect to the context of queueing theory, certain models are known to exhibit some kind of insensibility for the choice of the service distributions. The most prominent example is the $M/G/c/c$ loss system, which depends on mean values only. Another example is the $M/G/1$ queue, which depends on the service distribution only through the first two moments. This can be easily seen from the Pollaczek-Khintchine formula 3.71. Certainly, in both cases we would not even attempt to estimate the parameters and instead apply the sample moments right away. But they indicate, that moment based techniques provide a good starting point if one is interested in certain performance indicators. However, this does not free us from performing one or more verification steps to judge the quality of the chosen method in a specific environment.

Chapter 5

Interconnection of Call Centres

5.1 Types of Call Centre Integration

5.1.1 Distributed Call Centres

A *distributed call centre* uses network-based routing to interconnect disparate locations. Multiple sites are then managed as a single entity with universal call transferability. The implementation of distributed call centres is motivated by

- *Load sharing* - calls can be redistributed from a heavy loaded call centre to underutilized call centres. Very often static algorithms are implemented, so calls are transferred according to a transfer-and-forget rule. Some call centres perform *look aheads* before transferring the call, but in no case the call control remains in the originating call centre.
- *Backup* - In case of malfunction or outage of one of the components of a distributed call centre, calls can be rerouted to other sites.
- *Skill allocation* - If a call desires a certain type of expert, but the call centre does not have such an expert available, the call can be shifted to an available expert anywhere in the organization. Again, this is not a dynamic process, agents located at other sites have to be administered as *remote agents* at the originating call centres.

In being aware of the limitations, distributed call centres can still be a useful tool to implement. To provide a dynamic selection of skills, a kind of

expert negotiation has to be deployed between call centre locations. Further enhanced signaling can be also useful to overcome some problems of distributed resource management. In case proper signaling mechanisms together with a common control are introduced, the distributed call centre gets very close to become a virtual call centre.

5.1.2 Virtual Call Centre

A virtual call centre uses location-independent agents who telecommute from individual locations. Due to the unpredictable nature of equipment used by agents and supervisors, a generic interfacing approach has to be used. This can be achieved by a web-enabled PC in combination with a headset or a separate telephone. With the headset connected to the PC, voice over IP technology is employed to provide voice and data access over a single telephone line.

In order to overcome the limitations of distributed call centres described above in 5.1.1, a common control communicating with all involved call management components based on a client-server concept has to be introduced. To provide a high grade of reliability, one client can be promoted as a server. By taking over control this new server is now able to handle further client requests. An outage can be detected by the use of keep-alive messages, so that traffic can be diverted from the malfunctioning client to an operational one by the server. If the server itself is affected, it has to be replaced. To prevent data loss in such a situation, a backup server may be promoted together with the primary server. In case of a server outage, the backup server becomes the primary one, a new backup server is promoted and data are replicated between primary and backup server. A similar client-server concept is implemented in Microsoft Windows NT [1] as domain concept with browser service.

5.2 Signaling Systems

Signaling protocols may be shown from a user or network perspective. As the majority of protocols introduced in this section deals with access interfaces to the public network, the network view of signaling protocols is presented with an introduction for common channel signaling protocols in chapter 7. Even QSig, a protocol designed for interoperation of private branch exchanges may

be added to that group, as it heavily relates to the ISDN protocol. Limitations are similar, e.g. lack of operations and maintenance layers. In small to medium sized private networks, QSig has major advantages over public networking protocols. Implementation complexity is not as high as for protocol stacks like *Common Channel Signaling Protocol No.7* used in public networks. With IP based multimedia networks, splitting user and network view becomes more philosophical and less functional. Almost the same mechanisms are applied for communication between servers and clients. But TCP/IP does not provide quality of service guarantees, so a lot of work has been carried out to specify reliable transport mechanisms for several multimedia classes. When signaling information is affected by user data, IP based communication facilities may become rather unuseable due to unpredictable effects.

Transport protocols may be thought of as the heart of every communication, whereas signaling relates to the brain and the nerves. Major roles from a user perspective include:

- *User location* - Before a session can be established, the user locations have to be determined. In classic telephony, users are related 1:1 to their telephone number and are easily found. With the introduction of intelligent network services like number portability in the public network, this task becomes more sophisticated, as it usually involves several database queries. In IP networks, users may be assigned a temporary address using the *dynamic host configuration protocol (DHCP)*. As before several requests to directory servers may become necessary to serve the tasks of address resolution and user location.
- *Session establishment* - Before a session can be established, various conditions have to be considered. In classic telephony, calls may be blocked or redirected. Or consider a multimedia application, which demands a voice channel to be set up prior to the establishment of a video channel.
- *Session negotiation* - Modern networks often provide several bearer channels with different features and varying quality of service. Obviously some kind of negotiation has to take place according to pre-defined user preferences. This operation is usually carried out during session establishment. For example consider the ISDN audio bearer

service, which allows for sample rates of 3.1 kHz and 7kHz. For multimedia networks session negotiation becomes even more important, as demands for multiple media streams need to be satisfied and aligned with each other. As an example consider a combined voice and video transmission with lip synchronization. Also note the fact, that terminals with different capabilities want to get involved in a conversation, e.g. when a voice call is placed to a video phone. The process of finding a common base of understanding between them is also known as *capabilities exchange*.

- *Session management* - In classic telephony signaling information could not be exchanged during the conversation. With touch tones transmitted within the same frequency band as speech, no concurrent usage is possible. Features may be invoked by initiating a short interruption of line current (called a *flash*) followed by touch tone codes identifying the required service. Almost every other protocol mentioned in this chapter maintains a concurrent form of signaling, which enables the caller to place a request during conversation. However, more complex features such as alteration of quality of service, new members added to a session and existing members removed from a session, also demand for an advanced management functionality.
- *Feature invocation* - As shown above, features and services become an essential part of modern communication networks. The signaling system has to provide the functionality for invocation of these features and services. When not central to the basic communication abilities of a protocol, these features are often referred to as *supplementary services*.

In the following sections, we will discuss some of the most important signaling protocols in more detail. They will range from classic telephone networks to IP based multimedia networks. We have also chosen to include an account on the *extensible messaging and presence protocol (XMPP)*, which has been designed for instant messaging and presence systems. It shall serve as an example for a protocol, which goes beyond session handling by providing sufficient means for a unified transport method of signaling and application information in complex call centre configurations.

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Table 5.1: Multifrequency tones

5.2.1 Traditional Signaling

Traditional Signaling is mainly based on changes in line current and transmission of pulses or multifrequency tones. Two types of signaling are distinguished [4]

- *Inband signaling* - The same channel is used to convey user data and signaling information. A good example is the classic telephone call using a touch tone telephone. Signaling information and speech are transmitted in the same frequency band. Due to the simple structure of classic telephone calls, inband signaling remained sufficient for a long time. This changed with the introduction of advanced services and the possibility to activate some of them during an active call.
- *Outband signaling* - Different channels or frequency bands are used for transmission of user data and signaling information. Therefore, no interference can occur during the active phases of a call.

Although network infrastructure of many european countries is completely digital, analog equipment still plays a major role for user access and some interfaces used in the telephony field. The latter includes analog interfaces between call centre components such as switch and IVR. Signaling information is constituted of digits which are represented by multi frequency tones. In order to avoid failures in detection, two non-interfering frequencies are created at the same time. At minimum digits 0-9 had to be represented giving a mix of 4×4 possible combinations. All digits and related frequencies for Europe are shown in table 5.1.

The end user has been informed by *answer tones* about the current call state. These answer tones are still used today, but instead of being sent by the local exchange, they are generated by terminal equipment such as ISDN telephones. Very common tones include the *dial tone*, a continuous

tone and the *alerting signal*, an interrupted tone. Some of the signals used are based on the Morse alphabet, such as the *engaged signal*, which relates to Morse-E and *intercept tone* related to Morse-I [17]. Almost every analog equipment, e.g. fax and modem, has the ability to generate and detect special answer tones. The detection of answer tones also plays an important role in certain outbound call centre applications, i.e. predictive dialing. After being connected to the dialed number, the system has to distinguish fax, modem, answering machine from human voice to allocate the call to the desired resource.

5.2.2 ISDN

ISDN delivers a framework for transmission of various media streams including voice, video and data at high speeds over standard communication lines. Furthermore, it provides an open architecture, which allows for ongoing introduction of new services to the network. As far, as the user is concerned, these services are called *supplementary services*. By general, ISDN evolved from the traditional telephone network, which is focused on circuit switched operation. This is also true the majority of connections set up in an *integrated services digital network (ISDN)*. Even X.25 based packet oriented communication supported by ISDN requires a transparent channel to be set up by the network. In other words, the network does not know, that X.25 packets are transmitted. This has changed with the introduction of frame based bearer services, which moved packet oriented and circuit oriented operation closer together.

Although ISDN is an accepted framework, the national standards differ from country to country. The organization responsible for producing the ISDN standards is the *International Telecommunication Union - Telecommunication Standardization Sector (ITU-T)*, former called *Comité Consultatif International Télégraphique et Téléphonique (CCITT)*. The first set of ISDN recommendations was published in 1984 as *red books*. Prior to this publication, various geographical areas developed different versions of ISDN. After standardization, the *codeset* mechanism provided the degree of freedom required by these areas. A set of proprietary information elements was simply grouped into codesets. Examples of ISDN variants used in the United States include Bellcore's National ISDN 1 & 2 and AT&T's ISDN 5ESS. In Europe a variant called *Euro-ISDN* specified by the *European Telecommunications Standards Institute (ETSI)* has been adopted by almost all countries. Be-

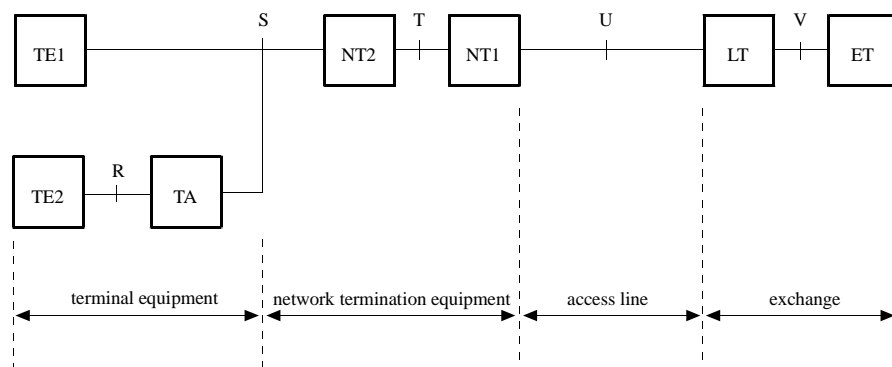


Figure 5.1: ISDN reference configuration

side Euro-ISDN, some countries still run national ISDN, examples are 1TR6 (Germany), VPN3/4 (France) and DASS-2 (United Kingdom). The following text will be focused on the specifications of the ITU-T and ETSI, differences will be explicitly stated.

Services

According to the *CCITT Blue Book*, the telecommunications services delivered by ISDN are divided into the following categories [8]:

- *Bearer services* describe the network transmission characteristics offered by the network at the interfaces between subscriber terminal and network. According to the ISDN reference model (see figure 5.1), these interfaces are specified for reference points S and T. In Europe, customer premises equipment is attached to the *network termination (NT)* at reference point S, as NT1 and NT2 are assembled into a single unit. Bearer services provide capability for information transfer between the communications functions of the terminals, as well as signaling between those functions and the network up to layer 3. Any end-to-end interactions adhering to user defined procedures are transparent to the service. Bearer services include definitions for circuit based 64 kbit/sec, 384 kbit/sec, 1536 kbit/sec and 1920 kbit/sec at 8kHz and packet mode based services.
- *Teleservices* are based on the functions of bearer services. Furthermore,

they extend these functions by including capabilities at layers 4 to 7. These include the specification of telex, G4 telefax, mixed mode, videotex, video conference and telephony services. Together with bearer services, teleservices are called *basic services*.

- *Supplementary services* modify or supplement basic services. Consequently, it cannot be offered to a customer as a stand alone service. The primary objective in specifying these services was to increase the level of convenience and user friendliness. Supplementary services are made available to customers on a subscription basis in association with a particular telecommunication service by the service provider. They have been divided in seven subgroups:
 - *Number identification supplementary services* - Direct dialing in, multiple subscriber number, calling line identification presentation, calling line identification restriction, connected line identification presentation, connected line identification restriction, malicious call identification, sub addressing
 - *Call offering supplementary services* - Call transfer, call forwarding busy, call forwarding no reply, call forwarding unconditional, call deflection, line hunting
 - *Call completion supplementary services* - Call waiting, call hold, completion of calls to busy subscribers, completion of calls on no reply
 - *Multiparty supplementary services* - Conference calling, three-party service
 - *Community of interest supplementary services* - Closed user group, private numbering plan, multi-level precedence and preemption services, priority service, outgoing call barring
 - *Charging supplementary services* - Credit card calling, advice of charge, reverse charging
 - *Additional information transfer supplementary service* - User-to-user signaling

From a practical standpoint, the ISDN network is mainly used for telephony applications. This is also reflected in the standards for basic services as

well as supplementary services. Only few of the latter are applicable to data equipment such as ISDN modems and routers. Of high importance are the subgroup of number identification services, especially the *multiple subscriber number (MSN)* service. It allows data equipment to be assigned a unique identification when attached together with other units such as telephones to the S0 bus (related to reference point S in figure 5.1). Such configuration is called *point-to-multipoint* in opposite to *point-to-point* configurations used to attach private branch exchanges (PBX) to the network. As only up to ten units may be assigned addresses, MSN is no longer feasible for that application. As an answer, *direct dialing in (DID)* allows forwarding of additional address information to the PBX. It enables the customer directly to call a specific subscriber attached to a PBX without intervention of an attendant. Clearly, the network does not perform any checking of subscriber extensions, this is entirely left to the PBX. As far as large PBXs are concerned, only less additional supplementary services are used. Being designed for analog networks as well as ISDNs, they resemble a service logic with similar functionality to ISDN supplementary services. In considering ISDN features for operations off net, severe resource savings may be introduced. When an incoming call is transferred to a mobile using internal resources, two channels are blocked on the trunk side. One channel occupied by the incoming call is joined with an outgoing channel connected to the mobile within the PBX by its service logic. If the call transfer request is served by the network, the incoming channel is released after the call has been connected to the mobile. Using the latter approach, less resources are required in the PBX. When considering distributed call centres based on a network of interconnected switches, call routing without route path optimization as in the scenario drawn above might render the call centre service unusable due to blocking. The easiest solution is to offload these operations to the public network by using ISDN call offering supplementary services to issue call transfer requests. Alternatively, a private network approach as described in section 5.5 may be used.

Protocol Overview

Due to the support of outband signaling and the ability to initiate services during a call, signaling information has to be separated from user information. This is reflected by the introduction of *control plane (C-plane)* and *user plane (U-plane)* to the ISDN framework. The U-plane is aimed to trans-

mit user information through ISDN according to the bearer service selected. This information may be transmitted transparently or may be manipulated, e.g. conversion between A-law and μ -law for voice transmissions. The C-plane includes all entities and protocols concerning any connection control information of the user plane. Consequently, specific channels have been allocated to these planes. User data are transmitted over *bearer channels* (*b-channels*), whereas signaling information is exchanged over the *signaling channel* (*d-channel*) using a three layer protocol stack called *digital subscriber signaling system no. 1 (DSS1)*. In Europe B-channels have been allocated a fixed bandwidth of 64 kbit/sec, whereas the d-channel allows for a fixed bandwidth of 16 kbit/sec (for basic rate access) or 64 kbit/sec (for primary rate access). A combination of two b-channels and one 16 kbit/sec signaling channel (2B+D) allows for basic rate access. Primary access allows for simultaneous reservation of up to 30 user channels using a single 64 kbit/sec signaling channel (30B+D). In the United States, a basic rate access variant using 56 kbit/sec bearer channels also exists. Primary rate access is defined as a combination of 23 b-channels and one signaling channel (23B+D) both having a fixed bandwidth of 64 kbit/sec. In addition to control and user plane, the *management plane* (*M-plane*) has been introduced. The M-plane includes layer management and system management entities. As can be seen from figure 5.2, user information is transmitted transparently over the public network. Signaling information is converted at the local exchange (LE) to *ISDN user part (ISUP)* messages, which are then transmitted using *common channel signaling system no. 7 (SS7)* to the terminating local exchange. Converted back to ISDN messages, signaling and user data are delivered to the destination terminal. Please note, that signaling information must not be transmitted over the same path as user data. Moreover, signaling information is carried over an own signaling network. For more information refer to chapter 7.2.

By using a virtual magnifying glass to give a closer look at the user-network interface shown in figure 5.2, components shown in figure 5.1 may be obtained. ISDN capable *terminal equipment* (TE1) is connected directly to the *S-bus* (called after reference point S). Any other equipment (TE2) has to be attached to a converter called *terminal adapter (TA)*, before being connected to the S-bus. The driving idea behind the introduction of the terminal adapter was to continue the use of analog telephones in digital networks. Network termination 1 (NT1) provides proper physical and electrical termination of the network. In addition it performs layer 1 multiplexing

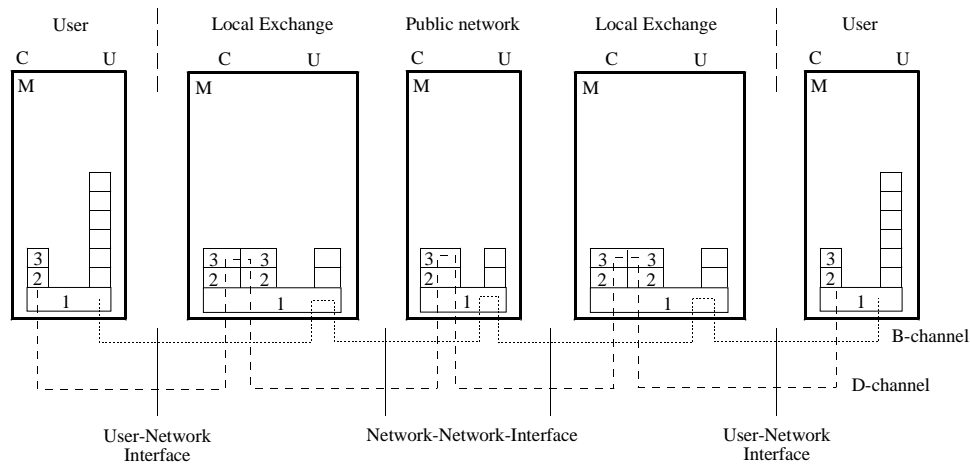


Figure 5.2: subscriber access reference model

and maintenance functions. Network termination 2 (NT2) is responsible for layer 2 and 3 protocol handling, multiplexing, switching and maintenance. In Europe, both devices are usually combined in a single unit. Sometimes the terminal adapter (TA) is added as well. In the United States, customer premises equipment such as PBXs may already contain NT1 and NT2 components. At the network provider the access line is terminated by a *line termination (LT)* interconnected with the *local exchange (LE)* [149]. In order to provide a clear positioning of each ISDN standard, they have been related to the reference points shown in figure 5.1:

- The *R reference point* describes the functional interface between a non-ISDN terminal and the terminal adapter. As already mentioned, the terminal adapter was thought to be an interim solution for migrating to ISDN.
- The *T reference point* defines the functional interface for customer premises equipment, when connected to network termination 2 (NT2).
- The *S reference point* is the function interface for customer premises equipment, when connected to network termination 1 (NT1). From a practical standpoint, reference point S and reference point T may be thought of as being identical, as a lot of ISDN standards concentrate on activities related to S/T reference points.

- The *U reference point* describes the functional interface of the access line with the network termination. These specifications are mainly required by manufacturers of network terminations and central office concentration modules.
- The *V reference point* becomes important, when *central office concentration modules* are used, as these modules are connected to the local exchange using special access protocols such as V5.1 and V5.2.

Physical Layer

At the physical layer most functions of the control and the user plane are merged into a single common entity. Two cooperating layer 1 entities interact on both sides of an S/T reference point to carry out certain tasks including [8]

- activation and deactivation of physical customer and network equipment
- access control for the signaling channel, if more than one terminal is connected
- performance of maintenance functions and equipment status indication for S/T reference points
- coding, multiplexing and synchronization of user and signaling information
- transmission and reception of user and signaling information over the physical medium between terminal and network termination

Basic connection and activation services are provided to the upper layer by the use of service primitives. The datalink layer requests data transmission on a peer-to-peer basis. With respect to the physical media type, the frame structure used in layer 1 varies with the access type:

- *Basic rate access* - User and signaling data are organized into recurring frames of 48 bits, each transmitted synchronously in 250 μ sec for an effective bandwidth of 192 kbit/sec. In total, 144 kbit/sec (2B+D) of bandwidth are received from the higher layers, the rest is used for

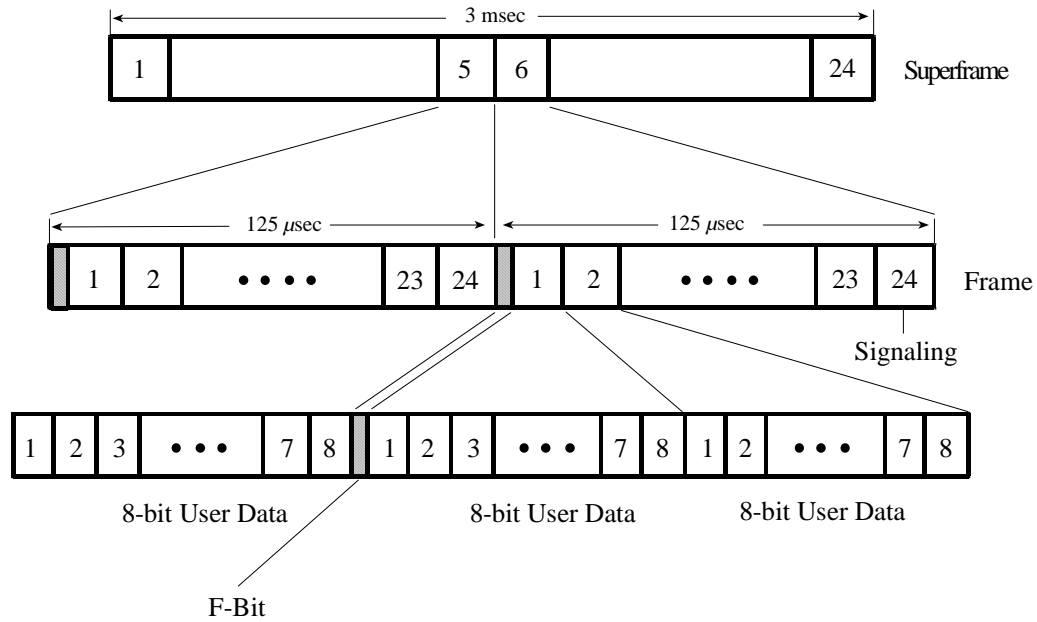


Figure 5.3: Layer 1 frame format - United States - 1544 kbit/sec

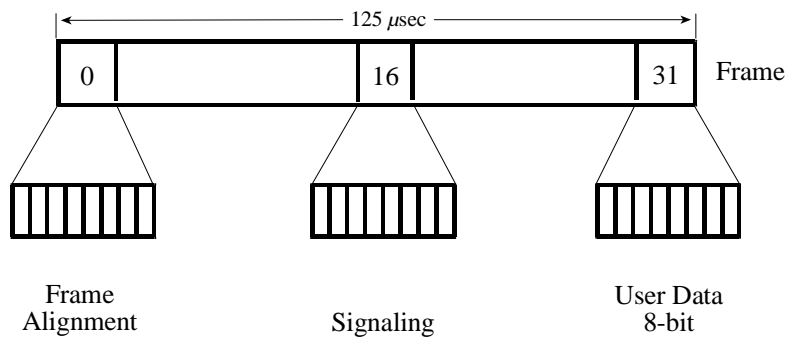


Figure 5.4: Layer 1 frame format - Europe - 2048 kbit/sec

frame alignment and access control. Within the frame structure every bit position is reserved for a certain usage, e.g. bits for bearer and signaling channel. Access control is performed by the use of echo bits within the frame structure. Before a station attempts to send data over the signaling channel, it transmits a nondisruptive sequence of bits. As every bit related to the d-channel is mirrored in a corresponding e-bit (forming an *e-channel*), the station is able to detect other stations communication over the signaling channel. The problem of contention does not appear for b-channels, as they are assigned to a certain station during a connection. In order to identify the interface at reference point S for basic rate access, it is often called *S0 interface*.

- *Primary rate access* - User and signaling data are transmitted using two incompatible schemes. The european version is based on PCM standards developed by the *European Conference of Posts and Telecommunications Administrations (CEPT)* [4]. As shown in figure 5.4, each frame consists of 32 groups of 8 bit blocks, numbered from 0 to 31. Frames are transmitted in 125 μ sec resulting in a bandwidth of 2048 kbit/sec. Each 8-bit block is transmitted in a time slot of 3.91 μ sec giving a rate of 64 kbit/sec per time slot. Signaling information is placed in time slot 16 of each frame, whereas the other time slots are allocated to bearer channels. Time slot 0 is used for frame alignment purposes. For more information on frame alignment refer to [8]. To identify the interface at reference point S for primary rate access, it has been given the name *S2m interface*. In the United States the frame structure is based on AT&T DS1 channel arrangement, which has been standardized by the CCITT in recommendation G.704. Each frame is composed of 24 groups of 8 bit blocks and the reserved *f-bit*. As in the european version, frames are transmitted in 125 μ sec. The function of the f-bit is determined by the position of each frame in the *superframe*. It is used for maintenance, frame alignment and error detection [8]. The signaling channel is usually located in time slot 24, as shown in figure 5.3. For more information, please refer to [17].
- *Wideband access* - When primary access is used, it is also possible to group bearer channels together to support greater transmission rates for applications such as video communications. These *wideband channels (h-channels)* are given as H0 for speeds of 384 kbit/sec, H11 for

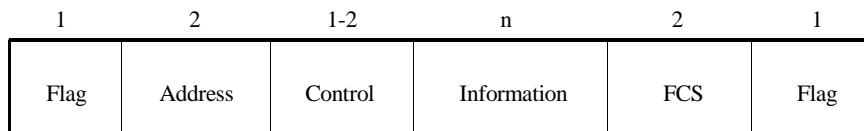


Figure 5.5: LAPD frame format

speeds of 1536 kbit/sec and H12 for speeds of 1920 kbit/sec. In order to switch these channels, a digital exchange must have the ability to establish connections using 6, 24 or 30 time slots. Please note, that H11 requires more than one 1544 kbit/sec link to support the signaling channel. However, h-channels and b-channels may be carried over the same physical links sharing the same d-channel.

Datalink Layer

The *link access protocol for the d-channel (LAPD)* is a layer 2 protocol defined in ITU standards Q.920/Q.921. It provides an efficient and reliable information transfer service to the network layer. Information is carried over one or more logical channels between peers such as terminal (TE), network termination (NT2) and exchange termination (ET). Please note, that line termination (LT) and NT1 are not included in the list, as these entities perform functions only extended to the physical layer. LAPD is based on the *asynchronous balanced mode (ABM)* of the *high level data link control (HDLC)* protocol. In ABM, no master/slave relationship exists between peers, as access control is performed by collision detection [193]. Each station may initialize, supervise, recover from errors and transmit frames at any time. *Data terminal equipment (DTE)* and *data communications equipment (DCE)* are treated as being equal. LAPD is designed for point-to-point operation as well as point-to-multipoint and broadcast communication between terminals and network termination.

The LAPD frame format is shown in figure 5.5. Each frame contains at least the flags, address, control and *frame check sequence (FCS)*. The length of each field is given in octets, which refer to groups of 8 bits each. A frame is considered invalid, if

- it does not contain a proper layer 2 address

- it is not bounded by two flags
- it does not contain a proper number of octets
- if bits have been altered during data transmission

Invalid frames are discarded at the receiving side without notification. In the following text, a closer look will be given to the field formats used in every frame.

- The *opening and closing flags* always contain a binary value of 01111110 (0x7E). To ensure, that such a bit pattern does not occur in the other fields within the frame, a technique known as *bit stuffing* is employed. With bit stuffing, consecutive 1's are avoided by simply inserting a logical 0 after each series of 5 consecutive logical 1's.
- The first two octets after the opening flag are known as address field, which is in turn divided into five subfields. The first field transmitted is called *extension address bit 0 (EA0)* and indicates that the next octet also contains addressing information. It is followed by the *command/response bit (C/R)*, which identifies the entire frame as command frame or response frame. Please note, that the bit value depends on the direction of flow. Frames sent by the user with C/R bit set to 0 are command frames. The same applies to frames sent by the network to the user with C/R bit set to 1. All other combinations identify response frames. The next field called *service access point identifier (SAPI)* has a length of 6 bits. It allows for specification of 64 distinct service access points.. Currently value 0 is used for circuit switched call control procedures, a value of 1 relates to packet mode operation based on ITU recommendation I.451. A value of 16 identifies packet mode communication based on X.25 level 3, whereas 63 is used for layer 2 management purposes. The SAPI is followed by *extension address bit 1 (EA1)*, which indicates the end of the address field. The last subfield transmitted called *terminal endpoint identifier (TEI)* is associated with the user side of the user-to-network interface. With TEI set to 127, a broadcast connection may be set up meant for all terminal endpoints.
- The *control field* has a maximum length of 2 octets depending on one of three possible frametypes. So called *information frames* carry messages between peers. *Supervisory and unnumbered frames* convey layer

2 control and management information. Certain unnumbered frames (*unnumbered information frames*) may also contain layer 3 information. Information frames also include relative *send and receive sequence numbers* allowing for window flow control. Although the length of seven bits supports windows sizes up to 128 frames, the windows size w is fixed at $w = 1$ for basic rate access and $w = 7$ for primary rate access [17]. Additional flow control is available through the use of the supervisory frames *receive ready (RR)*, *receive not ready (RNR)* and *reject (REJ)*. Each message also carries the relative receive sequence number to identify the last successful transmitted frame. Unnumbered frames are responsible for connection establishment, maintenance and release. A data link connection is requested by the *set asynchronous balanced mode extended (SABME)* message. In case of a successful attempt, the receiving station replies with an *unnumbered acknowledge (UA)*. Otherwise a *disconnect mode (DM)* response frame is returned to the sender. The *frame reject (FRMR)* response frame is used to report an error condition, which is not recovered by simply retransmitting an identical frame, e.g. when having received a frame with invalid control field. *Exchange identification (XID)* frames allow for transfer layer 2 management information such as windows size and timing thresholds.

- The *information field* carries user information, which indeed consists of embedded layer 3 messages. A maximum of 260 octets is allowed. The presence of the information field depends on the type of frame used. It is only available for information frames.
- The *frame check sequence (FCS)* allows for detection of transmission errors at the receiver. Prior to bit stuffing, a parity checking sequence with length of two octets is computed from the other fields within the frame. The receiver performs the same calculation for the received frame and then compares its result to the value contained in the FCS field. The calculations carried out are based on the use of *cyclic redundancy codes (CRC)*.

For sake of readability, the description of protocol flows and associated procedures were omitted. For further information please refer to [17] and [8]. The latter contains a detailed description of TEI assignment, checking, removal and information transfer procedures, whereas [17] depicts the most important protocol flows.

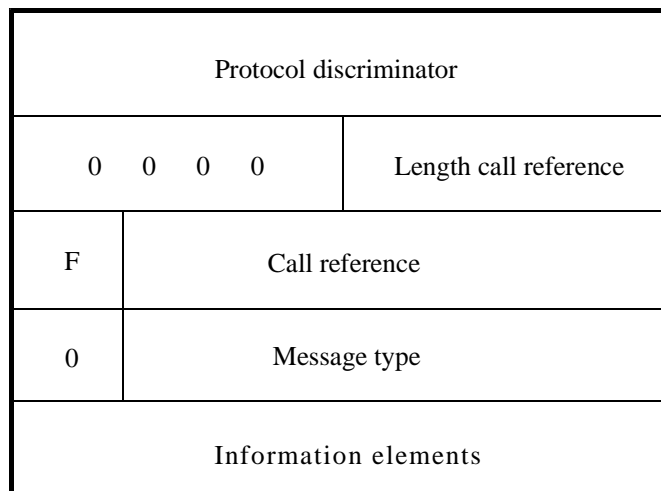


Figure 5.6: ISDN message format

Network Layer

The network layer is responsible for basic call control and control of supplementary services. *Basic call control* defined in ITU-T recommendations Q.930/I.450 and Q.931/I.451 comprises functions to establish, maintain and release circuit switched and packet switched connections. For all functions, operation is based on the services provided by layer 1 and layer 2.

All information - whether it relates to connection control, supplementary services control, user-to-user signaling or user information transfer - is conveyed across the user-network interface by means of *protocol messages*. These messages are supplied as complete messages for transfer in layer 2 information fields. As predetermined through LAPD specifications, the length of each message in octets may vary but is not allowed to exceed 260. As shown in figure 5.6, each message contains the prescribed protocol discriminator, a call reference value, the message type and a number of information elements.

The *protocol discriminator* identifies the layer 3 protocol used, e.g. pattern 00001000 is used to identify messages for user-network call control according to ITU-T recommendation Q.931. Other values include identifiers for user-user information elements, the X.25 protocol or national applications. The *call reference value* identifies a particular layer 3 signaling connection to which the message pertains. This relationship only applies to the relevant

layer 2 connection, it has no end-to-end significance. The call reference value is assigned to a call for its entire lifetime, it can only be reassigned to another call, if the call has been released before. Using such a unique identifier for each call, simultaneous signaling activities may be carried out over the same user-network interface. If multiple terminals are attached to a single line, one user may invoke services while the other is engaged in a conversation. In order to invoke services not associated to an existing bearer connection, e.g. call forward activation in offline state, a dummy call reference is used, i.e. no value is assigned to the call reference, the length indication is 0. The *global call reference* consists of two octets, the length indication set to 1 and the value octet to 0. Messages containing this value affect all layer 3 signaling connections associated with the given logical link layer connection. One example is the invocation of an equipment reset after transmission errors, because all connections are affected. A flag bit (shown as F in figure 5.6) identifies the origin of a call reference. The originating side sets the flag bit to binary 0. In call related messages from the remote end, the flag bit is always inverted to binary 1. The next part of an ISDN message is constituted by the *message type* field. The set of messages specified by the message type is divided into four subgroups for call establishment, call clearance, messages during the active call phases and miscellaneous messages. The codes for the individual message type octets are listed in table 5.2

The last part of each message consists of additional information elements assigned to the message type. Information elements contain the required information to setup a call or control a service. Each layer 3 message may carry one or more information elements or none at all. Whether they consist of one or more octets of data, information elements are called *single octet* or *multiple octet* information elements. The latter have been enhanced by a length indication to allow for variable length. Single octet information elements exist in two types:

- Type 1 consists of an information element identifier (bits 5 to 7) and a contents (bit 1 to 4) with various parameters. These parameters may, for example, indicate a code set changeover (SHIFT) or the congestion level.
- Type 2 consists of only an information element identifier. This corresponds to parameters like SENDING COMPLETE or MORE DATA. These information elements may be identified by having bits 5 to 7 set to 010.

message type octet coding								message name
8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	message type defined in subsequent octet
0	0	0	-	-	-	-	-	call establishment messages
			0	0	0	0	1	ALERTING
			0	0	0	1	0	CALL PROCEEDING
			0	0	0	1	1	PROGRESS
			0	0	1	0	1	SETUP
			0	0	1	1	1	CONNECT
			0	1	1	0	1	SETUP ACKNOWLEDGE
			0	1	1	1	1	CONNECT ACKNOWLEDGE
0	0	1	-	-	-	-	-	messages during active call phases
			0	0	0	0	0	USER INFORMATION
			0	0	0	0	1	SUSPEND REJECT
			0	0	0	1	0	RESUME REJECT
			0	0	1	0	0	HOLD
			0	0	1	0	1	SUSPEND
			0	0	1	1	0	RESUME
			0	1	0	0	0	HOLD ACKNOWLEDGE
			0	1	1	0	1	SUSPEND ACKNOWLEDGE
			0	1	1	1	0	RESUME ACKNOWLEDGE
			1	0	0	0	0	HOLD REJECT
			1	0	0	0	1	RETRIEVE
			1	0	0	1	1	RETRIEVE ACKNOWLEDGE
			1	0	1	1	1	RETRIEVE REJECT
0	1	0	-	-	-	-	-	call clearance messages
			0	0	1	0	1	DISCONNECT
			0	0	1	1	0	RESTART
			0	1	1	0	1	RELEASE
			0	1	1	1	0	RESTART ACKNOWLEDGE
			1	1	0	1	0	RELEASE COMPLETE
0	1	1	-	-	-	-	-	miscellaneous messages
			0	0	0	0	0	SEGMENT
			0	0	0	1	0	FACILITY
			0	0	1	0	0	REGISTER
			0	1	1	1	0	NOTIFY
			1	0	1	0	1	STATUS ENQUIRY
			1	1	0	0	1	CONGESTION CONTROL
			1	1	0	1	1	INFORMATION
			1	1	1	0	1	STATUS

Table 5.2: Codes for message types

Further examples of information elements include the call state, calling party number, calling party subaddress and bearer capability. The way how information elements are interpreted, depends on the *codeset*. The codeset may be changed by using the single octet information element SHIFT. As the last three bits of the SHIFT information element specify the codeset to be selected resulting in a total of eight codesets available. The codeset may be fixed, until changed by another SHIFT information element or reserved for the next information element only. Codeset 0 defined by the ITU-T is the default for every new connection. Codeset 5 defines information elements for Europe according to ETSI specifications. Codeset 6 and 7 are reserved for national use and for definition of application specific information elements. For a detailed list of information elements based on ITU-T specifications refer to [8] or [17]. The latter also states, which information elements are not used in Europe.

A typical call setup showing the message types and some important information elements is depicted in figure 5.7. Please note, that digits are not transmitted *en-bloc*, moreover after being prompted by an appropriate dial tone, they are sent to the exchange using one or more INFORMATION messages. This type of signaling is referred to as *overlap sending*. Due to space limitations, obvious information elements such as calling party number and called party number have been omitted. The BEARER CAPABILITY information element indicates the major service characteristics (discussed above in section 5.2.2) to the network. The SIGNAL information element is used to initiate the generation of certain tones and alerting signals related to an incoming call or the status of a call. The DISPLAY information element allows the network to convey information to be displayed by the end-user terminal. Information entered using the keypad of an end-user terminal may be transferred across the user-network interface embedded in the KEYPAD FACILITY information element. Both types of information, display information and keypad input are coded in *international alphabet no. 5 (IA5)*. The CHANNEL IDENTIFICATION information element relates to the type of interface between user and network and the particular b-, d-, or h-channel to be controlled by the signaling protocol.

In order to invoke supplementary services using layer 3 of the ISDN protocol, three protocol procedures have been defined.

- The *keypad protocol* uses information elements DISPLAY, SIGNAL and KEYPAD FACILITY to convey information across the user-network in-

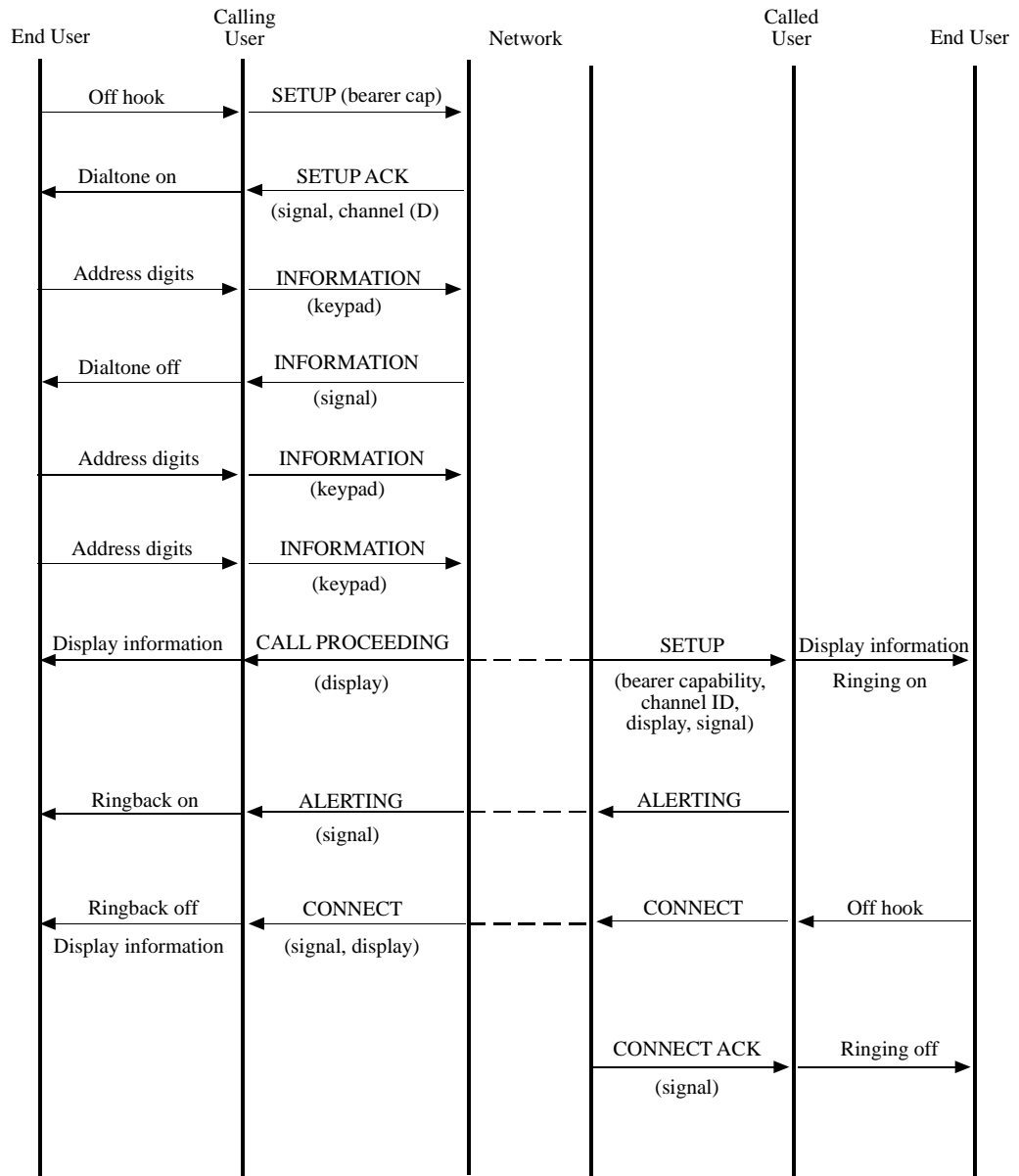


Figure 5.7: Call establishment in ISDN

terface. A service may be invoked by the local exchange at any time during establishment, operation and clearing phases of a call. As already mentioned above, keypad and display information are encoded in IA5. Services invoked have to be associated with a call, i.e. at least a call reference has to be established.

- The *feature key protocol* uses information elements FEATURE INDICATION and FEATURE ACTIVATION. Again, these information elements are carried in layer 3 messages such as SETUP, INFORMATION, ALERTING, RELEASE. In opposite to the keypad protocol, services may be invoked in association with a call or independently. Features invoked by the FEATURE ACTIVATION information element depend on the network provider. Usually user profiles are defined for each user allocating particular services to the *feature identification number* transmitted with the FEATURE ACTIVATION information element.
- The *functional protocol* is based on the usage of the FACILITY information element as well as messages such as FACILITY, HOLD and RETRIEVE. Services may be invoked by the use of explicit messages exclusively identified with a particular service (*separate message procedure*) or supplementary service information conveyed via the FACILITY information element (*common information element procedure*). Only six explicit messages have been defined up to now - HOLD, HOLD ACKNOWLEDGE, HOLD REJECT, RETRIEVE, RETRIEVE ACKNOWLEDGE, RETRIEVE REJECT - these are used to put an existing call to hold state and retrieve it at a later time. More services have been defined for usage with the FACILITY information element. For call independent services, this information element is carried by a FACILITY or REGISTER message, whereas services associated to calls may be transmitted in any of the basic call control messages. A complete list of supplementary services is given above in section 5.2.2.

Broadband ISDN

Based on the fact, that bandwidth demand raises with time, the need for services employing greater bit rates based on a new transport technology was already clear, when the I series recommendations were approved. Applications for broadband networks include high-speed file transfer, high resolution facsimile, TV distribution, broadband video telephony and videoconferencing.

Broadband ISDN (B-ISDN) relies on the *asynchronous transfer mode (ATM)* as transport technology. ATM uses very small packets of fixed length called *cells* to convey user and signaling information across the network. The term *asynchronous* refers to the fact, that cells allocated to the same connection may exhibit an irregular recurrence pattern in the context of a multiplexed transmission. In opposite to most TDM technologies such as the *synchronous transfer mode*, cells are filled according to the actual demand. ATM allows for

- transmission of multiple traffic types
- multiple classes of quality of service (QoS)
- dynamic QoS handling
- improved transmission utilization by use of statistical multiplexing
- guaranteed resource allocation including bandwidth

As ATM belongs to the group of connection oriented protocols, cells are associated to a *virtual channel* by a common identifier called *virtual channel identification (VCI)*. Similar to the concept of trunking in classic telephony, virtual channels are always grouped into *virtual paths* to be transmitted between logical ATM nodes. Each cell of a virtual channel is associated to a virtual path by the *virtual path identifier (VPI)*. Nodes devoted to switching of virtual paths are called *ATM crossconnects*, whereas *ATM switches* perform switching of virtual channels.

The protocol stack of B-ISDN based on ATM is shown in figure 5.8. Similar to narrowband ISDN, the logical model is composed of user plane, control plane and management plane. The *management plane* is further divided into two subplanes. Management functions relating to the system as a whole, belong to *plane management*. The *layer management* performs functions relating to resources and parameters of its protocol entities. This also includes the handling of layer specific *operations, administration and maintenance (OAM)* information flows. At the physical and the ATM layer, functions are the same for signaling and user information, i.e. there is not distinction between user plane and control plane.

The *physical layer* performs low level functions such as bit alignment, line coding and high level functions such as cell delinieation, cell rate decoupling,

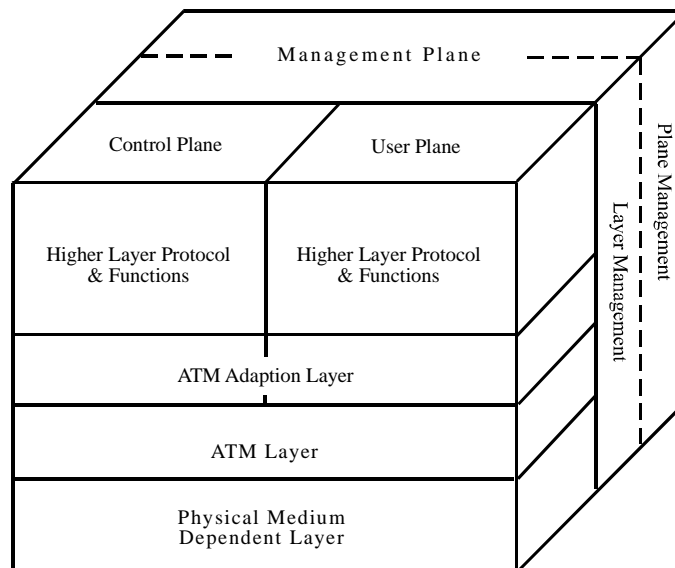


Figure 5.8: B-ISDN protocol stack

HEC sequence generation and transmission frame adaption. A *header error correction (HEC)* sequence is generated at the sender for each cell (refer to field HEC in figure 5.9) to be recalculated and compared at the receiving side. Please note, that error correction is restricted to the header only. *Transmission frame adaption* occurs, when cell flow is adapted to the payload structure of a transmission system such as *synchronous digital hierarchy (SDH)*.

The *ATM layer* performs medium independent functions such as VPI/VCI translation, call header extraction and generation applied to termination end-points, call multiplexing and demultiplexing and generic flow control. *VPI/VCI extraction and generation* occurs during translation of a VPI or a VCI at ATM switching or ATM crossconnect nodes. *Generic flow control (GFC)* supports the control of ATM traffic flow across the B-ISDN user-network interface. The *cell* is the basic element of the ATM layer. Each cell consists of a 5 octet header and a 48 octet user information field. The structure used for the B-ISDN user-network interface is shown in figure 5.9. In addition to the fields already discussed - VCI, VPI and HEC - each cell includes fields for *generic flow control (GFC)*, *payload type (PT)* and *cell loss priority (CLP)*. The GFC field is currently not supported, therefore its value is set to 0000.

Flow Control		VPI	
VPI		VPI / VCI	
VCI			
VCI	PT	CLP	
Header Error Control			
User Information			

Figure 5.9: ATM cell format

It was defined to support more broadband terminals connected to a single broadband network termination (B-NT2). The B-ISDN reference configuration is shown in figure 5.10. With CLP set to binary 1, messages may be discarded in network overload situations. The user may use the CLP field to identify low priority cells during peak times, whereas the network changes the value of the CLP bit only, if the traffic contract is violated. The payload type (PT) allows cells to be related to information streams for user information transfer, OA&M purposes and resource management.

In addition to low level functions such as *segmentation and reassembly*, the *ATM adaption layer (AAL)* adapts services provided by the ATM layer to those required by higher layers. In order to minimize the number of AAL protocols, a service classification has been defined based on timing relations, bit rates and connection modes [7]:

- *Class A* - Timing required, bit rate constant, connection oriented - Applications include circuit emulation, constant bit rate audio and video
- *Class B* - Timing required, bit rate variable, connection oriented - Applications include variable bit rate audio and video

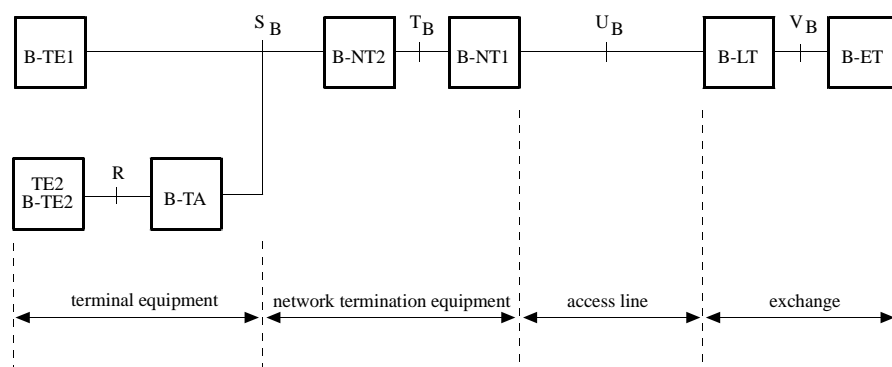


Figure 5.10: B-ISDN reference configuration

- *Class C* - Timing not required, bit rate variable, connection oriented - Applications include connection oriented data transfer
- *Class D* - Timing not required, bit rate variable, connectionless - Applications include connectionless data transfer

To support these traffic types and related applications, five AAL protocols have been defined:

- *AAL type 0* does not deliver any additional services to those already provided by the ATM layer. Its main application is the transport of *metasignaling* messages. Metasignaling is used at the user-network interface to establish, monitor and release *signaling virtual channels* [46]. Cells of the metasignaling channel are identified by having set the VCI to 0 with $VPI \neq 0$.
- *AAL type 1* provides connection oriented services with constant bit rate having specific delay requirements for class A applications such as circuit emulation including DS0 like links. AAL type 1 includes functions for segmentation, reassembly and indication of lost or errored information.
- *AAL type 2* provides connection oriented services with variable bit rate for class B and class C applications such as compressed speech data. No functions for segmentation and reassembly are included in AAL2, user information is transmitted in mini-cells.

- *AAL type 3/4* provides services for class C and class D traffic such as bursty data services. The functions of AAL 3/4 include segmentation, reassembly, error control and flow control. Reliable data transfer is supported as well as transfer of data on a best effort basis.
- *AAL type 5* provides similar services as AAL type 3/4. Error correction has been omitted, lost and corrupted cells are not retransmitted. Its strength lies in its simplicity, as overhead and processing requirements have been significantly reduced.
- The *signaling adaptation layer (SAAL)* is based on AAL type 5. It provides error correction and flow control for signaling channels.

Although layer 1 and layer 2 widely differ from those used in narrowband ISDN, the B-ISDN reference configuration as shown in figure 5.10 follows the same structure. For every reference point except reference point R, a broadband counterpart has been defined. In order to connect narrowband terminal (TE1) equipment, two solutions exist. Either terminals are connected to a broadband terminal adapter (B-TA) or the broadband network termination (B-NT2) provides an interface for the S reference point [35]. The protocol stack used by broadband terminal equipment to connect to the broadband ISDN at reference point S_B is called *digital subscriber signaling system no. 2 (DSS2)*.

The network layer of DSS2 has been specified by the ITU-T in recommendation Q.2931. Although it employs a similar structure as the network layer of DSS1, several modifications have been applied. With reference to figure 5.6, the *protocol discriminator* for Q.2931 is 00001001, the *call reference* has been extended in length and the *dummy reference* employs a different bit pattern. The message encoding as described in Q.2931 for *capability set 1 (CS1)* are shown in table 5.3. Please note, that messages PROGRESS, SETUP, ACKNOWLEDGE and INFORMATION have been defined to support interworking with narrowband ISDN. Peak cell rate modification messages as described in ITU recommendation Q.2963 are used to renegotiate an existing traffic contract of an active call. Information elements used basically follow recommendation Q.931 for narrowband ISDN, although new information elements have been included to capture the advantages provided by ATM. It has become possible to handle AAL service class, quality of service parameters and several broadband characteristics. For more details, please refer to [17].

message type octet coding								message name
8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	message type defined in subsequent octet
0	0	0	-	-	-	-	-	call establishment messages
			0	0	0	0	1	ALERTING
			0	0	0	1	0	CALL PROCEEDING
			0	0	0	1	1	PROGRESS
			0	0	1	0	1	SETUP
			0	0	1	1	1	CONNECT
			0	1	1	0	1	SETUP ACKNOWLEDGE
			0	1	1	1	1	CONNECT ACKNOWLEDGE
0	1	0	-	-	-	-	-	call clearance messages
			0	0	1	1	0	RESTART
			0	1	1	0	1	RELEASE
			0	1	1	1	0	RESTART ACKNOWLEDGE
			1	1	0	1	0	RELEASE COMPLETE
0	1	1	-	-	-	-	-	miscellaneous messages
			0	1	1	1	0	NOTIFY
			1	0	1	0	1	STATUS ENQUIRY
			1	1	0	1	1	INFORMATION
			1	1	1	0	1	STATUS
1	0	0	-	-	-	-	-	peak cell rate modification messages
			0	1	0	0	0	MODIFY REQUEST
			0	1	0	0	1	MODIFY ACKNOWLEDGE
			0	1	0	1	0	MODIFY REJECT
			0	1	0	1	1	CONNECTION AVAILABLE
0	1	1	-	-	-	-	-	messages for semipermanent connections
			1	0	1	1	0	SPC STATUS ENQUIRY
			1	0	1	1	1	SPC UPDATE STATUS
			1	1	1	1	0	SPC STATUS REPORT
			1	1	1	1	1	SPC UPDATE STATUS ACKNOWLEDGE

Table 5.3: Codes for message types

In the context of call centres it is worth to mention the *look ahead* procedure as defined in ITU recommendation Q.2964 for capability set 2.1. Classic call centres often perform a look ahead using mechanisms such as busy tone detection, before issuing a blind transfer. Broadband ISDN allows the transmission of LOOK AHEAD information elements embedded in a FACILITY message. Clearly, these information elements only carry the connection state of the desired terminal. Call Centre specific information like agent state still has to be exchanged using user-to-user signaling.

Multimedia options in ISDN and B-ISDN

So far, ISDNs and B-ISDNs in the light of classic telephone services have been explained. The following section attempts to provide an overview, how multimedia streaming applications may be implemented on top of the ISDN/B-ISDN framework. The relevant standard for narrowband ISDN has been specified by the ITU-T in recommendation H.320. As H.320 provides an open platform for multimedia communications, it refers to specific standards for audio, video and data streams. A rough overview according to recommendation H.320 is given in figure 5.11. To avoid incompatible terminals, each terminal must support a minimum set of media streams. For video coding and decoding, the relevant *video codec* is described in ITU recommendation H.261. The default *audio codec* is specified in recommendation G.711. Data capabilities are not mandatory for H.320 terminals. During connection establishment a *capabilities exchange* is performed by the terminals involved. Based on the capability sets provided by all parties, the desired codecs are selected and necessary bearer channels are established. Without additional components, H.320 terminals may communicate point-to-point. If more than two terminals should participate in a conference, a *multipoint control unit (MCU)* has to be added. It provides audio mixing and video switching capabilities as desired by each party. An MCU is not necessarily a separate unit, its functions may be included in H.320 terminals as well.

The *H.261 codec* supports two types of format, the *common intermediate format (CIF)* for a resolution of 352×288 and the *quarter common intermediate format (QCIF)* for a resolution of 176×144 . Video signals are compressed by reducing redundancy, which is inherent to these signals. Each picture is divided into *groups of blocks (GOBs)* and further in *macroblocks* and *blocks*, which allow for motion compensation and reuse from information of the previous picture (*inter coding*). Each block is transformed using a

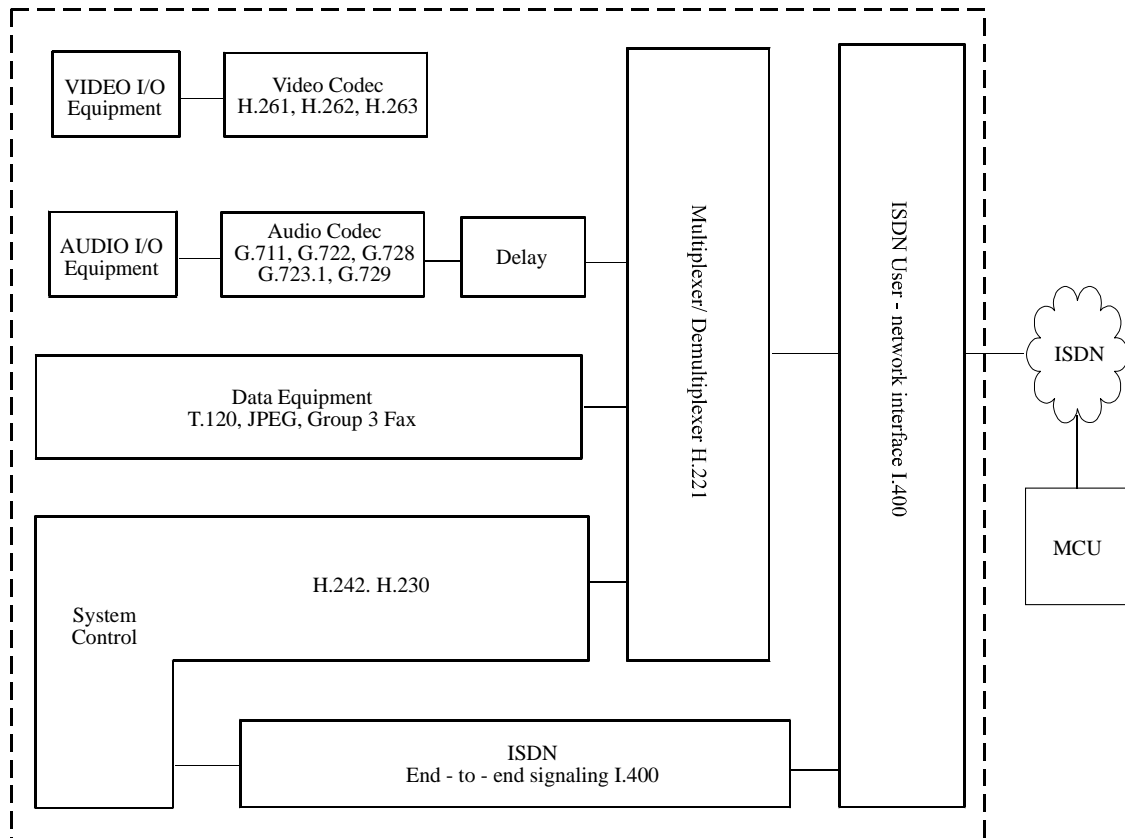


Figure 5.11: Interrelationship of H.320 recommendation series

discrete cosine transformation (DCT) to achieve better compression ratios.

As shown in figure 5.11, several audio codings may be used in H.320 terminals. G.711 refers to pulse code modulated (PCM) 3 kHz speech of 8000 samples per second, encoded to 8 bits per sample giving a transmission rate of 64 kbit/sec. Quality may be improved by using G.722 instead of G.711. Using sub-band adaptive differential PCM encoding 7 kHz audio streams results in a bandwidth demand of 64 kbit/sec. In order to alleviate waste of resources, G.728 has been defined. The transmission rate for 3 kHz audio is reduced to 16 kbit/sec using *codebook excitation linear prediction (CELP)*.

The H.320 standard allows for transmission of data along with audio and video information. The relevant communication protocol stack has been defined in the T.120 recommendation series. It allows for data applications such as still image transmission, multipoint fax and binary file transfer.

To provide a flexible, synchronized and economic solution, ISDN bearer channels are partitioned into subchannels of 8 kbit/sec. The first seven subchannels contain audio, video or data streams according to the media chosen. The eighth subchannel called *service channel* is used for signaling purposes. Please note, that the service channel is part of the bearer channel and does not affect the ISDN signaling channel at all. If multiple bearer and h0 channels are used, *multiframe numbering and alignment* has to be performed. The receiver uses multiframe numbering to synchronize the received signals. The service channel contains two parts

- Bits 1 to 8 are occupied by the *frame alignment signal (FAS)*. In addition to framing and multiframe information, it contains control and alarm information, as well as error check information to control end-to-end error performance and frame alignment validity.
- Bits 9 to 16 are allocated to the *bit-rate allocation signal (BAS)*. It contains commands for the receiver or a description of the capabilities of a terminal to structure the capacity of its channels.

The frame structure just described has been defined in ITU recommendation H.221, whereas details about signaling information for *control and indication* is contained in recommendations H.221 and H.230.

A similar structure has been adopted for multimedia communication using B-ISDN. Recommendation H.321 describes the adaption of *H.320 terminals*

to B-ISDN environments. The main specification is given in recommendation H.310. It is intended to support applications such as conversational services, retrieval services, messaging services, distribution services, video transmission and surveillance. H.310 terminals may be distinguished according to their audio and video capabilities. Unidirectional H.310 terminals are not required to support H.261 and H.263 video codecs, whereas bidirectional terminals have to support H.261 CIF and QCIF encoding to achieve interoperability with H.320 terminals. In relation to audio encoding, only bidirectional H.310 terminals have to stay compatible to H.320 terminals through the use of G.711 codecs. Data capability is not mandatory. Based on these specifications, cost effective solutions for applications such as surveillance and video distribution may be designed for broadband multimedia communication. For more information on videoconferencing and videotelephony for ISDNs and B-ISDNs, refer to [16].

5.2.3 QSig

Traditionally, *private branch exchanges (PBX)* were interconnected using analog tie lines between remote sites. Due to its nature, analog signaling is restricted to basic call control such as call establishment and call clearance. As an alternative, sites can be linked by ISDNs utilizing more sophisticated services. Unfortunately such an arrangement does not possess all the necessary characteristics required for *corporate telecommunication networks (CTN)* [14], such as

- reduced communication cost
- the ability to manage network and services according to business needs
- improved security and confidentiality
- increased number of facilities
- the use of customized features

To overcome these limitations, the QSig protocol suite has been defined to interconnect so called *private integrated services network exchanges (PINX)*. The term PINX covers various types of corporate networking equipment such as PBXs and multiplexers. QSig has been standardized as a part of the ISDN

standardization process by the *European Computer Manufacturer Association (ECMA)*. These standards have also been adopted by ISO/IEC and ETSI. The name QSig stems from the fact, that signaling occurs at the Q reference point. QSig as a inter-PINX signaling system provides [14]

- support of future developments by international standard bodies
- a harmonized, flexible and cost efficient method for interconnecting multi-vendor equipment
- synergy with the public ISDN and business applications developed for the public ISDN
- a platform for introduction of supplementary services within a heterogeneous environment

Services

Basic services, related signaling protocols and generic support for supplementary services have been defined in ECMA-106, ECMA-142, ECMA-143, ECMA-156, ECMA-161 and ECMA-165 based on related ISDN standards. Two groups of enhancements have been defined for QSig. Supplementary services provide benefits to a particular user, whereas *additional network features (ANFs)* improve handling of certain calls or the performance of the network. An example for the latter is the path replacement ANF, which allows for route optimization within corporate networks. Standards for supplementary services and additional network features (ANFs) include

- Calling line identification, connected line identification, calling / connected line identification restriction (ECMA-148, ECMA-157)
- Calling name identification, connected name identification, calling / connected name identification restriction (ECMA-163, ECMA-164)
- Call forward on busy, call forward on no reply, unconditional call forward, call deflection (ECMA-173, ECMA-174)
- Path replacement ANF (ECMA-175, ECMA-176)
- Call transfer (ECMA-177, ECMA-178)

- Call completion to busy, call completion on no reply (ECMA-185, ECMA-186)
- Call offer (ECMA-191, ECMA-192)
- Do not disturb and override (ECMA-193, ECMA-194)
- Call intrusion (ECMA-202, ECMA-203)
- Advice of charge (ECMA-211, ECMA-212)
- Recall (ECMA-213, ECMA-214)
- Call interception (ECMA-220, ECMA-221)
- Transit counter ANF (ECMA-224, ECMA-225)
- Message waiting indication (ECMA-241, ECMA-242)
- Common information ANF (ECMA-250, ECMA-251)
- Call priority interruption and protection (ECMA-263, ECMA-264)
- PUM registration (due date 1999)
- PUM call handling ANF (due date 1999)
- Call distribution ANF (due date 1999)
- CTM incoming call ANF (ECMA-215)
- CTM outgoing call handling ANF (ECMA-233)
- CTM authentication (ECMA-243)
- CTM location registration (ECMA-216)

Please note, that QSig also provides supplementary services and ANFs to integrate with cordless terminal mobility (CTM). Users will be able to move through the QSig network, make and receive calls after having registered their terminals.

Protocol Overview

QSig is based on ITU-T Q.93x recommendations for basic call services and generic functions and the Q.95x recommendations for supplementary services. Therefore compatibility is preserved between the public and private ISDNs. As shown in figure 5.12, the protocol stack is similar to the protocol stack of DSS1. Layers 1 and 2 are equal, whereas the network layers differ. As QSig is based on point-to-point operation between PINXs rather than to provide a user-network interface as defined for the ISDN framework, several parameters for layer two have been restricted:

- The terminal identifier (TEI) is always set to 0 (point-to-point operation)
- Unnumbered information frames are not used. In the ISDN framework these frames are related to point-to-multipoint operation such as TEI-management. As a consequence, layer 3 SETUP messages are embedded in a layer 2 information frame instead of using an unnumbered information frame.
- The command/response bit (C/R bit) is fixed, a distinction between user and network side is not necessary for PINX equipment.

The network layer of QSig is based on ITU-T recommendation Q.931 for DSS1, although some extensions and restrictions have been applied. Almost all messages used during the active phases of a call have been omitted. As can be seen from table 5.2, the majority of these messages is related to activities carried out by the local exchange only. Within a private network, these functions are performed by the PINX itself, therefore omitting the necessity of dedicated signaling messages. As shown in figure 5.12, the network layer is divided in three sublayers. The *QSig basic call sublayer* provides a symmetrical interface designed for point-to-point and transit operation. The *QSig generic functional procedure sublayer* is responsible to provide control of supplementary services and additional network features (ANFs) over corporate networks. Similar to the user-network interface of ISDN, supplementary services are either associated to a call or call independent. Unlike DSS1, the QSig generic functional procedure sublayer supports connection oriented and connectionless transport for call independent supplementary services and ANFs. The *QSig supplementary services sublayer* defines QSig procedures at the Q reference point for individual supplementary services.

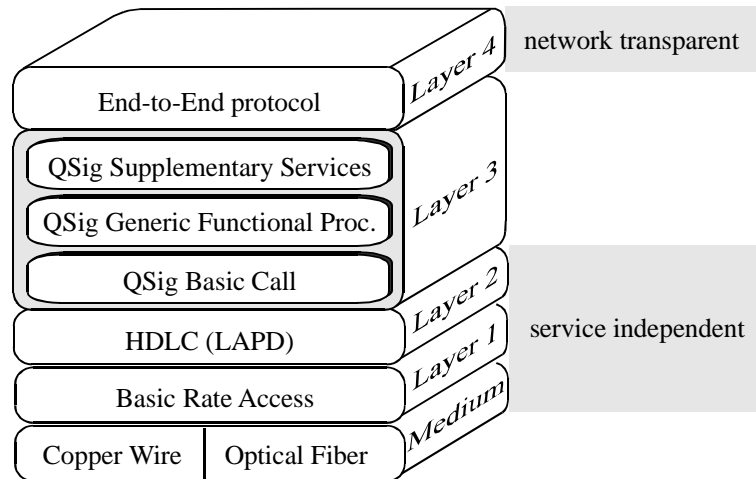


Figure 5.12: QSig protocol stack at the Q reference point

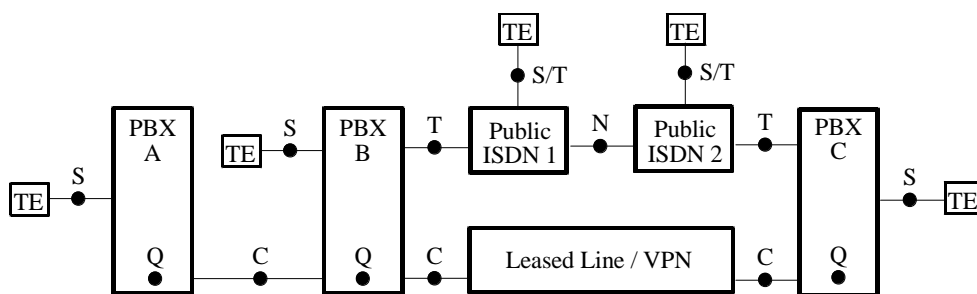


Figure 5.13: QSig reference configuration for corporate networks

Figure 5.13 presents an overview of the reference configuration used in public and private networks according to the ISDN/QSig framework [4]. Please note, that the Q reference point is rather a logical signaling point than a reference point for a physical interface such as the S/T interface for ISDN. The physical connection is made at the C reference point for leased lines or switched connections. When interconnected over the public network, PINX equipment is attached at the S/T reference point. In Europe, S0 and S2m interfaces are used to access the public network [17].

Broadband QSig

Similar to the evolution from narrowband ISDN to broadband ISDN, ECMA specified a framework for operation at the Q reference point between PINXs across a *broadband private integrated services network (B-PSDN)* employing ATM as transport technology. The protocol stack looks similar to the protocol stack for narrowband QSig as shown in 5.12, except that layer 1 and 2 have been replaced by related broadband layers. In order to achieve alignment between B-QSig specifications and the signaling part of the ATM Forum's *private network-network interface (PNNI)*, ECMA has cooperated with the ATM Forum during development of these standards. The basic control protocol (ECMA-266) is based on the signaling section of PNNI version 1.0 with several extensions required to capture some characteristics of QSig. Similar to QSig, B-QSig is designed to employ static hop-by-hop routing, whereas the PNNI specification provides dynamic source rerouting. The basic call control protocol runs on top of the signaling ATM adaption layer defined in ECMA-265. Bearer services are specified in ECMA-261, whereas generic support for supplementary services and additional network features (ANFs) is given in ECMA-254. For more information refer to the related standards, an overview of the protocol stack is given in [14].

5.2.4 IP Networks with Binary Signaling

Especially for IP based networks, a new flood of multimedia protocols is overwhelming the market. Some of them have been developed for the purpose of transporting multimedia streams over these networks. H.323, as recommended by the ITU-T in 1996 provides a foundation for audio, video, and data communications across IP-based networks, including the Internet. By complying to H.323, multimedia products and applications from multiple

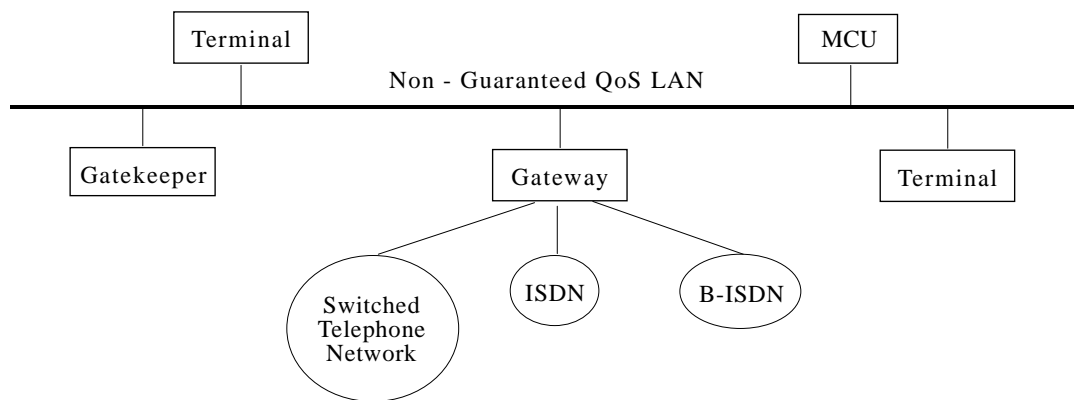


Figure 5.14: Reference configuration according to H.323

vendors can interoperate, allowing users to communicate without concern for compatibility. H.323 is an umbrella recommendation that sets standards for multimedia communications over *local area networks (LANs)* that do not provide a guaranteed quality of service (QoS). Therefore, the H.323 standard suite defines important building blocks for a wide range of collaborative, LAN-based multimedia applications.

As shown in figure 5.14, H.323 defines the following four components for a network-based communications system

- *Terminals* are the client endpoints on the LAN providing real time, two-way communications. The ability for voice communication is mandatory, whereas support for video and data is optional.
- *Gateways* provide an interface to other networks such as ISDNs, B-ISDNs and classic telephone networks. Clearly, terminals belonging to these networks have to support the relevant standards for multimedia communications such as H.320, H.310 and H.324. Each gateway performs translation between transmission formats and communication procedures. If necessary, audio and video streams created by the codecs of each terminal are converted.
- The *multipoint control unit (MCU)* supports conferences between three or more parties. It consists of a mandatory *multipoint controller (MC)*

and one or more optional *multipoint processors (MP)*. The latter mixes, switches, and processes audio, video, and data streams, whereas the former controls call flow and supports capabilities exchange between terminals. Please note, that MCU functionality may be combined with any of the other H.323 components in a single box.

- The *gatekeeper* acts as the central point for all calls within its zone and provides call control services to registered endpoints. The term *H.323 zone* refers to a collection of terminals, gateways and multipoint control units (MCUs) managed by a single gatekeeper. Functions to be performed by the gatekeeper include *admission control*, *address translation* and *bandwidth management*. Address translation is required to map an alias such as name or telephone number to a valid transport address, i.e. an IP address. As an optional feature, each gatekeeper may get involved into routing decisions, e.g. to provide balancing between multiple gateways.

Transport Network Protocol Overview

Although H.323 may also be implemented on top of protocols like IPX/SPX, it is primarily used in conjunction with TCP/IP. This is also reflected by the protocol stack shown in figure 5.15. Layers 1 and 2 have been omitted due to space limitations. For more information, refer to textbooks on Ethernet such as [56].

At the network layer, the *internet protocol (IP)* is responsible to route data across a network. At this layer, the service is connectionless. An endpoint is identified by a unique IP address, which reflects the network topology of the internet or an intranet. If dynamic addressing, e.g. the dynamic host configuration protocol (DHCP), is used, endpoints are identified by a unique IP address for the time being active. Especially these mechanisms impose some problems on the address resolution function of the gatekeeper. IP is a very essential part of the TCP/IP protocol suite, because almost all other protocols are built on top of IP. Exceptions include the *address resolution protocol (ARP)* and the *reverse address resolution protocol (RARP)*. In order to provide end-to-end transmission services to the layers above, the network layer has to perform *address resolution* and routing. The former is achieved by the use of ARP, which maps IP addresses to layer 2 *medium access control (MAC)* addresses. *Routing* refers to the selection of

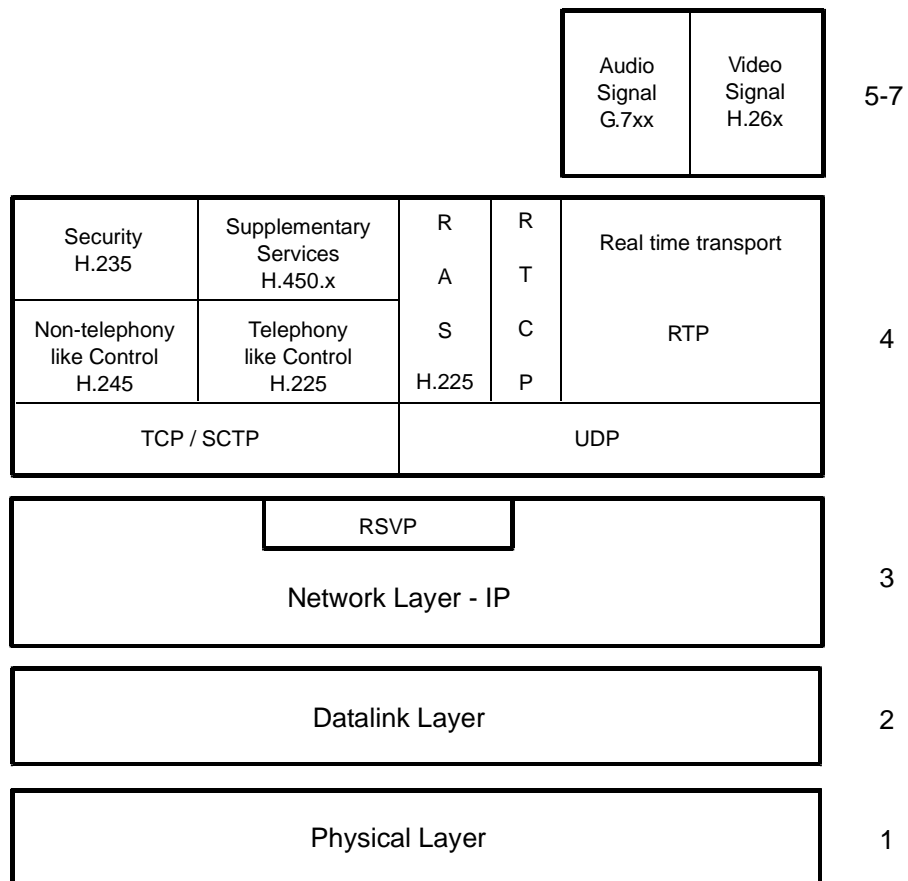


Figure 5.15: Voice over IP protocol stack based on H.323

paths performed by the network. In case of connection oriented transmission, the routing path is determined at the beginning of each conversation. If connectionless transmission also known as datagram transport is used, the routing path may differ for each packet.

As TCP/IP has been defined for pure data transport, it lacks several quality of service mechanisms used in transport networks such as ATM. Due to the nature of audio and video streams, admission control has to be introduced in IP networks. Although several approaches exist, only the *resource reservation protocol (RSVP)* is able to provide quality of service guarantees in an end-to-end manner. After RSVP has been deployed in the entire network, each network flow may reserve a specific amount of resources in the network. Please note, that RSVP does not improve the average performance of the network. The reservation process is initiated sending out a PATH message to the destination. Obviously, this message follows the normal routing path like any other packet. As the PATH message traverses through the network, the routing path is determined and several performance data are collected to be registered within the message. This entire set of data, as received by the destination is called *advertisement specification*. It includes information such as the [22]

- *number of hops* supporting RSVP along the routing path and information about routers, that do not support RSVP. The latter is gathered by RSVP enabled routers being aware of intermediate hops, which do not support RSVP.
- *minimum path bandwidth*, which is set by every router according to its resource status.
- *path latency*, which captures various delays such as processing delay and propagation delay in an additive manner.
- *maximum size* of an IP packet, which is not fragmented along the path.

In addition to these dynamic data, each PATH message also carries the *traffic specification* for the required application. The traffic specification consists of the following parameters [22]

- *peak rate* in bytes per second
- *bucket size* in bytes

- *bucket rate* in bytes per second
- *maximum datagram size* in bytes
- *minimum packet size* in bytes

By evaluating both specifications, the destination compiles a *reservation specification* embedded in a RESV message to be returned to the sender. Please note, that the reservation of resources is initiated by the receiver. As each reservation expires with time, it has to be frequently renewed by repeating the whole process. In an RSVP enabled network, three service classes are supported

- The *guaranteed service* introduces tight bounds on delay and throughput characteristics. If too many connections with guaranteed service are established in a network, the network performance may significantly suffer.
- The *controlled load service* is not as strict as the guaranteed service, as only average performance parameters are bounded. Obviously, the controlled load service has a better effect on the network performance.
- The remaining traffic, which is also found in networks without RSVP support, is called *best effort traffic*. In RSVP enabled networks, especially the service level of best effort traffic degrades if too many guaranteed and controlled load services are deployed.

Please note that RSVP is not included in the H.323 suite of protocols, but practice has shown, that proper use of RSVP performs well in H.323 based IP networks. For more information on RSVP refer to [54] and [22].

The *transmission control protocol (TCP)* as defined in RFC 793 was the first transport layer protocol associated with IP. With focus on transaction based data handling, i.e. connection oriented transport, TCP introduced the first quality of service mechanism to the TCP/IP protocol stack. It was designed to ensure robust delivery of data over unreliable communication channels. Besides *reliability*, the other main concepts introduced by TCP were multiplexing and flow control. *Multiplexing* refers to the process of segmentation and reassembly of packets at source and destination nodes. *Flow control* is based on a window mechanism. A window mechanism allows for transmission of packets until the number of unacknowledged packets exceed

a certain limit called *window size*. In case of retransmission, all packets after the last acknowledged packet are retransmitted. In order to support flow control and in-sequence transmission, the TCP header contains *sequence and acknowledgement numbers*. As a consequence no extra acknowledgement packets are required in a two way conversation, as packets in one direction contain the acknowledgement of packets sent on the return path. In order to support various applications on a single node, the concept of port numbers has been introduced. They refer to applications such as file transfer, mail and identification services. Port numbers are negotiated at the beginning of a conversation. This can be omitted, if a port has already been assigned by the *internet assigned numbers authority (IANA)*. Such permanent assignments are called well known ports, as a certain application or internet related protocol can be addressed without preceding negotiation. Examples include ports 20, 21 assigned to the *file transfer protocol (FTP)*, port 23 assigned to *Telnet* and ports 66, 150 allocated for SQL database purposes.

The connectionless counterpart of TCP is called *user datagram protocol (UDP)*. It provides efficient transmission of information lacking the need for reliable in-sequence delivery. Clearly, the protocol overhead is much less than in TCP, as only checksum, length information, source and destination ports are included in the UDP header. Please note, that UDP ports are different from those used in TCP, although the majority of internet related protocols has been assigned to the same well known ports in TCP and UDP. Due to space reasons, detailed descriptions have been omitted. For more information refer to TCP/IP textbooks such as [54], [76] and [84].

With the introduction of multimedia services, TCP and UDP were insufficient to support this new traffic profile. Being designed for classic IP traffic such as text documents and binary files, they had to be enhanced by a new protocol suitable for real time multimedia traffic. As a consequence, the *real time protocol (RTP)* and the *real time control protocol (RTCP)* have been defined in RFC 1889. Please note, that RTP and RTCP have been included in ITU recommendation H.225. RTP follows an integrated approach, as RTP protocol handlers are often integrated into applications rather forming a separate layer. At this point, it is important to point out, that RTP has not been defined for IP only, other protocols may be supported as well. As media streams are encoded using the same codecs as are used for multimedia operation in ISDN and B-ISDN (refer to section 5.2.2), RTP has to deliver the same flexibility as the protocols used there. Therefore, each RTP implementation is companioned by a *profile specification document*, which maps

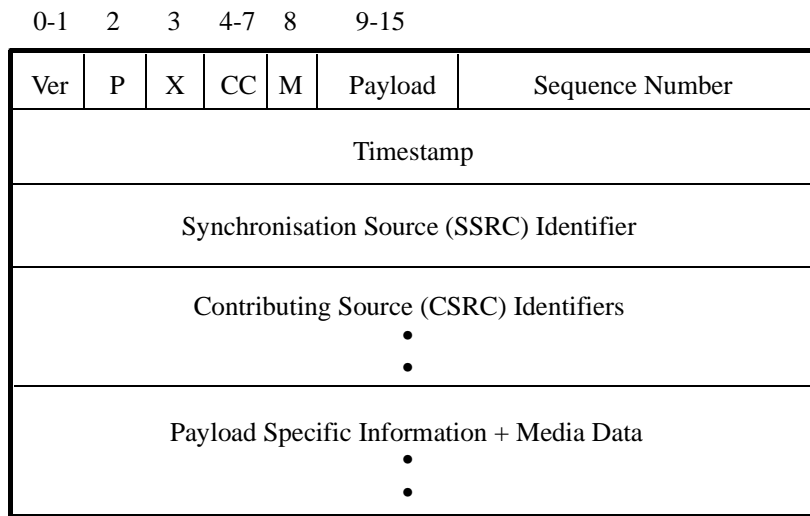


Figure 5.16: RTP message format

payload formats to their specification documents, e.g. H.261 and G.728. One example for such a profile is RFC 1890, which contains a profile for audio and video conferences with minimal control. RTP is built on top of UDP, as UDP provides necessary basic services like multiplexing and error detection. In opposite to RSVP, RTP does not reserve resources in the network, but it allows the receiver to recover data in the presence of loss and jitter. These functions are reflected by the RTP packet format as shown in figure 5.16. Please note, that only fixed RTP header fields have been included, payload specific information is given through the profile specification. Each RTP message includes

- *header information* with length of 2 bytes. The *version* (V) is currently set to 2. The *padding* (P) bit indicates the presence of padding bytes, whereas the last byte in each packet contains the number of padding bytes. If the *extension* (X) bit is set, an optional header extension follows the fixed header. The *contributing source count* (CC) indicates the number of contributing sources added to the payload. The *marker* (M) bit is specific to the profile, whereas bits 9-15 contain the *payload type* used to identify the profile used to interpret the data contained in the payload.

- a *sequence number*, which used for reordering, loss detection and compensation.
- a *timestamp* to allow synchronization and jitter compensation.
- the *synchronization source (SSRC) identifier* to determine the source.
- a *list of contributing sources (CSRC)* to identify contributing sources such as mixers.

In order to preserve flexibility and interoperability, the concept of mixers and translators was employed in the RTP framework. According to RFC 1889, a *mixer* describes an intermediate system, which combines packets of more sources and probably changes the data format, i.e. the encoding. One application might be the conversion of RTP audio or video streams into streams with lower quality to ensure continuous transmission over low bandwidth channels. Affected parties in a videoconference will then still be able to follow the call by receiving a multimedia stream with degraded quality of service, instead of being excluded from participation due to equipment limitations. In opposite to mixers, *translators* do not combine any media streams. Their applications include encoding conversion and multicast to unicast replication. Using translators, multicast multimedia streams may be tunneled through firewalls. Please note, that the use of mixers and translators is not supported by recommendation H.323.

With RTP being responsible for end-to-end transport of real time data, RTCP monitors quality of service and conveys control and session information. In particular, it provides

- *Quality of service feedback* - During conversation, each destination returns reports about the quality of their reception to each sender. This information including the number of lost packets, jitter, and round-trip delays is used by the source to adjust encoding rates and other parameters.
- *Intermedia synchronization* - In case, audio and video data are carried in separate packet streams, they need to be synchronized at the receiver. This type of synchronization is often referred to as *lip synchronization*.
- *Identification* - The source description of each participant is propagated through the network by means of RTCP. It contains a unique *canonical*

endpoint identifier as well as e-mail address, phone number and full name.

- *Session control* - This includes functions for session release and transmission of application dependent data. The former is achieved through the use of the BYE message, whereas the latter is based on the APP message. Exchanging user data allows for applications such as notifications between participants.

The quality of service feedback containing time sensitive information is sent periodically since it may become obsolete after some time. In order to prevent the network to become flooded by RTCP packets, RTCP allows the update period to be increased for larger multicast groups. RTCP packets are addressed to the same destination as the related RTP packets, but to a different port.

Another protocol worth to mention is the *real time streaming protocol (RTSP)* defined in RFC 2326. It has been defined to support media servers. Their main functions include the ability to play and record various media such as audio and video. Media servers can be seen as the counterparts of voice mail systems and IVRs. RTSP offers controls similar to those in a VCR remote control. A client can instruct the server to play, record, fast forward, rewind, and pause. Before usage the server has to be configured with the target IP addresses, target UDP ports and codecs to be used. Typically, media streams originating from the media server are embedded in RTP packets.

We have already mentioned the shortcomings of TCP and UDP with respect to the transport of real time traffic. A similar insufficiency also exists for the transmission of signaling information. Accordingly an alternative transport protocol has been considered in version 5 of the H.323 specification. This *stream control transport protocol (SCTP)* aims to provide similar transmission speeds like UDP at safety levels superior to TCP. As suggested by the name, SCTP is well suited to convey streamlined traffic. Furthermore it provides message transmission, *selective acknowledge (SACK)* and support of multiple subflows within a single connection. SCTP is a connection oriented protocol, which provides a four-way handshaking procedure to associate one SCTP endpoint to the other. As opposed to TCP or UDP, an SCTP endpoint may possess one or more IP addresses. SCTP associations are usually terminated using a three-way shutdown procedure.

type	name	description
00	DATA	user data
01	INIT	start SCTP association
02	INIT-ACK	ACK for INIT
03	SACK	ACK for user data
04	HEARTBEAT	keep-alive message
05	HEARTBEAT-ACK	ACK for keep-alive
06	ABORT	ungraceful terminate association
07	SHUTDOWN	graceful terminate association
08	SHUTDOWN-ACK	ACK for SHUTDOWN
09	ERROR	operation error
0A	COOKIE ECHO	pass cookie (during setup)
0B	COOKIE-ACK	cookie response
0C	ECNE	explicit congestion notification
0D	CWR	congestion windows reduced
0E	SHUTDOWN-COMPLETE	final message for SHUTDOWN

Table 5.4: SCTP chunk types

The protocol header consists of a common part and several optional blocks called *chunks*. Each SCTP chunk features type, flags and length information and are best compared to TCP messages such as SYN and ACK. According to the chunk type, it carries mandatory and optional chunk parameters. As the description of each chunk is too exhaustive to be provided here, we refer to the reference [21] for more details. A list of SCTP chunk types is provided in table 5.4, which also serves as an indication of the far reaching capabilities of SCTP. These include procedures for congestion management and keep-alive messages for idle associations.

However, the ITU-T also had to cope with connections between gatekeepers and gateways, which deserve features SCTP could not provide. As a consequence the *aggregate server access protocol (ASAP)* defined in RFC 5352 has been considered to address further reliability issues. Whereas SCTP operates at the link level, ASAP provides support for server pools and surviving nodes at the session level. Furthermore, abstraction of location, load sharing and plug-n-play capabilities have been introduced in ASAP.

Call Control

In the last section transport of multimedia streams and their quality of service control mechanisms were considered. The current section focuses on call related activities also found in telecommunications networks such as the ISDN. Call control is based on ITU recommendations Q.931 and Q.932, although some messages have been omitted. Table 5.5 lists the remaining messages used for call setup on LANs. Please note, that some messages are not used, while others are explicitly forbidden. An example for the former is the SEGMENT message, as segmentation and reassembly is automatically performed by lower layers.

Even the information elements used for call control in LANs conform to the information elements defined in the ISDN framework. Additional mandatory signaling information is conveyed by means of user-to-user signaling. Examples include additional address fields, conference data, reason codes and information about the multipoint control used.

Supplementary services have been defined the same way as for ISDN. The relevant recommendation for supplementary services based on H.225 is H.450.x. Currently the *call transfer* and *call diversion* services are supported. For call transfer, the *call transfer by rerouting* procedure was chosen as the primary one, as routing is inherent to LANs. In opposite to services for switched networks, the gatekeeper received special attention, because for call transfer and call diversion services the gatekeeper may be involved in the execution.

As the H.323 suite of protocols is designed for networks, which do not provide quality of service guarantees, Q.931/Q.932 signaling alone is not sufficient. Following the example of ATM in the B-ISDN framework, a protocol for admission control, address translation and bandwidth management had to be introduced. As a consequence, the *registration, admission and status (RAS)* protocol has been specified by the ITU in recommendation H.225.0. The entity mainly related to the RAS protocol is the gatekeeper. In order to find a gatekeeper, a terminal sends a gatekeeper request (GRQ) using unicast or multicast transmission. In case of the latter, the target is a certain multicast group address and a certain IP port. A terminal having received a gatekeeper confirm (GCF) will proceed by sending a registration request (RRQ) to the gatekeeper. If terminals are successfully registered with a certain gatekeeper, they perform an *admission control request* before initiating or answering a call. This is usually done by an admission request (ARQ)

message name	usage
call establishment messages ALERTING CALL PROCEEDING PROGRESS SETUP CONNECT SETUP ACKNOWLEDGE CONNECT ACKNOWLEDGE	mandatory conditional mandatory optional mandatory mandatory optional forbidden
messages during active call phases USER INFORMATION SUSPEND REJECT RESUME REJECT HOLD SUSPEND RESUME HOLD ACKNOWLEDGE SUSPEND ACKNOWLEDGE RESUME ACKNOWLEDGE HOLD REJECT RETRIEVE RETRIEVE ACKNOWLEDGE RETRIEVE REJECT	optional optional optional optional optional optional optional optional optional optional optional optional
call clearance messages DISCONNECT RESTART RELEASE RESTART ACKNOWLEDGE RELEASE COMPLETE	forbidden not used forbidden not used mandatory
miscellaneous messages SEGMENT FACILITY REGISTER NOTIFY STATUS ENQUIRY CONGESTION CONTROL INFORMATION STATUS	not used mandatory not used optional conditional mandatory forbidden optional mandatory

Table 5.5: Q.931 messages used in H.225.0

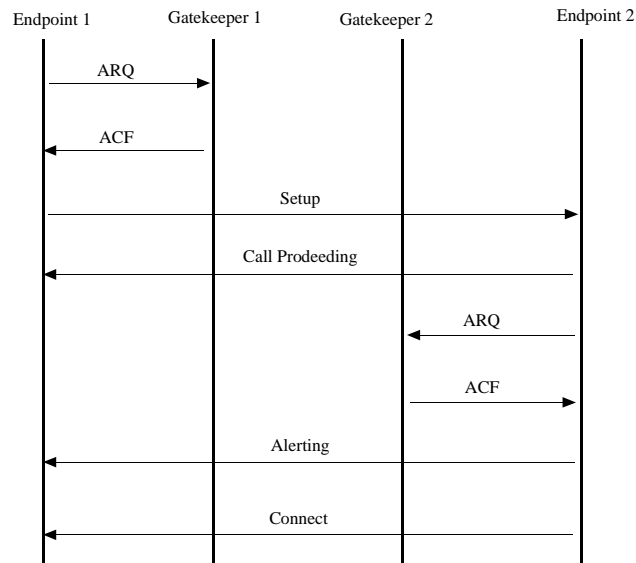


Figure 5.17: Successful H.323 call attempt (direct call model)

message. The gatekeeper responds either with an admission confirm (ACF) or admission reject (ARJ). In the former case the call setup procedure is continued, whereas in the latter case the call attempt did not succeed. Figure 5.17 shows a successful call attempt for two terminals registered at a gatekeeper. The same request/response scheme applies to the majority of RAS procedures. A complete list of RAS messages is shown in table 5.6. As the demand in control and management functions raises for multipoint conversations, the gatekeeper is often combined with the multipoint control unit (MCU) into a single unit.

In order to support signaling between endpoints, recommendation H.245 defines additional services and protocol entities. One *H.245 control channel* is established between two endpoints for each conversation, the endpoint is participating in. As messages required to open and close channels are defined in H.245, they do not apply to the control channel itself. As a consequence, channel 0 has been assigned as H.245 control channel. The same applies to RAS channels, which are established before any other channel between an endpoint and the gatekeeper. H.323 endpoints will support procedures for the following protocol entities:

RAS message	abbreviation
admission request	ARQ
admission confirm	ACF
admission reject	ARJ
bandwidth request	BRQ
bandwidth confirm	BCF
bandwidth reject	BRJ
disengage request	DRQ
disengage confirm	DCF
disengage reject	DRJ
gatekeeper request	GRQ
gatekeeper confirm	GCF
gatekeeper reject	GRJ
information request	IRQ
information request reponse	IRR
location request	LRQ
location confirm	LCF
location reject	LRJ
registration request	RRQ
registration confirm	RCF
registration reject	RRJ
unregistration request	URQ
unregistration confirm	UCF
unregistration reject	URJ

Table 5.6: RAS message types

- *Master-slave determination* - To avoid conflicts due to simultaneous events, endpoints follow the master-slave determination procedure as given in recommendation H.245. Actions performed by the chosen master include the choice of the relevant multipoint controller and the distribution of encryption keys for media channels according to the security framework specified in recommendation H.235.
- *Capability exchange* - During call establishment a capability exchange is performed to identify the ability of each endpoint to receive and decode multimedia streams sent by the other endpoint.
- *Logical channel signaling* - To support logical channel signaling, an acknowledged protocol has been defined for opening and closing audio, video and data channels. Bidirectional channels are supported as well. Assuming a channel is requested by both endpoints at the same time, master and slave have to be determined by following the procedure mentioned before. A close request may be issued by each endpoint. In case the receiver requests the closure of a channel, the transmitting endpoint may accept or the reject the closure request. A channel is closed by the receiver, if the carried information can not be decoded. Please note, that an association between the logical channel identifier and the synchronization source identifier (SSRC) used in RTP is provided by H.245 for audio and video streams.
- *Mode request* - As a result of the capabilities exchange procedure, both terminals are aware of the capabilities to encode, transmit, receive and decode certain media streams. After the sender has indicated its capabilities to transmit, the receiver may request a certain mode to be used for transmission by initiating a mode request. If the sender does not want to be controlled by the receiver, it simply does not send any transmission capabilities.
- *Round trip delay determination* - A mechanism to measure the round trip delay between endpoints is provided to support applications, which require to know these characteristics.
- *Maintenance loop signaling* - Maintenance loop messages are used by an endpoint to perform maintenance loop functions. After a loopback has been completed, all loops are disconnected and audio, video and data paths are restored.

The messages used to provide the listed functions follow a request/response scheme similar to RAS messages. *Commands and indications* have also been defined to support functions such as conferences, flow control, encryption, jitter and skew indications, audio inactive and video freeze signals.

As already mentioned before, the gatekeeper may participate more actively in call control. This more centralized approach may either be dictated by the application or become an essential feature in the communication across multiple domains. Centralization is achieved by forcing the client to use the *gatekeeper routed model* instead of the *direct call model*. In the former, the gatekeeper simply puts its own IP address in the ACF call signaling address, whereas for the latter only RAS messages are routed by the gatekeepers.

It has to be mentioned at this point, that *multipoint graphic communication* as defined in the T.120 series of recommendations has been omitted. Usually, the data management function is carried out by terminals and MCUs. Applications include still image transfer and multipoint binary transfer. For further information please refer to [16].

Extensions and Recent Developments

The experience with version 1 of recommendation H.323 triggered several new extensions and optimizations to the protocol. These include

- *Fast connect* - The conservative call setup shows a bad performance as it involves message round trips for admission process, setup-connect sequence, capabilities exchange, master-slave determination and logical channel setup. Further taking into account the delays caused by TCP to setup a session, delays sum up to a considerable amount. The effect might become worse in a WAN environment or when using the gatekeeper routed model. As an alternative endpoints may implement the fast connect procedure by including the fastStart parameter in the SETUP user-to-user information element. This parameter conveys the information required to set up the logical channels immediately. Furthermore the fast connect procedure allows for the integration with intelligent network services, as media exchange is supported even before receiving a CONNECT message. In the telephone network, a call is charged before sending a CONNECT message. Examples include calls to 09xx numbers and calling card services.

- *H.245 tunneling* - In order to reduce the number of TCP sessions, H.245 messages may be encapsulated in the Q.931 user-to-user information elements. Consequently, the Q.931 channel needs to remain open for the entire conversation. If no Q.931 message is pending to carry H.245 protocol data units, a FACILITY message is sent.
- *Tunneling of telephony protocols* - Appendices M.1, M.2 and M.3 of the H.323 specification provide details on how to convey DSS1, ISDN user part (ISUP) and QSig messages within the context of an established H.323 call.
- *H.235 security framework* - Service providers as well as users desired the implementation of authentication mechanisms and encryption. The former may be achieved by using certificates or simple challenge response schemes. The call signaling channel may be secured by deploying technologies such as IPSEC or *transport layer security (TLS)*. The media channels may be encrypted by introducing new capability types for each combination of codec and encryption mode. As the keys for encrypting the media channels are exchanged using the H.245 channel, which also has to be secured. The relevant method is then negotiated during the initial setup procedure.
- *H.332 conference model* - Conventional conferences treat each member of a conference as a participant, there is no difference between the *virtual panel* and the *virtual audience*. The latter consists of people only listening to the conversation, they do not participate actively in the conference. To overcome this shortcomings especially for very large conferences, recommendation H.332 introduces an alternative approach by treating active participants separately from the virtual audience. By using multicast technologies, transmission to the audience may be further optimized in terms of traffic and load.
- *Extended video capabilities and far end camera control* - Besides other enhancements, recommendation H.241 adds support for the H.264 codec family to H.323 systems. Far end camera control has already been specified in recommendation H.282 and H.283, but has been felt to be too costly and difficult to implement. This has been remedied with in Annex Q of the H.323 specification.

- *H.323 over UDP* - Due to the use of TCP as transport protocol for call signaling, the connection setup procedure has been found to add significant delay to the overall call setup time. As an option, UDP may be used instead. In order to prevent from severe impacts on reliability, several issues such as acknowledgement mechanisms and keep-alive messages have to be considered. As a consequence of the connectionless nature of UDP, a restart procedure indicating the termination of the previous session has been defined.
- *End-to-end quality of service and service priorities* - The former Annex N of the H.323 specification has been further enhanced and moved to recommendation H.361. QoS negotiation takes place before the called party receives an ALERTING signal. The corresponding information elements are either exchanged out-of-band or by use of H.245 tunneling.
- *Supplementary Services over HTTP* - H.323 Annex K defines the use of the hypertext transport protocol (HTTP) for H.323 entities. HTTP applications are not able to access the features thus allowing for web based customization and software uploads. Please note, that the term supplementary services has been used in a wide sense here.
- *Generic extensibility framework (GEF)* - The GEF provides a more flexible way to introduce supplementary services to the gatekeeper. Before a GEF service may be invoked by a remote entity, capabilities have to be negotiated according to the RAS protocol. Although not limited to a certain set of messages, RRQ, RCF and RRJ will be used most likely. At the time of writing, already 23 extensions have been defined. These include number portability, delayed call establishment, multi-level precedence, multiple message release sequence capability, location numbers, NAT traversal, messaging, presence, handling of geographic information, call signaling transport channel suspension and redirection.
- *Support of unintelligent terminals* - Typical for the majority of proprietary telephone sets, supplementary services are executed in the PBX rather than in the telephone itself. A similar centralized approach is taken in H.323 Annex L, where signaling between unintelligent H.323 terminals and the responsible feature server is described. The stimulus protocol introduced there can be seen as a big brother to the ISDN

keypad protocol. It covers control of audio units, presentation via status lamps and text display as well as service invocation via function keys, soft keys. Please note, that this type of operation has been suggested for but is not limited to so called *simple endpoint types (SET)* introduced in H.323 Annex F.

- *Communication between administrative domains* - In order to exchange information between administrative domains, H.225.0 Annex G has defined a new protocol, which provides the capability to perform address resolution, access authorization information and report resource usage for calls. It is also possible to use a clearinghouse, which aids in address resolution and settling billing between service providers without pre-arranged service agreements. For inter-domain communication a new functional entity, the so called *border element (BE)* has been defined.
- *Gatekeeper redundancy* - Targetting reliability of standard telephone switches, the concept of alternate gatekeepers has been introduced in the H.323 recommendation series. Version 4 introduced new details on the procedures that endpoints may follow to increase system robustness and reliability. Version 6 extended these concepts even further to allow for an assigned gatekeeper to be chosen from a list of alternate gatekeepers.
- *H.323 URLs* - With H.323 Version 4 the universal resource locator (URL) "h323" has been introduced thus allowing for consistent access to users and services. A typical URL for H.323 calls looks like h323:user@host. Since version 5 gatekeepers have to advertise themselves also as *session initiation protocol (SIP)* gateway to allow for extended address resolution.

Apart from the standardization sector, a lot of effort has been put on the network infrastructure used for voice over IP. With respect to the various address translations such as *native address translation (NAT)* and *port address translations (PAT)*, signaling channels have to be actively monitored to allow for proper communication. Care has to be taken for H.245 signaling and RAS, especially if the network scope is extended beyond the firewall. As far as quality of service is considered, *differentiated services (DiffServ)* evolved as the architecture of choice. Although DiffServ does not provide end-to-end guarantees like RSVP, it is easier to implement within and between network

domains. Traffic is classified and assigns a *differentiated services code point (DSCP)*, which is stored in the six most significant bits of the *type of service (TOS)* field in the standard IP packet. Another networking technology called *multi protocol label switching (MPLS)* emerged recently. MPLS provides quality of service guarantees by establishing a controlled path through the network by interfacing with layer 2. MPLS has been extended to *generalized multiprotocol label switching (GMPLS)* to cover issues of heterogeneous networks as well. With MPLS, each packet in a traffic stream is assigned a label by the edge router. Before arriving at the target system the label is removed again. During its travel time, the packet may pass several network regions with different configurations. In order to deal with these issues, a label stacking mechanism has been defined. By stacking addresses, tunnels are built through foreign network regions. MPLS labels are simply pushed at the ingress to each tunnel and popped at the tunnel's egress. For more information on quality of service in networks, DiffServ and MPLS, please refer to [210].

A thorough treatment on voice over IP may be found in [136], [65] and [66]. In all books, interoperability played a major role. Douskalis focuses on public network integration, whereas in [136] the authors emphasize more on data communication network issues.

5.2.5 A New Millenium of Signaling

A new era of signaling emerged with the introduction of the *session initiation protocol (SIP)* in 2002, a text based protocol for IP multimedia communication. Unlike the H.323 suite of protocols, SIP does not stick to the concepts of ISDN anymore. The ISDN framework has been considered as unsuitable and too complex. In fact we know from H.323, that complementary protocols such as RTP and SCTP became necessary to cope with the requirements of the underlying transport network. This had been a first step deviating from ISDN concepts. As a consequence of this omission, SIP radically differs from what has been developed before. This is best seen from the definition of signaling gateways. When mapping SIP to the *ISDN user part (ISUP)* of the *common channel signaling system no. 7 (SS7)*, a significant amount of information gets lost in each direction (This problem was partially solved by *ISUP tunneling* through SIP based networks). But there is more to the design of SIP and similar protocols of this new era ! It is not only leaving the ISDN framework, it is a completely new approach to multimedia protocol design.

And it is a policy of small steps: By keeping complexity at a manageable level, the overall failure rate necessarily decreases. This strategy has been in place already for years and has been commonly applied to the definition and extension of IP based data networking protocols. This led to the adoption of the *Internet Engineering Task Force (IETF)* as a replacement of the traditional standardization bodies such as the ITU and ETSI. In fact, the rules of the IETF allowed for rapid introduction of de-facto standards. Some more recent signaling protocols such as the *inter-Asterisk exchange protocol (IAX)* [2] omitted the IETF as well and were released as an industrial specification.

Since the new millenium of signaling has begun, there has been an unsolved controversy on how a signaling protocol shall be designed. On one hand we do feel the flexibility of data networking and on the other hand protocols such as SIP and IAX lack an adequate supplementary service mechanism. Furthermore there has been a discussion on the reliability of lately introduced systems and protocols. Some have argued, that SIP as the reinvention of plain old telephony caused an erosion in reliability and feature richness. But we already know from the world of mobile telephones, that reliability is a relative measure. Thinking further, almost all new protocols have been the answer to rapidly changing demands of customer and industry. In the old millenium of signaling the major players were incumbent providers, so the change in philosophy goes hand in hand with a change in economic structure.

The choice of architecture and protocol for communication and call centre platforms has become more than ever a question of requirements, a fact we aimed to emphasize in this short interlude.

5.2.6 IP Networks with Text Based Signaling

In order to focus on text based signaling for multimedia networks, we now turn attention to the *session initiation protocol (SIP)*. Similar to H.323, SIP does not stand for its own. Instead several additional protocols are required to enable multimedia communication. As this is only a brief overview on text based signaling, we will only discuss some of the core protocols typically met in SIP enabled networks. The prominent position of SIP in the world of voice over IP could be explained by the fact, that it has been adopted by the *third generation partnership project (3GPP)* as a vital part of their mobile networking architecture.

The SIP protocol is a close relative to the *simple mail transfer protocol*

(*SMTP*) and the *hypertext transfer protocol (HTTP)*. As such it has inherited some features and ideas, which have been in place for years.

The major entity in a SIP network is the so called *user agent (UA)*, which generates requests or responds to them. Although abstract in nature the user agent is best seen as a piece of software embedded within a networking device. Two types are distinguished:

- The *user agent client (UAC)* originates a request and triggers session establishment. This typically occurs on initiation of a new call.
- The *user agent server (UAS)* responds to the requests issued by UACs.

These functions do not exist exclusively, in most cases both of them will be implemented in a single device. Each SIP device has to support a minimum set of SIP messages and parameters, which assures a minimum level of communication between them. Although the industry adheres to these rules, some vendors have chosen to enhance their devices by proprietary functions not supported by other vendors. Unfortunately this led to a slow development of more advanced features such as supplementary services.

Besides the user agent the following entities are important for SIP networks:

- The *proxy server* acts on behalf of a user agent. Being concerned with routing of messages, it aids in topologizing the network. Typically the proxy server implements a user agent, as it actively participates in communication. In some cases, it also splits requests for routing to multiple destinations (*forking proxy*).
- The *redirect server* or *redirector* provides an alternative address upon request. This might occur on looking up a specific target, during execution of the call forward service or when balancing the load between devices.
- The *registration server* or *registrar* is responsible to authenticate a new user and to record its current location. Location information is usually passed to the *location server*, which often is a part of the registration server.

These entities should not be understood as physical ones, as they are often combined into single devices - the so called *SIP servers*. Very often we

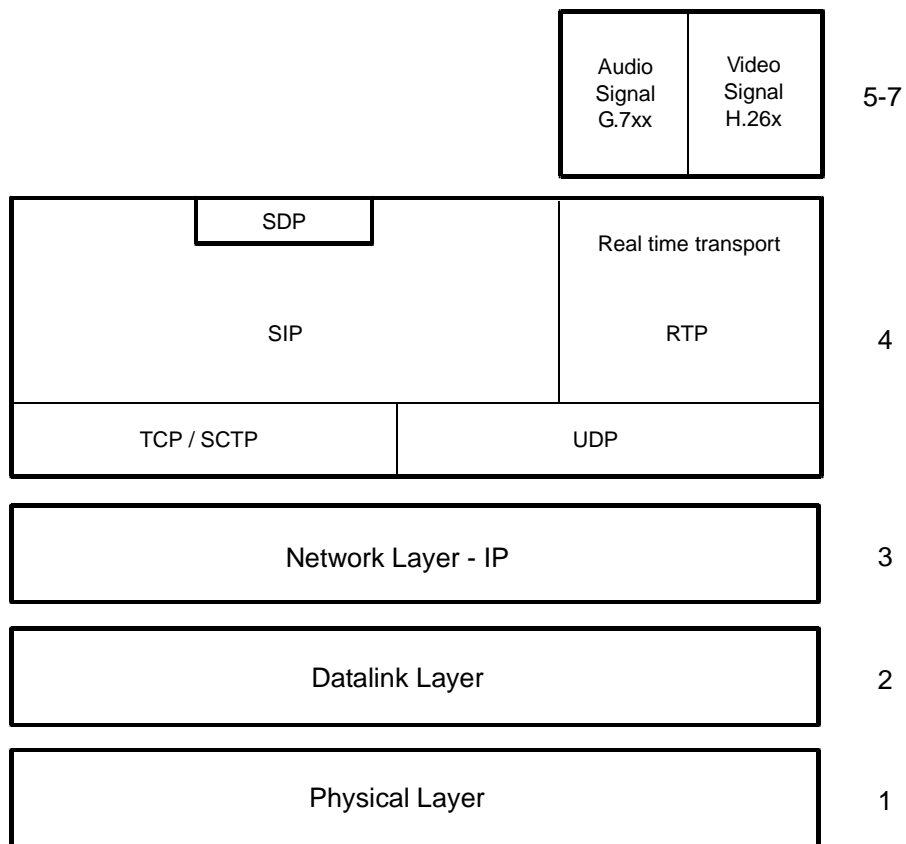


Figure 5.18: Voice over IP protocol stack common to SIP networks

do find a combination of redirect server and registrar. For more information on SIP entities please refer to the book [178] of T. Russell.

Transport Network Protocol Overview

SIP has been designed to work on top of the TCP/IP protocol stack. This is also reflected by the protocol stack shown in figure 5.18. We have already treated layers 1-3 and the protocols TCP, SCTP and RTP in the context of H.323. So we refrain from repetition and instead refer the reader to section 5.2.4.

field	description	usage
v	protocol version	mandatory
o	owner/creator and session ID	mandatory
s	session name	mandatory
i	session information	optional
u	uniform resource ID (URI)	optional
e	email address	optional
p	phone number	optional
c	connection information	mandatory
b	bandwidth information	optional
z	time zone adjustment	optional
k	encryption key	optional
m	media information	optional
a	media attributes, attribute lines	optional
t	session start and stop time	mandatory
r	repeat times	optional

Table 5.7: SDP protocol elements

Session Control

Call or session control in SIP networks is mainly exercised by the SIP protocol. The *session description protocol (SDP)* is used to describe the session. This includes information on codecs, the phone number and other types of connection information. SDP already existed some time before SIP has been defined. It played a vital part in the description of multicast sessions used in the *internet's multicast backbone (MBONE)*. SDP protocol elements are encoded as text in property assignment style, i.e. they adhere to the syntax `field=value`. A list of SDP fields is provided in table 5.7. In order to assure a minimum session description, some fields are mandatory. As each line ends with CRLF, SDP messages are readable to humans without any further parsing. For more details on SDP and its parameters refer to the books [10] and [178] by A.B. Johnston and T. Russell.

SIP is also a purely text based protocol similar to HTTP or SMTP, which implements a classical request/response communication. The request messages are also called methods in SIP nomenclature, but we will stick to the more general terms request and response. If a SIP message can be conveyed in a single packet, UDP is used for transport. Otherwise a TCP or SCTP ses-

SIP request	description
INVITE	establish new session
REGISTER	register/authenticate device
BYE	release session
ACK	acknowledge
CANCEL	cancel pending request
OPTIONS	query capabilities from UA
REFER	request UA to access other resource
SUBSCRIBE	establish subscription for notification
NOTIFY	notify an event
MESSAGE	convey an instant message
UPDATE	modify session state
INFO	send information
PRACK	provisional acknowledge

Table 5.8: SIP request messages

SIP response class	description
1xx	provisional
2xx	successful
3xx	redirect
4xx	client failure
5xx	server failure
6xx	global failure

Table 5.9: SIP response classes

sion is established to transport the signaling information. SIP also supports multicast transmission, which is very useful for conference calls.

A SIP request follows the syntax `REQUEST URI VERSION`, where the URI adheres to the convention `SIP: user@domain`. Again we do see a close relation to the protocols used for mail and the world wide web. A list of possible SIP requests is shown in table 5.8. We omit the description of parameters, as this would be beyond the scope of an overview. Instead we once again refer to the books [10] and [178] by A.B. Johnston and T. Russell. We also do not delve into the details of SIP responses. Instead we provide an overview about the possible classes in table 5.9.

As an example for a typical SIP conversation consider the message flow

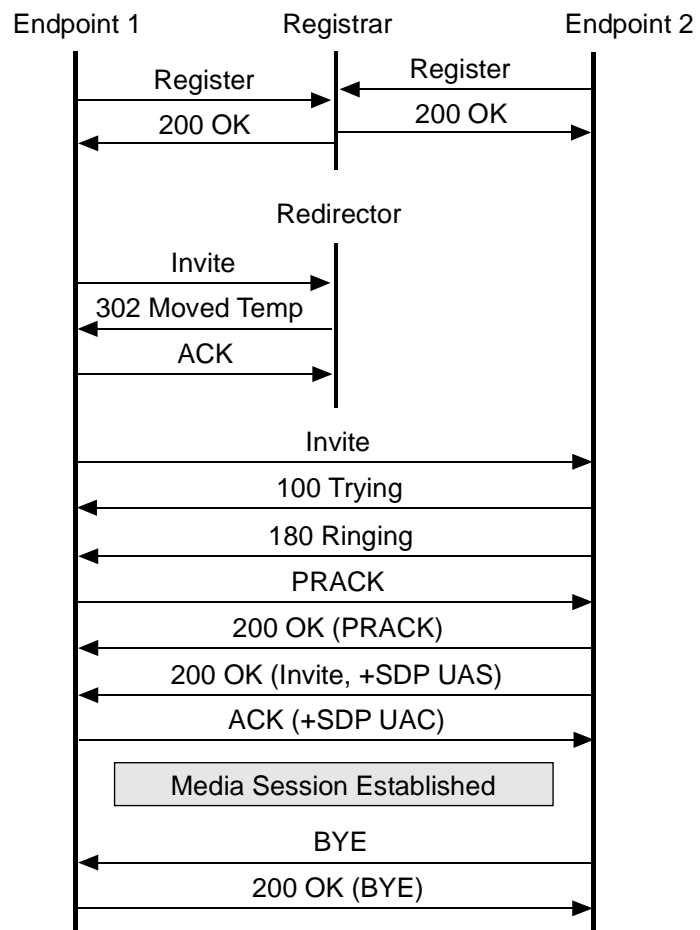


Figure 5.19: Typical SIP session

in figure 5.19. Prior to the first communication establishment an endpoint declares himself as evident by sending a REGISTER request to the registrar. The IP address is recorded and provided to proxies on request for routing purposes. Each registration request is acknowledged by a 200 OK response. Endpoint 1 is not aware of the location details for endpoint 2 and issues an INVITE request to the redirector. The redirector sends back a 302 Moved Temporarily response containing the desired address. In opposite to a proxy server, the redirector does not forward requests, it only responds to them. In other protocols we often find a specific address resolution request to perform a similar operation. After sending an acknowledge to the redirector, endpoint 1 initiates the session establishment procedure to endpoint 2 by sending an INVITE request. The called party responds with 100 TRYING and 180 RINGING. As provisional responses of the 1xx class are usually not acknowledged, the calling party issues a PRACK request. In case the 180 RINGING response got lost, it would have been resent by endpoint 2. Without the optional PRACK method there are situations, which leave the caller without any feedback. The PRACK request is acknowledged with the 200 OK response. Now the session negotiation starts. The called party issues a 200 OK response containing an SDP block, which describes the capabilities of endpoint 2. A similar SDP block is conveyed in the ACK message from endpoint 1. It carries the final choice of media and connection properties. Now the conversation can start and RTP takes over. After some time the called party decides to end the call by sending a BYE request to endpoint 1, which is acknowledged by a 200 OK response.

Although we have dealt with a more complex example, we only touched on some of the possibilities provided by the SIP protocol. There is more to say about scenarios involving proxy servers, instant messaging, gateways and PSTN interworking. Many of them are found in reference [10].

Remarks and Extensions

As almost any other voice over IP protocol SIP is steadily modified and enhanced. Some of these modifications are within the core SIP framework, while others are related to external IETF protocols. We now list some of these features, which are certainly a subjective choice of the author.

- There are still problems for SIP and RTP to traverse firewalls and network entities, which perform *native address translation (NAT)*. In

order to solve these problems (which are not specific to SIP), the IETF proposed several supporting solutions such as *simple traversal of UDP through NATs (STUN)* and *traversal using relay NAT (TURN)*. The former is specified in RFC 3489, while the latter was still in draft state at the time of writing. When TCP is used for SIP messaging, a proxy server commonly deals with the issues of firewall traversal.

- Encryption occurs on a hop-by-hop basis using *transport layer security (TLS)*. If this is not sufficient and end-to-end encryption needs to be applied, S/MIME is the right choice.
- SIP gains its major importance from the efforts of the 3GPP, especially from the *IP multimedia subsystem (IMS)*. Accordingly several extensions have been proposed. These range from charging and billing to packet cable extensions. For more information refer to [178].
- Signaling information from traditional telephony protocols are only insufficiently conveyed across SIP networks. In case of ISUP an appropriate tunneling mechanism has been defined for SIP. For further reference please consult [18].
- Within certain limits SIP allows for CTI third party control, as only standard SIP messages are used [18]. Alternatively one may follow the SIP Servlets approach [3], which from programming perspective is more related to a protocol API than to a CTI API.

At the beginning we have restricted ourselves to a presentation of the core protocols only. By including several not so close protocols such as RTCP, RTSP and MGCP, any SIP based network may achieve great benefits. The former two protocols allow for a reach out in the direction of unified messaging applications, while the latter adds a distributed gateway architecture to SIP networks. For more information on these topics please refer to the books [65] and [66] by B. Douskalis.

5.2.7 IP Networks with XML Based Signaling

In the preceding sections we have discussed two flavours of telephony signaling. Whereas the H.323 protocol suite is hierarchically structured, SIP follows an evolutionary approach. While the former certainly poses some

limits on the development, a feature rich protocol compliant to the ideas of ISDN has evolved. Accordingly it is perfectly suited for enterprise needs. On the other hand, SIP is readable by humans and very flexible, but has also been considered as too limited to cope with complex requirements. However, both protocol suites are derived from classic signaling systems. From an architectural perspective, they are still isolated from the application. When approaching *extensive markup language (XML)* based signaling for IP networks, we will meet a protocol suite, which has initially not been designed with telephony signaling in mind. Instead it started from an application viewpoint, which is radically different from what has been developed until now. In fact, agents, messaging servers, IVR ports and the like are seen as devices, which provide services to each other. Although the *extensible messaging and presence protocol (XMPP)* is often described in the context of instant messaging, we will adopt a call centre perspective in this section.

Adhering to the client-server paradigm, XMPP is aware of the following two entities

- The *XMPP client* requests service from the XMPP server. It is simple in design and usually communicates with a single XMPP server called the *home server*. XMPP clients may be further classified into interactive and automated clients. The latter group is also known as *bots*.
- The *XMPP server* either processes or forwards the request issued by the XMPP client. In the latter case the target SMPP server is now responsible to carry out the necessary operations.

Very often XMPP is concerned with the interconnection of two or more XMPP clients. Messages are transported in a store-and-forward manner, i.e. there are no intermediate servers as common to SMTP driven email networks. The XMPP architecture follows a very flexible design, as services may be implemented as bots or extensions to the server (so called *server components*).

Transport Network Protocol Overview

XMPP is best described as a technology for streaming XML [15]. As such it relies on transport technologies such as TCP or SCTP. This has also been indicated in figure 5.20. Similar to H.323 and SIP, XMPP uses an RTP

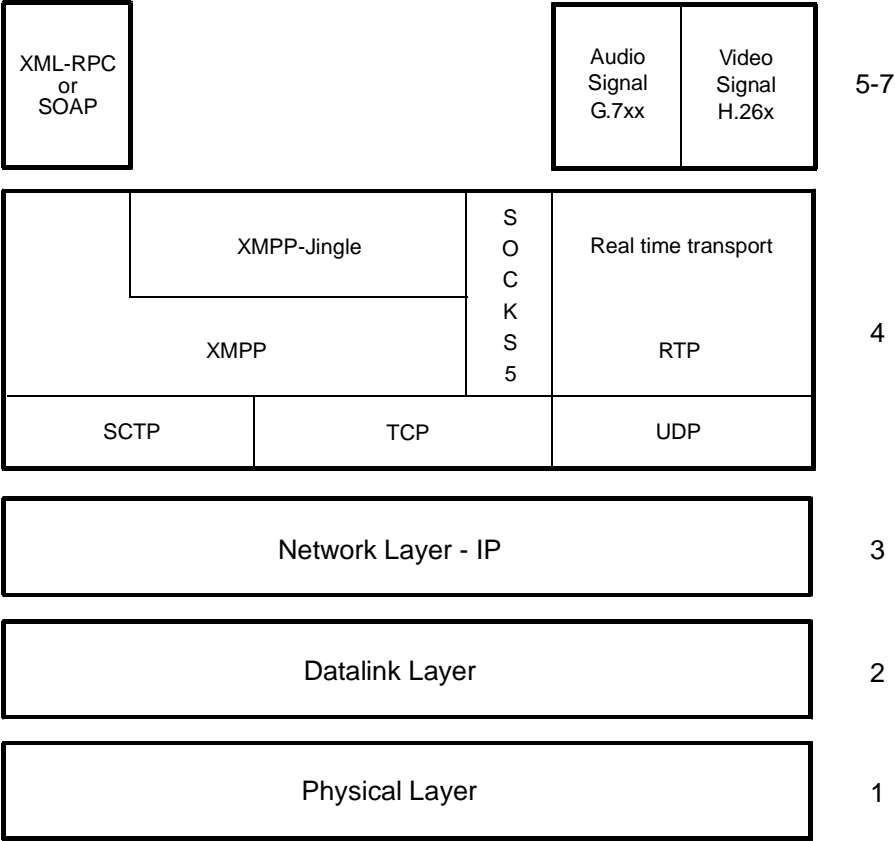


Figure 5.20: XMPP protocol stack

stream to convey voice samples across the IP network. Furthermore XMPP is able to process data transmission requests of any size by operating a SOCKS5 over TCP session. At the time of writing, SOCKS for SCTP has just started development and we expect it to be finalized in the next years. For more information on the protocols TCP, SCTP and UDP please refer to section 5.2.4 and the references stated there.

Session Control

As XMPP is not a voice over IP protocol in the classical sense, describing call or session control in this sections restricts our discussion to a single *XMPP extension protocol (XEP)* called *XMPP-Jingle*. We will touch on the remaining features of XMPP later in the text. XEPs are best thought of as modular extensions to the XMPP protocol. They provide a flexible way to introduce new protocol elements to XMPP. In the same way XMPP-Jingle formerly called *Google Talk* joined the set of XMPP specifications and became a XEP.

Similar to SIP and H.323, external addressing is based on the *uniform resource identifier (URI)* concept. Accordingly a typical address is stated in `xmpp:user@domain`. But XMPP also specifies an internal address called the *Jabber ID (JID)*, which is unique to each XMPP entity. XMPP also supports the extended *unicode* character set for address URIs. Name resolution is carried out by using the classic *domain name service (DNS)*, where the SRV record is queried to find an appropriate XMPP server. Besides addressing the discovery and negotiation of service capabilities is an important issue. Accordingly XMPP specifies an extensive service discovery mechanism, which is implemented as a two step process. First the entities known to the XMPP server and referred to as *items* are queried before details on their capabilities can be requested. These capabilities are referred to as *info*. In exactly this way a device may find out, if another device supports XMPP-Jingle signaling and is able to handle voice, video or data transmission.

The designers of XMPP have assigned the name *stanzas* to their communication primitives. Typically encoded as XML tag they constitute the basic unit of communication [15]. Each stanza is further distinguished by the *type* XML attribute. We have listed the basic stanza-type combinations in table 5.10. It suggests three main groups of stanzas. Whereas the *message stanza* mainly deals with sending short messages, alerts and notifications, the *information query (iq) stanza* is concerned about message handling in

stanza-type	description
message-normal	send normal message
message-chat	process bidirectional real time message
message-groupchat	convey multiuser messages
message-headline	send unacknowledged alert or notification
message-error	error indication for previous message
iq-get	retrieve information
iq-set	provide information
iq-result	return information or acknowledge iq-set
iq-error	error indication for iq-get or iq-set
presence	convey presence information
presence-subscribe	subscribe to presence service
presence-subscribed	acknowledge presence subscription

Table 5.10: XMPP stanza-type combinations

action	description
session-initiate	initiate new session
session-accept	accept offered session
session-info	provide session information
session-terminate	release session
transport-info	offer transport alternatives
transport-replace	redefine transport in use
transport-accept	accept transport offer
transport-reject	reject transport offer
content-add	add new content or media stream
content-remove	remove content or media stream
content-modify	modify content attributes
description-info	provide application information

Table 5.11: XMPP-Jingle actions

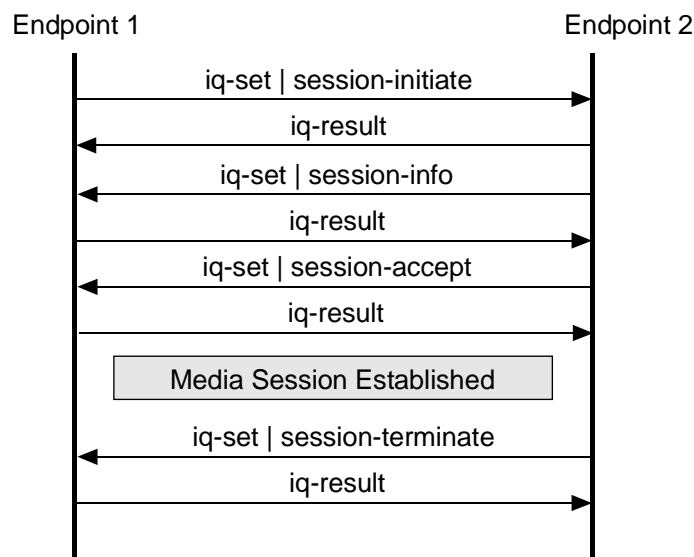


Figure 5.21: Typical XMPP multimedia session

a request/response style. The *presence stanza* as the least structured one is responsible to handle presence information of any kind. With respect to XMPP-Jingle we will only meet the second group, because all XMPP-Jingle messages are conveyed across the network using *iq* stanzas. Accordingly XMPP-Jingle is best considered as running on top of XMPP. This already has been indicated in figure 5.20. XMPP-Jingle communication primitives are encoded as information in the *action* property of the *jingle* XML tag. These so called *actions* are summarized in table 5.11. For a detailed discussion of XMPP stanzas and XMPP-Jingle actions please refer to the book [15] of P. Saint-Andre, K. Smith and R. Tronçon.

As before for SIP and H.323 we will proceed with an illustrative example. Consider the message flow in figure 5.21. As noted above, all XMPP-Jingle messages are embedded in an *iq-set* unit. Accordingly all messages are acknowledged by *iq-result* or rejected by *iq-error*. As we are concerned with a successful call attempt, the latter unit does not appear in the example. However, the *iq-result* unit does not convey an XMPP-Jingle answer to the request, as it operates on another layer. Instead responses are represented by XMPP-Jingle messages as well. The handshake procedure is carried out by use of *session-initiate* and *session-accept*.

Both carry additional data such as codecs, speed and connection information. The latter is typically represented by a combination of *application type* (e.g. RTP) and *connection method* (e.g. UDP). The parameters stated in the `session-accept` message represent the final choice of communication parameters as a result of capability negotiation. During the handshake procedure further messages may occur. Less preferred parameters may be conveyed using the `transport-info` message. In example 5.21 we have shown a `session-info` message, which acts as a ringing indication. In general, `session-info` may contain almost any information relevant for the application. Examples include the description for a data chunk or the window size for a video stream. We have noted above, that `session-accept` finalizes the handshake. Accordingly the conversation starts at this point. The session properties may be altered during conversation by use of the XMPP-Jingle messages `content-add`, `content-remove`, `content-modify` and `transport-replace`, `transport-reject`, `transport-accept`. Whereas the former group of messages handles the application type the latter is responsible for the connection method. Either party may issue a `session-terminate` request to end the session. In the current example endpoint 2 has finished the conversation.

Please note, that each endpoint connects to a XMPP server. Assuming distant clients, two servers are engaged in forwarding the signaling information. However, this is not true for the media channels based on RTP or SOCKS sessions. Both are established point-to-point between clients. From a network routing perspective signaling and user data are conveyed across the IP network on distinct paths.

Remarks and Extensions

Up to now we have only touched on some features of the XMPP core and XMPP-Jingle. Compared to other protocols XMPP is much more modular, a fact also exhibited by the number of protocol extensions. At the time of writing, there have been more than 100 XEPs in place. Furthermore we have not covered the entire functionality of the XMPP core protocol. Knowing, that XMPP originated as an instant messaging protocol, this becomes immediately evident. Accordingly we will now indicate some of its features and extensions.

- Handling of presence information is central to XMPP. The corresponding data format is not restricted and so any type of information can

be considered to be presence information. This leads to the *rich presence* concept. In call centre environments an enriched set of presence information may include the status of IVR ports or call centre agents.

- XMPP allows for pending requests, i.e. more of them may be open at the same time.
- Large chunks of data are usually carried out-of-band using XMPP-Jingle. However, there also exists an inband data service similar to *user-to-user signaling (UUS)* in ISDN. And as for UUS the data chunks should be of reasonable size.
- Contact management is well developed. In addition to manual interaction, contacts may evolve automatically based on presence status. The corresponding data can be stored in hierarchical fashion in so called *rosters*. At the time of writing, the contact format followed the *vCard MIME directory profile* specified in RFC 2426.
- XMPP supports handling of data forms similar to the HTML forms. This enables a standardized interaction between clients and leads to generic workflow handling. Call centre environments would benefit from a single signaling channel for media signaling, screen pop handling and data transmission.
- Event management is supported by a classic publish-subscribe mechanism.
- Security is a central concept, e.g. TLS encryption is requested by the `<starttls>` XMPP tag. Authentication may be carried out using classic mechanisms such as DIGEST-MD5.
- Multiple parties are supported on a media independent basis. The corresponding extension is called *multi-user chat (MUC)*.

In the millenium XML based signaling becomes more and more important. This is also underlined by the fact, that the ITU considers developing a new signaling protocol based on XML. Once it will be known as H.325 group of standards. However, at the time of writing this initiative has just started and it is much too early to present any facts about it. It is hoped by the author, that some lessons learned from XMPP, SIP and H.323 will provide fruitful contributions to the project.

5.2.8 Notes on Performance Engineering

Signaling performance is an important topic and has a long history in queueing theory. When the first digital networks appeared, complex protocol architectures have been developed. Many of them are based on state machines, which are responsible to handle protocol state transitions in a consistent manner. These protocols lend themselves well to be modeled as queueing network or matrix exponential queueing system. More recently we do also find stateless protocol implementations. Either the protocol is simple in design or it is associated to a stateful entity. In the latter case one is well advised to perform an analysis of the stateful device, whereby propagation delay and thinking times of the stateless entity should be incorporated into the model. Simple protocols are best modeled as a classic queueing system with an appropriate queueing discipline.

In order to derive an appropriate performance model, one first has to identify the system under consideration. With respect to signaling systems one frequently encounters

- *signaling networks*, examples include ISDN and IP based networks
- *signaling components* such as telephone switch, PBX and communication platform
- *service architectures*, which rely on signaling

A good resource for all of these are the proceedings of the *international teletraffic congress (ITC)*. They contain a lot of material on ISDN switch modeling, ISDN network performance and service quality. More recently they also focused on IP based networks and their components. In general the analysis of IP based multimedia architectures has become more difficult than has been for telephone systems and networks. In considering signaling networks only, we leave behind the problems related to real time traffic. Concerned with a classic data network we may apply the well known results found in textbooks such as [124]. In most cases signaling messages fit into a single TCP or UDP packet eliminating the need to incorporate effects of segmentation into the model. If TCP is used to convey signaling information across a loaded IP network, congestion may occur and significantly affect the results of signaling performance. It may become necessary to extend the analysis to lower protocol layers. A valuable reference is the book [20] by R. Srikant, which discusses TCP and internet congestion in detail.

5.3 From Distributed to Virtual Call Centres

As the path from distributed call centres to virtual call centres is a stony one, developments are achieved only piecewise. Although the real virtual call centre as described in the beginning of this chapter does not exist today, several interesting scenarios have been developed. These include *post-routing* and *pre-routing*. The prefixes pre and post indicate, that an arriving call is routed before or after it has arrived at the switch. In both cases a central *call router* is responsible to allocate the calls to the call centre resources. Switch, ACD, IVR and other resources interface with a *telephony gateway* using a CTI link. The telephony gateway translates route requests and gathers resource statistics from switch, ACD and other resources. In post-routing configurations, the route request is issued by the ACD and forwarded by the telephony gateway to the call router. After resource statistics of the call centre have been retrieved, the call router returns a routing decision to the telephony gateway, which forwards it to the ACD for execution. Accordingly the call is connected to the appropriate resource by the switch. Such an architecture is shown in figure 5.22. Please note, that the scenario just drawn is only one out of many possible architectures. As no standards exist, the entire architecture except some telephony and CTI interfaces is completely vendor specific.

If a routing decision is to be performed, before the call has been connected to the switch, the call centre requires an interface to the public network. To be more specific it needs to communicate with the so called *service control point (SCP)*. This entity is part of the intelligent network and performs services such as number translation and routing. Please note, that the SCP does not perform any switching in the public network, it only provides the necessary information. To allow for external control through the call router, the SCP is attached to a *network gateway*. As before for the telephony gateway, the network gateway translates route requests and gathers statistics from the public network. The call flow is similar to the call flow in post-routing, except that route requests are now issued by the SCP instead of one of the ACDs.

As an illustration consider the architecture shown in figure 5.23. The corresponding signaling activities are as follows

1. As a call arrives, a route request is originated by the local switch to the SCP within the intelligent network.

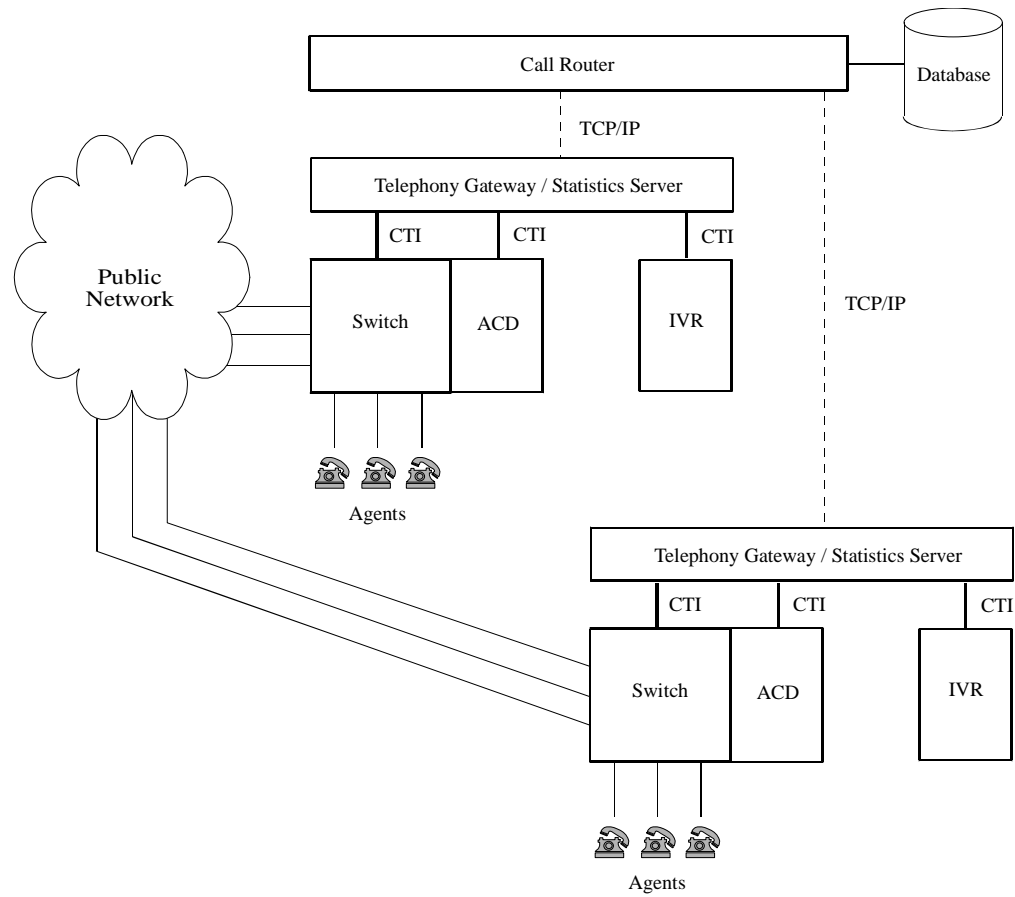


Figure 5.22: Call centre architecture with post-routing

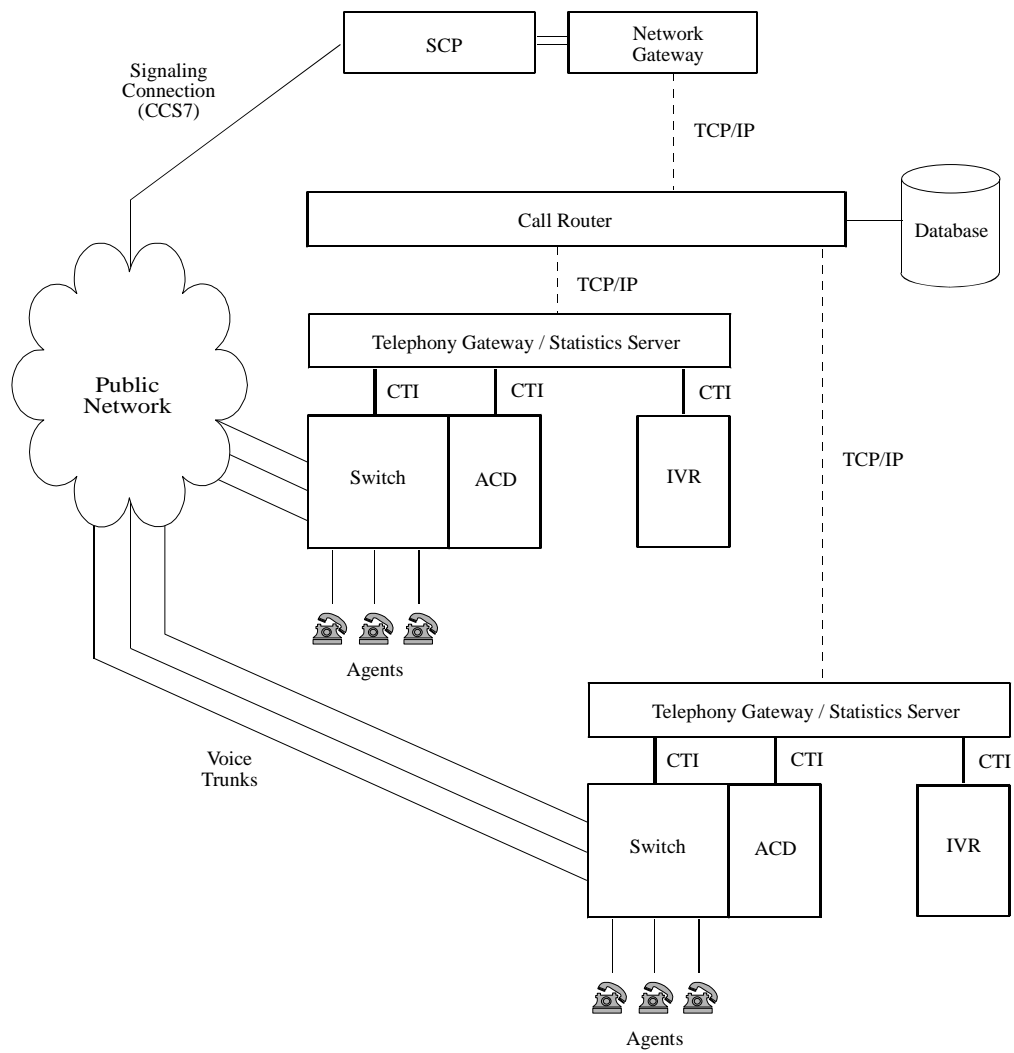


Figure 5.23: Call centre architecture with pre-routing

2. The route request is passed to the call router for processing. The network gateway translates the network description of the call including point of origin, dialed number or customer input into signaling data understood by the call router.
3. The call router selects the appropriate routing script, assesses the current real time status of the call centre and generates a routing decision based on internal and external data. Examples include the point of origination, the number dialed, the time of the day, call centre load conditions and resource statistics. The latter two have to be requested from the telephony gateways located within the call centre. The transactions necessary to carry out such a routing decisions may reach a certain grade of complexity.
4. The routing response is passed back to the public network to be executed. As a consequence the call is connected to the switch, where it is parked for queueing or redirected to the desired resource.

This has been a rather simple example with respect to the components involved. We now indicate some possible enhancements to the pre-routing architecture.

- Any desktop integration has been omitted up to now. Applications such as screen pop and outbound call management become possible by using the telephony gateway as a CTI server. This often leads to an integration of multimedia and internet services based on TCP/IP. Please note, that the same enhancements may also be applied to post-routing scenarios.
- If additional information such as account data or pin codes are required before the call is connected to one of the switches, a *network IVR* should be considered.
- As figure 5.23 indicates, post-routing and pre-routing may be combined in a single call centre.

With pre-routing, the traffic load on public and private network trunks significantly decreases. If post-routing is used, ISDN and QSig services such as *call transfer by rerouting* and *path replacement* will become an essential factor of stability and efficiency. Both procedures will be described in more

detail later in section 5.5. Without these services, a single call may block a significant number of trunk resources.

As both, pre-routing and post-routing architectures rely on CTI links, they have to receive special attention. Obviously, the efficiency of the call router depends on the set of events passed over the CTI links. Each CTI link can only carry a certain amount of messages and therefore the amount of simultaneous calls is limited. A short hand solution to bottleneck problems is to deploy additional CTI links, telephone switches, ACDs or IVRs in the call centre. If other channels such as web, mail and chat are handled over the same link, similar considerations apply. Furthermore they extend to multimedia communication platforms, mail handlers, web servers and the like. Potential solutions include the introduction of *load balancing* and *proxy servers*.

Pre-routing is also an issue for provider-connected communication platforms similar to the one described in section 2.3.4. However, a radically different terminology is used. If several media gateways are spread across the landscape, an optimal route needs to be determined prior to call establishment. Media and *signaling gateways* have to be coordinated by a softswitch platform to provide the necessary directives. This approach is indeed similar to what we have called pre-routing before. Unfortunately these components are not flexible enough to handle call centre specific routing decisions and so the concept is often implemented twice.

5.4 Signaling between Call Centres

5.4.1 Classic Telephony

As both architectures, distributed and virtual call centres, are attached to the public network using end-user protocols, e.g. the ISDN based *digital subscriber signaling system 1 (DSS1)* in Europe, these call centres are network users rather than being part of the public network. In order to establish a signaling path between call centre nodes, a separate signaling channel has to be maintained during the entire time of the call centre operation. With ISDN, a separate bearer channel may be used for such purposes. A more elegant solution is the use of *user-to-user signaling (UUS)*, a supplementary service of ISDN, specified by ITU-T in recommendation I.257.1. User-to-user signaling allows an ISDN user to send and receive a limited amount of

information (without modification or interpretation of its contents) to/from another ISDN user over the signaling channel. If associated to a circuit switched connection involving bearer channels, the service is regarded as a *supplementary service of the first case*. Otherwise the service is regarded as bearer service called *user signaling bearer service (USBS)* [19]. User data transfer capability is broken up into the following three user-to-user signaling supplementary services:

- Service 1 (UUS1): User information is transferred during the setup and the release phase of a call. Data are embedded with the call control messages, e.g. SETUP and DISCONNECT-RELEASE.
- Service 2 (UUS2): User information is transferred during the setup phase of the call independently of call control messages. From the sender's point of view, data are sent prior to the active phase of the call, i.e. between messages ALERT and CONNECT. UUS2 is only relevant in point-to-point arrangements and limited to a maximum of two messages in each direction.
- Service 3 (UUS3): User information is transferred during the active phase of a call. Any number of messages in each direction are allowed. Anyway, the rate of submission of these messages is limited and may be further restricted by the use of the CONGESTION CONTROL message.

User-to-user signaling may be specified different from country to country due to national regulations. Often affected are the type of charging and the length of user packets (e.g. 35 or 131 octets) [46]. Especially the design of multi-provider or multinational networks suffer from these hard limitations. The effective throughput of user data is on the average less than 1kbit/sec and message transfer delays are in the order of 1 second. Anyway, for most classic call centre applications, these values are sufficient.

As DSS1 specifies the user interface, messages have to be mapped to those used in common channel signaling networks by the local exchange. Today, Common Channel Signaling No. 7 (SS7) is used and the messages used to carry DSS1 signaling information are given by the *ISDN user part (ISUP)*. Data transferred using UUS1 are simply mapped from the SETUP and DISCONNECT-RELEASE messages to *initial address message (IAM)* and *release (REL)* by the local exchange and vice versa [177]. With UUS2 and UUS3 two different types of user data transfer are possible:

- Link-by-Link: User data are embedded in *user information messages (USR)* similar to DSS1 and carried over the network by the *message transfer part (MTP)*. These messages have to traverse the same path as circuit related signaling resulting in increased delays and increased loads on MTP signaling links due to inefficient routing.
- End-to-End: User data are transferred by using a sequenced connectionless service provided by the *signaling connection control part (SCCP)*. An association is kept between the ISUP controlled circuits and the SCCP connection.

The latter method is preferred, as UUS2 and UUS3 are supported in a more efficient manner. In case SCCP is not supported end-to-end, a fallback to the link-by-link method is performed [19].

Another option for transporting user data over the signaling channel of ISDN using packet oriented services is the use of *ISDN packet-mode bearer services (PMBS)*. These services are also specified for the use with bearer channels. When used with the signaling channel, PMBS allows point-to-point communication between users using X.25 encoding. According to CCITT Recommendation X.1 classes 8-10 speeds between 2400 bit/sec and 9600 bit/sec are supported for basic rate access. With primary rate access, class 11 allowing for a speed of 48000 bit/sec might be used in addition to class 8-10 [47]. X.25 signaling information is transmitted together with user data on the bearer channel, after a circuit has been established using the standard ISDN call setup procedure. Therefore X.25 signaling data are not visible to the ISDN signaling system. Another approach has been adopted with ISDN frame mode bearer service. User data may be transmitted over both, the bearer and the signaling channels according to ITU-T recommendation Q.922 specifying *link access protocol F (LAPF)* for frame based services. Exchange of signaling data over the signaling channel of ISDN is based on ITU-T recommendations Q.933 and Q.921 (LAPD) [17].

As stated above, 1 kbit/sec throughput may be sufficient for most classic call centre applications. Then ISDN user-to-user signaling might be the system of choice. Call selection criteria and universal caller identifications helping a call tracking or call management system to associate an arriving call to their internal statistics may be sent embedded in ISDN layer 3 SETUP and DISCONNECT messages to the target switch. This applies for both, public and private networks. In distributed call centres routing decisions have to be

made before setting up a new call or transferring an existing call. Then each call centre site may provide dummy extensions for pure information transfer. Consider an example with two interconnected call centre sites based on a distributed architecture. A call is waiting at site A to be serviced by an agent with certain skills. Site A estimates the expected wait time and determines, that thresholds given through strategy rules could not be met. As another expert resides at site B, the expected wait time for that agent is requested from site B. As spare resources have to be available for call bursts at site B, a penalty is applied to the expected wait time before sent back to site A. Having received the answer, site A deduces, that still the expected wait time at site B is lower, it initiates a call transfer to the desired expert at site B. With such mechanisms, skill allocation becomes a dynamic process. Great care has to be taken, when adjusting penalties for certain call selection criteria.

Even though interoperability has been established between private branch exchanges using standards based signaling systems as given for ISDN and QSig, there are no agreements for call centre purposes. Information is exchanged using a simple request/response scheme on a vendor specific proprietary basis. Such information might include

- a *universal identification* associated to a specific call to support call management systems in tracking the call
- an *identification for the application*, the call is related to
- *call path related information* such as digits entered by the caller in accordance with an announcement
- *call statistics* such as current wait time and the position in queue
- *resource statistics* such as the expected wait time or the number of calls waiting in a given split or skill, or the number of available agents

5.4.2 Multimedia Call Centres

The treatment of multimedia streams depends mainly on the system architecture of the call centre. Based on the evolution of these systems, two types may be distinguished. Call centres based on traditional voice technology often introduce the concept of *phantom calls*. Phantom calls are associated to

non-voice media such as fax, mail and web and act as placeholders, while the call progresses through the system. Phantom calls attain queue positions, suffer from wait time and are finally assigned to the desired resource similar to a voice call. If the final target is found, the ACD informs the related adjunct system to connect the non-voice call to the determined destination. An incoming email would therefore be held by a mail server, until a delivery request for a specific agent has been issued by the ACD. Definitions for resources such as split groups, skills and agents are also extended to support the new media types. While emails are handled in the back office, front office agents are engaged with voice calls. It is also possible, that both applications are supported concurrently by means of media blending. As an enhancement to traditional voice technology the concept of phantom calls is often found in call centre systems with integrated ACD.

If an adjunct ACD is used, these systems are capable of handling various media types natively without the use of phantom calls. If voice calls are to be switched, the ACD requests the switch to perform the necessary switching, while mail, fax and web applications are routed over the local area network using protocols such as TCP/IP or IPX/SPX. In order to support fax over network, each fax has to be converted to an email message or the like. Put to an extreme, also voice streams may be packetized and transported over the local area network by use of protocols related to SIP, H.323 or similar. In that case, call switching is replaced by call related signaling activities and routing inherent to the protocols used.

In any case, a signaling path has to be established between ACD sites over the local area network. This allows for load balancing and exchange of statistics. The corresponding procedures are either vendor specific or based on CTI protocols conforming to existing standards. In the latter case it is very likely, that these CTI protocols have to be enhanced by additional information elements for inter-ACD communication. In order to preserve interoperability between ACDs located in different types of networks (e.g. H.323 based networks and ISDN), a *gateway* has to convert the necessary information (e.g. user-to-user information elements) in addition to media data. Such an approach requires a tight integration between gateway and ACD. Although H.323 supports various types of information transfer such as user-to-user signaling through H.225 or APP messages carried by RTCP, none of them is applicable for the communication between gateway and ACD. The reason for this unapplicability stems from the fact, that both mechanisms still depend on media transport. Instead a separate data channel is required

for application signaling purposes. Unfortunately protocol suites related to H.323 and SIP are not flexible enough to provide means for such a communication. This is different for XMPP, which allows an ACD related signaling path to be established either inband or out-of-band. The former is especially useful, when the number of open TCP sessions should be kept to a minimum.

5.5 Call Centre to Network Signaling

5.5.1 Classic Telephony

In order to provide load balancing for distributed call centres, interaction with a public or private network is required. This can be achieved by the use of supplementary protocols as have been specified for ISDN and QSig. After having identified the resource statistics for agents, splits and skills, active calls have to be transferred or rerouted to the destination. A call centre connected to the public network by means of an ISDN interface may use the *call transfer supplementary service* to perform that task. This is different for private networks. Invoking subsequent transfers for the same call by interconnecting input and output lines soon leads to a waste of trunk resources. Instead the call path should be optimized after each transfer. One possibility is to use the *call transfer by rerouting supplementary service* as specified in ITU Standard ISO 13869. If not applicable, a combination of call transfer by join (ITU-T Standard ISO 13869) and path replacement (ITU-T Standard ISO 13874) provides the same result. A sample scenario is shown in figure 5.24. No error conditions are assumed, i.e. all call transfer and path replacement attempts succeed. With three switching nodes involved, the call is directed to an inactive destination. Supplementary services information elements are embedded in the ISDN layer 3 messages FACILITY, SETUP, CONNECT, ALERT, DISCONNECT and RELEASE COMPLETE. The message flow depicted in figure 5.24 is now described in more detail:

1. User A invokes the call transfer supplementary service on his telephone by entering a feature code or simply pressing a feature button. A CTIN-VOKE REQUEST message is sent to exchange A. According to QSig specifications such an exchange is called a *private integrated service network exchange (PINX)*.
2. PINX A informs PINX B and PINX C about the transfer request.

3. User B and user C are notified by their home switches about the transfer by CTNOTIFY INDICATION message.
4. Both, PINX B and PINX C issue an CTUPDATE INVOKE message delivering network information including connected number, call history and alerting indication to PINX A.
5. PINX A passes messages from PINX B to PINX C and vice versa. This information will be delivered later to the terminals by a CTNOTIFY INDICATION message.
6. Inbetween, user B initiates a SUBADRTFR REQUEST message to PINX B carrying information not stored by the network, i.e. connected subaddress.
7. The connected subaddress is delivered with the SUBADRTFR INVOKE message to PINX A to be routed to user C.
8. Now, CTNOTIFY INDICATION messages carrying information known to the network arrive at the terminals of user B and user C. This may result in a state change of the terminals involved, e.g. display updates.
9. PINX A forwards the SUBADRTFR INVOKE message to PINX C.
10. The terminal of user C is informed of user B's connected subaddress by a SUBADRTFR INDICATION message.
11. A setup response message is sent to PINX C from user C as an answer to the altering condition of his terminal.
12. PINX C issues an ISDN layer 3 CONNECT message to PINX A.
13. PINX B is informed, that an answer has taken place following an alerting transfer by the use of a CTACTIVE INVOKE message originated at PINX A.
14. A CONNECT ACKNOWLEDGE message is sent back to PINX C responding to the CONNECT request. At the same time, user B is informed about the current status by a CTNOTIFY INDICATION delivering information extracted from the CTACTIVE INVOKE information element. The call transfer is completed as can be seen from the bearer channel diagram in figure 5.24.

15. In order to optimize bearer channels, a path replacement is initiated by the use of a PATH REPLACEMENT PROPOSITION INVOKE message sent from PINX B to PINX A.
16. The PATH REPLACEMENT PROPOSITION INVOKE message is forwarded to PINX C.
17. This results in reserving a direct line by an ISDN layer 3 SETUP message carrying a PATH REPLACEMENT SETUP INVOKE information element sent back from PINX C to PINX A.
18. As an answer, CALL PROCEEDING is replied.
19. A CONNECT request carrying the PATH REPLACEMENT SETUP RETURN RESULT is issued by PINX A to PINX C.
20. With the connection attempt being successful, a CONNECT ACKNOWLEDGE message is returned to PINX A.
21. Now a new connection has been established and the call is switched by both ends to the new connection making the old connection obsolete. Therefore a DISCONNECT message is issued to PINX A by PINX C.
22. PINX B also receives a DISCONNECT message in order to release the obsolete circuit.
23. A RELEASE message is returned to PINX C by PINX A as an answer to the DISCONNECT request.
24. A RELEASE message is returned to PINX A by PINX B as an answer to the DISCONNECT request.
25. A RELEASE COMPLETE message is replied to PINX A as an acknowledge for a successful release of the circuit between PINX A and PINX C.
26. A RELEASE COMPLETE message is replied to PINX A as an acknowledge for a successful release of the circuit between PINX A and PINX C. Now the call has been rerouted, which is shown in the bearer channel diagram of figure 5.24.

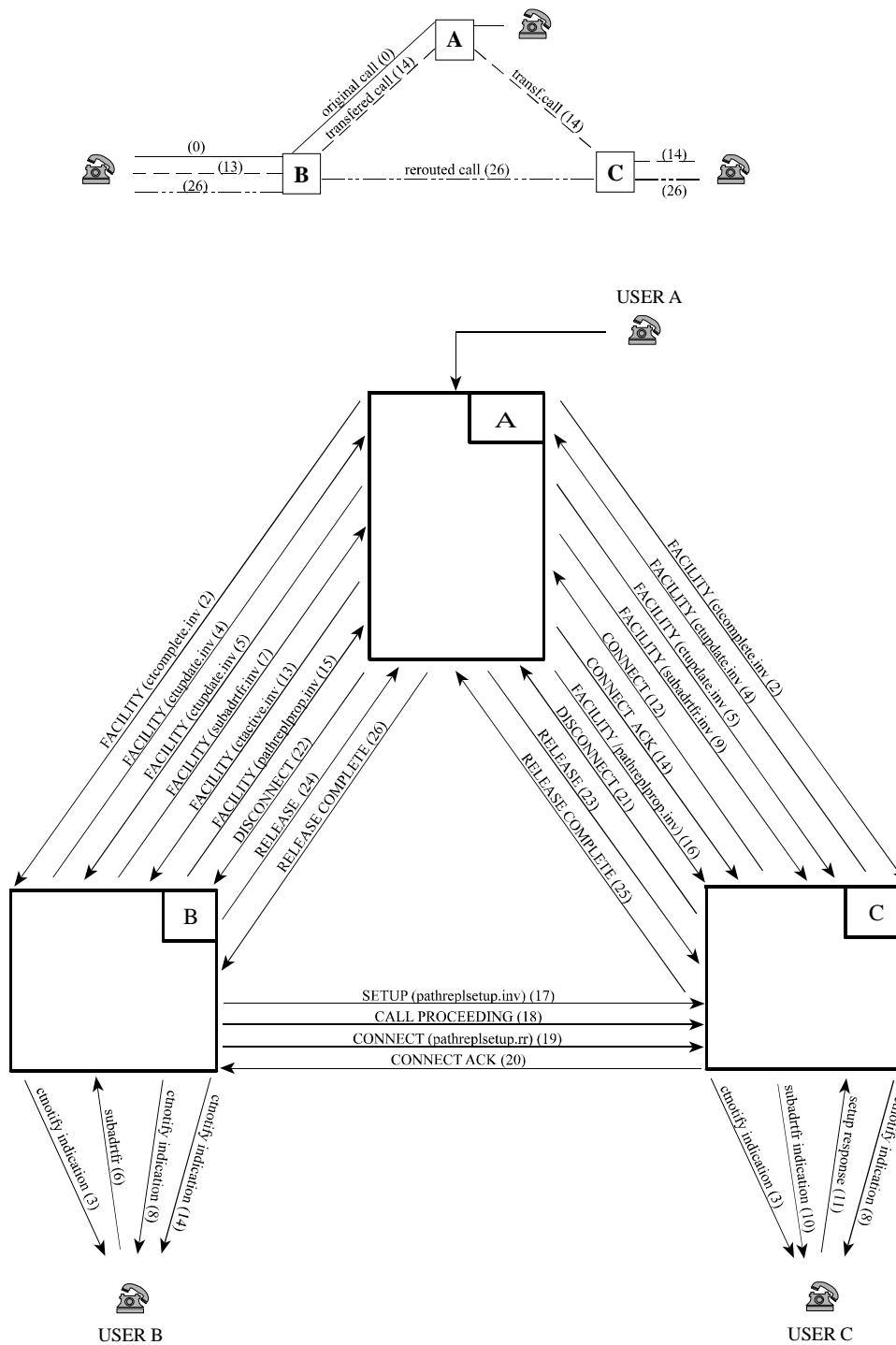


Figure 5.24: Sample scenario for use of supplementary services in private networks

Reviewing the specification for call transfer by rerouting raises the question, why to use a combination of two supplementary services instead of a more efficient service defined for such purposes. The answer becomes evident, when considering the call centre as a whole. In order to provide exact statistics about calls, each single call has to be tracked from its arrival at the call center to its termination. Many call tracking and call management systems are call centre adjuncts and so they only register certain events from the telephone switch. Furthermore in a distributed call centre configuration these systems do have access only to the information supplied by their associated switches and ACDs. In other words, call tracking and call management systems never see the whole picture. For example it may become extremely difficult to distinguish between normal and transferred call both with calling party number suppressed, when only scarce information is provided by the switch or ACD. It becomes essential to reconstruct the network state in sufficient detail based on the information delivered by distant locations. Sometimes this is not possible to the extent required, so the network is forced to apply a step by step scheme for some services. In our example there is only a single fact unknown at the same time, as call path and destination do not change simultaneously. For the call transfer by rerouting supplementary service this is not the case, i.e. an adjunct system may not be able to track the call anymore. One solution to this problem is to associate a *universal call identification* with the call by means of user-to-user signaling, which has been described above in 5.4.1.

Similar problems for call tracking and call management systems arise with the use of other supplementary services such as *call forward* and *conference call*. On the other hand, ISDN supplementary services like *closed user group*, *private numbering plan* and the *multilevel precedence and preemption service* have the potential to offload some tasks performed by a switch, PBX or integrated ACD to the public network.

5.5.2 Multimedia Call Centres

For multimedia call centres, network signaling depends on the type of media conveyed across the network. Separate transport networks lead to several user-to-network interfaces, each with different design and protocol. One example might be the distribution of facsimiles between agents located in several sites. Either the fax is routed through the voice network and delivered to a dedicated agent fax machine or it is centrally processed by a fax

server. In the former case the signaling procedures described in the preceding section apply, while for the latter each fax is carried over the data network using protocols such as TCP/IP. As these networks define procedures for identification, address resolution and routing, no further definition of signaling procedures is necessary. Voice is still delivered using classic telephony lines, while email, ftp and web services are provided over the data network. Clearly, the separated approach is less flexible and less efficient in terms of resource usage, but it takes advantage of the characteristics of each transport network. Although the price-performance ratio may be bad for low to medium sized call centres, it often gets better for large architectures. This effect mainly results from the fact, that different scalings may be applied to components such as switches and servers. Furthermore, single components allow for selective scalability. So an increase in voice traffic might lead to a switch upgrade or demand the introduction of an expensive transport network such as ATM to carry the packetized voice streams. The latter also exhibits, that certain cost factors may be well hidden in the architecture of distributed communication platforms. So the economic situation does not only depend on the number of components involved, it also depends on their level of interaction and cohesion.

Following an integrated approach means to carry all media streams over a single transport network. Much effort has been undertaken to enhance the IP protocol suite to provide such services. Considering the IP layer only, load balancing leads to a form of central server operation. Although servers may be interconnected or clustered, they appear as a single entity to their clients. If more layers are involved, the server cluster has to be synchronized on an application level. Otherwise it may loose track of session and conversation information. For call centres based on distributed communication platforms and interfacing to several networks the gateway plays a major role. As an example consider the call transfer supplementary service defined in H.450.2. Here the gateway acts as a transferred endpoint or transferred-to endpoint for H.323 based networks. Here we could repeat our considerations from above for the ISDN. But the gateway may also connect to a public or private telephone network. In either case the distributed communication platform has to provide means for tracking calls and network related activities. More recently we also find providers offering voice over IP services based on SIP. Unfortunately one can rely on basic call handling only, i.e. the supplementary services architecture is almost unuseable for call centre purposes. Complex transactions have to be implemented on a manual base by joining input and

output sessions. The determination and enforcement of optimal routes is now under complete responsibility of the call centre.

The integrated approach leads to a shift from hardware to software, which results in higher flexibility for the range of small to medium sized call centres. However, large call centre installations still demand a significant amount of hardware to handle the requirements of performance and reliability. In this respect technologies such as IP over ATM transport, *multiprotocol label switching (MPLS)* and B-ISDNs are worth considering.

Chapter 6

Analytical Call Centre Modeling

This chapter focuses on analytical techniques suitable for resource optimization and call centre performance engineering. Having introduced the philosophy of call centre modeling we will show how to apply the methods of queueing theory in this specific domain. Although analytical models provide less transparency, they can be solved at low cost. Furthermore the use of pseudo equivalence and transform techniques is encouraged leading to some impressive results. On the other side, simulation is often more transparent and the behaviour of the system under consideration is projected into the computing domain at a high level of detail. It is also possible to set up hybrid models and gain the advantages of each approach. However, for the current chapter we will stick to the way of mathematical model building as the method of choice. Furthermore we will restrict ourselves to the discussion of inbound traffic in a call surplus situation. Outbound management is more related to the optimization of schedules based on marketing data rather than queueing theory. In fact, the operator exercises complete control with respect to the calls processed. This is different for inbound traffic, because calls approaching to the call centre are subject to statistical fluctuations, which perfectly fit into a queueing theory framework.

A combination of both strategies is suggested for the treatment of call and media blending. These can be associated with priority models, which prefer inbound traffic. We know from L. Kleinrock and others [123], that high priority customers are unaffected by calls or sessions of lower priority. The outbound traffic is then analyzed using classical methods in a dynamic

environment. In other words, idle resources are scheduled to perform tasks under control with respect to their time of occurrence. Accordingly, any results valid for inbound traffic remain useful for call and media blending scenarios as well.

By capturing all important aspects we attempt to provide an abstract but complete view of a call centre. Therefore, we will also include the description of some methods related to the analysis of call flows and call distribution. Especially the latter has been a topic of active research in the last ten years. Whenever no details are provided, we will refer to the corresponding papers and books.

6.1 Call Centre Perspective

Setting up and analyzing a model is often performed with respect to a certain target. The same applies to resource modeling. One may be concerned with the determination of an optimal number of agents for a specific time of day or the choice of an IVR system with suitable capacity. To cope with these and similar demands, we have to attain a view specific to the type of problem. Fortunately there are not much of them and so we may identify on a general level of detail the following perspectives:

- call view
- agent view
- system view
- network view

The *system perspective* is the one we usually attain, when dealing with questions of capacity or system architecture. Analyzing a single component, the history of a call is barely of interest. As an example, consider a capacity assessment for an IVR system. After measuring the number of call attempts for a certain period of time, the driving parameters are identified and an appropriate model is chosen. There is no need to distinguish between calls originating from outside or the ACD (redirected calls or overflows). In some cases, one is more interested in a specific part of the system. Sticking to the IVR example, this might be a specific service or a dedicated technical resource. In fact, this is only a matter of definition.

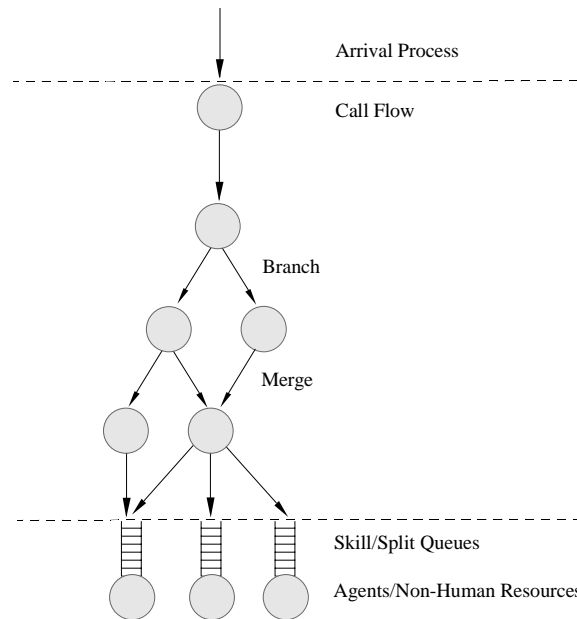


Figure 6.1: Typical call processing scenario

If one is more interested in the interaction of system components, he has to attain the *call view*. One chooses a typical call scenario and records the components passed by a tagged call when traversing through the network. In other words, one assumes the view of the customer and the effects he is exposed to. Due to their setup, a call scenario commonly results in a tandem configuration similar to a conveyor belt. By considering all representative call scenarios and combining them to a single perspective one arrives at the *network perspective*. A typical call processing scenario is shown in figure 6.1. Accordingly, queueing network analysis has been considered a natural choice. Necessarily all resources are specified as Markovian nodes in a queueing network. For some applications the resulting degree of approximation is considered insufficient and so the affected resources have to be analyzed separately. This immediately leads to the *agent perspective*. Allowing for the most flexible instruments of queueing theory, such a view has become ideal for problems of resource optimization.

Attaining a single view might not be sufficient for some problems. Whereas call, system and network perspective are highly relevant in technical engi-

neering, the resource view is very interesting from a business point of view. As such, the corresponding methods may be applied on a regular basis to predict waiting times, work volumes and similar parameters of interest for workforce management.

6.2 Stochastic Traffic Assessment

The purpose of this section is to bridge the gap between the real world scenario under consideration and the corresponding mathematical model. After describing some typical call centre data sources and approaches to measurement, we will discuss how to transform the measured data into a stochastic description.

6.2.1 Data Sources and Measurement

According to G. Koole and A. Mandelbaum [129], we may distinguish the following types of call centre data:

- *Operational data* are collected by the ACD and the underlying communications platform. Almost all key performance indicators discussed in section 1.6 belong to this class. From a technical perspective call detail and agent data records are collected and aggregated by the *management information system (MIS)*, which is commonly implemented as an adjunct system to the ACD. The aggregated data are usually presented in the form of a call centre *report*, either of real time or historical nature. The former is well suited to support the call centre supervisor, whereas the latter is more appropriate for long term management decisions.
- *Marketing data* are associated with the CTI or the *customer relationship management (CRM)* platform. The majority of data, which is delivered to the call centre agent in form of a screen popup belongs to this class. Examples include customer data, bank accounts and purchased items. However, they all depend heavily on the application.
- *Psychological data* express the perceptions of customers and call centre employees. They are deduced from customer surveys and employee assessments. Psychological data can be seen as a secondary source of

BCMS TRUNK GROUP REPORT											
Switch Name: Switch				Date: 12:27 pm MON DEC 20, 1999							
Group: 2											
Group Name: PSTN				Number of Trunks: 30							
TIME	CALLS	ABAND	TIME	CCS	CALLS	COMP	TIME	CCS	%ALL	%TIME	
5:00- 5:30	0	0	0:00	0.00	2	2	15:29	18.57	0	0	
5:30- 6:00	0	0	0:00	0.00	1	0	4:31	2.71	0	0	
6:00- 6:30	1	0	0:46	0.46	0	1	0:00	0.00	0	0	
6:30- 7:00	3	1	0:45	1.36	7	7	2:43	11.39	0	0	
7:00- 7:30	3	1	0:14	0.42	15	11	2:07	19.09	0	0	
7:30- 8:00	18	0	0:21	3.86	31	15	0:37	11.48	0	0	
8:00- 8:30	34	0	0:47	16.06	74	46	1:15	55.14	0	0	
8:30- 9:00	72	0	0:52	37.71	150	81	0:57	85.55	1	0	
9:00- 9:30	75	1	0:34	25.66	169	95	1:13	123.02	0	0	
9:30-10:00	67	3	0:36	24.22	208	130	1:21	168.05	0	0	

Figure 6.2: A typical trunk group report

information in the modeling process. As such, they provide an indication on how far the selected model deviates from reality. Consequently one may adapt the model based on operational data according to the drift suggested by psychological data.

Being concerned with statistical models, we are in need of quantitative data. As marketing data are more of qualitative nature we are left with operational data. As mentioned above, these data are provided in form of reports by the management information system. As an example consider the trunk group report shown in figure 6.2. We may use it to carry out a capacity analysis based on the Erlang loss formula. Trunks are usually found in circuit switched platforms and belong to the class of technical resources. However, they are also important for call centre operation. Imagine a call centre with insufficient trunk resources. Arriving customers are blocked and prohibited from entering the system. Accordingly, they cannot contribute to the waiting time of a certain split or skill group. Consequently, the average waiting time decreases and the service level increases. This effect is inferior to the targets of call centre management and underlines the importance of trunks and related abandoned call statistics.

Having identified types and sources of data, one could ask for appropriate sampling techniques in call centre environments. Although there is no recommendation for call centres, there are some for classic telephony. According to ITU Recommendation E.500 [105], traffic statistics should be measured for the significant period of each day of the whole year. For trunk groups

they suggest

- the mean of the 30 highest days during a 12 month period to measure normal traffic load
- the mean of the 5 highest days during a 12 month period to measure a high traffic load

Obviously the same rules may be applied to other resources as well. By balancing economical considerations and effects of blocking, this average of high volume periods achieves a low probability of loss. However, this was not enough and so the concept of *busy hour measurements* has been introduced. The *busy hour* is simply the busiest hour during the day in terms of traffic load. By averaging busy hours, one arrives at the *average bouncing busy hour* as introduced by AT&T. The term bouncing emphasizes the fact, that each busy hour may occur at a different time. Considering our trunk group example shown in figure 6.2, we have to locate the pair of adjacent half hours featuring the highest load. Recall from section 3.1.7, that *centum call seconds (CCS)* are related to Erlangs by $\rho_{ccs} = 36\rho_{erl}$. By visual inspection the busy hour may be found to range from 08:30 to 09:30 and 09:00 to 10:00 for incoming and outgoing traffic. Because of its importance, the busy hour concept has also been included in later revisions of ITU Recommendation E.500 [105]. The ITU defines the *time consistent busy hour* as four consecutive quarter hours during an average day, which maximizes the sum of the corresponding observed values. The higher resolution allows for an improved identification of traffic peaks and leads to more conservative estimates for the blocking probability.

All the concepts introduced so far may be and have been applied to call centre installations. The concept related to the average bouncing busy hour has become very attractive, as it suggests immediate access to reliable performance indicators. However, this is not true for the startup phase. Although it has been expressed explicitly only for one case, measurements have to be gathered continuously. Otherwise no justification on the spread of data can be made.

6.2.2 Traffic Processes

In probability theory the notion of traffic is usually expressed in terms of stochastic processes. Such a description has been commonly applied to ar-

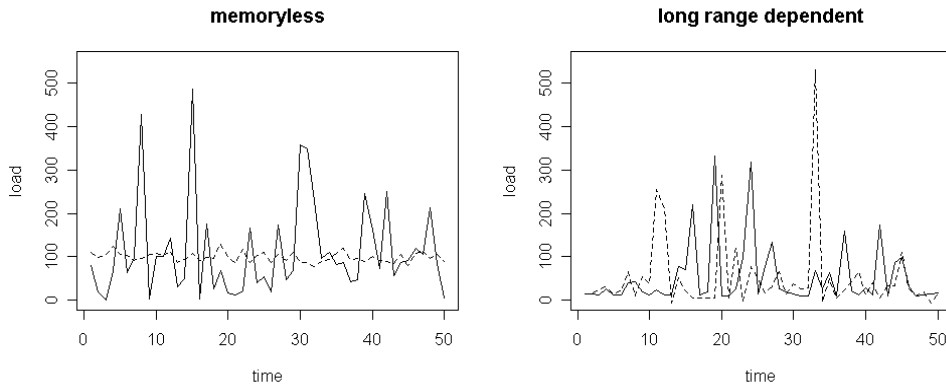


Figure 6.3: Behaviour of traffic

rivals, departures, abandoned calls, retrials, serviced and queued customers. Any set of resource, call detail or agent data records collected by the ACD constitutes a realization of the corresponding stochastic process. As the type of each stochastic process is determined through the chosen model, the task of finding an appropriate representation is reduced to an estimation of parameters. This becomes even simpler in queueing theory. Very often renewal processes are assumed for external and predetermined traffic processes describing arrivals, abandoned calls, retrials, services and the like. In further asserting homogeneity for the renewal distribution, we are only left with estimating the corresponding parameters. However, this does not tell us how to select the model according to traffic patterns. Fortunately we can make use of some results of telephony and data networking. Being aware, that many queueing models use moments as input parameters, we need to check, if they admit a useful representation. If not, the *bursty* behaviour shown over a wide range of time scales might corrupt the averaging process. Now consider the traffic patterns shown in figure 6.3. In both cases, 5000 data records have been collected. The solid line represents the first 50 entries of both data sets, whereas the dashed line shows the 50 average values taken in chunks of 100 entries each. For the memoryless process on the left we can see, that the averaging process yields a reduction in variation. But this is not true for the right side. In fact, this *self similar process* exhibits a similar burstiness on different scales. So far we are concerned with an aggregation of

memoryless and long range dependent sources, respectively. The latter are often found in IP networks and the internet [163]. As an example, consider a workstation downloading the latest Madonna video. Due to the length of the download session, there is a certain correlation between a packet sent now and another one transmitted 10 minutes later. In estimating the density function, we would encounter heavy tails. These *heavy tailed distributions* possess polynomial rather than exponential decay rates. One example is the *Pareto distribution*.

One might argue, that almost all well known heavy tailed distributions belong to the class of absolutely continuous distributions and thus are well represented by a suitable phase type distribution. We know from section 3.1.5, that this is indeed correct, but very often the resulting approximation features a large number of phases [72]. This yields numerically expensive models and therefore justifies an alternative approach based on heavy tailed distributions.

The theory of self similar processes led to some new results in teletraffic engineering [185]. As a matter of fact, we might still apply traditional models to packet networks carrying *constant bit rate traffic* such as H.261 video, G.711 and PCM voice streams. In using codecs featuring compression and *voice activity detection (VAD)*, we encounter talk spurts following a heavy tailed distribution. By combining an infinite number of these *ON/OFF sources*, the aggregated traffic stream becomes long range dependent. Empirical studies have shown such effects on the packet level for media streams conveyed across IP networks after being encoded according to MPEG, G.723.1, G.729A or GSM 06.10 [185].

However, in classic telephony the effects of long range dependence are only rarely encountered, so that traditional approaches to teletraffic modeling still apply. As such, we can also apply some of these results to call centre environment. For IP based communication platforms we can neglect the effect of long range dependence on the call level provided the underlying network is sufficiently sized and not shared with other applications. At the packet level we can expect reasonable results only for constant bit rate traffic. This leads to the classic definition of telecommunication traffic, which is stated in terms of the *variance to mean ratio* or the coefficient of variation c . Sticking to the latter we classify traffic as *peaked*, *purely random* or *smooth* according to $c^2 > 1$, $c^2 = 1$ and $c^2 < 1$. Also refer to figure 6.4 for a typical representation of each traffic pattern. Purely random traffic is directly related to the Poisson process and is usually encountered for arrivals. Peaked or

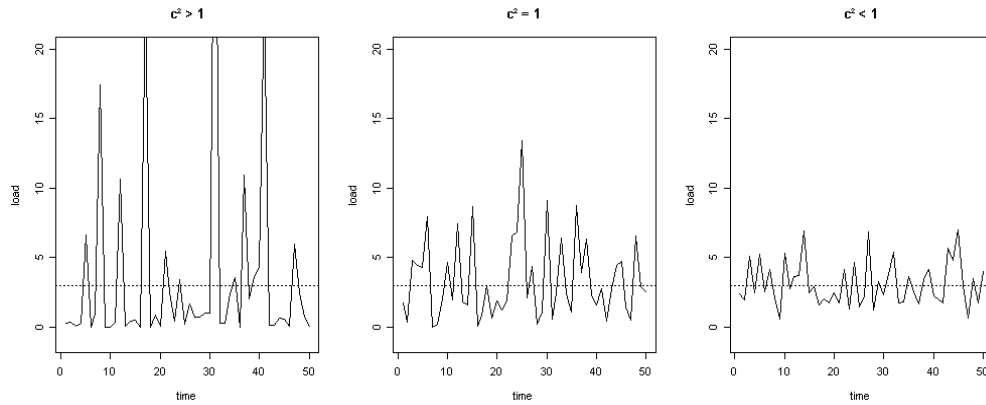


Figure 6.4: Types of traffic

rough traffic is typical for overflows from telephone trunks or split groups. It is best represented by the class of hyperexponential or Cox distributions. As opposed to purely random traffic there has to be some compensation for the increased variation. This led to the development of *equivalent random theory* in teletraffic engineering [79]. Smooth traffic is commonly encountered for a limited number of sources and best represented by the Erlang distribution. These considerations render the family of phase type distribution an ideal instrument for the purpose of describing various traffic types. In fact, we can apply the methods of chapter 4 to estimate the parameters of the interevent distribution and conclude about the corresponding renewal process.

So far we have classified patterns of traffic and suggested appropriate representations in terms of stochastic processes on a rather general level. To be more specific, we need to consider the application as well.

Arrivals

When discussing the arrival process to a call centre, one usually thinks of a representation for the traffic originated by calling customers in the first run. Adopting some nomenclature from classic telephony, the corresponding intensity is referred to as *offered load*. We have to be aware, that the arrival process is different to the input process, as calls might be rejected due to capacity limitations or retrials may occur. Ignoring the presence of repeated call attempts for the moment, we may describe the arrival process

as an aggregation of the input and the abandoned call process. The intensity related to the former is called the *carried load*. By cutting off the peaks, the carried traffic delivered to the system as input process is smoothed. We may introduce retrials by delaying a certain percentage of blocked calls before merging them with the arrival stream. If we decide to offer the overflow traffic to another group of resources, trunks or agents as offered load, we have to consider its increased peakedness.

It is suggested by example 5, that the Poisson process is an adequate description of the arrival process describing the traffic offered to a call centre. Empirical assessments have shown this to be true for voice traffic encountered in larger installations. This raises the opportunity to use queueing models of the $M/M/c$, $M/G/c$ or $M/ME/c$ type. Provided the traffic does not exhibit the memoryless property, we have to switch to more advanced models. In a phase type context we would describe the system as a quasi birth death process. However, for peaked traffic an application of the *equivalent random method* provides an alternative. The idea is similar to the one, which led to the Allen-Cunneen formula for $G/G/c$ systems. More specific, one attempts to compensate the increased variation by asserting a higher intensity. Such a reasoning obviously depends on the choice of the model and therefore we will delay its discussion to section 6.5.1.

So far we have assumed stationary models and a stable environment during busy hours. If these assumptions do not hold, the choice of the homogeneous Poisson process to describe a purely random arrival pattern is no longer justified. However, not everything is lost and we still can make use of some memoryless results by considering an inhomogeneous Poisson process instead. One such approach has been carried out by Jongbloed and Koole in [110]. Assuming a random arrival intensity they allow for a traffic description in terms of Poisson mixtures. For an estimation of the mixing distribution an application of the maximum likelihood method is suggested.

Service Process

In describing the service time, we have to do so from a specific viewpoint. In the past, agent talk times have often been identified with service times. But this means nothing else than to attain a technical perspective. To be more specific, this is either the call, the systems or the network view. From an agent perspective the service might not have been completed upon hanging up. Instead he might have to perform some after call work. Being related

BCMS SPLIT SUMMARY REPORT											
Switch Name: Switch Time: 12:30-13:00						Date: 1:29 pm MON OCT 2, 2000					
SPLIT NAME	ACD CALLS	AVG SPEED ANS	ABAND CALLS	AVG ABAND TIME	AVG TALK TIME	TOTAL AFTER CALL	FLOW IN	FLOW OUT	TOTAL AUX/ OTHER	AVG STAFF	% IN SERV LEVL
Tec Hotline	14	2:03	10	1:19	3:40	5:19	0	5	94:54	7.0	0
Bus Hotline	16	2:09	7	1:34	3:52	4:42	0	6	94:54	7.0	10
SUMMARY	30	2:06	17	1:25	3:46	10:01	0	11	189:48	7.0	5

Figure 6.5: A typical split group report

to the call, the corresponding wrap up or after call work time definitely contributes to the service time. Furthermore one might accept the auxiliary times as part of the service time for modeling purposes. Such a reasoning leads to a phase type description of the service time. In the simplest case, we have to estimate the rates of exponential service nodes related to talking, after call work and auxiliary states. As suggested by the maximum likelihood estimator for the exponential distribution derived in example 32, we only have to assume the reciprocal of the corresponding mean times.

As an example consider the report shown in figure 6.5. Describing the service time by a two state phase type distribution with talking and after call work states, we would readily compute the corresponding rates as follows

$$\mu_{talk} = \frac{1}{\text{AVG TALK TIME}}, \quad \mu_{acw} = \frac{\text{ACD CALLS}}{\text{TOTAL AFTER CALL}}$$

However, such a report might not be sufficient when assuming distributions different from the exponential one. This is due to the fact, that most common reports lack values for empirical moments other than the mean. If the management information system allows for the creation of custom reports, one might include the desired statistics right away. Otherwise they have to be calculated manually from call detail or agent data records for the busy hour period.

Several authors recommended the *logarithmic normal distribution* as an appropriate description of the talking time \check{T} [143][53]. This is nothing else than assuming $\ln \check{T}$ to follow a normal distribution with mean μ and variance

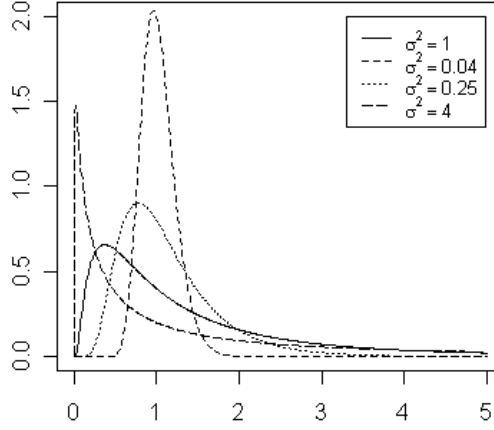


Figure 6.6: Logarithmic normal density function

σ^2 . This leads to the density function [108]

$$f(t) = \frac{1}{\sigma t \sqrt{2\pi}} \exp \left\{ -\frac{\ln t - \mu}{2\sigma^2} \right\}$$

and to the n -th moment

$$\mathbb{E}T^n = \exp \left\{ n\mu + \frac{\sigma^2 n^2}{2} \right\}$$

As shown in figure 6.6, the ascent of the logarithmic normal density is adjusted by modifying the value of σ^2 . This second parameter allows for more flexibility in matching a given data set. Assuming observations t_1, t_2, \dots, t_n of size n , the maximum likelihood estimates for μ and σ^2 are given by

$$\hat{\mu} = e^M, \quad \hat{\sigma}^2 = \frac{\sum_{i=1}^n (\ln t_i - M)^2}{n}$$

where

$$M = \frac{\sum_{i=1}^n \ln t_i}{n}$$

Obviously the lognormal distribution cannot be embedded in the phase type concept suggested above. However, E. Ishay has shown [103], that the lognormal distribution is already well represented by a phase type distribution of order 3. Assuming a specific phase configuration $PH(\beta, \mathbf{B})$, one may equate the moments, i.e.

$$\begin{aligned}\hat{\mu} &= \ln(-\beta \mathbf{B}^{-1} \mathbf{1}) - \frac{\hat{\sigma}^2}{2} = \ln \Psi[-\mathbf{B}^{-1}] - \frac{\hat{\sigma}^2}{2} \\ \hat{\sigma}^2 &= \ln \left(\frac{2\beta \mathbf{B}^{-2} \mathbf{1}}{(-\beta \mathbf{B}^{-1} \mathbf{1})^2} \right) = \ln \frac{2\Psi[\mathbf{B}^{-2}]}{\Psi^2[-\mathbf{B}^{-1}]}\end{aligned}$$

From these formulas Ishay proceeded to derive representations in terms of hyper- and hypoexponential distributions. One has to be aware, that this representation of the logarithmic normal distribution is stated in terms of the first two moments only and nothing has been said about higher moments or the shape of the distribution. If one is concerned about the latter, an improved match could be established by one of the methods described in chapter 4.

The same applies to more general distribution structures. Provided sufficient data records are available, the phase type approach leads to acceptable estimates for the performance indicators in a rather general setting. Furthermore the analytical difficulties prevalent in models of the $M/G/c$ and $G/G/c$ type point in the same direction.

Customer Impatience and Retrials

Although retrials and impatience constitute a part of the other processes, we have devoted an own section to these phenomena. This decision is motivated by two factors: First, to maintain the relation to queueing theory and second, to emphasize its special role with respect to parameter estimation. As opposed to other traffic processes discussed so far, there are no operational data available describing the retrial behaviour or impatience of customers. For the latter case something can be done. When viewed as incomplete data problem, the patience survival function $\bar{G}(\cdot)$ may be calculated by means of the *product limit estimator* first introduced by E.M. Kaplan and P. Meier [139]. To be more specific, the patience time is the minimum of the virtual waiting time and the time to abandon. Assume an ordered sample of distinct abandonment times a_1, a_2, \dots, a_m of size $m \leq n$, where n denotes the

total number of waiting customers. Furthermore let A_k denote the number of abandonments at a_k and denote the number of customers still present and uncensored prior to a_k by B_k . Following the common procedure given in [139] and [103], this leads to the product limit estimator of the patience survival function

$$\bar{G}(t) = \prod_{k: B_k \leq t} \frac{B_k - A_k}{B_k} = \prod_{k: B_k \leq t} (1 - \hat{h}_k)$$

where $\hat{h}_k := A_k/B_k$ is the estimated hazard rate of patience. From there an estimator for the patience intensity δ may be calculated as follows

$$\hat{\delta} = \left(\int_0^\infty \bar{G}(t) dt \right)^{-1}$$

Both estimates may be applied right away to common retrial models such as the $M/M/c + M$ and $M/M/c + G$ queueing systems. One has to be careful, when applying the product limit estimator in presence of heavy censoring or when the last observation is censored. This leads to biased estimates, which might affect the model under consideration. Alternatively one may follow the classic approach. That is to choose a specific distribution and estimate its parameters. Palm suggested the use of the *Weibull distribution* for that purpose. For more information on the Weibull distribution refer to [108].

Although retrial models are well established from a queueing theory perspective, there is only an ample evidence of empirical aspects found in the literature. Before proceeding, we need to clarify the term retrial. It is commonly associated with a repeated attempt after being blocked by a service facility. But this is not the only interpretation. Retrials may also occur in terms of follow up calls after having successfully completed service. In either case the retrial behaviour may be assessed from call detail records provided the calling party number, disconnect cause and time of occurrence have been tracked. This allows for the specification of the percentage of repeated attempts in terms of abandoned or serviced primary calls. Obviously such an assessment is a long term activity. In the past the retrial phenomenon has often been neglected. But this leads to misleading results, especially in the case of a non-stationary analysis [23]. In this work we have only discussed retrial models for abandonments with retrial times following an exponential or phase type distribution. In both cases one of the methods introduced in chapter 4 may be applied to assess the corresponding parameters.

6.3 Call Flow Model

The call flow can be considered either alone or as part of a larger model also describing call distribution. This raises the question for the better strategy. Before answering this question, we should locate, where queueing occurs within the call flow. First note, that human resources are better modeled as queueing systems, whereas technical resources are best represented by loss systems. One reason for that lies in the implementation of the latter. If the desired resource is not available, the call is either blocked or diverted. With respect to human resources, queueing usually occurs right before the call is directed to the agent. At that point the caller has been fully classified by the IVR system or a similar device and the caller experiences some type of entertainment. No further routing decisions are carried out and the caller is only diverted to another split group in case of overflow. Accordingly the call flow should be splitted in a classification and an entertainment part. The same separation can be carried out from a modeling perspective. The entertainment part overlaps with the queueing time of the customer and so one is only concerned to provision enough resources for an undisturbed operation. Therefore only the system view needs to be adopted. But there is also a structural difference between the classification and the entertainment part of the call flow. While the former possesses a tree-like structure, the latter is usually implemented as an endless loop. Accordingly only the call flow steps used for classification are shown in call flow diagrams similar to the one presented above in figure 6.1.

Before proceeding the discussion, consider the following implementation common to AT&T, Lucent Technologies and Avaya PBX systems and communication platforms. Please note, that some textual ammdements have been made to increase the readability of the script.

```
01 goto step 17 if time-of-day is all 18:00 to all 07:59
02 goto step 17 if time-of-day is fri 18:00 to mon 07:59
03 collect 1 digits after announcement 79001 'Menu'
04 goto step 07 if digits = 1
05 goto step 11 if digits = 2
06 route-to number 80000 with cov y if unconditionally
07 queue to skill 05 pri 1
08 goto step 15 if staffed-agents in skill 05 = 0
09 wait 30 seconds hearing announcement 79002 then music
```

```
10 goto step 08 if unconditionally
11 queue to skill 06 priority
12 goto step 15 if staffed-agents in skill 06 = 0
13 wait 30 seconds hearing announcement 79003 then music
14 goto step 12 if unconditionally
15 disconnect after announcement 79004 'Emergency'
16 stop
17 disconnect after announcement 79000 '00BusyHours'
18 stop
```

The first two steps ensure, that call processing only occurs within business hours Monday to Friday between 08:00 and 18:00. In case someone calls outside the regular business hours, the call proceeds at step 17 triggering an appropriate announcement. Otherwise the caller is presented the voice menu implemented by steps 03 to 06. If neither 1 or 2 is selected in the menu, the call is redirected to an external resource residing at extension 80000. In choosing 1 or 2, the call proceeds at step 07 or 11 to be queued to skill group 05 or 06. At this point the call classification task has been completed and the entertainment part begins. Please note, that the `queue to` command is to be understood as an instruction to the call queueing engine operating in background. There is no significant delay in execution, as the call is only added to the queue of the specified skill group. In classic programming languages we would identify this step with some sort of interprocess communication. Accordingly execution of the subsequent steps starts immediately. While steps 08 and 12 provide an escape in case of emergency, steps 09 and 13 constitute the heart of the entertainment part. They provide advertisements and music on hold to a waiting customer. Also note, that steps 10 and 14 implement the loop, which is so characteristic for the entertainment part. In case an agent becomes free, script execution is interrupted and the call is redirected to that agent (interruptive assignment). This again is an implicit action not visible to the script language. If no customers are waiting for the specified skill group, the script is already terminated when the corresponding `queue to` step is encountered.

Turning attention to the entertainment part first, it becomes evident, that it is overshadowed by the queueing process. In case there are no customers waiting it does not become active at all. Provided, that all necessary resources are available to serve the customers at high quality, we do not gain any further information out of the entertainment part. In fact, when know-

ing the calls offered to the queueing process we are also in the position to determine the capacity of system resources such as media ports, voice and video announcements.

For the discussion of the classification part we can safely omit the entertainment steps 07 to 14. Giving a closer look to the rest one immediately gets aware of the tree like structure mentioned above. Adopting the viewpoint of a call traversing through the steps we can also see, that the majority of steps possesses deterministic holding times. The only step, which exhibits a statistical nature, is step 03 showing the `collect digits` command. Its holding time depends on the user, but is also bounded by the system through the so called *interdigit timeout*, which is usually set to 4 seconds. As such we are facing a truncated service time distribution with almost negligible length compared to the time consumption of the announcement steps. In assuming these steps to be of deterministic nature we can expect the error to remain within reasonable bounds. In the above example it is an easy task to derive the overall classification time per skill group. Provided we know the call load offered to the entry point associated with the above script as well as the routing probabilities for steps 04 to 06, we are able to determine the call loads offered to extension 80000 and skill groups 05 and 06. Assuming, that no abandonments occur during call classification we thus arrive at a (delayed) Poisson arrival process for the corresponding call distribution models associated to steps 11 and 14. In presence of abandonments, we are facing a filtered Poisson process, which is best modeled in terms of overflow traffic.

There are other approaches and most of them are based on loss or queueing networks. But this often requires exponential service times to be assumed, which has to be considered a coarse approximation to deterministic holding times. In a similar fashion one may include the classification steps as part of a larger analysis based on phase type queues, but again the deterministic nature of some nodes may introduce analytical and numerical difficulties.

The same reasoning carries over to networked ACDs operating in a distributed call centre environment. As there is no structural difference in the call flow one may proceed as described above and separate classification from entertainment part. In some cases we may expect additional steps, which are responsible for the exchange of data between locations and the redirection of calls. Sites are loosely coupled and so the latter is often implemented as classic call transfer or basic call. The corresponding address information is nothing else than a valid telephone extension or an URL.

Especially in distributed call centre environments one has to be very careful about the terms used. As an example consider a call centre implementation featuring a unified queue, that is a single queue assigned to a skill group spreaded over several locations. Here queue control and call distribution is usually exercised by the device also responsible for call flow processing. Apparently the physical description differs from the conceptual one presented in this text. Usually this does not pose a problem, as for analytical purposes the equivalency of systems is sufficient.

6.4 Call Distribution

The effects of call distribution mechanisms are best analyzed by considering the waiting time distribution under different queueing disciplines. It turns out, that only some disciplines are suitable for call centre purposes. Unfortunately there is no standard naming convention for call distribution policies and queueing disciplines. For the latter we adhere to the notation introduced in table 3.1, which has become the most common one in last years. If not otherwise stated, we will restrict ourselves to Markovian systems only. In other words, we assume Poissonian arrivals and exponential service times.

6.4.1 Classic Disciplines

From the insensibility of the steady state distribution with respect to the queueing discipline and Little's law we know, that the average waiting time exhibits the same robustness. But in general this is not true for higher moments and the entire distribution. In fact, it has been shown for the $M/M/c$ queueing system [59], that the conditional waiting time \check{W}_q provided an arrival has to wait is exponentially distributed, i.e.

$$\Pr \left\{ \check{W}_q > t \mid \check{W}_q > 0 \right\} = \exp \{ - (c - \rho) \mu t \} \quad (6.1)$$

where $\rho := \lambda/\mu$ the traffic intensity, λ the arrival rate and μ the service rate. Please note, that we have implicitly assumed the *first come first serve* (FCFS) queueing discipline. The unconditional waiting time distribution is given by [90]

$$\Pr \left\{ \check{W}_q > t \right\} = 1 - p_d \exp \{ - (c - \rho) \mu t \}$$

where

$$p_d = \frac{p_0 \rho^c}{c!(1 - \frac{\rho}{c})}$$

is the *Erlang C formula* as given by expression 3.24. Obviously the FCFS discipline is a natural choice for a split group following some simple policy in a call surplus situation. As long as there are customers waiting, the first one is next to be scheduled for service when an agent becomes free. Even with no calls present, the $M/M/c$ model and its variants provide valid results for the average performance. Adopting a system or network perspective, the FCFS discipline is commonly accepted for technical implementations. As such, it becomes the most widely used discipline for resource engineering purposes including assessments for trunk groups, tone detectors and other system components.

If customers are picked in random order, we are talking about *random selection for service (RSS)*. F. Pollaczek derived an exact expression for the conditional waiting time distribution [59]

$$\Pr \left\{ \check{W}_q > t \mid \check{W}_q > 0 \right\} = 2(1-u) \int_0^\pi \exp \{-At\} \frac{B \sin x}{1 + \exp \{\pi \cot x\}} dx \quad (6.2)$$

where

$$\begin{aligned} A &= 1 + u - 2\sqrt{u} \cos x \\ B &= A^{-2} \exp \left\{ x + 2 \frac{\arctan \sqrt{u} \sin x}{1 - \sqrt{u} \cos x} \right\} \cot x \end{aligned}$$

and $u = \rho/c$ the utilization. According to J. Riordan [171], the conditional waiting time distribution may be approximated as follows

$$\Pr \left\{ \check{W}_q > t \mid \check{W}_q > 0 \right\} \approx \frac{1}{2} \alpha e^{-\alpha(c-\rho)\mu t} + \frac{1}{2} \beta e^{-\beta(c-\rho)\mu t} \quad (6.3)$$

where

$$\alpha = 1 + \sqrt{\frac{1}{2}u}, \quad \beta = 1 - \sqrt{\frac{1}{2}u}$$

Expression 6.3 provides reasonable results for $u < 0.7$. Generalizing the single server result of P.M. Morse [151], its first four moments match with the corresponding ones of the exact expression. Although the RSS discipline assumes a random selection of customers for service, it still might provide a

reasonable approximation for call distribution policies, which assign the most idle or least occupied agent to the next arrival.

As will be seen later, the *last come first serve (LCFS)* queueing discipline is more of theoretical interest. Together with FCFS it will assume the role of a boundary discipline with respect to the effect caused by a rather arbitrary discipline. The corresponding expression of the conditional waiting time has been derived by Riordan and is given as follows

$$\Pr \left\{ \check{W}_q > t \mid \check{W}_q > 0 \right\} = \sqrt{\frac{c}{\rho}} \int_0^{c\mu t} \frac{e^{-(1+u)x} I_1(2x\sqrt{u})}{x} dx \quad (6.4)$$

where $I_1(\cdot)$ denotes the Bessel function of the first kind [187]. The evaluation of the integral and the computation of the Bessel function in expression 6.4 may be avoided by considering the following approximation

$$\Pr \left\{ \check{W}_q > t \mid \check{W}_q > 0 \right\} \approx \frac{1}{2} \alpha e^{-\alpha(c-\rho)\mu t} + \frac{1}{2} \beta e^{-\beta(c-\rho)\mu t} \quad (6.5)$$

where

$$\alpha = 1 + \sqrt{u}, \quad \beta = 1 - \sqrt{u}$$

For more details refer to the books [132][171] by L. Kosten and J. Riordan. As for the RSS approximation expression 6.5 can be considered exact with respect to the first four moments. Furthermore it has been shown in [171], that the LCFS policy can be considered worst with respect to the delay function $\Pr \left\{ \check{W}_q > t \mid \check{W}_q > 0 \right\}$.

The conditional waiting time distributions under FCFS, LCFS and RSS queueing disciplines for the $G/M/c$ model have been derived in [59]. We omit their treatment here, because the Poisson assumption is often justified for arrival patterns occurring in call centre environments.

The delay or conditional survival functions $\Pr \left\{ \check{W}_q > t \mid \check{W}_q > 0 \right\}$ of the policies FCFS, RSS and LCFS for a typical Markovian queueing system are shown in figure 6.7. When operating under the LCFS discipline, the chance of exceeding a certain waiting time threshold is higher than for the FCFS or RSS discipline. Furthermore FCFS seems to provide the best results. This suggests the FCFS and LCFS policies as boundary disciplines. Indeed, it has been shown in [111], that this is true for all work conserving disciplines with respect to the waiting time W_q and the expect sojourn time before abandoning the queue of a $G/G/c + M$ queueing system with impatient

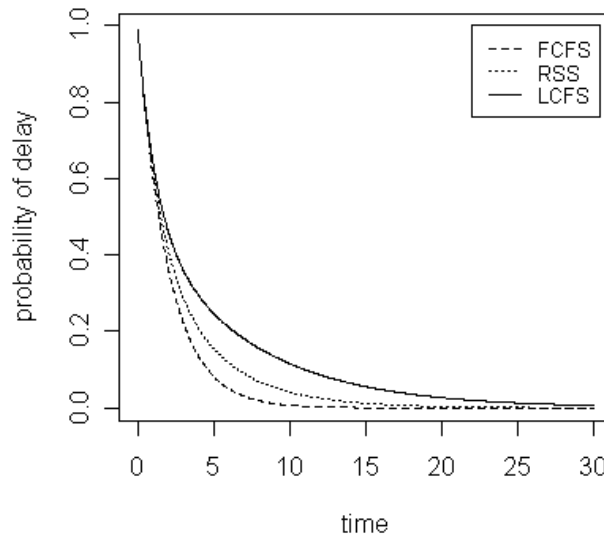


Figure 6.7: Probability of delay for the FCFS, RSS and LCFS policy

customers. A work conserving discipline assures, that no server is idle while customers are waiting. Otherwise the upper bound for the probability of delay under the LCFS regime does not necessarily hold, because the server idle times add to the waiting time of each customer. However, we might expect the FCFS policy to be the optimal choice for systems featuring a single queue. This explains, why it has been chosen so often for technical and organizational implementations. In fact, queueing systems operating under the FCFS regime are sufficient for the analysis of many real world scenarios, when adopting call, system or network view. Viewed from the agent perspective, the policies FCFS and LCFS provide lower and upper bounds on certain waiting and sojourn time distributions.

Some ACDs allocate calls to agents in strictly circular fashion, that is every c -th arrival is allocated to the same agent provided there are c agents staffed. This strategy is called *channel allocation (CA)* and leads to a description of the arrival process in terms of a filtered Poisson process. As such, the input stream is of the Erlangian type and so the system is best modeled as an $E_c/G/1$ or $E_c/M/1$ queue. For more details on this interesting account refer to [179].

If the service demand of each customer is known in advance, the average waiting time may be reduced as well. In that case one can make use of time dependent queueing disciplines such as *shortest processing time first (SPT)* and *shortest remaining processing time first (SRPT)* [124][203]. Although such information might be available for some technical resources, we cannot be certain in advance about the holding time of a call or agent. Although SPT and SRPT are non-feasible policies for that purpose, a similar idea has been formulated in context of priority classes. When properly configured, this leads to a reduction in the average waiting time for calls, agents and similar resources. This will be described next.

6.4.2 Call Priorities

The effect of priorities on queueing system has mainly been studied in the context of single server systems. This leads to simpler analytical expressions, while many insights gained from these models remain valid for multiserver systems as well. The main reference in the field is still the book [106] by Jaiswal. Priority policies allow for a classification of customers. Each class is processed according to the FCFS regime. This concept leads to virtual queues for each class. Customers belonging to a high priority class are allowed to overtake low priority customers. To allow for a more detailed characterization of priority policies, one has to consider the customer in service. According to Jaiswal, the following disciplines may be distinguished

- *preemptive disciplines* allow for interruption of service, if a customer with higher priority than the one currently in service arrives.
- *non-preemptive* or *head-of-the-line (HOL) disciplines* allow the current customer to complete his service. An arriving customer may advance the queue to be placed ahead of customers with lower priority.
- *discretionary disciplines* allow the server to decide whether he continues or discontinues the current service on arrival of a high priority customer.

There is further classification for preemptive policies, which has been motivated by the way the preempted unit proceeds service. As telephony applications aim to avoid interruptions of any kind, there are only few use cases for preemptive priority regimes. Therefore we refer to [106] for a complete treatment. We only want to note, that customers served at top priority do

not suffer any delay from customers belonging to a lower priority class. As such, their performance indicators are of the same structure as the ones for the corresponding non-priority model with the overall arrival rate replaced by the arrival rate for the top priority class. With respect to an application to call centre environments we will restrict the following discussion to non-preemptive queueing disciplines.

For a purely Markovian single server queueing system featuring two heterogeneous classes P.M. Morse [151] showed, that the total average queue length and the total average waiting time change by the factor

$$\frac{1 - \delta \bar{u}}{1 - \delta \rho} \quad (6.6)$$

on introduction of priorities. Here $\bar{u} = \rho [\delta + (1 - \delta) \beta^{-1}]$ denotes the *effective utilization* and $\rho = \lambda/\mu$ the traffic intensity assuming arrival rates $\delta\lambda$, $(1 - \delta)\lambda$ and service rates μ , $\beta\mu$ for class 1, 2 customers, respectively. When expression 6.6 is greater than unity, we can expect a higher waiting time and more customers waiting. As a matter of fact, this undesirable result is assumed for $\beta > 1$. In other words, if priority is assigned to classes of customers, which tend to have a faster service rate, the overall average waiting time and the total average queue length decrease. This result has been generalized to arbitrary service times in [106]. Provided, there are k priority classes each with a service demand μ_i , $i = 1 \dots k$ and cost per unit delay K_i , $i = 1 \dots k$ the optimal priority assignment is achieved for a descending order of K_i/μ_i . In assuming equal cost and exponential service times we arrive at the result of Morse. Please note, that we have assumed a non-preemptive priority policy. When considering preemptive disciplines, the above result remains valid only for exponential service times.

In call centre environments, priorities are assigned to calling customers according to their status. Typical classifications range from premium to normal, gold to bronze customers, emergency to normal calls. The above results may help to judge the effect on the average waiting time of a priority assignment. On the other hand, they provide guidelines on how to split single groups to allow for a gain in performance. However, priority assignments become dangerous for overloaded split groups, because low priority customers may be completely locked out from service and the potential waiting time increases ad infinitum. To avoid such deadlocks, overflows to other split groups, multiskill configurations and reserve agents have been considered as an alternative. This will be described next.

6.4.3 Multiple Skills

According to Koole [131], we are facing two types of problems when considering multiple skills. The first is concerned with the staffing problem, that is to determine the number of agents and their skill configuration required to reach a certain level of service. The other problem is related to call distribution. While only the latter is tackled here, the former will be briefly discussed in section 6.6.

In order to analyze the effects of call distribution, skill based routing is best approached by models featuring multiple queues. In some way they are related to priority models, because calls are classified and assigned to the target queue. However, by introducing a milder regime, deadlock situations are avoided. The first account on such a system has already been given by Morse in his book [151], which has been written before 1958. A more advanced model has been considered by G. Koole, P.D. Sparaghis and D. Towsley in [127]. They show, that given the service time distribution possesses an *increasing likelihood ratio (ILR)*, the decision to *join the shortest queue (JSQ)* minimizes the average queue length and the overall average waiting time. Having assumed arbitrary arrivals, which may not depend on the system state, this is a rather general result. Let \check{S}_t denote the remaining service time given that the customer under consideration has already received t units of service. Furthermore assume that \check{S}_t possesses density f_t . Then the (continuous) service time distribution is said to belong to class ILR, if for $x_1 < x_2$ and $t_1 \geq t_2$ the following property holds:

$$f_{t_1}(x_1) f_{t_2}(x_2) \geq f_{t_1}(x_2) f_{t_2}(x_1)$$

One example is the exponential distribution. A similar configuration with Poisson arrivals and exponential service times has been considered by D.R. McDonald and S.R.E. Turner in [147]. They compare the effects of three policies, namely JSQ, join the queue with shortest actual wait (JSAW) and join the queue with shortest expected wait (JSEW). It turns out, that JSAW performs better than JSEW, which in turn should be preferred against JSQ. In other words, balanced waiting times reduce the probability of a potential overflow to occur from one skill group to another one. Whereas JSAW balances the actual wait, JSEW only affects the expected waiting time and JSQ does not even do that. This result is valuable for both multiskill configurations and distributed call centre architectures. Some vendors offer call distribution and load balancing between locations based on waiting times.

According to [147], JSAW should be the preferred policy. Furthermore for the JSEW discipline the expected waiting time needs to be estimated in advance. Compared to the actual waiting time, this estimate increases the uncertainty and has to be considered less stable. The actual waiting time is best measured by considering the waiting time of the oldest call waiting in the queue. The best way to implement JSAW is to queue an arriving customer simultaneously at each location or for each skill group. When assigned to service, the customer is removed from all queues. Physically the call needs to be parked in an interactive voice response (IVR) system or similar device. This type of implementation inherently provides a high reliability for distributed call centre architectures. If one location fails, the call is still queued in the remaining locations. This is a clear advantage when compared to single queue systems commonly used in virtual call centre architectures.

So far we have assumed homogenous resources. Focusing on call centre agents, heterogeneity is often desired from a business point of view. One example is the introduction of cross-trained agents or the classification of agents in generalists and specialists. While specialists serve a specific application, generalists usually handle calls overflowing from specialists belonging to several skill groups. The corresponding model based on Markov chain analysis has been considered by R. Stollatz in his book [191]. His model features impatient customers and two types of calls handled by three skill groups, the generalists and one group of specialists for each type of call. He shows, that the total average waiting time weighted by the arrival rates for each call type is below the average waiting time of the corresponding $M/M/c/K$ queue with c the total number of agents and K the total number of trunks. Furthermore he discusses the effect of call priorities on the average waiting time, which are similar to the results described above. However, there are other aspects of heterogeneity, which have to be considered. In fact, an agent with multiple skills provides a higher degree of flexibility in a rapidly changing environment. He is better suited to work on different tasks during the day and to smooth effects of seasonality. Analytical models can help to judge the trade off between performance objectives and flexibility.

Another topic to be discussed is the instance of a reserve agent. As opposed to overflow systems, reserve agents are staffed for groups in overload situations. A commonly applied technique to detect an overload is based on threshold policies. From [126] we may conclude on its effect with respect to the queue length and the waiting time. The results state the optimality of the threshold policy provided the reserve agents do not operate as fast

as the mainstream agents. This is usually the case, when backoffice staff is considered to be used as reserve agents. However, if an accelerated operation is considered for excessive requests, this might violate the optimality of the threshold discipline.

Summarizing, there are many results on the optimality of call distribution regimes, which allow for the identification of bounds on the mean queue length and the average waiting time. In order to analyze a specific discipline, it has to be explicitly included into the model. However, a call centre operator has to consider other factors as well. For example, the call centre should be prevented from getting overskilled. This might affect the performance in an undesired way and raises the cost of education and training. It is also not advisable to employ only generalists, as the advantage introduced by multiskill configurations is limited [131].

6.4.4 Call and Media Blending

In order to analyze call or media blending, one has to separate real time requests from non-real time ones. The former class includes instant messages, video and voice calls, whereas the latter constitutes any postponable task or background work. The corresponding system is best described by a preemptive priority queueing model, where top priority is assigned to real time requests. From the underlying theory [106] we are aware, that customers of lower priority classes do not have any effect on the average waiting time of a top priority customer. In other words, they are invisible. With only one real time class available, we may switch to a classic queueing model with FCFS queueing discipline for analysis. Based on the utilization factor, additional background work may be performed by the call centre agents when no real time requests are queued. In most cases, this simplification is also an improvement. While real time requests are sensible to delays, background work is postponable. With respect to the latter one is concerned with keeping a certain level outstanding work to compensate for idle times and increase the overall agent utilization. One way to tackle with this problem is by means of inventory control.

From a design perspective, blending raises the question for the optimal policy. In their paper [41] S. Bhulai and G. Koole considered a Markovian queueing model with two blended traffic classes. If no background jobs are present, the model is equivalent to the $M/M/c$ queueing system. As opposed to our description above, no preemptions are allowed. By imposing

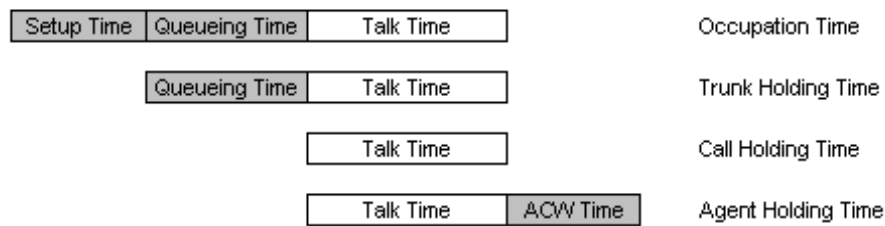


Figure 6.8: Different interpretations of holding time

a threshold on the maximum number of background jobs $m < c$ served by a split group of size c its effect on the throughput of background work and the average waiting time for real time requests is studied. While the former increases rather linearly the latter grows very slow for values of m not close to c . In other words, an increase in agent productivity comes at very low cost in terms of waiting time experienced by video or voice calls. Furthermore there is some empirical evidence, that this threshold discipline is very close to the optimal policy even for heterogenous service rates. For more details please refer to the paper [41] by Bhulai and Koole.

6.5 Resource Modeling

Resource modeling plays an essential role in the performance analysis of call centres. Queueing models have been used for a long time to describe resources of any kind. It is felt by the author, that in the past technical resources have received more attention than human ones. As such an emphasis is placed on the description of call centre agents and similar resources. We will start with the discussion of classic telephony and queueing approaches in the context of call centres and then proceed to more recent models. For the latter we follow the phase type approach to describe models based on the agent state diagram introduced in section 1.4.1.

For resource modeling it is essential to be aware of the perspective adopted. As an example consider the term *holding time*. From the viewpoint of a call the so called *call holding time* is nothing else than the talking time. In attaining a system perspective we are more concerned about the occupation time of technical resources and so we might have to include call setup and

queueing time as well to arrive at a proper description of the holding time. This becomes different when adopting an agent perspective. Obviously the agent is blocked from accepting a new customer when he is currently on the phone or performing some after call work. This leads to an *agent holding time* as shown in figure 6.8. We will proceed to enhance common terms such as the holding time in the way just presented to underline the viewpoint implicitly assumed. In some cases we will explicitly refer to one or the other perspective to allow for a clear understanding.

If not otherwise stated, we will assume a *first come first serve (FCFS)* policy. This has been motivated by its natural importance and the fact, that FCFS serves as a boundary discipline for work conserving queueing disciplines. Keeping track of several policies would only clutter the text and barely add something new. For more information of the effects of call distribution policies refer to section 6.4.

6.5.1 Classic Models

The classic Erlang and Engset models have been widely accepted for use in telephony and call centre applications. Being equivalent to the $M/M/c/c$ queueing system with infinite and finite sources, both have been discussed in a larger context in section 3.3.2 and 3.3.6 already. For an infinite number of customers the Engset model approaches the Erlang model. Both models are concerned about the steady state distribution and the probability of being blocked or getting lost. The underlying theory is entirely Markovian although the $M/M/c/c$ queue exhibits an invariance property with respect to the service time distribution. Accordingly the Erlang model may be considered as an equivalent of the $M/G/c/c$ queueing system. When talking about Erlang models one implicitly assumes the Erlang loss system as has been done above. Note, that there also exists an Erlang delay system, which is equivalent to the classic $M/M/c$ queue. In call centre application one is usually concerned with both loss and delay. While loss is common to trunk lines, customers shall have the possibility to queue for an agent. This leads to the combined $M/M/c/K$ queueing model, but such a generalization comes at a price. Loosing the invariance property, we have to assume exponential service times. Assuming c agents and $K > c$ trunk lines, we may calculate the probability of loss and delay. On the contrary we may also set the latter to derive the former. To be more specific, consider the graphs shown in figure 6.9. Assuming a call centre setting described by a $M/M/c/K$ model with

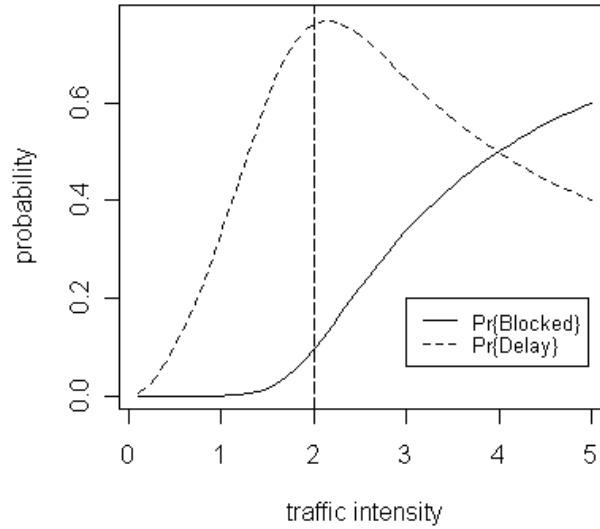


Figure 6.9: Loss and delay probabilities for various traffic intensities

$c = 2$ agents and $K = 10$ trunk lines, the loss and delay probabilities are considered as a function of the traffic intensity. As expected, the system remains stable even in overload situations, that is for $\rho \geq 2$. While the blocking probability steadily increases, the delay probability exhibits a maximum and even starts to decrease in case of overload. This typical behaviour results mainly from the fact, that customers more likely encounter a busy system rather than being queued. Adapted to the call centre situation, one can also decrease the average waiting time by reducing the number of available trunk lines! But this comes at the price of an increased chance, that customers are blocked by a fully loaded system. Accordingly the best strategy is to monitor both the number of waiting customers as well as the customers abandoned from the system. It is a common strategy to exclude those customers, which hang up within the first 10 seconds of waiting. Unless the underlying communication platform is able to distinguish customers being blocked due to resource limitations from impatient ones, such a strategy leads to a significant loss of information. Furthermore any optimization based on censored performance indicators easily leads to a degradation of service quality.

In analyzing a specific parameter constellation, the steady state distri-

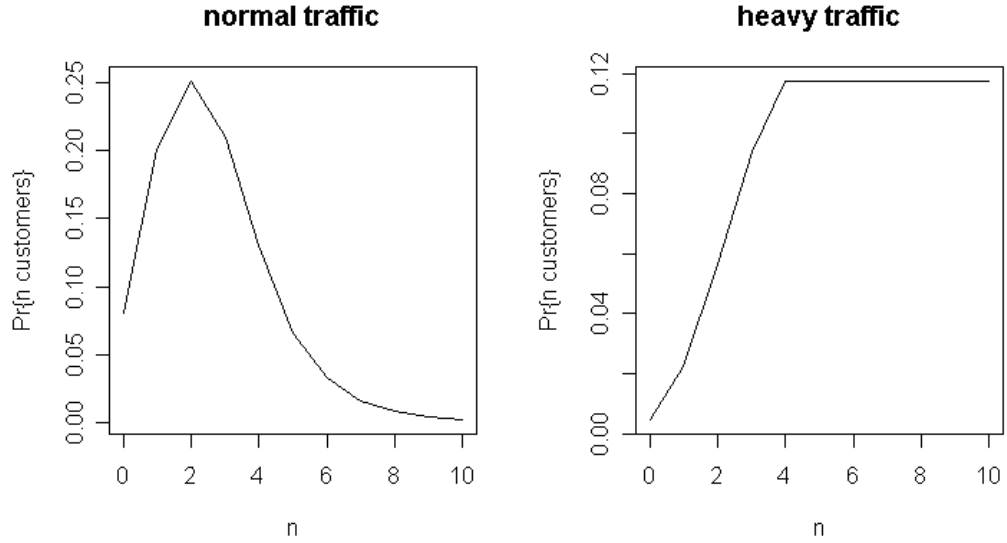


Figure 6.10: Steady state distribution under normal and heavy traffic conditions

bution function may provide a higher level of detail. As shown in figure 6.10, under normal traffic conditions the equilibrium probability function is more balanced. As the rightmost probability p_K also constitutes the blocking probability in the $M/M/c/K$ queueing model, one can see, that doubling the traffic does not necessarily lead to a doubled blocking probability. Using a discretized version based on sample proportions commonly found in call centre data records and checking the right tail of the resulting empirical density function provides a simple indication for the growing insufficiency in trunk capacities.

In a similar fashion more advanced variants of the $M/M/c/K$ model are to be considered. In incorporating the customer impatience into the model, one arrives at Palm's $M/M/c/K + M$ or the more general $M/M/c/K + G$ queueing system as described in section 3.3.4 and 3.4.2. Another extension also related to abandonments assumes that customers retry after being rejected by the system. The corresponding retrial systems are described in section 3.4.3.

The above mentioned models have been applied in call centre environ-

ments as well. It is obvious from the $M/M/c/K$ model description, that call holding times and agent holding times have been assumed to be the same. When modeling a split group featuring ACW times, this might result in highly inaccurate results. One solution to that dilemma has been to use separate $M/M/c$ and $M/M/c/c$ models for agents and trunks thus adopting an agent view for the former and a call or system perspective for the latter.

6.5.2 Impact of Overflow Traffic

Overflow traffic has been studied for more than 50 years in the context of telephone traffic engineering. Considering the aggregated traffic overflowing from multiple Erlang loss systems, pioneers such as L. Kosten, R.I. Wilkinson and C. Palm were able to derive approximations for the blocking probability of the target system. The traffic overflowing from a single Erlang loss system with c servers has been characterized as renewal process by Descloux. Furthermore he showed, that the Laplace transform of the *overflow density* $o_c(\cdot)$ is given by the recursion

$$\begin{aligned}\bar{o}_0(s) &= \frac{\lambda}{\lambda + s} \\ \bar{o}_n(s) &= \frac{\lambda}{\lambda + s + c\mu(1 - \bar{o}_{n-1}(s))}\end{aligned}\quad (6.7)$$

Accordingly the distribution of the inter-overflow time belongs to the family of $c + 1$ order hyperexponential distributions H_{c+1} [171][24][129]. The corresponding mean load and variance is given by

$$\rho_O = \rho E_c(\rho) \quad (6.8)$$

$$\sigma_O^2 = \rho_O \left(1 - \rho_O + \frac{\rho}{1 + c + \rho_O - \rho} \right) \quad (6.9)$$

where $E_c(\rho)$ is given by the classic Erlang loss formula 3.28 [132][171]. In order to emphasize the dependency on the two parameters c and ρ , we have changed the notation to the one common in teletraffic engineering. By using the above formulas we are now able to assess the *peakedness* of overflow traffic by the variance to mean ratio

$$z_O = \frac{\sigma_O^2}{\rho_O} \quad (6.10)$$

This is another difference to classic queueing theory. Instead of the coefficient of variation the variance to mean ratio has become more popular in teletraffic engineering. Please note, that both ratios equal one, if exponential interevent times are considered. One might expect, that the above theory also generalizes to combined delay and loss systems. Unfortunately this is not true, as the overflow process from a $M/M/c/K$ queue is not of the renewal type anymore.

We will now turn attention to some simple approximations. As above consider traffic overflowing from several Erlang loss systems to a single target. Assuming independence we may add the individual means and variances to arrive at a description in terms of ρ_O and σ_O^2 for the compound overflow stream. Wilkinson aimed to replace this renewal stream by an equivalent Poisson process. Instead of describing the stream itself, he parametrized the system generating the traffic with an equivalent load ρ and an equivalent number of servers c . This also coined the name *equivalent random theory* [215]. One approach to calculation is to iterate equations 6.8 and 6.9. The other one resorts to approximation and has been carried out by Y. Rapp. He derived the following expressions, which have been called *Rapp formula* to underline his efforts [37][24]:

$$\begin{aligned}\rho &\approx \sigma_O^2 + 3z_O(z_O - 1) \\ c &\approx \frac{\rho(\rho_O + z_O)}{\rho_O + z_O - 1} - \rho_O - 1\end{aligned}\tag{6.11}$$

As noted above, the inter-overflow time follows a $c + 1$ order hyperexponential distribution. For a large number of servers this leads to a large number of parameters. By matching moments, one may easily shrink them to an acceptable level. This leads to descriptions in terms of exponential [128], H_2 [80], C_2 and matrix exponential [156] distributions as well as *interrupted Poisson processes (IPP)* [24] and *Markov modulated Poisson processes (MMPP)*. This led to a variety of models each with its unique features. For example, in [128] the exponential approximation allows for the definition of a loss network with rather arbitrary (but deterministic) routing rules. In this respect it overcomes one of the shortcomings of the equivalent random theory, which may only be applied to tandem and tree-like structures. The method has been further improved to allow for a H_2 approximation of the overflow traffic [80].

An alternative to the equivalent random theory is the *equivalent congestion method* derived by A.A. Fredericks and W.S. Hayward. While in the

former multiple groups are replaced by a single equivalent group, the latter considers each server group separately. Furthermore the equivalent congestion method is capable of handling non-random traffic as well. Assume, that traffic with offered load ρ_O and peakedness z_O overflows to a secondary loss system featuring c exponential service channels. According to Fredericks and Hayward the blocking probability of the second system may be approximated by [215][107]

$$p_c \approx E_{\frac{c}{z_O}} \left(\frac{\rho_O}{z_O} \right)$$

Note, that the classic Erlang loss formula can not be used anymore, as most likely non-integer terms appear. Instead one has to apply expression 3.31. As for the equivalent random method, the equivalent congestion method adheres to recursive application, as the overflow traffic from the secondary system is $\rho_O p_c$.

It is also possible to perform a more detailed analysis of the scenario just described. This has been carried out by C. Palm and L. Takacs. They were able to derive exact expressions for the distribution function and the transforms of the binomial moments. Furthermore they provided approximations for the binomial moments. For more details please refer to the book [171] by J. Riordan.

Another alternative to the equivalent random and the equivalent congestion method is the GI approximation introduced by Akimaru and Kawashima in [24]. The traffic overflowing from other systems is considered as renewal process and fed into a $H_2/M/c/c$ queue. The overflow is described as an interrupted Poisson process (IPP), which leads to H_2 distributed interarrival times of the overflow process. Matching moments and applying the Rapp formula 6.11 supplies the required IPP parameters. Transformed into the corresponding H_2 parameters, the Laplace transform $\bar{a}(\cdot)$ of the H_2 density function is inserted into formula 3.99 for the blocking probability of the $G/M/c/c$ queue, which yields

$$p_c = \left[1 + \sum_{n=1}^{c-1} \binom{c}{n} \prod_{k=1}^n \left(\frac{\rho_O}{k\mu} + (z_O - 1) \frac{\rho_O + 3z_O}{\rho_O + 3z_O + s - 1} \right)^{-1} \right]^{-1}$$

As before the overflow traffic is specified in terms of ρ_O and z_O , while the exponential service facility has been assumed to operate at rate μ . It is also possible to consider other approximations of the overflow process. R.W. Wolff

aimed to capture the irregularity inherent to superposed overflow processes by using a batch Poisson process. He treated the target system as batch versions of the $M/D/c/c$ and $M/M/c/c$ queues and supplied exact results based on the equivalence of both to classic $M/D/k/k$ and $M/M/k/k$ queues, where k equals to c divided by the batch size. Obviously this approach can only be applied, if k is an integer value. However, he managed to provide a simple way to justify the performance of the equivalent random and the equivalent congestion method [215].

In order to derive bounds for the blocking probability, A.A.N. Ridder [170] considers the traffic overflowing from a $M/PH/c_1/c_1$ to a $G/C_2/c_2/c_2$ queueing system. By noting, that the service distribution of the latter is well represented by a phase type distribution we assume representations $PH(\beta, \mathbf{B})$ and $PH(\tau, \mathbf{T})$. For details refer to section 3.8.1 and 4.3.3. Applying expression 3.116 for the moments of a phase type distribution immediately leads to the mean times

$$\mu_1^{-1} = \Psi[-\mathbf{B}^{-1}], \quad \mu_2^{-1} = \Psi[-\mathbf{T}^{-1}]$$

Further assuming Poisson arrivals at rate λ to the first system, Ridder has calculated the following bound in terms of the Erlang loss formula

$$p_{c_2} \leq E_{c_1}(\lambda\mu_1) E_{c_2}(\lambda\mu_2)$$

Please note, that the overflow traffic does not depend on the server or the customers. These dependencies have been considered for the case of exponentially distributed service times at the secondary station in [170]. Another approach has been followed by N.M. van Dijk and E. van der Sluis when introducing their *call packing bounds*. For more details refer to the report [205].

So far we have discussed methods, which characterize the overflow traffic and associated blocking probabilities for loss systems only. As such we have found a set of adequate models for the description of technical resources and facilities without queues. In a typical call centre environment we are also concerned about combined loss and delay systems. Unfortunately the evidence for such models in the literature is very scarce. On the other hand we are aware from section 6.5.1, that the blocking probability decreases, when queueing is introduced to the system. Accordingly we may use the blocking probability results contained in this section as rough bounds. In this context a fewer number of waiting places means a closer bound. But one has to keep

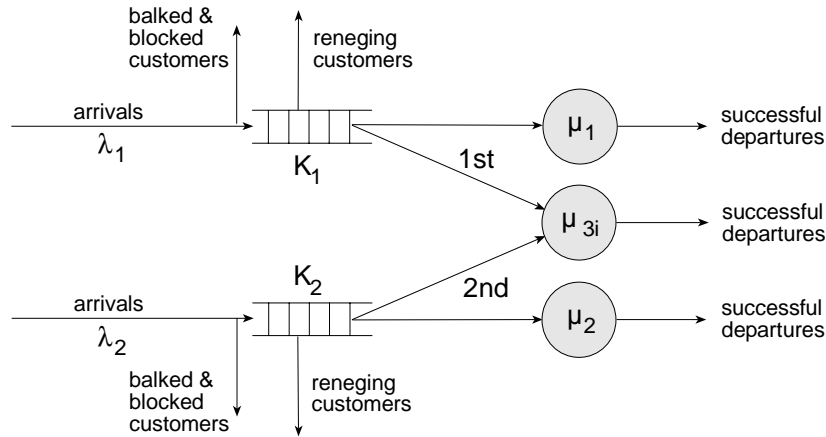


Figure 6.11: Call centre model with two customer classes and three skill groups

in mind, that in case of excessive wait the blocking effect becomes negligible and a pure delay model becomes the better choice for analysis.

As an alternative to the approximations and bounds contained in this section one might consider to carry out an exact analysis based on an extended state model. This leads to an application of Markov methods, which are treated next.

6.5.3 Markovian State Analysis

Among the most powerful methods we find the Markovian state analysis, which is based on the theory presented in appendix A.3. In some respect we may speak of a brute force approach to solution, as the specific structure of the transition matrix is only rarely taken into account. To preserve the flexibility while moving towards queueing theory one should also consider a phase type queueing model. If this is not possible, one can still stick to Markovian state analysis.

In [191] R. Stollatz introduced a Markov chain model of an inbound call centre featuring two caller classes, two dedicated and one general split group. The corresponding model is shown in figure 6.11. Calls are assumed to arrive according to a Poisson process with rates λ_1 and λ_2 . Upon arrival a customer of class $i \in \{1, 2\}$ is either blocked due to an exhausted queue, leaves the

system with probability β_i or joins the queue. In the latter case the customer might still renege and leave the system. The corresponding patience time is assumed to follow an exponential distribution with rate r_i . The queue sizes are limited to K_1 and K_2 calls and blocked customers are lost, i.e. there are no retrials. All service times are exponentially distributed and the dedicated agents are assumed to operate at rate μ_i , while the general ones serve customers of class i at rate μ_{3i} . Customers of the first class are treated at high priority. In other words, if no dedicated agents are free and calls of both classes are waiting, class 1 calls are given priority. The call distribution implements a FCFS policy. Furthermore we assume, that the split groups are staffed with c_1 , c_2 dedicated and c_3 general agents. The model does not distinguish between call and agent holding time, as no ACW handling has been included.

This model is well suited to investigate the interaction between dedicated and general agents. However, the design may not be appropriate for all purposes, as some vendors insist on having a queue placed in front of a split or skill group. Very often some mechanism exists to reassign the call to another queue even while waiting. This is called *jockeying* in queueing theory and has not received much attention in the literature. One solution to the problem might be a simultaneous assignment to all queues. For some notes on these topics please refer to section 6.4.

Before the model can be solved by some standard method, the state space has to be defined. Usually there is more than one way to configure the state space and R. Stollitz has decided to use the quadruple (i, j, k, l) where

- k is the number of general agents serving calls of class 1 subject to $0 \leq k \leq c_3$
- l is the number of general agents serving calls of class 2 subject to $0 \leq l \leq c_3 - k$
- i is the number of class 1 calls in the system subject to $k \leq i \leq K_1$,
 $i - k \leq c_1$ if $k + l < c_3$
- j is the number of class 2 calls in the system subject to $l \leq j \leq K_2$,
 $j - l \leq c_2$ if $k + l < c_3$

Based on the state representation the balance equations are derived in terms of $p_{i,j,k,l}$ by use of the global balance principle. The resulting system

of equations is then solved by the so called uniformization approach. We omit the lengthy details here and instead refer to the book [191] by R. Stollatz. In order to calculate performance indicators of interest one has to isolate a specific region of the state space and add the corresponding equilibrium probabilities $p_{i,j,k,l}$. This leads to the queue lengths

$$\begin{aligned} L_q^{(1)} &= \sum_{k=0}^{c_3} \sum_{i=c_1+k+1}^{K_1} \sum_{j=l}^{K_2} (i-k-c_1) p_{i,j,k,l} \\ L_q^{(2)} &= \sum_{k=0}^{c_3} \sum_{i=k}^{K_1} \sum_{j=c_2+l+1}^{K_2} (j-l-c_2) p_{i,j,k,l} \end{aligned}$$

where the superscript denotes the call classification. The corresponding blocking probabilities are given by

$$\begin{aligned} p_{K_1} &= \sum_{k=0}^{c_3} \sum_{i=l}^{K_2} p_{K_1,j,k,l} \\ p_{K_2} &= \sum_{k=0}^{c_3} \sum_{i=k}^{K_1} p_{i,K_2,k,l} \end{aligned}$$

It is also possible to apply Little's law to derive the average queueing time for both call types. This requires the determination of the effective arrival rate for each class. By noting, that balking can only occur when a queue has formed one arrives at

$$\begin{aligned} \bar{\lambda}_1 &= \lambda_1 \left[\Pr \left\{ \check{W}_q^{(1)} = 0 \right\} + (1 - \beta_1) \left(1 - p_{K_1} - \Pr \left\{ \check{W}_q^{(1)} = 0 \right\} \right) \right] \\ \bar{\lambda}_2 &= \lambda_2 \left[\Pr \left\{ \check{W}_q^{(2)} = 0 \right\} + (1 - \beta_2) \left(1 - p_{K_2} - \Pr \left\{ \check{W}_q^{(2)} = 0 \right\} \right) \right] \end{aligned}$$

where

$$\begin{aligned} \Pr \left\{ \check{W}_q^{(1)} = 0 \right\} &= \sum_{0 \leq k \leq c_3, 0 \leq l \leq c_3 - k, k \leq i < c_1 + k, l \leq j \leq c_2 + l, k+l \neq c_3} p_{i,j,k,l} \\ \Pr \left\{ \check{W}_q^{(2)} = 0 \right\} &= \sum_{0 \leq k \leq c_3, 0 \leq l \leq c_3 - k, k \leq i \leq c_1 + k, l \leq j < c_2 + l, k+l \neq c_3} p_{i,j,k,l} \end{aligned}$$

denotes the probability, that calls of class $i \in \{1, 2\}$ proceed to service immediately. Now Little's law may be readily applied, i.e.

$$W_q^{(1)} = \frac{L_q^{(1)}}{\bar{\lambda}_1}, \quad W_q^{(2)} = \frac{L_q^{(2)}}{\bar{\lambda}_2}$$

Based on these and other performance indicators several numerical experiments are carried out in [191]. It turns out, that the pooling effect also applies to the current system under consideration. In fact, more cross-trained agents lead to an improved performance in terms of queue sizes, average waiting times and agent utilization. However, for an individual call centre installation the corresponding cost factors have to be taken into account too. These and other economical aspects have been considered by Stolletz as well. Another interesting feature of the model is the dependency of the blocking probability on the trunk size limits for both classes of calls. Raising the limit for high priority calls may have a negative effect on the blocking probability and the agent utilization of class 2 calls. Furthermore it has been shown, that general statements on how to staff the split groups are rarely available. Even for this Markovian model it is necessary to consider each case separately.

An alternative to the Markovian state analysis is the queueing network approach. The graphical representation of the system under consideration is transformed to a *product form solution* by means of an appropriate algorithm. This leads to the equilibrium probabilities, which may be used to derive the desired performance indicators. Such an approach has been followed by K. Polina in [167], which deals with the analysis of a call centre featuring an interactive voice response (IVR).

6.5.4 Matrix Exponential Approach

A call centre is a complex entity, that combines voice and data communications technology to enable organizations to implement critical business strategies or tactics aimed at reducing cost or increasing revenue. At an organizational level, costs are highly dependent on capacity management of human and technical resources. This introduces the need for exact or well approximated key performance indicators from which cost and revenue may be inferred. As noted in the introduction to this chapter, a single view might not be sufficient for that purpose. Especially for human resource management it becomes necessary to adopt several perspectives. This becomes immediately evident when reviewing the agent state model shown in figure 1.1. There is a stringent requirement to capture the effects of more than only the talking time of a call centre agent. We need to take care about all the phases of a typical workflow. This suggests the use of more complex techniques such as Markovian state analysis or matrix exponential queueing systems. We have chosen the latter, because the multiserver systems commonly used to

describe a call centre setup naturally fit to the underlying methods. While requesting little more theory, the matrix exponential approach generates the state space rather automatically. For Markovian state analysis we need to define the entire infinitesimal generator, while for a matrix exponential queue we only specify the corresponding interevent distribution. For an example of the efforts necessary to construct the state space of a more complex queueing system consult [191].

In this section we will focus on the agent perspective and describe the corresponding call centre setup as $M/ME/c/K$ queueing system, where we assume

- Poisson arrivals
- c agents staffed in the system
- a maximum of K available trunk lines

Obviously the most challenging part is the definition of the matrix exponential service distribution. In looking for a proper approach we have to face the problem of how to embed fictitious nodes into a physical representation. Otherwise we would be restricted to associate each agent state to a single exponential node only, which prevents us from incorporating some important effects into the model. As described in section 3.8.1 we may add some fictitious nodes to allow for approximation of the distribution of the agent state under consideration. This in turn can be achieved by one of the methods introduced in chapter 4. To underline the need for such approximations consider the effect caused by a so called *timed after call work (TACW)* mechanism. In case a split or skill group requires some sort of wrap up time, the transition to the ACW state is triggered by the termination of the corresponding inbound call. In a TACW scenario, the transition out of the ACW state occurs after a predefined time interval. Accordingly the distribution of the ACW time has to be considered a deterministic one. This is different from the classic version, where the ACW time constitutes a random variable. Therefore we need to find a node replacement behaving like a deterministic node to justify the effects of TACW. This is indeed possible by using a fixed node approximation as introduced in chapter 4. But before this can be done we need to find a way on how to embed a matrix exponential distribution representing node behaviour into a larger configuration describing the physical configuration.

Layered Matrix Exponential Distributions

We will now adopt a micro-macro approach to explain the concept of embedding one matrix exponential distribution into another. For the micro level we assume N matrix exponential distributions $ME(\boldsymbol{\beta}^{(j)}, \mathbf{B}^{(j)})$ with $1 \leq j \leq N$, while for the macro level there is only one phase type distribution $PH(\boldsymbol{\tau}, \mathbf{T})$ of order N with service rate matrix $\mathbf{M} \equiv \mathbf{I}$. By formula 3.109 this is equivalent to having defined the pair $(\boldsymbol{\tau}, \mathbf{P})$, where the transition matrix is given by $\mathbf{P} = \mathbf{T} + \mathbf{I}$. Obviously there is no need for service rates at the macro level, because their influence will be exercised by the corresponding matrix exponential distributions.

Merging models would be a simple task, if we assume that $\beta_{k_n+1}^{(j)} = \mathbf{1} - \boldsymbol{\beta}^{(j)}\mathbf{1} = 0$. Unfortunately this is not the case and so we have to foresee this instantaneous jumps and incorporate them as contributions to both parameters of the compound matrix exponential distribution. Consider the modification of the entry vector $\boldsymbol{\tau} = (\tau_1, \tau_2, \dots, \tau_N)$ first. Then we may calculate the probability of immediate completion γ_{N+1} by simply summing up all possible paths to the absorbing state $N+1$, i.e.

$$\begin{aligned} \gamma_{N+1} = \tau_{N+1} &+ \sum_{i=1}^N \tau_i \beta_{k_i+1}^{(i)} t_i \\ &+ \sum_{i=1}^N \tau_i \beta_{k_i+1}^{(i)} \left(\sum_{j=1}^N P_{i,j} \beta_{k_j+1}^{(j)} t_j \right) \\ &+ \dots \end{aligned} \quad (6.12)$$

where $t_j, 1 \leq j \leq N$ are the elements of the exit vector $\mathbf{t} = -\mathbf{T}\mathbf{1} = \mathbf{1} - \mathbf{P}\mathbf{1}$. Defining a row vector $\mathbf{u}^{(N+1)} = (\tau_i \beta_{k_i+1}^{(i)})$ and a (substochastic) matrix $\mathbf{V} = (P_{i,j} \beta_{k_j+1}^{(j)})$ for $1 \leq i, j \leq N$, the preceding expression becomes

$$\begin{aligned} \gamma_{N+1} &= \tau_{N+1} + \mathbf{u}^{(N+1)} (\mathbf{I} + \mathbf{V} + \mathbf{V}^2 + \dots) \mathbf{t} \\ &= \tau_{N+1} + \mathbf{u}^{(N+1)} (\mathbf{I} - \mathbf{V})^{-1} \mathbf{t} \end{aligned}$$

The successful inversion is guaranteed by the substochastic nature of \mathbf{P} and the exclusion of a non-proper transition matrix. The latter prevents endless loops resulting in an infinite interevent time. An extreme example for a non-proper transition matrix is the identity matrix \mathbf{I} . Also note, that \mathbf{V} has been

assembled from the substochastic matrix \mathbf{P} and the non-negative vector $\boldsymbol{\beta}^{(j)}$ for $1 \leq j \leq N$.

In a similar fashion we may calculate the probability of entering the j -th micro level γ_j , i.e.

$$\gamma_j = \tau_j + \mathbf{u}^{(j)} (\mathbf{I} - \mathbf{V})^{-1} \mathbf{P}_j \quad (6.13)$$

where \mathbf{P}_j is the j -th column of the transition matrix \mathbf{P} and $\mathbf{u}^{(j)}$ for $1 \leq j \leq N$ is given by

$$\mathbf{u}^{(j)} = \left(\tau_1 \beta_{k_1+1}^{(1)}, \dots, \tau_{i-1} \beta_{k_{i-1}+1}^{(i-1)}, 0, \tau_{i+1} \beta_{k_{i+1}+1}^{(i+1)}, \dots, \tau_N \beta_{k_N+1}^{(N)} \right)$$

In other words, $\mathbf{u}^{(j)}$ has always the same structure except for the j -th column, which equates to zero. Assembling these vectors into a matrix of dimension $(N+1) \times N$, that is by defining

$$\bar{\mathbf{U}} = \begin{pmatrix} \mathbf{u}^{(1)} \\ \mathbf{u}^{(2)} \\ \vdots \\ \mathbf{u}^{(N)} \\ \mathbf{u}^{(N+1)} \end{pmatrix} = (u_{ij})$$

where

$$u_{ij} = \begin{cases} 0 & i = j \\ \tau_i \beta_{k_i+1}^{(i)} & i \neq j \end{cases}$$

and setting $\bar{\mathbf{P}} = (\mathbf{P}, \mathbf{t})$ yields the compact expression $\boldsymbol{\gamma} = (\gamma_i)$ with

$$\gamma_i = \tau_i + (\bar{\mathbf{U}} (\mathbf{I} - \mathbf{V})^{-1} \bar{\mathbf{P}})_{i,i}, \quad 1 \leq i \leq N+1 \quad (6.14)$$

The entry vector for the compound matrix exponential distribution $ME(\boldsymbol{\beta}, \mathbf{B})$ is then given by

$$\boldsymbol{\beta} = (\gamma_1 \boldsymbol{\beta}^{(1)}, \gamma_2 \boldsymbol{\beta}^{(2)}, \dots, \gamma_N \boldsymbol{\beta}^{(N)}, \gamma_{N+1}) \quad (6.15)$$

If no instantaneous exit occurs at the micro level, we have $\beta_{k_n+1}^{(j)} = 0$ for all $1 \leq j \leq N$. This leads to the trivial values for \mathbf{u} and \mathbf{V} and by expression 6.12 and 6.13 to $\boldsymbol{\gamma} = \boldsymbol{\tau}$.

So far we have done nothing else than to compensate for immediate transitions, which have been caused by the possibility to reach the absorbing state

at the micro level. A similar reasoning also applies to the matrix parameter \mathbf{B} of the compound matrix exponential distribution $ME(\boldsymbol{\beta}, \mathbf{B})$. Any completion at the micro level is associated with an exit vector $\mathbf{b}^{(j)} = -\mathbf{B}^{(j)}\mathbf{1}$, $1 \leq j \leq N$. By replacing the absorbing state with a multitude of subsequent states, we need to distribute the rate among the new targets. First note, that \mathbf{B} must be of the form

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}^{(1)} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,N} \\ \mathbf{B}_{2,1} & \mathbf{B}^{(2)} & & \mathbf{B}_{2,N} \\ \vdots & & \ddots & \\ \mathbf{B}_{N,1} & \mathbf{B}_{N,2} & & \mathbf{B}^{(N)} \end{pmatrix} \quad (6.16)$$

where $P_{i,j} = T_{i,j} + \delta(j-i)$ are the elements of \mathbf{P} . The diagonal elements represent the immediate entrance to the j -th sublevel, while for the non-diagonal elements we need to distribute the rates of the exit vectors $\mathbf{b}^{(j)}$, i.e.

$$\begin{aligned} B_{i,j} &= \mathbf{b}^{(i)} \left(P_{i,j} + \sum_{n=1}^N P_{i,n} \beta_{k_n+1}^{(n)} P_{n,j} + \dots \right) \boldsymbol{\beta}^{(j)} \\ &= \mathbf{b}^{(i)} \boldsymbol{\beta}^{(j)} [(\mathbf{I} + \mathbf{V} + \mathbf{V}^2 + \dots) \mathbf{P}]_{i,j} \\ &= \mathbf{b}^{(i)} \boldsymbol{\beta}^{(j)} [(\mathbf{I} - \mathbf{V})^{-1} \mathbf{P}]_{i,j}, \quad i \neq j \end{aligned} \quad (6.17)$$

where \mathbf{V} and \mathbf{P} are defined as before. Also note, that the entry vector $\boldsymbol{\beta}^{(i)}$ is responsible to distribute the rate within the j -th micro level. The exit vector of the compound matrix exponential distribution is given by the common formula $\mathbf{b} = -\mathbf{B}\mathbf{1}$. Considering the trivial case from above, that is by assuming $\beta_{k_n+1}^{(i)} = 0$ for all $1 \leq i \leq N$ we immediately arrive at

$$B_{i,j} = \mathbf{b}^{(i)} P_{i,j} \boldsymbol{\beta}^{(j)}$$

as expected. Although the trivial case is widely applied, it is not very illustrative with respect to the current presentation. Accordingly we will present another simple example.

Example 34 Consider the configuration shown in figure 6.12. At the micro level we have a generalized and an ordinary exponential node. Adhering to our notation the latter is fully parametrized by

$$\begin{aligned} \beta^{(2)} &= 1, \quad \beta_2^{(2)} = 0 \\ \mathbf{B}^{(2)} &= \mathbf{M}(\mathbf{P} - \mathbf{I}) = \mu_2(0 - 1) = -\mu_2 \\ \mathbf{b}^{(2)} &= -\mathbf{B}^{(2)}\mathbf{1} = \mu_2 \end{aligned}$$

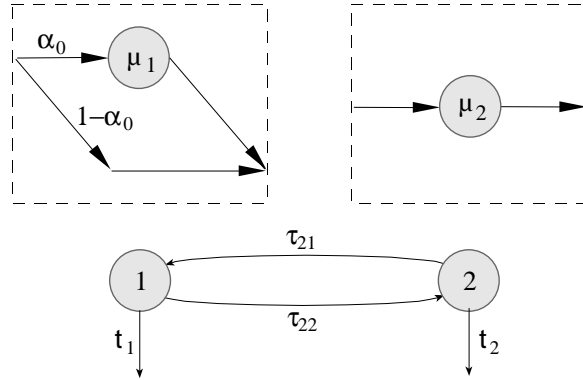


Figure 6.12: A micro-macro level example

while the former is given by

$$\begin{aligned}\beta^{(1)} &= \alpha_0, & \beta_2^{(1)} &= 1 - \alpha_0 \\ \mathbf{B}^{(1)} &= -\mu_1 \\ \mathbf{b}^{(1)} &= \mu_1\end{aligned}$$

For the macro level we deduce the following parameters

$$\mathbf{P} = \begin{pmatrix} 0 & P_{1,2} \\ P_{2,1} & 0 \end{pmatrix}, \quad \boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)$$

Preparing the matrices, i.e.

$$\begin{aligned}\bar{\mathbf{U}} &= \begin{pmatrix} 0 & \tau_2 \beta_2^{(2)} \\ \tau_1 \beta_2^{(1)} & 0 \\ \tau_1 \beta_2^{(1)} & \tau_2 \beta_2^{(2)} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ \tau_1 (1 - \alpha_0) & 0 \\ \tau_1 (1 - \alpha_0) & 0 \end{pmatrix} \\ \mathbf{V} &= \begin{pmatrix} 0 & P_{1,2} \beta_2^{(2)} \\ P_{2,1} \beta_2^{(1)} & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ P_{2,1} (1 - \alpha_0) & 0 \end{pmatrix}\end{aligned}$$

Noting, that $\mathbf{V}^2 = 0$ saves an inversion step. By using expression 6.14, we

arrive at

$$\begin{aligned}
\gamma &= \left(\tau_i + (\bar{\mathbf{U}}(\mathbf{I} + \mathbf{V})\bar{\mathbf{P}})_{i,i} \right) \\
&= \left(\tau_i + \left(\begin{pmatrix} 0 & 0 \\ \tau_1(1-\alpha_0) & 0 \\ \tau_1(1-\alpha_0) & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ P_{2,1}(1-\alpha_0) & 1 \end{pmatrix} \bar{\mathbf{P}} \right)_{i,i} \right) \\
&= \left(\tau_i + \left(\begin{pmatrix} 0 & 0 \\ \tau_1(1-\alpha_0) & 0 \\ \tau_1(1-\alpha_0) & 0 \end{pmatrix} \begin{pmatrix} 0 & P_{1,2} & 1-P_{1,2} \\ P_{2,1} & 0 & 1-P_{2,1} \end{pmatrix} \right)_{i,i} \right) \\
&= \left(\tau_i + \begin{pmatrix} 0 & 0 & 0 \\ 0 & \tau_1(1-\alpha_0)P_{1,2} & \tau_1(1-\alpha_0)(1-P_{1,2}) \\ 0 & \tau_1(1-\alpha_0)P_{1,2} & \tau_1(1-\alpha_0)(1-P_{1,2}) \end{pmatrix}_{i,i} \right) \\
&= (\tau_1, \tau_2 + \tau_1(1-\alpha_0)P_{1,2}, \tau_3 + \tau_1(1-\alpha_0)(1-P_{1,2}))
\end{aligned}$$

This leads to the compound entry vector

$$\begin{aligned}
\boldsymbol{\beta} &= (\gamma_1\boldsymbol{\beta}^{(1)}, \gamma_2\boldsymbol{\beta}^{(2)}, \gamma_3) \\
&= (\tau_1\alpha_0, \tau_2 + \tau_1(1-\alpha_0)P_{1,2}, \tau_3 + \tau_1(1-\alpha_0)(1-P_{1,2}))
\end{aligned}$$

A simple check may be carried out by adding the components of $\boldsymbol{\beta}$. As we are still concerned with probabilities, their sum must equal to one. Alternatively the model may be validated by tracing all possible paths. Giving a closer look to sublevel 1 reveals the possibility for bypassing that sublevel. This happens with probability $1 - \alpha_0$. Accordingly the process may start there but is instantaneously redirected to sublevel 2 or the absorbing state. Obviously we need to take care of this contribution and include it in our calculations for β_2 and β_3 . No such thing can happen for β_1 and so we are left with the product of the corresponding entry probabilities. Turning attention to the second parameter \mathbf{B} , we now proceed to determine its value by substituting the above matrices into expression 6.17. Some matrix algebra yields

$$\begin{aligned}
B_{1,2} &= \mathbf{b}^{(1)}\boldsymbol{\beta}^{(2)}[(\mathbf{I} + \mathbf{V})\mathbf{P}]_{1,2} \\
&= \mu_1 \left[\begin{pmatrix} 1 & 0 \\ P_{2,1}(1-\alpha_0) & 1 \end{pmatrix} \begin{pmatrix} 0 & P_{1,2} \\ P_{2,1} & 0 \end{pmatrix} \right]_{1,2} \\
&= \mu_1 \left(\begin{pmatrix} 0 & P_{1,2} \\ P_{2,1} & P_{2,1}(1-\alpha_0)P_{1,2} \end{pmatrix} \right)_{1,2} \\
&= \mu_1 P_{1,2}
\end{aligned}$$

and

$$\begin{aligned} B_{2,1} &= \mathbf{b}^{(2)} \boldsymbol{\beta}^{(1)} [(\mathbf{I} + \mathbf{V}) \mathbf{P}]_{2,1} \\ &= \mu_2 \alpha_0 P_{2,1} \end{aligned}$$

which in turn leads to the second parameter of the compound matrix exponential distribution

$$\mathbf{B} = \begin{pmatrix} -\mu_1 & \mu_1 P_{1,2} \\ \mu_2 \alpha_0 P_{2,1} & -\mu_2 \end{pmatrix}$$

Please note, that the factor α_0 represents the entry vector $\boldsymbol{\beta}^{(1)}$. The result can be verified immediately without the need for tracing the paths through the model. Actually the matrix \mathbf{V} was not able to contribute anything substantial to the second parameter \mathbf{B} of the compound matrix exponential distribution.

The task of assembling layered matrix exponential distribution is best implemented as a computer program. Although the necessary matrix calculations do not exhibit a certain mathematical beauty, they are close to being optimal in terms of required memory and CPU time. It is not advisable to carry out the calculations by hand. Assuming a positive value for $\beta_2^{(2)}$ in the previous example forces the inversion step back into the game and prevents any manual path tracking.

However, the concept of layered matrix exponential distributions enables us to freely choose from any suitable composition of exponential nodes for the purpose of approximation. In case the coefficient of variation, the variance to mean ratio or some higher moment changes we may simply replace the corresponding micro level and repeat the model analysis. There is no need to manually adapt the parameters of the compound matrix exponential distribution or incorporate a complex approximation technique capturing all possible changes of the underlying data.

Representations of the E_2 or C_2 type

We now proceed to introduce the simplest model for a call centre featuring after call work. In the context of the agent state model discussed in section 1.4.1 this means, that we consider states available (idle), active (talking) and ACW only. Furthermore we assume only primary agent states. Accordingly there is only one active and one ACW state. When transformed to the language of queueing this leads to a model with a structure similar to a

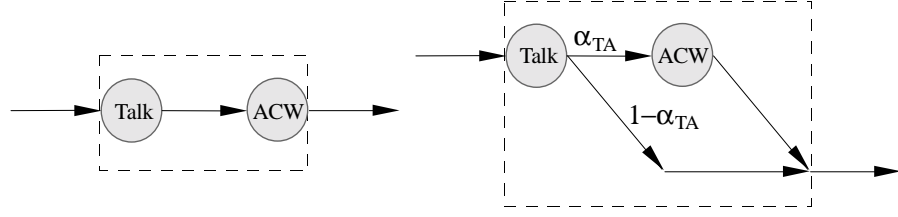


Figure 6.13: Staged approach to the agent occupation time

$M/E_2/c/K$ or $M/C_2/c/K$ queueing system as shown in figure 6.13. Please note, that we still go with the Poisson assumption for the arrivals. If each state is modeled by an exponential service time distribution we are obviously concerned with the models themselves. If not, we may choose to approximate each state distribution by the methods described in chapter 4. By utilizing the method of layered matrix exponential distributions discussed above we are able to embed the corresponding approximation in the macro model of E_2 or C_2 type.

The decision on whether to choose the Erlangian or the Coxian setup depends on the call centre configuration. If the system automatically triggers a transition to the ACW state on call termination, the former is sufficient. If the agent is allowed to skip after call work or the system is able to detect the necessity of it, a corresponding branching probability α_{TA} needs to be introduced into the model. In order to use the model then such behaviour has to be reflected by reporting data as well. As an example consider the report shown in figure 6.5. Assuming exponential service times for each node the data presented by this report are sufficient for the Erlangian scenario but do not provide enough information for the Coxian one. In fact we are missing statistics which enable us to estimate the branching probability α_{TA} . These may either be given in terms of transition counts n_{ij} or state counts n_j observed within a representative time period. For the former we have

$$\hat{\alpha}_{TA} = \frac{n_{TA}}{n_{TA} + n_{TI}}$$

where n_{TI} denotes the number of transitions from talking to idle state. If only state counts are available, one might ignore any temporal effects and use the rough estimator

$$\hat{\alpha}_{TA} = \frac{n_A}{n_T}$$

To capture developments over time we use the common balance argument

$$n_j(t) = \sum_{i=1}^S n_i(t-1) \alpha_{ij} + \varepsilon_j(t)$$

where S is the total number of states and $\varepsilon_j(t)$ is the error amounting for the difference of actual and estimated occurrence of $n_j(t)$. Proceeding further this relation may be formulated as regression problem in matrix form, which allows for the estimation of $\hat{\alpha}_{ij}$. However, not any solution is valid. In fact each $\hat{\alpha}_{ij}$ is an estimator for the branching probability and so we must have $0 \leq \hat{\alpha}_{ij} \leq 1$ and $\sum_{j=1}^S \hat{\alpha}_{ij} = 1$. While the latter can be shown to be satisfied without further assumptions the former needs to be taken care of. As is evident from the notation, these arguments remain valid for more complex models as well. Also note, that these calculations at the macro level are not affected by the choice of the service node approximation provided that the branching probabilities are assumed to be independent from the structure of the service nodes. In practice this will always be the case, if after call work is always performed after the call. If the system offloads work to non-peak times, two separate models for each scenario might be required.

Having completed the model setup a first glance on the performance indicators is provided by the approximation methods for the $M/G/c$ model in section 3.5.3. Provided K is large enough and the squared coefficient of variation c_S^2 is at hand, one is led to immediate results. As we are concerned with matrix exponential distributions, c_S^2 is readily deduced from expression 3.116 as

$$c_S^2 = 2 \frac{\Psi[\mathbf{B}^{-2}]}{\Psi^2[\mathbf{B}^{-1}]} - 1$$

for the agent holding time described by a matrix exponential distribution with representation $ME(\boldsymbol{\beta}, \mathbf{B})$. As an example consider an Erlangian configuration with exponentially distributed talking and ACW times. Then the squared coefficient of variation evaluates to $c_S^2 = \frac{1}{2}$ and the overall queue length becomes the average of the queue lengths of the corresponding $M/M/c$ and $M/D/c$ models.

In order to derive exact results, no closed form solution exists even for the simple configurations featuring exponential node distributions. This has already been indicated in standard text books such as [90]. Accordingly an iterative approach such as the one introduced in the context of the $M/ME/c/N$ queueing system needs to be used.

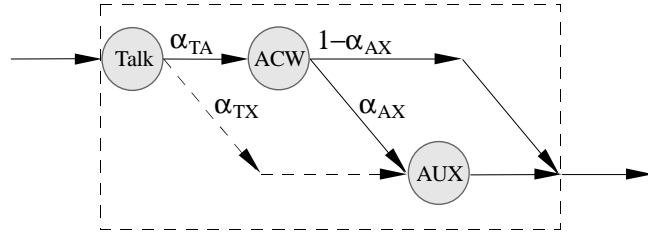


Figure 6.14: Staged configuration with one auxiliary state

Advanced Agent State Models

In adopting an agent perspective we have enhanced the classic model by an additional stage. This has been intuitively appealing due to the fact, that any after call work performed by the agent is directly related to the call and therefore contributes to the agent occupation time. This is obviously not the case for the auxiliary states, which may describe anything from an ordinary break to tasks not related to the call. However, the phase type approach is flexible enough to deal with this situation as well. From an application point of view we simply assume, that any agent state may be treated as if it is related to the call. In most cases this suggestion is also supported by the reporting data available through the management information system (MIS). Whatever state is chosen, the data are often represented in terms of counts or time lengths. From a theoretical point of view we need to shed some light on the matrix exponential description. This will be done shortly in the context of a more complex model.

First consider the staged configuration shown in figure 6.14. Here we have assumed only one auxiliary state, which is treated as being related to the call. By adjusting the branching probabilities α_{TX} and α_{AX} we are able to control the chance of the AUX state being entered during a typical work cycle. When does it make sense to include the auxiliary state? This depends on the modeling requirements. In order to derive an appropriate estimate for the number of agents to be staffed in a specific time period, any contribution to the work cycle needs to be captured. In most countries breaks of defined length and frequency are enforced by local government. Although not productive these breaks are equivalent to the work performed from a modeling perspective. Accordingly the effect of these breaks on the key performance indicators is captured by the model.

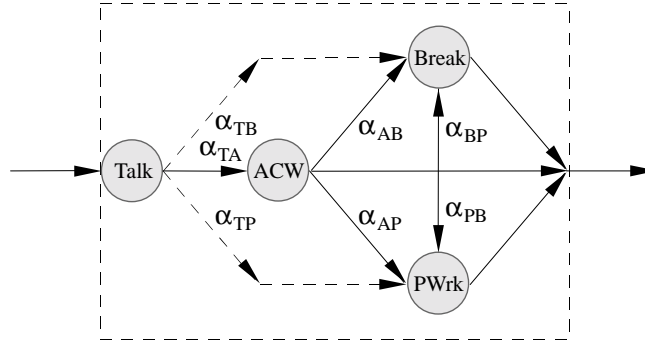


Figure 6.15: Staged configuration with two auxiliary states

Please note, that for mandatory after call work the branching probability $\alpha_{TX} = 0$. We have decided to use a dashed line instead of a solid one to emphasize the common nature of this specific feature.

It might be tempting to include the LOGOFF periods as well to capture even more effects such as training times and illness periods. One has to be careful here, because usually the LOGOFF times are not tracked by the MIS. Instead they form the difference between the length of the measurement interval and the aggregated occupation time of all states. Insofar the LOGOFF periods violate the independence assumption. In bringing in business related data provided by an external source this dependency could be avoided. Examples include average absence statistics and estimates of training frequency and length.

Now consider the staged configuration shown in figure 6.15 featuring two auxiliary states. Whereas BREAK is self-explanatory, the state PWRK describes any paperwork not related to a specific call. Examples might include letter orders and written complaints.

Turning attention to the probabilistic description of the matrix exponential distribution related to the graph shown in figure 6.15, we first deduce the transition matrix

$$\mathbf{P} = \begin{pmatrix} 0 & \alpha_{TA} & \alpha_{TB} & \alpha_{TP} \\ 0 & 0 & \alpha_{AB} & \alpha_{AP} \\ 0 & 0 & 0 & \alpha_{BP} \\ 0 & 0 & \alpha_{PB} & 0 \end{pmatrix}$$

Setting $\mathbf{M} = \text{diag}(\mu_T, \mu_A, \mu_B, \mu_P)$ we immediately obtain the service rate

matrix

$$\mathbf{B} = \mathbf{M}(\mathbf{P} - \mathbf{I}) = \begin{pmatrix} -\mu_T & \alpha_{TA}\mu_T & \alpha_{TB}\mu_T & \alpha_{TP}\mu_T \\ 0 & -\mu_A & \alpha_{AB}\mu_A & \alpha_{AP}\mu_A \\ 0 & 0 & -\mu_B & \alpha_{BP}\mu_B \\ 0 & 0 & \alpha_{PB}\mu_P & -\mu_P \end{pmatrix}$$

and the exit vector

$$\begin{aligned} \mathbf{b} &= -\mathbf{B}\mathbf{1} = (\alpha_T, \alpha_A, \alpha_B, \alpha_P) \\ &= (0, (1 - \alpha_{AB} - \alpha_{AP})\mu_A, (1 - \alpha_{BP})\mu_B, (1 - \alpha_{PB})\mu_P) \end{aligned}$$

As indicated by figure 6.15, we have assumed that $\alpha_{TA} + \alpha_{TB} + \alpha_{TP} = 1$, i.e. there is no chance for absorption from the talking state. In suggesting to introduce a dependency between the auxiliary states and the call we have implicitly switched to an approximation. Obviously this approximation will perform quite well, if call arrivals dominate the remaining demands. Expressed in mathematical terms this condition becomes

$$\lambda = \lambda_T \gg \lambda_B + \lambda_P \quad (6.18)$$

We now think of the auxiliary states as being triggered by a call arrival as part of a single work cycle and set $\lambda_B = \lambda_P = 0$. Accordingly the effect caused by arrivals to the BREAK and PWRK states has to be captured by the corresponding branching probabilities. For a system featuring mandatory after call work their estimates are given by

$$\hat{\alpha}_{AB} = \frac{n_B}{n_A}, \quad \hat{\alpha}_{AP} = \frac{n_P}{n_A}$$

where n_B , n_P , n_A and n_T are the visit counts to states BREAK, PWRK, ACW and TALK. Please note, that for mandatory after call work we have $n_A = n_T$. As we only deal with call arrivals the entry vector assumes the following simple form

$$\boldsymbol{\beta} = (\beta_T, 0, \beta_B, \beta_P) = (1, 0, 0, 0)$$

In a sufficiently loaded call centre operation the above model is a suitable choice, which pays regard to the impact of breaks or minor working tasks. But this is not always the case. Most contact centre installations process several channels and media types each with considerable demand. Condition

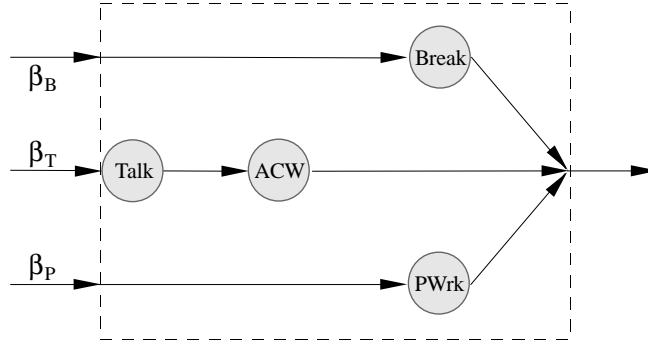


Figure 6.16: Staged configuration with isolated auxiliary states

6.18 ceases to hold and so we have to find another alternative. Fortunately this can be done by introducing some minor modifications to the model.

Assuming Poisson arrivals for the auxiliary states as well we are now concerned with an overall arrival rate

$$\lambda = \lambda_T + \lambda_B + \lambda_P$$

The corresponding entry vector becomes

$$\boldsymbol{\beta} = (\beta_T, 0, \beta_B, \beta_P)$$

where $\beta_T, \beta_B, \beta_P > 0$. By setting $\alpha_{TB} = \alpha_{TP} = \alpha_{AB} = \alpha_{AP} = 0$ we are led to a representation as the one shown in figure 6.16. For illustrational purposes we have also included the entry probabilities. The problem of parameter estimation has been shifted from the branching probabilities to the entry probability vector. Rough estimates are provided by

$$\begin{aligned} \hat{\beta}_T &= \frac{n_T}{n_T + n_B + n_P} \\ \hat{\beta}_B &= \frac{n_B}{n_T + n_B + n_P} \\ \hat{\beta}_P &= \frac{n_P}{n_T + n_B + n_P} \end{aligned}$$

We have implicitly assumed the availability of state counts from the MIS data. Otherwise one may employ one of the methods discussed in the context of E_2 and C_2 representations.

So far we have presented a method for systematic analysis of call centre configurations from an agent perspective. Although the phase type approach is well suited the suggested techniques may be adapted to be applied in the framework of Markovian state analysis. The latter may become necessary when considering models with separate queues and/or priority disciplines [191]. In some cases the more general QBD approach leads to results as has been exercised in [188].

As before the service times for exponential nodes are easily estimated from the data means, while non-exponential nodes may be approximated by one of the methods of chapter 4. The resulting matrix exponential distribution can be embedded into the macro model by utilizing the concept of layered matrix exponential distributions as suggested in this chapter.

Timed After Call Work

We will now provide some information on how to match a deterministic distribution by using the fixed node approximation method suggested in section 4.3.4. This is especially useful for modeling timed after call work and will also be used in the next section. It is always possible to use a brute force approach and calculate the first three moments from a suitable generating function or transform of the deterministic density function. However, this requires some advanced methods and so we choose an indirect path. First note, that the moments of the normal distribution $\mathcal{N}(\zeta, \sigma^2)$ with mean ζ and variance σ^2 are given by

$$\begin{aligned}\mu_{\mathcal{N}}^{(1)} &= \zeta \\ \mu_{\mathcal{N}}^{(2)} &= \zeta^2 + \sigma^2 \\ \mu_{\mathcal{N}}^{(3)} &= 3\zeta\sigma^2 + \zeta^3\end{aligned}$$

In accepting a value of 0 for the variance, we arrive at a deterministic distribution with mean $\zeta := s$ and moments

$$\begin{aligned}\mu_{\mathcal{N}}^{(1)} &= \zeta = s \\ \mu_{\mathcal{N}}^{(2)} &= \zeta^2 = s^2 \\ \mu_{\mathcal{N}}^{(3)} &= \zeta^3 = s^3\end{aligned}$$

where s denotes the service time of the corresponding facility. Using this moments $\mu_{\mathcal{N}}^{(r)}$ instead of $\mu_F^{(r)}$ for $r = 1, 2, 3$ in the system of equations 4.4 and

expressing the result in terms of the parameters yields

$$\begin{aligned}\hat{\alpha}_1 &= \frac{2(6-6-1)s^3}{(3-1)s^3} = -1 \\ \hat{\alpha}_2 &= \frac{2(18+15-9-18-1)s^6}{(12-9+1)s^3(3-1)s^3} = \frac{5}{4} \\ \hat{\mu} &= \frac{(1-3)s^3}{(2-3)s^4} = \frac{2}{s}\end{aligned}$$

The conditions for a valid approximation are satisfied, because $\hat{\mu} > 0$ and

$$\mu_{\mathcal{N}}^{(1)}\mu_{\mathcal{N}}^{(3)} = s^4 < \frac{3}{2}s^4 = \frac{3}{2}\left(\mu_{\mathcal{N}}^{(2)}\right)^2$$

for $s > 0$. From the configuration shown in figure 4.3 we may deduce the matrix exponential components immediately, i.e.

$$\begin{aligned}\beta &= (1, 0) \\ \mathbf{P} &= \begin{pmatrix} 0 & 1 - \hat{\alpha}_1 \\ 1 - \hat{\alpha}_2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ -\frac{1}{4} & 0 \end{pmatrix} \\ \mu &= (\hat{\mu}, \hat{\mu}) = \left(\frac{2}{s}, \frac{2}{s}\right)\end{aligned}$$

Obviously we are dealing with an improper transition matrix \mathbf{P} , because its elements do not belong to the interval $[0, 1]$. This is beyond the context spanned by the classic theory of phase type distributions. Fortunately we are using a matrix exponential framework, which is capable to cope with such a configuration.

6.5.5 Comparison

In call centre performance engineering we often meet the $M/M/c$ queueing and the $M/G/c/c$ loss model or their relatives Erlang delay and Erlang loss formula. Although not suitable in any case, they are still widely applied in practice. Unfortunately we still find them stated as the one-and-only tool in performance engineering trainings held for call centre managers. In this section we aim to compare the more general $M/M/c/K$ model with its matrix exponential counterpart. For all systems we assume a configuration with c

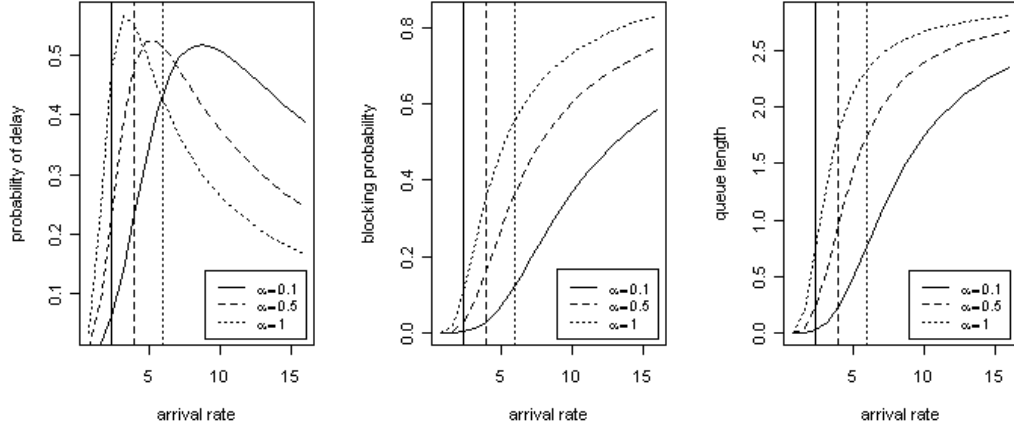


Figure 6.17: Impact of α_{TA} on delay probability, blocking probability and queue length

agents and K trunks. We have chosen the $M/M/c/K$ model rather than the so common combination of $M/M/c$ and $M/M/K/K$ system, because the latter configuration omits the dependency between wait and loss. Such a decoupling leads to severe errors for all estimates in heavy and overload traffic situations. The second idea commonly met in practice is motivated by the memoryless nature of simple queueing systems, i.e. the ability to sum the service time parameters of separate states. One application is the inclusion of the after call work time as part of the overall service time. So we aim to provide an impact analysis on this issue as well. For simplicity and not to clutter the graphs too much, we have decided to use E_2 representations for the majority of matrix exponential models presented here. For a thorough understanding we also need to get an idea about the impact caused by the branching probability α_{TA} in a C_2 like configuration. In practical terms we are discussing the impact of optional after call work. As indicated in figure 6.13, the E_2 representation relates to the case of mandatory after call work (ACW). For the system under consideration we have assumed, that on the average ACW requires twice the time of a typical call. Iterating over a wide range of arrival rates, we created the graphs shown in figure

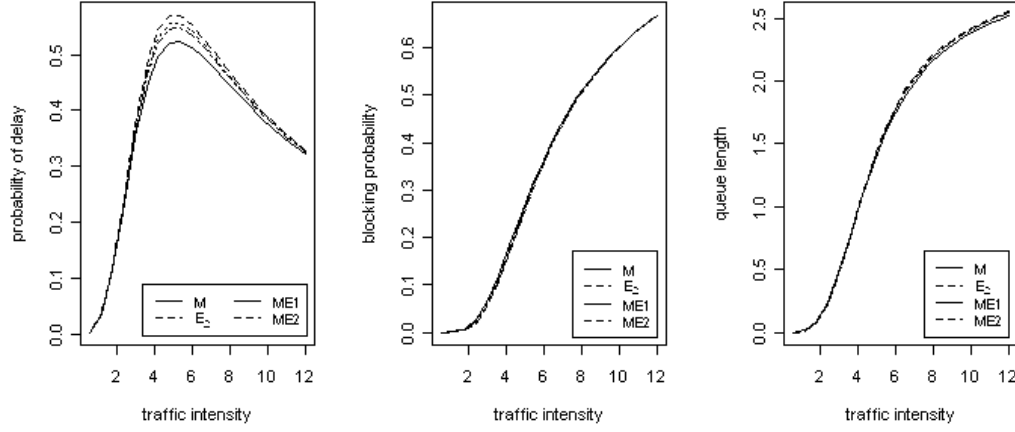


Figure 6.18: Effect of ACW on delay probability, blocking probability and queue length

6.17. As might have been expected, the sojourn time of the service facility increases with raising values for α_{TA} . As a consequence the probability of delay, blocking probability and queue length curves are shifted to the right, i.e. the related performance indicators experience an increase as well. The attentive reader might have noticed the vertical lines in each graph. These are the demarkation points for the boundary between heavy and overload traffic, i.e. for a utilization $u = 1$. For practical concerns we can safely assume, that α_{TA} is a probability and so we may conclude, that the E_2 representation forms a natural boundary for the more general C_2 type configurations.

For the remaining examples we will switch back to the traffic intensity on the x-axis. Turning attention to the effect caused by an introduction of after call work, we first assume all nodes to follow an exponential distribution with overall average service time $s = 0.5$. The resulting graph is shown in figure 6.18. The solid curves correspond to the Markovian case with $\mu = 2$ and $c_S^2 = 1$, whereas for E_2 we have chosen the parameters $\mu_1 = \mu_2 = 2\mu = 4$ with $c_S^2 = 0.5$. Additionally we have provided curves for the hypoexponential distributions with parameters $\mu_1 = 5\mu = 10$, $\mu_2 = (1 - \frac{1}{5})^{-1}\mu = 1.25\mu = 2.5$ denoted by *ME1* and parameters $\mu_1 = 1.142857\mu$, $\mu_2 = 8\mu = 16$ denoted by *ME2*. The corresponding squared coefficients of variation are $c_S^2 = 0.68$

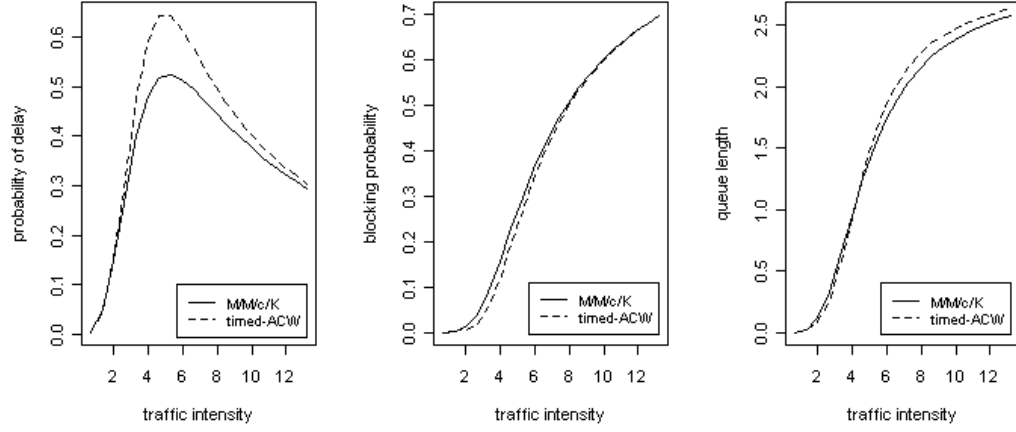


Figure 6.19: Effect of timed ACW on delay probability, blocking probability and queue length

and $c_S^2 = 0.78125$ for $ME1$ and $ME2$, respectively. As indicated by c_S^2 , the extremes are found to be the curves for M and E_2 . Further tests for a wide range of values for K lead to similar pictures. All curves stick close together, suggesting a certain robustness for exponentially distributed talking and ACW times. However, taking this robustness for granted might lead to unexpected results. For the next example we consider exponential talking times and deterministic ACW times in an E_2 setting. The corresponding rates are given by $\mu_T = 5$ and $\mu_A = 0.5$. From an applied point of view, we are discussing the effects of timed ACW. Using the fixed node approximation suggested in section 4.3.4, we arrive at a compound service facility with mean sojourn time of 2.2 and squared coefficient of variation $c_S^2 = 0.008264463$. The resulting graph is shown in figure 6.19. Although the curves still exhibit a certain similarity, deviation is more significant than for the previous example. Simply adding the service time parameters of subsequent states may indeed lead to significant errors.

For the above examples the squared coefficient of variation provided an indication for the applicability of the $M/M/c/K$ queue as an approximation to a more complex matrix exponential system. It appears, that based on our judgement of the squared coefficient of variation it is sufficient to supply

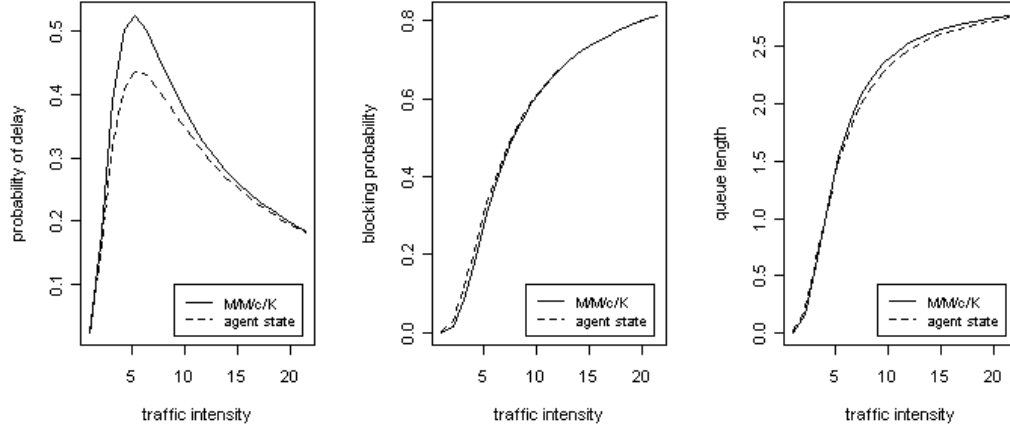


Figure 6.20: A more complex example

the mean sojourn time of the matrix exponential service distribution to the common $M/M/c/K$ model. This hypothesis will now be validated for a fancier setup than the ones we have used so far. Consider the model shown in figure 6.16 and parametrized as follows

$$\beta = (\beta_T, \beta_A, \beta_B, \beta_P) = \left(\frac{7}{10}, 0, \frac{1}{10}, \frac{2}{10} \right)$$

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mu = (\mu_T, \mu_A, \mu_B, \mu_P) = \left(10, 1, \frac{1}{10}, 10 \right)$$

After having accounted for all internal transitions, we end up with agents talking 70%, performing paperwork 20% and taking a break 10% of their total staffed time. The overall service time distribution now features a mean sojourn time of 1.79 and a squared coefficient of variation $c_S^2 = 5.728254$. Obviously our artificial choice of service times has led to large value for c_S^2 . Taught by experience we expect the $M/M/c/K$ model to provide an

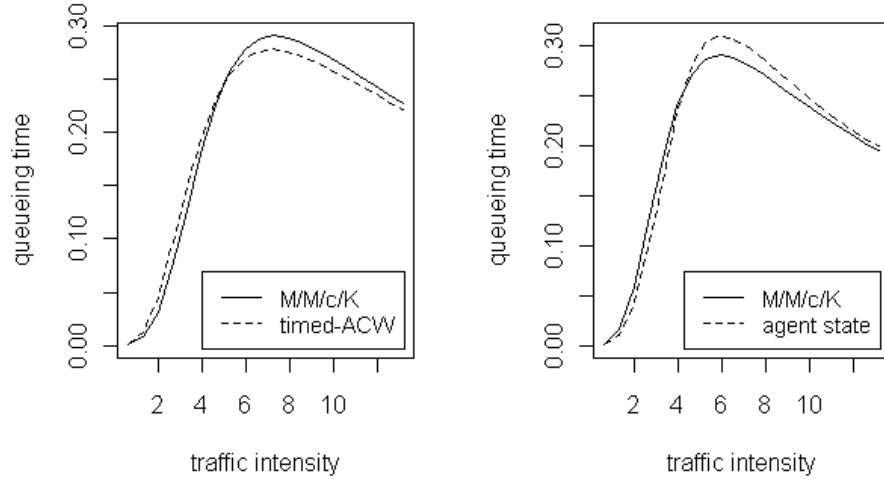


Figure 6.21: Waiting times for matrix exponential and Markovian models

acceptable approximation. The result is shown in figure 6.20. So we have to conclude, that even when taking into account the effects caused by auxiliary states the $M/M/c/K$ approximation provides an acceptable choice.

Before getting too enthusiastic, we should adopt a more pedantic view for two reasons. First, our choice of scale might have been too coarse. So some details could have been missed from the graphs. Second, up to now we only looked at the queue length, not the waiting time. Especially for an arrival rate $\lambda < 1$ the deviation in waiting time becomes a magnified version of the queue length error caused by using a $M/M/c/K$ approximation. This can be easily seen from Little's law. So we have to expect larger relative errors for the waiting time in lightly loaded systems. For heavy loaded systems the $M/M/c/K$ approximation proves to be more robust. This also agrees with what we might expect from heavy traffic theory [124][30]. To remedy both deficiencies, we have provided graphs for the waiting time of the timed-ACW and the advanced agent state example in figure 6.21. By close inspection we are able to detect a significant deviation of the $M/M/c/K$ approximation from the exact model. In fact, for some points the relative error is larger than 20% in both examples. This effect is mainly caused by the steep ascent

of the waiting time curves in a lightly loaded region.

From the above examples we have to conclude, that the $M/M/c/K$ approximation provides a reasonable approximation for blocking probability and queue length on a coarse scale, for heavy traffic and overload situations. In case of a lightly loaded system, we experience significant deviations especially for waiting time related performance indicators. The $M/M/c/K$ approximation fails and the exact matrix exponential model should be used. This will also avoid errors caused by the systematic omission of structural information.

Furthermore the relation between delay and blocking probability exhibited in all graphs explains the failure of the separated modeling approach based on $M/M/c$ and $M/G/K/K$ queueing systems for assessment of split group size and trunk capacity. As a matter of fact, it is the relation between delay and blocking probability, which has a significant impact on the results. This also justifies our choice of the $M/ME/c/K$ model for a simultaneous assessment of split group and trunk sizes.

6.5.6 Extensions and Open Issues

Although the theory of matrix exponential queueing systems is young compared to other representatives in the field, a broad spectrum of models has already been worked out. In this section we briefly discuss the potential of these extensions, their pros and contras. The presented selection is a rather subjective one based on the author's experience and attitude.

- One extension would be the inclusion of multiple priority classes into matrix exponential queueing models. These priority classes would directly relate to different classes of calls, e.g. bronze, silver and gold calls. As pointed out in section 6.4.2, the use of call priorities requires a lot of care and very likely leads to adverse effects in overall call centre performance. This has been the main reason for omitting queueing models with priorities.
- Due to space limitations we have left out the incorporation of multiple classes into matrix exponential queueing models for further study. This extension would be extremely beneficial for contact centre models as it would allow for decoupling and explicit modeling of tasks and media streams. As an example consider email, fax and voice call handling.

- It would be very desirable to have the overflow process of a general $M/ME/c/K$ queueing system described in terms of a matrix exponential distribution. Unfortunately this does not work out. Recalling from section 6.5.2, that even for the simple $M/M/c/K$ queue the overflow traffic is not of renewal type, we have to specify the entire overflow process rather than its renewal distribution. However, we still can apply the methods introduced there to find some suitable approximations. For call centre purposes it might be more fruitful to combine originating and target split group into a single matrix exponential or Markovian model. This avoids the specification of a complex non-renewal overflow process. But one has to be very careful about how to assess the overflow condition. If stated in terms of the queue length, we still work in the domain of an ordinary queueing system capacity limit. If expressed as a waiting time condition, the translation into a queue length description by use of Little's law may fail, if there is high fluctuation in the arrival rate. Furthermore the structure of the model now depends on the arrival rate. If automated by a computer program, the assembly of the phase configuration has to occur each time the mean arrival rate for the busy hour changes.
- Another possible extension to the agent state models suggested above is the introduction of retrials similar to the model presented in section 3.8.6. This has been left for further study, because some efforts are still necessary to adapt the model to a matrix exponential framework. However, the calculation may be simplified tremendously by proceeding as follows. Given a system with c agents and $K > c$ trunks, we first use the classical orbit model to derive the retrial arrival rate r and then use the compound arrival rate $\lambda + r$ as an input to the $M/ME/c/K$ queueing model. Recalling results from section 3.4.3, the retrial arrival rate is calculated from $E\left(\frac{\lambda+r}{\mu}, K\right) = \frac{r}{\lambda+r}$, where $E(.,.)$ denotes the Erlang loss formula and μ^{-1} is the average talking time of an agent. The main difficulty lies in the estimation of the arrival rate λ , as all retried calls have to be excluded from the underlying sample. Although this is only an approximation, it has an appealing interpretation. It enables us to determine trunk capacities based on a given service level and estimated arrival, retrial and service rate followed by an assessment of agent resources from a given service level and further estimated rates for the agent states. Following our terminology introduced at the be-

ginning of this chapter, we construct an overlay of separate perspectives to describe a complex configuration. In our case we are concerned with trunk and agent view.

- In some cases it might be fruitful to include the call flow as part of the arrival process, which results in the more general $ME/ME/c/K$ queue. With some care the methods suggested for the quasi birth-death process in section 3.8.4 may be applied. In accordance with section 6.3 we have to note, that most components of the call flow are of deterministic nature. Insofar we have to make heavy use of fixed node approximation techniques to find a tractable representation for the arrival process. In a similar way we may also tackle the truncated service time distributions used to model the interdigit timeouts. However, due to the differences in shape of the original distribution and its approximation one might expect significant deviations in the resulting performance indicators. Due to space limitations such robustness considerations have been left for further study.

At this point we have to mention the remarkable work [188] of D.A. Stanford and W.K. Grassmann concerned with bilingual call centres. Although this is a very special application, it often occurs in common call centre installations. The authors provide a solution procedure, waiting time distribution and average delay together with numerical examples. The method of solution is based on matrix geometric techniques as introduced by M. Neuts in [153].

6.6 Remarks and Applications

The major task of this chapter was to present a collection of methods for the calculation of call centre performance indicators in terms of queueing theoretic results. For the genuine matrix exponential approach we also presented a comparison to commonly applied models leading to statements of strength and weakness. With all these results at hand one might ask how they fit into a larger operational framework.

A major application will be found in the context of workforce management. Roughly speaking, the central target of workforce management is to have the right number of agents answer a forecasted volume of calls at the desired service level. Using given values for some performance indicators, the number of agents of a split group and the required trunk capacities may

be calculated. In some cases derivations could not be reversed, but applying an iterative approach may lead to the desired results. With all computations carried out in pseudo real time and by allowing for input parameter adaption a call centre manager may easily create scenarios and compare the predicted results. Introducing adaptations to the original model will lead to a tuned algorithm for prediction of workload. The forecasted traffic volumes may then serve as a framework for shift scheduling. It has been an accepted strategy to let the agents decide about their assignment to a shift. Practice has proven this to be a flexible and efficient way to incorporate the complex organizational requirements into shift planning.

In our description of workforce management we mentioned that agent resources shall be provisioned at a desired service level. The service level is a metric measuring the quality of service in a call or contact centre. Unfortunately its definition is not unique. The most common one is given by the percentage of callers, which have to wait no longer than a predefined waiting time threshold called *acceptable waiting time (AWT)*. Obviously it does not account for the amount of waiting time experienced by the caller provided he or she has already waited beyond the AWT. This in turn may lead to adverse management decisions, which bear a further increase of this excessive waiting time. To avoid these drawbacks an improved service level concept has been introduced by G. Koole in his paper [130]. He discussed a waiting time metric called the average excess (AE), which leads to the following definition for the service level

$$\mathbb{E}(\check{W}_q - AWT) = \frac{p_d e^{-(c\mu - \lambda)AWT}}{c\mu - \lambda}$$

where p_d is the probability of delay 3.24 encountered in the $M/M/c$ queue. Its simple form mainly rests on the memoryless property of the exponential distribution.

The process of determining an appropriate number of agents or workforce is also called *staffing*. Several approximations have been developed for this problem. Best known is the so called *square root staffing principle*, which has been used as rule of thumb by call centre managers for years. More recently it has been treated rigorously by M. Reimann, A. Mandelbaum, W. Whitt and others, e.g. see [48]. In its simplest form the square root staffing principle is given by

$$c = \rho + k\sqrt{\rho} \tag{6.19}$$

where ρ is the offered load and $k\sqrt{\rho}$ is the *safety staffing*. The value for k is determined by operational requirements. For example, one may use a positive function of the ratio of staffing and delay cost [129]. In accepting the $M/M/c$ queue as a framework, the factor k may also be derived from thresholds for common performance indicators. Assuming an objective related to the waiting probability one proceeds as follows. First note, that the Erlang loss formula 3.28 may be stated in terms of the Poisson distribution, i.e.

$$p_b = \frac{\Pr \left\{ \check{X} = c \right\}}{\Pr \left\{ \check{X} \leq c \right\}}$$

with \check{X} denoting the corresponding random variable. Substituting expression 6.19 and applying the normal approximation to the Poisson distribution leads to [201]

$$p_b = \frac{1}{\sqrt{\rho}} \frac{\varphi(k)}{\Phi(k)}$$

where $\varphi(k)$ and $\Phi(k)$ denote density and distribution function of the normal distribution. Noting the formal relation between Erlang delay formula 3.24 and Erlang loss formula 3.28, the probability of delay may be calculated as follows

$$p_d = \frac{cp_b}{c - \rho + \rho p_b} \approx \frac{(\rho + k\sqrt{\rho}) \frac{\varphi(k)}{\Phi(k)}}{\rho \left(k + \frac{\varphi(k)}{\Phi(k)} \right)}$$

For large values of ρ we may safely ignore the term $k\sqrt{\rho}$, which leads to

$$p_d \approx \frac{\frac{\varphi(k)}{\Phi(k)}}{k + \frac{\varphi(k)}{\Phi(k)}} = \left(1 + \frac{k\varphi(k)}{\Phi(k)} \right)^{-1}$$

Provided the probability of delay is bounded by a value α to ensure quality of service for the split group under consideration, the value of k is given by the solution of

$$\alpha = \left(1 + \frac{k\varphi(k)}{\Phi(k)} \right)^{-1}$$

The square root staffing principle has been shown to be robust even in presence of abandonments and for the more general $M/G/c$ queueing model for square integrable service times [129]. However, we still remain in the realm

of infinite queueing models and so we cannot assess the number of agents and the trunk capacity in a single step. This may lead to an overestimation of split or skill group sizes in presence of severely limited trunk resources. One may argue, that the determination of trunk sizes belongs to operational planning rather than workforce management. This is indeed true, but the previous section has shown stringent dependencies to exist. When omitted, this may lead to corrupted results.

Broadening context, we encounter another application. Operational planning is concerned with the efficient allocation of resources leading to a maximization of revenue. Targets are best formulated by the use of objective functions, which may differ with respect to organization or application. For example consider a freephone number. The overall trunk cost, which are in turn assembled from the trunk size K and the average cost per trunk port, contribute negatively to the revenue. This is different for value added services, where trunk usage causes returns. Any performance indicator discussed in this chapter may readily become part of an objective function, which allows the call centre management to monitor their economical and organizational objectives. For a detailed exposition on operational planning please refer to the book [191] by R. Stollatz.

Chapter 7

Call Centre Solutions in Public Networks

The concepts behind the *intelligent network (IN)* changed the way we think of the *public switched telephone network (PSTN)*. A service based view has been introduced to the almost technical driven networks. Due to the ongoing convergence process of voice, data, wireline and wireless networks, the IN shall be seen more as a concept of network intelligence than a physical network. Finally, convergence will result in a broader view of the basic IN approach to service control. Consequently, the aim of this chapter is to review IN concepts with respect to call centre applications. Several solutions are developed based on their level of integration. Starting with standard pre-routing as used in existing call centres today, a path up to the tightest coupling possible, the application specific *service control point (SCP)* will be established.

7.1 The Classic Public Network

Section 5.2.2 introduced the ISDN protocol as the user interface of the public network, more accurate the integrated services digital network. Please note, that the abbreviation ISDN is interchangeably used for the protocol and the entire network itself. Whereas the ISDN protocol resides at the *user-network interface (UNI)*, SS7 is defined for the *network-network interface (NNI)*.

Figure 7.1 presents a simplified view of the relation between components of the integrated services digital network. Please note, that signaling infor-

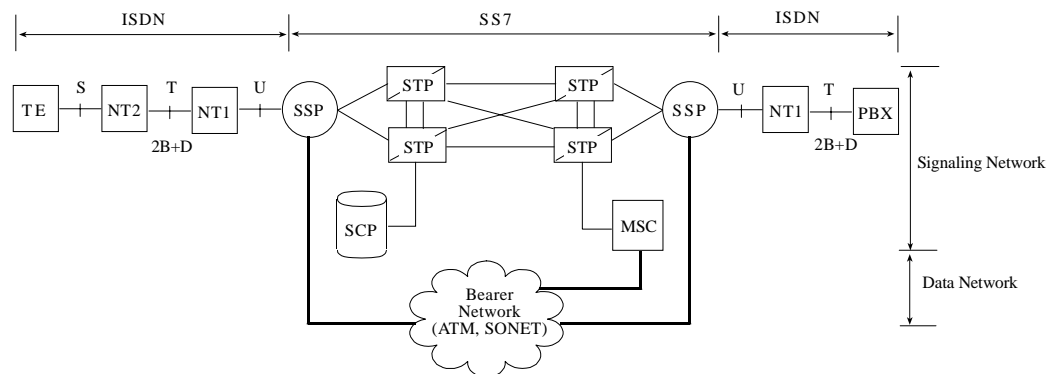


Figure 7.1: Simplified view of the ISDN

mation and user data may be conveyed on totally separate links within the public network.

7.2 Common Channel Signaling

This section shall provide an introductory treatment of SS7. SS7 provides transport and transaction services for IN. The following items shall be reviewed

- Functional entities - SP, SSP and SCP
- Protocols - MTP1-3 and SCCP
- Application parts - TCAP
- User parts - ISUP

7.3 The Intelligent Network

This section shall provide an introduction to the concepts and architecture of the Intelligent Network (IN). Covered items will include

- Conceptual model
- Planes and relation between (Global functional plane to distributed functional plane, SIB's and operations, information flows and INAP protocol messages)
- Capability sets

7.4 IN and TCP/IP (optional)

In this section, interfaces to the IP world shall be reviewed. Potential candidates include

- TALI Interface (RFC) and relatives - Delivery of IN services through IP networks
- Approaches for handling H.323 based IN calls - proposed extensions to the IN call models
- Gateways to the Public Network - MGCP and relatives, their impact on IN

7.5 Prerouting Revisited

This section shall review the current state of the art in building network based call centres. A basic overview has already been provided in section 5.3. As IN related aspects have not been covered there, a more thorough treatment is presented here. Obviously, pre-routing architectures provide a good starting point for the development of a purely IN based call distribution service.

7.6 Prerouting Extended

The minimalistic approach of prerouting shall be relaxed. The aim of this section is to review the interface between adjunct (AD) and SCP and to reveal more functionality by approaching a tighter coupling at that interface. As before for standard prerouting, non call related information such as agent states are still conveyed over IP.

7.7 IN based Call Handling and Distribution

Based on the experience of the preceding chapters, an attempt to convey context specific information using the telephone network shall be made. The target is to investigate the feasibility of such an approach. It shall be determined, if the standards provide sufficient support or if extensions have to be proposed. In case of extensions, other proprietary alternatives shall be considered as well. Questions to be answered include

- Based on the agent state model presented in section 1.4.1, what is the best way to incorporate an agent state request / response scheme in the intelligent network ?
- What approach can be undertaken to provide outbound functionality ?
- Agent and call selection describe two scenarios under agent and call surplus. How do these impact design ?
- Which open issues will remain, e.g. operation of mixed VOIP and classical call centres, ACD-to-ACD communication over gateways

7.8 Performance Analysis

In this section, the impact of each solution presented on performance shall be derived. These performance values may allow for future comparison with less centralized approaches like PBX based distributed call centres. Possible scenarios include

- Impact of INAP link traffic

- SSP load analysis

Appendix A

Stochastic Processes

A.1 Introduction

If one wishes to model real world phenomena and gain deeper insight into them, an adequate set of mathematical and probabilistic tools is required. One such tool is the theory of stochastic processes concerned with the abstraction of empirical processes. Examples include the flows of events in time and evolutionary models in biological science. Associated with the concept of a stochastic process are the *state space* and the *parameter space*. Whereas the latter is often identified with time, the former contains all values the process can assume. If the parameter space is not limited to time only, i.e. possesses a higher dimension, stochastic processes are also called *random fields*. In the following discussion, the parameter space is limited to the one dimensional case and treated as time.

Definition 35 *Let T denote the parameter space. A stochastic process is a family of random variables $\{X(t) : t \in T\}$ defined on the same probability space.*

Definition 36 *A stochastic chain is a stochastic process with countable (discrete) state space. It will be denoted by $\{X_t : t \in T\}$.*

As stated above, it is very important, that each random variable $X(t)$ assumes the same state space. Flipping a coin and throwing a dice are independent experiments. Their combination alone does not constitute a stochastic process.

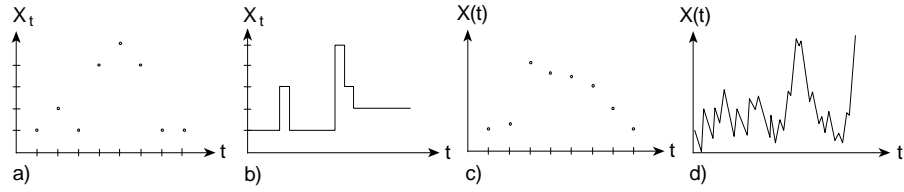


Figure A.1: Examples of stochastic processes

Figure A.1 shows examples for each combination of state and parameter space. Notation has been chosen according to above definitions specifying two stochastic chains on the left and the more general processes with continuous state space on the right. These graphs are called sample graphs, because they reflect one possible realization of a stochastic process over time. By keeping the time fixed and considering the possible realizations one arrives at another view, that is the process as a random variable for a certain point in time. These dualistic perspectives are central to the theory of stochastic processes and give rise to concepts such as *ergodicity*. Ergodicity deals with the problem of determining measures for stochastic processes. For example, the ergodic theorem states, that under certain conditions, the time average equals the ensemble average (almost sure). The time average is deduced from a single realization over infinite time and the ensemble average is the mean over all possible realizations for a certain point in time.

A central concept to the theory of stochastic processes is stationarity. It releases the requirement of time dependence allowing for a steady state view of certain processes.

Definition 37 A stochastic process $X(t)$ will be called stationary, if its joint distributions are left unchanged by shifts in time, i.e. $(X(t_1), \dots, X(t_n))$ and $(X(t_1 + h), \dots, X(t_n + h))$ have the same distribution for all h and t_1, \dots, t_n .

The general discussion will end here and we will turn to examples important to queueing theory. Proceeding further would require an introduction to measure theory. For readable accounts refer to [115][116][58][102]. The standard reference in the field is [64].

A.2 Markov Processes

General stochastic processes may exhibit a complicated dependence structure. By restricting the dependence of the future to the present, not allowing for any influence from the past, one arrives at what is called a *Markov process*. At first sight such a restriction seems to be a serious one, but this is not necessarily the case. Considering effects of births and deaths to determine tomorrow's distribution of a population is only one example. Compared to general stochastic processes, Markov processes are mathematically more tractable. Furthermore they may serve as approximations to more elaborate models.

Definition 38 *A stochastic process $\{X(t) : t \in T\}$ is called Markov process, if for any set of n time points $t_1 < \dots < t_n$ the conditional distribution of $X(t_n)$, given the values $X(t_1), \dots, X(t_{n-1})$, depends only on $X(t_{n-1})$, i.e. for any x_1, \dots, x_n*

$$\begin{aligned} & \Pr \{X(t_n) \leq x_n | X(t_1) = x_1, \dots, X(t_{n-1}) = x_{n-1}\} \\ &= \Pr \{X(t_n) \leq x_n | X(t_{n-1}) = x_{n-1}\} \end{aligned} \quad (\text{A.1})$$

According to the type of parameter space, Markov processes are classified in *discrete parameter Markov processes* and *continuous parameter Markov processes*. They are determined by the *transition probabilities* $P(s, t, x, A)$ and an initial distribution. Here $P(s, t, x, A)$ describes the probability of the transition from state x to a state $y \in A$ within time $|t - s|$. In case A is finite or at most countable, it can be calculated by

$$P(s, t, x, A) = \sum_{y \in A} \Pr \{X(t) = y | X(s) = x\} \quad (\text{A.2})$$

If the transition probability depends only on the difference of s and t , i.e. $P(s, t, x, A) = P(|t - s|, x, A)$, the Markov process is called a *(time-)homogenous Markov process*. The case of discrete parameter space will be discussed in more detail in the next section. The literature focusing on Markov processes is highly based on measure theory and functional analytic concepts, one classical reference is [174].

A.3 Markov Chains

Markov chains provide the means to calculate limiting distributions under relatively mild conditions. As such they find wide applications in modeling real world phenomena met in engineering and science. The ease in calculation is reached by restricting the state space.

Definition 39 *A Markov process with countable (discrete) state space is called Markov chain. Alternatively it can be seen as stochastic chain, which satisfies equation A.1.*

Markov chains are classified by their parameter space thus appearing either as *discrete time* or *continuous time* variants. Based on the definition for Markov processes, a Markov chain will be called *(time-)homogenous*, if the transition probabilities do not depend on time. In the following sections only homogenous Markov chains will be discussed.

A.3.1 Homogenous Markov Chains in Discrete Time

As both state space S and parameter space T are now discrete, the transition probabilities given in A.2 may be simplified to $p_{ij} = \Pr \{X_n = j | X_{n-1} = i\}$. These probabilities may be assembled to the *matrix of transition probabilities of stage 1*:

$$\mathbf{P} = (p_{ij})_{i,j \in T} = \begin{pmatrix} p_{11} & p_{12} & \cdots \\ p_{21} & p_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Please note, that the term matrix is used in the wide sense. It includes the possibility of infinite dimension. A short introduction to the algebra of denumerable matrices is given in [119]. The probability of reaching state j starting from state i within m steps is denoted by $p_{ij}^{(m)}$ and calculated using the *Chapman Kolmogorov equations*

$$p_{ij}^{(m)} = \sum_{k \in T} p_{ik}^{(m-1)} p_{kj}$$

for $m \geq 2$. In other words, the m -step transition probabilities are recursively defined by the single step transition probabilities. The idea behind the Chapman Kolmogorov equations is, that before proceeding to state j within

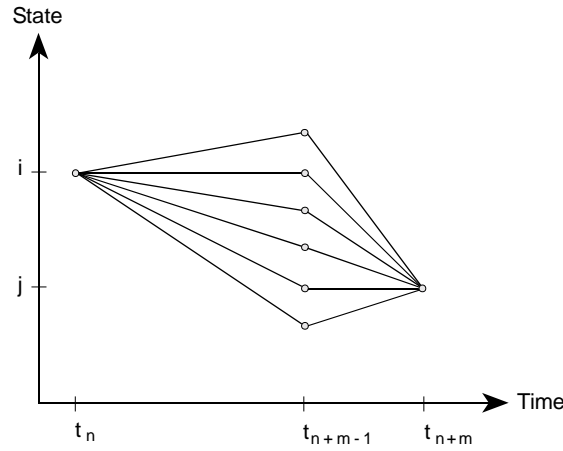


Figure A.2: Chapman Kolmogorov Equations

a single step, state k has to be reached from state i within $m - 1$ steps. As state k can be any state in state space, one has to sum up all probabilities. This is also shown in figure A.2. In matrix notation, the system of Chapman Kolmogorov equations is reduced to the power operation, that is $\mathbf{P}^{(m)} = \mathbf{P}^m$. Given a *vector of initial distributions* \mathbf{a} , the state of the process after m steps is given by $\mathbf{a}\mathbf{P}^m$. So a Markov chain is fully determined by an initial distribution and the transition probabilities. This we had expected already from the more general framework of Markov processes.

Based on the experience with empirical processes one may ask for a steady state following a startup phase. In mathematical terms steady state is described by a *limiting distribution*, which is independent of any initial distribution. For such a limit to exist, certain restrictions become necessary. One of the more obvious conditions is, that the Markov chain under analysis shall not consist of independent subchains. Visualized as graph, there must be a path of positive probability between each pair of states.

Definition 40 A Markov chain is *irreducible*, if all its states communicate, i.e. any $i \neq j \in S$ satisfies

$$p_{ij}^{(m)} > 0$$

Otherwise it is called *reducible*.

If a chain is reducible, it can be splitted up into subchains with each of

them analyzed separately. Mathematically expressed irreducibility defines an equivalence relation resulting in a certain grouping of states. It turns out, that the properties discussed below are shared by all members of an irreducible Markov chain. One of them is periodicity. As an example consider a bistable flip-flop toggling between states 0 and 1 at every time instant. This behaviour is clearly periodic, as the process returns to the starting state every second step. The initial distribution is preserved to infinity and no steady state can be achieved.

Definition 41 *A state is called aperiodic, if the greatest common divisor of*

$$\left\{ m | p_{ii}^{(m)} > 0 \right\}$$

equals 1. If not, the state is called periodic. For an irreducibility class, the period is the same for all class members.

In order to assume a steady state, we have to assure, that the Markov chain does not drift to infinity or get stuck in a group of states. The visiting behaviour of each state has to be inspected more closely.

Definition 42 *Let $f_i^{(m)}$ describe the probability to return to state i within m steps, i.e.*

$$f_i^{(m)} = \Pr \{ X_m = i, X_k \neq i : k = 1, 2, \dots, m-1 | X_0 = i \}$$

and define f_i to be the probability to return to state i in a finite or infinite number of steps, that is

$$f_i = \sum_{m=1}^{\infty} f_i^{(m)}$$

then state i is called recurrent, if $f_i = 1$. For $f_i < 1$ it is called transient. In case of a recurrent state

$$m_i = \sum_{m=1}^{\infty} m f_i^{(m)}$$

is called the mean recurrence time. For $m_i < \infty$ state i is called positive recurrent, otherwise it is called null recurrent. For an irreducibility class, these properties extend to all members.

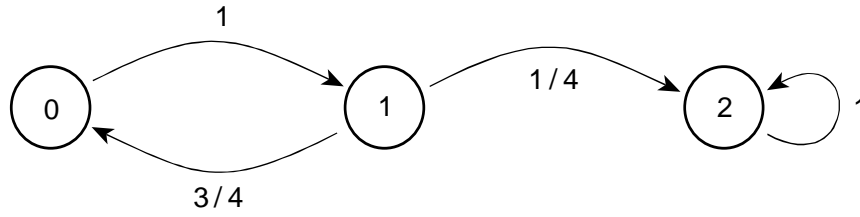


Figure A.3: A simple Markov chain example

Instead of assuming an almost sure return to state i for recurrence, we could also ask for an almost sure infinite number of visits to state i in the above definition. Indeed, it turns out that both conditions are equivalent:

Theorem 43 Given $\nu_i = \#(n > 0 : X_n = i)$ the number of visits to state i , the following conditions are equivalent: $f_i = 1 \Leftrightarrow \Pr\{\nu_i = \infty\} = 1 \Leftrightarrow \mathbb{E}\nu_i = \infty$.

For a proof, please refer to [58]. Combining all the properties discussed so far leads to the concept of *ergodicity*.

Definition 44 An irreducible, aperiodic and positive recurrent Markov chain is called *ergodic*.

Before proceeding to the existence of the limiting distribution, consider the example shown in figure A.3. State 0 may be reached from state 1 and vice versa. These states communicate and are periodic. State 2 acts like a trap - once entered the process can not escape. Such a state is called an *absorbing state*. A little calculation shows that $f_0^{(2)} = f_1^{(2)} = \frac{3}{4}$ and $f_0^{(m)} = f_0^{(m)} = 0$ for $m \neq 2$. Hence $f_0 = f_1 = \frac{3}{4} < 1$. Both states are transient, whereas state 2 is positive recurrent. The properties are not shared among states, so the chain is not irreducible. This can also be seen from the fact, that there is no communication from state 2 to state 1.

The limiting distribution assumed by a Markov chain in steady state will be denoted by $\pi_i = \lim_{n \rightarrow \infty} \Pr\{X_n = i\}$. These probabilities are sometimes called *stationary*. If they exist, no transition of the underlying Markov chain affects the probability vector $\boldsymbol{\pi} = (\pi_i)_{i \in S}$. In matrix form, the following equilibrium conditions hold

$$\boldsymbol{\pi} = \boldsymbol{\pi} \mathbf{P}, \quad \sum_{i \in S} \pi_i = 1 \quad (\text{A.3})$$

We are now ready to summarize the main results on the existence of the limiting distribution $\boldsymbol{\pi}$:

Theorem 45 *Given an aperiodic Markov chain in discrete time, the limits $\pi_i = \lim_{n \rightarrow \infty} \Pr \{X_n = i\}$ for all $i \in S$ exist. For an irreducible and aperiodic Markov chain the following expression holds*

$$\pi_i = \frac{1}{m_i}$$

These limits are independent of the initial distribution but do not necessarily constitute a probability distribution, because m_i might become infinite. In case the underlying Markov chain is ergodic, the vector $\boldsymbol{\pi} = (\pi_i)_{i \in S}$ represents a valid probability distribution.

Turning attention to the requirement of ergodicity for the existence of a stationary distribution it turns out, that irreducibility and aperiodicity are easy to verify. Proving recurrence often becomes cumbersome. So one often starts from the opposite direction, that is calculating the solution to A.3 first. By the existence of the stationary distribution vector $\boldsymbol{\pi}$, the underlying discrete time Markov chain may be assumed to be positive recurrent. Another useful fact is, that every irreducible Markov chain with a finite number of states is also positive recurrent. Furthermore standard matrix calculus may be applied to derive the solutions.

Although a Markov chain is per definition memoryless, it may be applied to a wider class of models. Therefore a stochastic process is observed only, when state transitions occur. These occurrences are called *regeneration points*. The resulting process satisfies the definition of a discrete time Markov chain. It is called an *embedded Markov chain* and proves to be useful especially in the theory of queues.

Markov chains in discrete time have been widely explored, so there exists a vast amount of literature. Classics include [115], [116], [58] and [64]. Numerical aspects relevant for applied Markov chains are discussed in [190]. A very detailed treatment is found in [119].

A.3.2 Homogenous Markov Chains in Continuous Time

There are several approaches to the analysis of Markov chains in continuous time. We will follow the traditional approach, because it nicely relates to the

methods used for discrete time Markov chains. With some slight modifications in notation to reflect the continuity of the parameter the *transition probabilities* are defined as $p_{ij}(s, t) = \Pr \{X(t) = j | X(s) = i\}$. Time-homogeneity allows us to write $p_{ij}(s, t) = p_{ij}(0, t - s) =: p_{ij}(t - s)$. Please note, there is no such concept like a single step transition probability, because a dedicated time unit does not exist. Consequently infinitesimal calculus has to be applied to gain results in continuous time. This in turn requires additional restrictions to be imposed on the *transition rate matrix* $\mathbf{P}(t)$:

Definition 46 A matrix $\mathbf{P} = (p_{ij})_{i,j \in S}$ is called *stochastic*, if $p_{ij} > 0$ for all $i, j \in S$, $\sum_j p_{ij} = 1$ for all $i \in S$ and at least one element in each column differs from zero.

Definition 47 $\mathbf{P}(t)$ is called a *transition semigroup* on state space S , if $\mathbf{P}(t)$ is a stochastic matrix, $\mathbf{P}(0) = \mathbf{I}$ and $\mathbf{P}(t + s) = \mathbf{P}(t)\mathbf{P}(s)$.

The last condition assumed for $\mathbf{P}(t)$ to be a transition semigroup is the continuous time equivalent of the system of *Chapman Kolmogorov equations*. Furthermore assume, that the transition probabilities are continuous at 0, that is $\lim_{t \rightarrow 0} \mathbf{P}(t) = \mathbf{P}(0) = \mathbf{I}$. This in turn implies

$$\lim_{t \rightarrow 0} p_{ij}(t) = p_{ij}(0)$$

and

$$q_{ij} := \lim_{t \rightarrow 0} \frac{p_{ij}(t) - p_{ij}(0)}{t} \quad (\text{A.4})$$

with $0 \leq q_{ij} < \infty$ for $i \neq j$ and $q_{ii} \leq 0$. Rewritten in matrix notation $\mathbf{Q} := (q_{ij})_{i,j \in S}$ one arrives at the *infinitesimal generator*. The matrix equivalent of A.4 is

$$\mathbf{Q} = \lim_{t \rightarrow 0} \frac{\mathbf{P}(t) - \mathbf{P}(0)}{t} = \lim_{t \rightarrow 0} \frac{\mathbf{P}(t) - \mathbf{I}}{t}$$

Based on the infinitesimal generator we are able to define further properties for the transition semigroup $\mathbf{P}(t)$:

Definition 48 $\mathbf{P}(t)$ is called *stable*, if $-q_{ii} < \infty$ for all $i \in S$. $\mathbf{P}(t)$ is called *conservative*, if $-q_{ii} = \sum_{j \in S, j \neq i} q_{ij}$ for all $i \in S$.

The latter probability derives from the *conservation equality* $\sum_{j \in S} p_{ij}(t) = 1$ for fixed t . In other words, any work performed by the process is preserved.

Rewriting the system of Chapman Kolmogorov equations as $\mathbf{P}(t+s) - \mathbf{P}(t) = \mathbf{P}(t)\mathbf{P}(s) - \mathbf{P}(t)$, dividing by s

$$\frac{\mathbf{P}(t+s) - \mathbf{P}(t)}{s} = \frac{\mathbf{P}(t)\mathbf{P}(s) - \mathbf{P}(t)}{s} = \mathbf{P}(t)\frac{\mathbf{P}(s) - \mathbf{I}}{s} \quad (\text{A.5})$$

and passing to the limit $s \rightarrow \infty$ one arrives at *Kolmogorov's forward differential system*

$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{P}(t)\mathbf{Q}$$

Extracting $\mathbf{P}(t)$ in A.5 to the right side results in *Kolmogorov's backward differential system*

$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{Q}\mathbf{P}(t)$$

In traditional notation, these systems may be written as

$$\begin{aligned} \frac{d}{dt}p_{ij}(t) &= p_{ij}(t)q_{jj} + \sum_{k \in S, k \neq j} p_{ik}(t)q_{kj} \\ \frac{d}{dt}p_{ij}(t) &= q_{ii}p_{ij}(t) + \sum_{k \in S, k \neq i} q_{ik}p_{kj}(t) \end{aligned} \quad (\text{A.6})$$

By embedding a discrete time Markov chain the concepts of irreducibility, communication, transience, recurrence and positive recurrence are inherited. Therefore one has to note, that for each $c > 0$, $Y_n = X(t)$ with $t = cn$ describes a Markov chain in discrete time [88]. Obviously there is no aperiodicity, as we miss a dedicated time unit for continuous time Markov chains.

Now we are in the position to calculate the steady state distribution of a Markov chain in continuous time. Let $p_j(0) = \Pr\{X(0) = j\}$ denote the *initial probability* for state j and define $\mathbf{p} = (p_j(0))_{j \in S}$ as the *initial probability vector*. Choosing equation A.6 and applying the law of total probability $p_j(t) = \sum_{i \in S} p_{ij}(t)p_i(0)$ results in

$$\frac{d}{dt}p_j(t) = q_{jj}p_j(t) + \sum_{k \in S, k \neq i} p_k(t)q_{kj} \quad (\text{A.7})$$

Irreducibility now assures the existence of the *limiting probabilities* $p_j = \lim_{t \rightarrow \infty} p_j(t)$. Assuming an equilibrium, there is no variation in $p_j(t)$, that is $\frac{d}{dt}p_j(t) = 0$. Equation A.7 now becomes

$$0 = q_{jj}p_j(t) + \sum_{k \in S, k \neq i} p_k(t)q_{kj}$$

or in matrix notation

$$\mathbf{0} = \mathbf{p}\mathbf{Q} \quad (\text{A.8})$$

This system of equations is often associated with the concept of *global balance*. Forcing the p_j to form a valid probability distribution by imposing the additional restriction

$$\sum_{j \in S} p_j = 1$$

one has successfully derived the *stationary probabilities* with \mathbf{p} the *stationary probability vector*. A similar derivation also exists for Kolmogorov's backward differential system. Knowing how to calculate steady state distributions, one may ask, under what circumstances such solutions remain valid. Based on the definitions of stability and conservativity we are able to state two simple conditions:

Theorem 49 *Given a conservative continuous time Markov chain, Kolmogorov's backward differential system is valid. Kolmogorov's forward differential system applies for a stable Markov chain in continuous time.*

Please note, that the global balance equations also remain valid for the discrete case, as the infinitesimal generator may be constructed as $\mathbf{Q} = \mathbf{P} - \mathbf{I}$. In either case they have an intuitive interpretation. The flow out of a certain state has to equal the flow into that state. Clearly this concept is related to conservativity and stability. For further reading on the topics discussed we recommend [51] and [190]. Proofs, which were skipped, are found in the former reference.

Appendix B

Computer Programs

Through the entire text we have used computer programs to generate graphs and calculate the more demanding examples. Some routines were solely developed for illustrational purposes while other ones have already been applied in practice. We have compiled the most useful algorithms and utility functions into some form of library. The corresponding source codes are now presented and explained in this appendix.

We have used two programming platforms for development, which are the open source programming language R (<http://www.r-project.org>) to be used on any standard personal computer and the Casio Classpad 330 table calculator [55]. The majority of queueing routines has been implemented on both systems revealing the far reaching power of todays' table calculators. However, in some cases not enough processing power is available and so the corresponding procedures have been developed for R only. This includes some routines from the queueing library as well as the entire set of procedures devoted to the approximation of lifetime distributions. We have to note, that especially the R source code has not been optimized for performance. Instead we aimed to provide a code library, which allows for simple manipulation and usage. Furthermore we tried to stick close to the underlying mathematical apparatus and not to use high sophisticated packages available from libraries or the like. We hope, that all these undertakings aid in repeating the experiments, tracing the results, generating new results and extending the framework.

B.1 Library Description

As mentioned above, the queueing library has been implemented in R and for the Classpad 330 table calculator. Although the implementations differ, the function signatures are similar for both systems. Accordingly the function names and parameter sets do not change. The following list provides an overview of all procedures and functions.

- **qmmck**(r, c, k) is an exact implementation of the $M/M/c/K$ model described in section 3.3.2. The input parameters specify traffic intensity r (ρ), number of servers c (c) and system size k (K). The corresponding variables as used in mathematical expressions have been put in brackets. This function calculates the steady state probabilities **p** (**p**) and various steady state performance indicators such as average queue length **Lq** (L_q), average system size **L** (L), blocking probability **PrBlock** (p_b) and delay probability **PrDelay** (p_d).
- **qerlb**(r, c) is a straightforward implementation of the Erlang loss formula 3.28. Accordingly it calculates the blocking probability p_b for the $M/G/c/c$ queueing system.
- **qxhlp**(r, c) is a utility function, which calculates $\varsigma(\rho, c)$ as given by expression 3.81. This function is called by **qxdlmc**(r, c) and **qxmdc**(r, c).
- **qxmdc**(r, c) provides a suitable multiplication factor, which relates W_q and L_q of the $M/M/c$ model to those of the $M/D/c$ queueing system. This function belongs to the framework set forth by Whitt, Cosmetatos, Krämer and Langenbach-Belz. It is called by **qggc**(r, c, ct, cs).
- **qxdlmc**(r, c) provides a suitable multiplication factor, which relates W_q and L_q of the $M/M/c$ model to those of the $D/M/c$ queueing system. This function belongs to the framework set forth by Whitt, Cosmetatos, Krämer and Langenbach-Belz. It is called by **qggc**(r, c, ct, cs).
- **qggc**(r, c, ct, cs) implements several approximations for the $G/G/c$ queueing system. In addition to the parameters described above the squared coefficient of variation for the interarrival time **ct** (c_T^2) and the service time **cs** (c_S^2) are accepted as well. Based on our treatment in section 3.7.2, the function first calculates idle probability **MMc-P0** (p_0), delay probability **MMc-Pd** (p_d) and average queue length **MMc-Lq** (L_q).

$(L_q^{(M/M/c)})$ for the $M/M/c$ queue. From there it proceeds to derive MDc-Lq ($L_q^{(M/D/c)}$) and DMc-Lq ($L_q^{(D/M/c)}$) as intermediate results. Finally it computes the Allen-Cunneen approximation for the average queue length AC-Lq (L_q) as well as the Kimura and Page approximations Kim-Lq and Page-Lq (L_q). According to the value for c_T^2 , a suitable approximation is chosen from the latter two. The Allen-Cunneen approximation is provided independently. In order to produce meaningful results, the stability condition $\rho < 1$ has to be satisfied.

- `qmmeqn(la, b, my, P, c, N)` provides an exact solution for the matrix exponential $M/ME/c/N$ model by using the results from section 3.8.2 and 3.8.3. Due to its complexity, this algorithm has only been implemented in R. The function accepts as input parameters the arrival rate `la` (λ), an entry vector `b` (β), a vector of service rates `my` ($\mu = (\mu_1, \mu_2, \dots, \mu_k)$), a transition matrix `P` (\mathbf{P}), the number of servers `c` (c) and system capacity `N` (N). The tuple (β, μ, \mathbf{P}) is an equivalent specification for the matrix exponential service distribution $ME(\beta, \mathbf{B})$, because $\mathbf{B} = \text{diag}(\mu)(\mathbf{P} - \mathbf{I})$. If multiple servers are specified, the routine first assembles the state space `SS` and the matrices `M`, `L` and `E`. The latter are compound versions of the matrices \mathbf{M}_n , \mathbf{L}_n and \mathbf{E}_n for $0 < n \leq c$. Although this is far from being computationally efficient, this implementation also allows for experiments with Markov chain solvers. As a next step the auxiliary and steady state probability vectors are calculated. This is a direct extension of the algorithm for $c = 2$ given by L. Lipsky in his book [142]. Finally the performance indicators are computed and returned in a format similar to `qmmck(r, c, k)`.
- `qmemn(b, my, P)` is a supporting routine, which computes the mean service time from the parameters of a matrix exponential distribution. It has been designed to simplify comparison with results from the $M/M/c/K$ or $G/G/c$ queueing model. This function is only available in R.
- `qmecv(b, my, P)` is a supporting routine, which computes the squared coefficient of variation for the mean service time. It has been designed to simplify comparison with results from the $G/G/c$ queueing system. This function is only available in R.

This concludes the set of functions dedicated to the solution of classic and matrix exponential queueing systems. To be of practical value, the parameters required to be fed into the queueing models are expected to reflect the relevant features common to real life situations. Finding a suitable matrix exponential representation for an arbitrary lifetime distribution will become an essential factor. Accordingly we have provided an illustrative set of procedures and functions devoted to this problem.

- `qdgam(x, a, l)` is a simple function wrapper for the gamma density function `dgamma` of R. It avoids a misspecification of the rate parameter `l`. This ancilliary function is only available in R.
- `qmgam(a, l, n)` returns an array containing the first n moments μ_m for $1 \leq m \leq n$ of the gamma distribution identified by shape parameter `a` and rate parameter `l`. This ancilliary function is only available in R.
- `qtaylor(m)` converts the set of moments m (μ_n for $1 \leq n \leq \#m$) of an arbitrary lifetime distribution to an array of coefficients γ_n for the Taylor approximation to the Laplace transform of the corresponding density function. Details are laid out in section 4.3.4. This function is only available in R.
- `qexpm(m, n=20)` calculates the matrix exponential of matrix `m` based on the first n terms of the exponential series expansion. The second parameter `n` is optional and defaults to 20. This function is only available in R.
- `qhankel(ar, k, n)` generates Hankel's matrix $H_k^{(n)}$ from a set of coefficients `ar`. The parameters `k` and `n` are intrinsic to the definition of Hankel matrices, for the theoretical background please refer to section 4.3.4. This function is only available in R.
- `qcfrac(ar)` determines the continued fraction representation for a given set of Taylor coefficients `ar` (γ_n for $1 \leq n \leq \#ar$). The corresponding coefficients are returned as an array (d_n for $1 \leq n \leq \#ar$). This function is only available in R.
- `qrlt(d)` calculates the rational Laplace transform based on the continued fraction `d`. The coefficients `b` for numerator (b_1, \dots, b_k) and `a` for the denominator (a_1, \dots, a_k) are returned in compound form. Here $k = \frac{\#d}{2}$, respectively. This function is only available in R.

- `qgenpar(r)` returns a matrix exponential representation $ME(\beta, \mathbf{B})$ for a given rational Laplace transform r . This function is only available in R.
- `qmeapprox(m)` returns a matrix exponential representation $ME(\beta, \mathbf{B})$ for a given set of moments \mathbf{m} (μ_n for $1 \leq n \leq \#\mathbf{m}$). From a functional perspective it is a combination of the preceding three functions. This function is only available in R.
- `qdme(x, mb, b)` operates similar to other common density functions in R. Corresponding to vector \mathbf{x} this function returns a set of values of a matrix exponential density represented by the parameters \mathbf{mb} (\mathbf{B}) and \mathbf{b} (β). The output is well suited for plotting purposes. This function is only available in R.
- `qdgme(x, ma, a)` computes density values for generalized matrix exponential representations. This function is currently unused and only available in R.
- `qmme(n, mb, b)` derives the first n moments of the matrix exponential distribution represented by the parameters \mathbf{mb} (\mathbf{B}) and \mathbf{b} (β). This function is only available in R.
- `qmeshow(m, x)` is a supporting function, which calculates the matrix exponential representation for a given set of moments \mathbf{m} . Besides, it prints an exhaustive set of details including all coefficients, original and approximated moments, absolute and relative errors. Furthermore the density is plotted to the screen for a given range of values \mathbf{x} . This function is the best way to try out several parameter sets without the need of getting to deep into programming details. This function is only available in R.

The thoughtful reader might have realized, that all routine names start with `q`. We hope, that this prevents the R workspace and the Classpad 330 variable manager from becoming cluttered by some weird names spread over the entire alphabet. For the latter, this concept has also been extended to the name assignment of global variables.

B.2 R

The easiest way to install the new functions to your workspace is by using the internal editor. First copy the file `queueing.r` to your work directory, then enter `edit(file='queueing.r')` in the R command line. After the editor has opened, press `CTRL-A` to select the entire file contents followed by `CTRL-R` to execute the selection. Then close the editor. Now each function can be invoked by simply entering its name and the corresponding parameters. Proceed in a similar fashion with the file `qmeapprox.r`. Whenever a function does return more than a single value, all results are embedded in a data frame. This compound variable type is unique to R and allows for simple access of its contents by name of the element.

The available functions of the queueing library have been arranged in two groups. Utility and helper functions have been assigned to the first group, whereas the model implementations for the $M/M/c/K$, the $M/ME/c/N$ and the $G/G/c$ queueing model belong to the second group. The sole purpose for this arrangement has been the gain in readability of the source code. There is no impact on the execution.

```
#
# helper functions
#

iif=function(c,t,f) if (c) return(t) else return(f)
fac=function(n) return(gamma(n+1))
qerlb=function(r,c) return(r^c/fac(c)/sum(r^(0:c)/fac(0:c)))
qxhlp=function(r,c)
  return(min(0.2499975, (1-r/c)*(c-1)*(sqrt(4+5*c)-2)/16/r))
qxdmc=function(r,c)
  return((0.5-2*qxhlp(r,c))*exp(-2*(c-r)/3/r))
qxmdc=function(r,c) return((1+qxhlp(r,c))/2)
qmemn=function(b,my,P)
  return(-sum(b%%solve(diag(my)%(P-diag(length(my))))))
qmecv=function(b,my,P) {
  h=solve(diag(my)%(P-diag(length(my))))
  return(2*sum(b%%h%%h)/sum(b%%h)^2-1)
}
```

```

#
# model implementations
#

qmmck=function(r,c,k) {
  # r ... rho
  # c ... no of servers
  # k ... no of customers in system k>=c

  stopifnot(k>=c)
  u=r/c;                # utilization
  kk=k+1

  n=seq(0,c-1)
  p=rep(0,kk)
  p[1]=1/(sum(iif(r==c,c^n,r^n)/fac(n))
    +r^c*iif(r==c,kk-c,(1-u^(kk-c))/(1-u))/fac(c))
  for (i in 2:kk)
    p[i]=p[i-1]*r/min(i-1,c)

  n=seq(0,k)
  l=sum(n*p)
  lq=sum((n[(c+1):(k+1)]-c)*p[(c+1):(k+1)])
  pd=sum(p[(c+1):k])      # pr{delay}
  pb=p[k+1]              # pr{blocked}

  d=data.frame(l,lq,pd,pb,t(p))
  colnames(d)<-c('L','Lq','PrDelay','PrBlock',as.character(n))
  return(d)
}

qmmecn=function(la,b,my,P,c,N) {
  # b ... vector of init probabilities for service fac
  # my ... vector of my's (service rates) for service fac
  # P ... transition probability matrix for service fac
  # c ... number of service facilities

```

```

# N ... number of customers

stopifnot(N>=c)
stopifnot(require(gtools))
k=unique(dim(P)) # number of stages
stopifnot(length(k)==1 && length(b)==k && length(my)==k)

# define exit vector
x=if(length(my)==1, my, diag(my))%*(diag(k)-P)%*rep(1, k)

if (c>1) {

# construct state space
SS=combinations(k+1, c, c(0:k), set=F, repeats=T)[-1, c:1]

# assemble matrix of service rates M
h=NA
for (i in 1:nrow(SS)) h=c(h, sum(my[SS[i,]]))
M=diag(h[-1])

# assemble matrix E, L and PP
E=matrix(0, nrow(SS)-choose(k+c-1, c), nrow(SS))
L=matrix(0, ncol(E), nrow(E)); L[1:k, 1]=x/my
PP=matrix(0, nrow(SS), nrow(SS))
l=1; u=k
for (n in 1:c) {
  v=u+choose(k+n, n+1)
  for (i in l:u) {
    hi=rep(0, k+1)
    hi[sort(unique(SS[i,]))+1]=table(SS[i,])
    if (i<u) for (j in (i+1):u) {
      hj=rep(0, k+1)
      hj[sort(unique(SS[j,]))+1]=table(SS[j,])
      h=hj[-1]-hi[-1]
      if (length(h[h!=0])==2 && min(h)==-1 && max(h)==1) {
        ii=which(h==-1); jj=which(h==1)
        PP[i, j]=P[ii, jj]*sum(SS[i,]==ii)*my[ii]/M[i, i]
        PP[j, i]=P[jj, ii]*sum(SS[j,]==jj)*my[jj]/M[j, j]
      }
    }
  }
}

```

```

    }
  }
  if (n<c) for (j in (u+1):v) {
    hj=rep(0, k+1)
    hj[sort(unique(SS[j, ]))+1]=table(SS[j, ])
    h=hj[-1]-hi[-1]
    if (length(h[h!=0])==1 && sum(h)==1) {
      jj=which(h==1)
      E[i, j]=b[jj]
      L[j, i]=x[jj]*sum(SS[j, ]==jj)/M[j, j]
    }
  }
}
l=u+1; u=v
}

# postcondition check
stopifnot(all((PP+L%%E)%%rep(1, nrow(PP))-1<1e-10))

# assemble matrix B
B=M%%(PP-diag(nrow(PP)))

# calculate R(c, n) for n=0...N-c
u=nrow(B); l=u-choose(k+c-1, c)+1; j=l-choose(k+c-2, c-1)
Rc=list(); Rc[1]=list(-la*solve(B[l:u, l:u]))
hi=la*diag(u-l+1)-B[l:u, l:u]
hj=M[l:u, l:u]%%L[l:u, j:(l-1)]%%E[j:(l-1), l:u]
if (N-c>=1) for (n in 1:(N-c))
  Rc[n+1]=list(la*solve(hi-Rc[[n]]%%hj))

# calculate R(n, N-c) for n=c...1
Rn=list(); Rn[c]=Rc[N-c+1]
u=nrow(B)
if (c>1) for (n in (c-1):1) {
  j=u-choose(k+n, n+1); l=j+1; i=l-choose(k+n-1, n)
  hi=la*diag(l-i)-B[i:j, i:j]
  hj=E[i:j, l:u]%%Rn[[n+1]]%%M[l:u, l:u]%%L[l:u, i:j]
  Rn[n]=list(la*solve(hi-hj))
}

```

```

    u=j
  }

  # assemble aux vectors
  rn=list(); rn[1]=list(b); rn[2]=list(b**Rn[[1]])
  l=1
  if (c>1) for (n in 2:c) {
    i=l+choose(k+n-2, n-1); u=i-1; j=u+choose(k+n-1, n)
    rn[n+1]=list(rn[[n]]**E[l:u, i:j]**Rn[[n]])
    l=i
  }
  rc=list(); rc[N-c+1]=rn[c+1]
  if (N>c) for (n in (N-c+1):2)
    rc[n-1]=list(rc[[n]]**Rc[[n-1]])

  # assemble (unnormalized) steady state prob vector
  p=c(sapply(rn, sum), rev(sapply(rc, sum))[-1])
  p=p/sum(p)

} else {

  # algorithm for c=1
  B=i if(length(my)==1, my, diag(my))** (P-diag(k))
  hi=solve(diag(k)-B/l a-matrix(rep(1, length(b))))**b
  hj=diag(k)-l a*solve(B)
  if (N>1) for (j in 2:N)
    hj=diag(k)+hi**hj
  p=rep(0, N+1)
  p[1]=1/(b**hj**rep(1, length(b)))
  hj=diag(k)
  if (N>1) for (j in 2:N) {
    hj=hj**hi
    p[j]=p[1]*b**hj**rep(1, length(b))
  }
  p[N+1]=-l a*p[1]*b**hj**solve(B)**rep(1, length(b))

}

```

```

# calculate performance indicators
n=seq(0,N)
l=sum(n*p)
lq=sum((n[1:(N-c+1)])*p[(c+1):(N+1)])
pd=sum(p[(c+1):N])      # pr{del ay}
pb=p[N+1]               # pr{bl ocked}

d=data.frame(l,lq,pd,pb,t(p))
col names(d)<-c(' 'L'',' 'Lq'',' 'PrDel ay'',' 'PrBl ock'',' as.character(n))
return(d)
}

qggc=function(r,c,ct,cs) {
  # r ... rho = service_time / interarr_time
  # c ... no of servers
  # ct ... squared coeff of variation interarr_time
  # cs ... squared coeff of variation service_time

  stopifnot(r<c)
  u=r/c;                      # utilization
  p0=1/(r^c/fac(c)/(1-u)+sum(r^(0:(c-1))/fac(0:(c-1))))
  pd=p0*r^c/fac(c)/(1-u)
  lq=pd*u/(1-u)
  lqd=lq*qxdmc(r,c)
  lqm=lq*qxmdc(r,c)
  lqa=r*pd*(ct+cs)/2/(c-r) # allen-cuneeen
  lqk=(ct+cs)/((1-ct)/lqd+(1-cs)/lqm+(2*(ct+cs-1))/lq)
  lqp=cs*(1-ct)*lqd+ct*(1-cs)*lqm+ct*cs*lq
  lqg=if(c>1,lqp,lqk)        # choose Kimura/Page

  d=data.frame(lqg+r,lqg,p0,pd,lq,lqm,lqd,lqa,lqp,lqk)
  col names(d)<-c(' 'L'',' 'Lq'',' 'MMc-P0'',' 'MMc-PrDel ay'','
                  ' 'MMc-Lq'',' 'MDc-Lq'',' 'DMc-Lq'',' 'AC-Lq'','
                  ' 'Page-Lq'',' 'Kim-Lq'')
  return(d)
}

```

We will now proceed to show some examples, which at the same time provide instructions on how to match the different models. The first example demonstrates the relation between the $M/M/4/7$ and the $M/ME/4/7$ queueing system with $ME = M$, $\lambda = 1$, $\mu = 10$.

```
> la=1; c=4; N=7
> my=10; b=1; P=matri x(0)
> c(l a, c, N); b; my; P
> qmmecn(l a, b, my, P, c, N)
```

	L	Lq	PrDel ay	PrBl ock	0
1	0.1000001	9.914332e-08	3.866766e-06	5.890868e-11	0.9048374
	1	2	3	4	5
1	0.09048374	0.004524187	0.0001508062	3.770156e-06	9.42539e-08
	6	7			
1	2.356347e-09	5.890868e-11			

```
> qmmck(l a/my, c, N)
```

	L	Lq	PrDel ay	PrBl ock	0
1	0.1000001	9.914332e-08	3.866766e-06	5.890868e-11	0.9048374
	1	2	3	4	5
1	0.09048374	0.004524187	0.0001508062	3.770156e-06	9.42539e-08
	6	7			
1	2.356347e-09	5.890868e-11			

For the next example we assume the service times to follow an Erlang distribution of order 3, i.e. we observe an $M/E_3/4/10$ queueing model. The corresponding parameters are given by $k = 3$, $\mu = 0.5$, $c = 4$, $K = 10$ and $\lambda = 1$. Obviously the system capacity will not lead to an excessive value for the blocking probability, so that we can compare our results with the corresponding $M/E_3/4$ queue. Please note, that similar results have been tabulated by F.S. Hillier and O.S. Yu in [96] (The model under consideration can be found on page 24 for $k = 3$ and $RHO = \frac{\lambda}{c\mu} = 0.5$).

```
> la=1; my=rep(3*0.5, 3); c=4; N=10; b=c(1, 0, 0)
> P=cbi nd(c(0, 0, 0), c(1, 0, 0), c(0, 1, 0))
> qmmecn(l a, b, my, P, c, N)
```

	L	Lq	PrDel ay	PrBl ock	0	1
1	2.121604	0.1223329	0.1687947	0.0003643759	0.1292421	0.2598127
	2	3	4	5	6	7
1	0.2625361	0.1792501	0.0950686	0.04376911	0.01853454	0.007455592

```

      8      9      10
1 0.002892440 0.001074375 0.0003643759
> qggc(length(my)*la/my[1], 4, 1, 1/length(my))
      L      Lq      MMc-P0 MMc-PDelay      MMc-Lq      MDc-Lq
1 2.123820 0.1238197 0.1304348 0.1739130 0.1739130 0.09877301
      DMc-Lq      AC-Lq      Page-Lq      Kim-Lq
1 0.02037781 0.1159420 0.1238197 0.1259904

```

For the last example we have prepared an arbitrary phase type model. It also shows how to calculate traffic intensity and squared coefficient of variation from the parameter specification for a matrix exponential distribution by use of the functions `qmemn(b, my, P)` and `qmecv(b, my, P)`.

```

> la=1; c=4; N=7
> my=c(10, 20); b=c(0.4, 0.6); P=cbind(c(0, 0.8), c(0.9, 0))
> qmmecn(la, b, my, P, c, N)
      L      Lq      PrDelay      PrBlock      0
1 0.4859411 0.0002281278 0.001621487 2.734215e-06 0.6152294
      1      2      3      4      5
1 0.2988257 0.0725718 0.01174894 0.001423462 0.0001761238
      6      7
1 2.190071e-05 2.734215e-06
> qggc(la*qmemn(b, my, P), c, 1, qmecv(b, my, P))
      L      Lq      MMc-P0 MMc-PDelay      MMc-Lq
1 0.4859433 0.0002290350 0.6152289 0.001623947 0.000224448
      MDc-Lq      DMc-Lq      AC-Lq      Page-Lq      Kim-Lq
1 0.0001402797 9.020976e-12 0.000230564 0.0002290350 0.0002280781

```

Obviously all these examples are artificial in nature and do not reveal the true power of matrix exponential queueing models. Instead they have been chosen to demonstrate library usage and the relation to more common queueing systems. A more applied view will be adopted in section 6.5.5.

The library containing the procedures and functions necessary for the approximation of lifetime distributions is similar in architecture to the preceding one. In browsing downward the source code, one smoothly transitions from helper and utility functions to the main part of the library. Furthermore we have included some ancillary functions related to the gamma distribution. If removed, there is no disruption of functionality. In fact, they have been included to support the examples used throughout the text.

```

#
# functions for the gamma distribution
#

qdgam=function(x,a,l)      # gamma density wrapper
  return(dgamma(x,shape=a,rate=l))

qmgam=function(a,l,n) {    # gamma moments
  # a ... shape, l...rate
  # n ... number of gamma moments
  #
  ar=rep(a/l,n)
  for (i in 2:n)
    ar[i]=ar[i-1]*(a+i-1)/l
  return(ar)
}

#
# helper functions
#

iif=function(c,t,f) if (c) return(t) else return(f)
fac=function(k) return(gamma(k+1))
lshft=function(ar) return(c(ar[2:length(ar)],0))

qtaylor=function(m)        # taylor coeffs
  # m ... vector of moments
  return(c(1,m/fac(1:length(m))
           *rep(c(-1,1),trunc(length(m)/2))))

''%^%''=function(m,n) {    # matrix power
  if (n<0) return(solve(m)%^%-n)
  else if (n==0) return(diag(dim(m)[1]))
  else return(m%*(m%^%(n-1)))
}

qexpm=function(m,n=20,mo=m,no=n) {

```

```

if (n<1)
  return(diag(dim(mo)[1]))
else
  return(diag(dim(mo)[1])+(mo%%qexpm(m, n-1, mo, no))/(no-n+1))
}

#
# core part
#

qhankel=function(ar,k,n) {
  # ar ... arr of Taylor series coeffs
  # k ... aka matrix size
  # n ... first index
  #
  # returns: ma ... hankel matrix
  #
  if (k<1) return(matrix(1,1,1))
  ma=matrix(0,k,k)
  for (i in 1:k)
    ma[i, 1:k]=ar[(n+i):(n+k+i-1)]
  return(ma)
}

qcfrac=function(ar) {
  # ar ... arr of Taylor series coeffs
  #
  # returns: d ... (d_1,d_2,...,d_2n), d_0 not included !
  #
  stopifnot(length(ar) %% 2 == 1)
  n=trunc(length(ar)/2)
  d=rep(0,n*2)
  for (k in 1:n) {
    # generate determinants
    mhk0=det(qhankel(ar,k,0))
    mhkm0=det(qhankel(ar,k-1,0))
    mhkp0=det(qhankel(ar,k+1,0))
    mhk1=det(qhankel(ar,k,1))
  }
}

```

```

    mhkm1=det(qhankel(ar, k-1, 1))
    d[2*k-1]=-mhk1*mhkm0/mhkm1/mhk0
    d[2*k]=-mhkp0*mhkm1/mhk0/mhk1
  }
  return(d)
}

qr1t=function(d) {
  # d ... arr of coeffs of continued fraction
  #
  # returns: data frame with r1t coeffs
  # numerator $b, denominator $a
  #
  stopifnot(length(d) %% 2 == 0)
  k=trunc(length(d)/2)
  ppp=rep(0, k+1)
  pp=qpp=q=c(rep(0, k), 1)
  for (i in 2:(2*k)) {
    p=lshift(ppp)*d[i-1]+pp
    q=lshift(qpp)*d[i-1]+qp
    qpp=qp          # prev-prev q
    qp=q           # prev q
    ppp=pp         # prev-prev p
    pp=p           # prev p
  }
  df=data.frame(rev(p)/q[1], rev(q)/q[1])
  colnames(df)=c("b", "a")
  return(df)
}

qgenpar=function(r) {
  # r ... data frame with coeffs of r1t
  #   numerator $b, denominator $a
  #
  # returns: mb=as.matrix(df[-1]); b=t(df$b)
  #
  k=dim(r)[1]-1
  ma=rbind(cbind(rep(0, k-1), diag(k-1)), -r$a[1:length(r$a)-1])

```

```

a=r$b[1:length(r$b)-1]
mt=rbind(c(1/r$a[1]+1, rep(0, k-1)), cbind(diag(k-1),
      rep(0, k-1)))-diag(k)
mb=solve(mt)%*%ma%*%mt
b=a%*%mt
df=data.frame(t(b), mb)
colnames(df)[1]="b"
return(df)
}

qmeapprox=function(m)
# m ... array of moments
#
# returns: mb=as.matrix(df[-1]); b=t(df$b)
#
return(qgenpar(qrlt(qcfrac(qtaylor(m))))))

qmedensity=function(x, mb, b)
return((-b%*%expm(mb*x)%*%mb%*%rep(1, length(b)))[1, 1])
qdme=function(x, mb, b) { # vector version
# mb ... "transition matrix" of ME
# b ... entry vector of ME
#
# returns: density vector
#
stopifnot(require(Matrix))
return(sapply(x, qmedensity, mb, b))
}

qgmedensity=function(x, ma, a)
return((a%*%expm(ma*x)%*%c(rep(0, length(a)-1), 1))[1, 1])
qdgme=function(x, ma, a) { # vector version
# ma ... "transition matrix" of generalized ME
# a ... entry vector of generalized ME
#
# returns: density vector
#
stopifnot(require(Matrix))

```

```

    return(sapply(x, qgmedensity, ma, a))
}

qmemoments=function(n, mb, b)
  return(fac(n)*b**%*(-mb**%-n)**%rep(1, length(b)))
qmme=function(n, mb, b) # vector version
  # mb ... "transition matrix" of ME
  # b ... entry vector of ME
  #
  # returns: vector of ME moments
  #
  return(sapply(n, qmemoments, mb, b))

qmeshow=function(m, x) {
  # m ... array of moments
  # x ... display values for graph
  #
  n=length(m)
  d=qcfrac(qtayl or(m))
  r=qrlt(d)
  df=qgenpar(r)
  mb=as.matrix(df[, -1])
  b=t(df$b)
  cat("Continued fraction: \n")
  print(d)
  cat("Rational LT: \n")
  print(r)
  cat("ME parameters: \n")
  print(df)
  cat("Original moments: \n")
  print(m)
  cat("ME moments: \n")
  mm=qmme(1:n, mb, b)
  print(mm)
  cat("Absolute Difference (moments): \n")
  print(m-mm)
  cat("Error in % (moments): \n")
  print(trunc(10000*abs(m-mm)/m)/100)
}

```

```

me=qdme(x, as.matrix(df[, -1]), t(df$b))
plot(me~x, type="l", xlab="", ylab="", lty=2)
}

```

The thoughtful reader might have realized, that the function `qexpm()` is not used by any other function. Due to numerical instabilities in the series approximation of the matrix exponential, we have chosen to use `expm()` of the `Matrix` package instead. In fact, the numerically most endangered parts of the library are associated to the matrix power and the matrix exponential. Both routines are not used throughout the core part. They are only required for the calculation of moments and for plotting the density function. As will turn out, these instabilities show up very quickly when increasing the level of approximation. Proceeding to some examples, our first attempt is concerned with the calculations of section 4.3.4. Following the first example we may calculate a 3-stage matrix exponential representation for the deterministic distribution using the following instructions

```

> qmeshow(rep(1,6), 1:100/50)
Continued fraction:
[1] 1.0000000 -0.5000000 0.1666667 -0.1666667 0.1000000
[6] -0.1000000
Rational LT:
      b  a
1  60 60
2 -24 36
3   3  9
4   0  1
ME parameters:
      b X1  X2 X3
1 -23 60 -60  0
2  27 60 -61  1
3  -3 97 -88 -8
Original moments:
[1] 1 1 1 1 1 1
ME moments:
[1] 1.0 1.0 1.0 1.0 1.0 1.1
Absolute Difference (moments):
[1] 6.661338e-15 -9.037215e-14 1.101341e-13 1.744827e-12
[5] -8.375522e-13 -1.000000e-01

```

```
Error in % (moments):
[1] 0 0 0 0 0 10
> abline(v=1)
```

The call to `abline()` marks the location of the unit pulse in the graph and is repeated here for convenience. The printed results are best compared with the corresponding expressions in the first example of section 4.3.4. To create plots similar to the ones presented in the second example for the gamma distribution assume a parameter set of $\alpha = 0.5$ and $\lambda = 2$. A 4-stage approximation is best attempted by

```
> m=qmgam(.5, 2, 8)
> x=1:100/500
> qmeshow(m, x)
Continued fraction:
[1] 0.250 0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125
Rational LT:
      b      a
1 2048 2048
2 1536 2048
3  320  640
4   16   64
5    0    1
ME parameters:
      b    X1    X2    X3    X4
1 1537 2048 -2048    0    0
2 -1216 2048 -2049    1    0
3  -304 2048 -2049    0    1
4   -16 4097 -3457 -576 -63
Original moments:
[1] 0.2500000 0.1875000 0.2343750 0.4101562 0.9228516
[6] 2.5378418 8.2479858 30.9299469
ME moments:
[1] 0.2500000 0.1875000 0.2343746 0.4101560 0.9228506
[6] 2.5378437 8.2479389 30.9250042
Absolut Difference (moments):
[1] -1.274820e-10 -1.869644e-09 4.430244e-07 2.334079e-07
[5] 9.958207e-07 -1.899438e-06 4.698399e-05 4.942691e-03
Error in % (moments):
```

```
[1] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01
> mg=qdgam(x, a, l)
> lines(mg~x)
```

The last two lines are again responsible for the inclusion of the original density function in the graph. Vector x allows for an adjustment of the plotting range. From the underlying theory we know, that the first seven moments have to match. Any other result would indicate numerical instability. There is no condition on the eighth moment, but in the current example it exhibits a deviation of 0.01% only. As will be supported by more examples, the error in the last moment may serve as a performance indicator in the selection of the right number of stages.

To justify the optimal level of approximation we once again have to face the principle of Ockham's razor. On one hand we can be sure to match more moments by simply increasing the number of stages. On the other hand we may run into numerical difficulties. As an example consider a gamma distribution with shape $\alpha = 0.5$ and rate $\lambda = 4$. Attempting a 5-stage approximation and proceeding as above yields the following moment statistics

Original moments:

```
[1] 0.12500000 0.04687500 0.02929688 0.02563477
[5] 0.02883911 0.03965378 0.06443739 0.12082011
[9] 0.25674272 0.60976397
```

ME moments:

```
[1] 0.12500230 0.10564748 -0.02578457 0.42122105
[5] 2.05814563 -1.06551325 -5.03615881 -6.30917110
[9] -18.89725123 0.53043734
```

Absolut Difference (moments):

```
[1] -2.303620e-06 -5.877248e-02 5.508144e-02 -3.955863e-01
[5] -2.029307e+00 1.105167e+00 5.100596e+00 6.429991e+00
[9] 1.915399e+01 7.932663e-02
```

Error in % (moments):

```
[1] 0.00 -125.38 188.01 -1543.16 -7036.64 2787.04
[7] 7915.58 5321.95 7460.38 13.00
```

To avoid the text from becoming cluttered, we decided to show only the interesting part of the output. Indeed, the relative error indicates severe numerical instabilities. Although the results (β, \mathbf{B}) may still be valid, we adopt a more conservative approach by reducing the level of approximation. Considering 3 stages as being sufficient, we arrive at

Original moments:

```
[1] 0.12500000 0.04687500 0.02929688 0.02563477
[5] 0.02883911 0.03965378
```

ME moments:

```
[1] 0.12500000 0.04687500 0.02929687 0.02563477
[5] 0.02883911 0.03956795
```

Absolut Difference (moments):

```
[1] 6.824763e-12 -1.393374e-11 1.083400e-11 -1.027378e-11
[5] -5.202505e-11 8.583061e-05
```

Error in % (moments):

```
[1] 0.00 0.00 0.00 0.00 0.00 0.21
```

Now the numerical instabilities have vanished and the result looks quite reasonable. For illustrational purposes, we perform a 2-stage approximation to compare the statistics

Original moments:

```
[1] 0.12500000 0.04687500 0.02929688 0.02563477
```

ME moments:

```
[1] 0.12500000 0.04687500 0.02929688 0.02490234
```

Absolut Difference (moments):

```
[1] -8.881784e-16 -3.941292e-15 -2.914335e-15 7.324219e-04
```

Error in % (moments):

```
[1] 0.00 0.00 0.00 2.85
```

The rise in the fourth moment indicates, that our selection resulted in a coarse approximation. According to the moment errors, 3 stages result in an optimal matrix exponential representation. This is also underlined by the density functions shown for all three cases in figure B.1. The next example shows, that it is not sufficient to review the density function or moments alone. In fact, they always go hand in hand when choosing the level of approximation. Consider a gamma distribution with shape $\alpha = 5$ and rate $\lambda = 2$. Reviewing the moment errors for 3, 4 and 5 stages does not reveal any deeper problem caused by numerical instability. We will only state the result for the last case

Original moments:

```
[1]      2.50      7.50     26.25    105.00    472.50
[6]  2362.50 12993.75 77962.50 506756.25 3547293.75
```

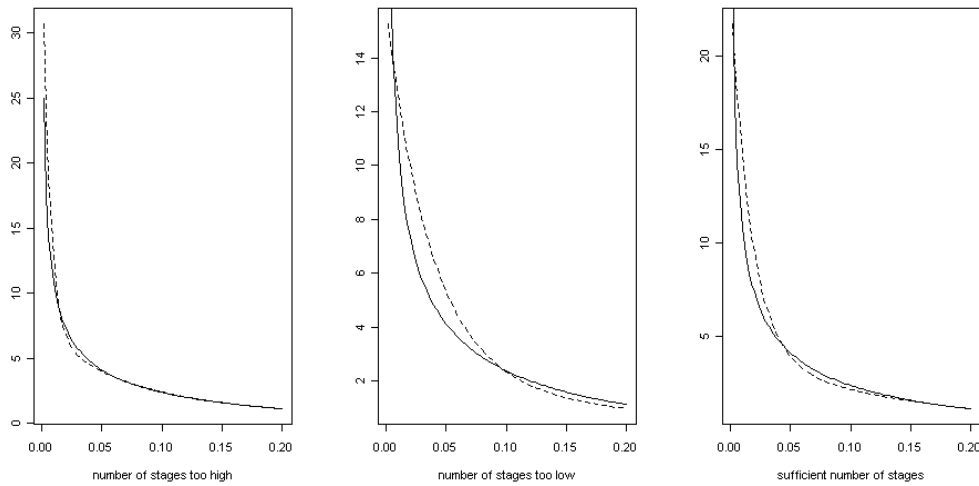


Figure B.1: Impact of the number of stages on the quality of approximation

ME moments:

[1]	2.50	7.50	26.25	105.00	472.50
[6]	2362.50	12993.75	77962.50	506756.25	3547293.75

Absolut Difference (moments):

[1]	5.151435e-13	4.225953e-12	4.751719e-10	4.731731e-09
[5]	5.767384e-08	3.204436e-07	2.173651e-06	1.352619e-05
[9]	1.020455e-04	9.288909e-04		

Error in % (moments):

[1]	0	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---	---

Comparing the density functions in figure B.2 yields a different picture. Only the matrix exponential representation based on a 5-stage approximation admits a valid probability distribution. Raising the number of stages led to a valid result. Although this is not always the case and success cannot be taken for granted, it lays out a potential strategy. But increasing the level of approximation far beyond the optimal level not only leads to numerical instabilities, it may also result in fluctuating approximations. This hysterical behaviour should be avoided under all circumstances.

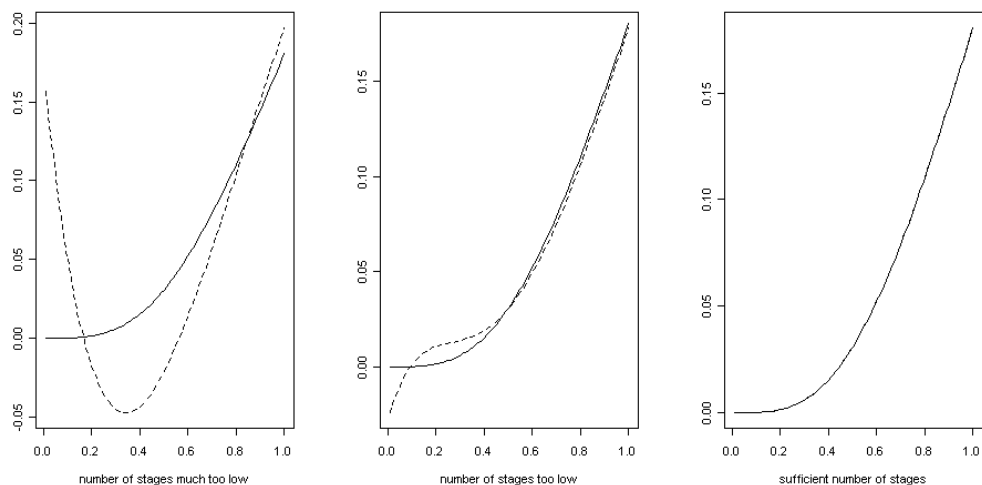


Figure B.2: Impact of the number of stages on the validity of the approximation

B.3 Classpad 300

At the time of writing a new operating system for the Classpad 300 series of table calculators has been released. This OS 3.03 has been the first, which allows return variables to be passed from a subroutine (not a function) to the calling routine. Although this feature has become available, we decided to store a history of parameters and results in global variables of the list type. As these lists are immediately available in the statistics application of the Classpad 330, this allows for printing charts and graphs without an additional line of code. However, due to a bug in all operating systems of version 3, the chunks of code responsible for building up the history could not be relocated to a common subroutine. Accordingly it has been replicated in procedures $\text{qmmck}(r, c, k)$ and $\text{qggc}(r, c, ct, cs)$. It turned out, that these code chunks adversely affect the readability of the source code and so we decided to show only the essential part of the algorithm. In order to provide executable routines we simply used a list argument as return value. It is up to the reader to choose the reduced program versions presented in figure B.3, B.4, B.5, B.6 and B.7 or the equivalent ones with history buildup, which are available from our website.

qmmck	N r,c,k	
<pre> setdecimal:local u,i,h1,h2,ll,lq,pb,pd r/c⇒u:0⇒pd seq(qn,qn,0,c-1,1)⇒qn:seq(0,qp,0,k,1)⇒qp if r=c:then c^qn⇒h1:k+1-c⇒h2 else r^qn⇒h1 (1-u^(k+1-c))/(1-u)⇒h2 ifend 1/(sum(h1/qn!)+h2*r^c/c!)⇒qp[1] for 2⇒i to k+1 qp[i-1]*r/min(i-1,c)⇒qp[i] next seq(qn,qn,0,k,1)⇒qn:sum(qn*qp)⇒ll sum(sublist(qn,1,k-c+1)*sublist(qp,c+1,k+1))⇒lq if c<k:then sum(sublist(qp,c+1,k))⇒pd ifend:qp[k+1]⇒pb return ((pb,pd,ll,lq,qp)) </pre>		

Figure B.3: Classpad 330 implementation of the $M/M/c/K$ queueing model

qxhlp	F r,c	
$\min(0.2499975, (1-r/c) \cdot (c-1) \cdot (\sqrt{(4+5 \cdot c)-2}) / 16/r)$		

Figure B.4: Supporting routine

qxmdc	F r,c	
$(1+qxhlp(r,c))/2$		

Figure B.5: Supporting routine

qxdmc	F r,c	
$(1/2-2 \cdot qxhlp(r,c)) \cdot e^{(-2 \cdot (c-r)/3/r)}$		

Figure B.6: Supporting routine

q99c	N r,c,ct,cs	
------	-------------	--

```

setdecimal
local n,u
local lq,pb,pd,lqm,lqd,lqk,lqp,lqa
r/c⇒u
1/(r^c/c!/(1-u)+Σ(r^n/n!,n,0,c-1))⇒qp[1]
qp[1]×r^c/c!/(1-u)⇒pd
pd×u/(1-u)⇒lq
lq×qxmdc(r,c)⇒lqd
lq×qxmdc(r,c)⇒lqm
r×pd×(ct+cs)/2/(c-r)⇒lqa
(ct+cs)/((1-ct)/lqd+(1-cs)/lqm+(2×(ct+cs)-1)/lq)⇒lqk
cs×(1-ct)×lqd+ct×(1-cs)×lqm+ct×cs×lq⇒lqp

if ct>1:then
  lqp⇒lq
else:lqk⇒lq
ifend

return ({lq,lqa})

```

Figure B.7: Classpad 330 implementation of the $G/G/c$ queueing model

The Classpad 300 programs are called from the main application very much the same way as the corresponding R functions. They accept all parameters in the same order, so we will not repeat any details here. In opposite to the R functions all parameters and performance indicators are recorded each time a certain program is invoked. By default up to 15 values are stored per parameter (This can be changed by modifying the variable E in each program). Each history is held in a global variable of the list type. The input parameters r , c and k are repeated in qrr , qc and qk , respectively. The performance indicators p_b , p_d , L_q and L are recorded by the list variables qpb , qpd , qlq and ql . These variables may be used right away in the statistics application to generate graphs comparing any performance indicators of interest. A similar mechanism is also available for the steady state probabilities computed by the $qmmck(r, c, k)$ program. The recent probability vector is stored in qp , whereas the former one may be accessed using the list variable qpp . The corresponding indices are maintained in the variable qn . This allows the steady state distribution of two separate $M/M/c/K$ models to be compared. However, it is advisable to calculate the results for the model with a smaller value for K first. Otherwise some tail probability values might be omitted when generating a graph. According to our experience

the `xyline` type of graph produces the best results. To get a better understanding of the history mechanism, just run several models from the main application and check the contents of the history variables by selecting the variable manager from the leftmost menu.

B.4 Source Codes

The entire set of source codes for both platforms is available from the website <http://www.telecomm.at>. For some background information, please browse to the **Research** section. If you are only interested in downloading the libraries, please directly proceed to the **Resources** section. The R implementation is provided in the form of standard text files `queueing.r` and `qmeapprox.r` respectively, whereas the Classpad 330 programs and functions are contained in a single `queueing.vcp` file. The latter may be opened by the accompanying software or the Classpad Manager application. In case of suggestions, improvements or any other feedback, the author can be reached at the email address given in the **Contacts** section of the website.

Appendix C

CTI Protocols and APIs

In this appendix we will deal with several approaches to *computer telephone integration (CTI)*. This suggests the existence of two different worlds, which have to be aligned with each other. In fact, traditional telephony and computer environments have been separated by architecture for the last decades. More recently we do also meet communication platforms, which are entirely built of common computing components. Some claim, that this is the end of CTI, because the core elements of a telephone system are assembled from standard computer components. Furthermore we do find application programming interfaces such as SIP servlets, which are closer related to an API for a protocol stack rather than a CTI API. However, we do not agree to an ending of CTI. In fact, we are still concerned with finding loosely coupled interfaces between the central communication platforms and adjunct systems. In solving this or similar problems we can greatly benefit from the inheritance of CTI. Accordingly we will broaden our scope and review CTI from a functional perspective. Typical tasks performed by a CTI protocol include

- *routing* or rerouting of calls and sessions
- *event management*
- *call or session control* for telephone and/or multimedia conversations
- *invocation of features* such as call forward, conferencing or instant messaging
- *administrative and maintenance operations*

From the above items, call or session control is of great importance. In traditional environments we do often meet the so called *third party call control*, which is often found in large call or contact centre installations. An adjunct system takes over control of the call and acts on behalf of the user. Examples include IVR applications, voice dialing systems and of course call and contact centre client applications. The typical point of influence is the communication platform or switching system. This is different for *first party call control*, where the computer component directly interacts with the telephone. In the new millenium this scenario has become more important due to the merging of computing with telephone devices. As an example consider a mobile phone, which automatically dials a number on request of a *personal digital assistant (PDA)*. First party call control is also more prevalent in modern communication platforms, e.g. consider the SIP servlets approach. It has become much simpler to directly interact with the communication entities and protocols than has been with traditional telephone systems. But the same approach also led to an omission of safety mechanisms as a consequence of tighter integration. With respect to reliability and security it has become essential to prevent malicious control entities from causing damage to the primary communication.

As we cannot discuss all CTI protocols of interest we have chosen two representatives, i.e. the *adjunct switch application interface (ASAI)* developed by AT&T and the *call processing language (CPL)* used by recent communication platforms implementing the SIP or H.323 protocol suite. Although the former is proprietary, it perfectly illustrates the functionality demanded from a typical CTI protocol. The latter has been selected, because it well demonstrates the capabilities of third party control for distributed communication platforms. Both protocols belong to the mainstream of CTI protocols, but our description given above allows for more. Accordingly we will give some indications on two extreme representatives in the last section of this appendix. These are the *intelligent network application protocol (INAP)* and the SIP servlet technology.

C.1 Adjunct Switch Application Interface

The *adjunct switch application interface (ASAI)* has been specified by Lucent Technologies former AT&T and is now an intellectual property of Avaya. ASAI is strongly related to TSAPI. Several vendors, e.g. Genesys, Hewlett

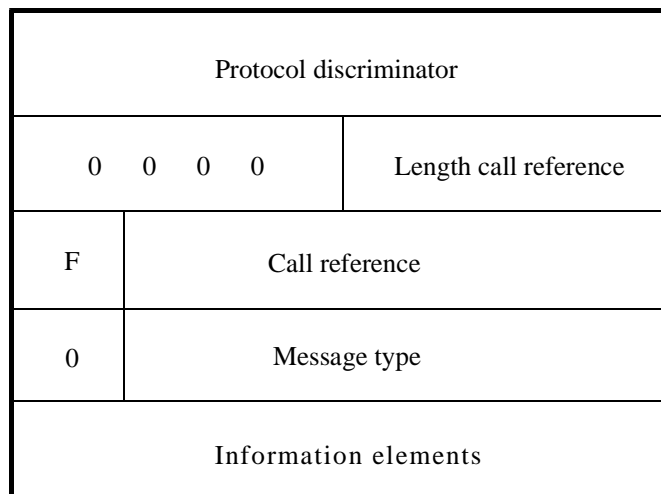


Figure C.1: ASAI message format

Packard and IBM implemented the ASAI protocol stack in their own products to be compatible to Avaya telephone switches. ASAI is strongly related to ISDN as used in the United States. Up to 1996, the ASAI protocol was transported over up to four 56kbit/sec data channels using an ASAI-ISDN interface in the switch and a basic rate ISDN board in the adjunct host. With the increasing demand in TCP/IP based communication, newer connections facilitate IP to encapsulate the ASAI BRI channels. The switch-side ASAI platform has changed and the host uses a simple twisted pair 10Mbit Ethernet card. The number of channels carried by such an interface have been increased to 8.

But ASAI is not only a protocol, Lucent Technologies also offered an application interface called *Call Visor ASAI*, which will not be discussed in this text.

C.1.1 Protocol Overview

As already mentioned, ASAI follows the ISDN framework used in the United States. Therefore layers 1 and 2 will be omitted in the following description. If ASAI information is conveyed across a TCP/IP connection, this connection is treated as layer 1 by ASAI. In order to determine, if this layer is active, ICMP requests using ping are transmitted between switch interface and host.

message type octet coding								hex	message name
8	7	6	5	4	3	2	1		
0	0	0	0	0	0	0	0	\$00	ESCAPE CODE, refer to table C.2
0	1	0	0	0	1	1	0	\$46	RESTART
0	1	0	0	1	1	1	0	\$4e	RESTART ACKNOWLEDGE
0	1	0	1	1	0	1	0	\$5a	RELEASE COMPLETE
0	1	1	0	0	0	1	0	\$62	FACILITY
0	1	1	0	0	1	0	0	\$64	REGISTER
0	1	1	1	1	1	0	1	\$7d	STATUS

Table C.1: Codes for message types

message type octet coding								subsequent octet								message name
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	MGMT INFO

Table C.2: Message type encoding for the subsequent octet

Layer 3 follows ITU recommendation Q.931 very closely. The message format is shown in figure C.1. For information on each field, refer to the network layer of ISDN in section 5.2.2, as only the specifics of ASAI are discussed here.

The *protocol discriminator* identifies the ASAI protocol stack and is always set to binary 00001000. The *call reference* used to associate messages transmitted has a length of either 1 byte or 2 bytes. As in the ISDN framework, the flag identifies the direction of message flow and therefore avoids the assignment of the same call reference value on each side. Only few Q.931 messages are used in ASAI, all possible *message types* are summarized in table C.1. Please note the two-byte network specific message used for exchange of management information. Network specific messages are identified by bit 8 set to 1 in the subsequent octet. The *information elements* are divided in two subgroups, general information elements and application specific information elements. The former are based on Q.931 and utilizes codeset 0, whereas the latter belong to codeset 6. For details, refer to tables C.3 and C.4.

In order to describe the operation of the ASAI protocol, it is important to understand the concept of *associations*. An association is the exchange of one or more messages across the interface between switch and adjunct host.

IE identifier coding								hex	information element name
8	7	6	5	4	3	2	1		
1	-	-	-	-	-	-	-	\$9x	single-byte information elements SHIFT (L=1 locked, XXX=codeset)
	0	0	1	L	X	X	X		
0	-	-	-	-	-	-	-		variable length information elements
	0	0	0	1	0	0	0		\$08 CAUSE
	0	0	0	1	1	0	0		\$0c CONNECTED NUMBER
	0	0	1	0	0	0	0		\$10 CALL IDENTITY
	0	0	1	0	1	0	0		\$14 CALL STATE
	0	0	1	1	1	1	0		\$1e PROGRESS INDICATOR
	0	1	0	1	0	0	1		\$29 DATE/TIME
	1	1	0	1	1	0	0		\$6c CALLING PARTY NUMBER
	1	1	1	0	0	0	0		\$70 CALLED PARTY NUMBER
	1	1	1	0	1	0	0		\$74 REDIRECTING NUMBER
	1	1	1	0	1	1	0		\$76 REDIRECTION NUMBER
	1	1	1	1	0	0	1		\$79 RESTART INDICATOR
	1	1	1	1	1	1	0		\$7e USER-TO-USER INFORMATION

Table C.3: ASAI Codeset 0 information elements

An ASAI association is identified by the *call reference value (CRV)*. Three of the messages stated above, are used to control associations

1. The REGISTER message initiates an ASAI association
2. The RELEASE COMPLETE message releases an ASAI association
3. The FACILITY message is used to convey information elements without initiating or releasing an association

The remaining messages are used for maintenance purposes. The following types of ASAI associations exist

- *Third Party Call Control* - The adjunct may initiate, terminate, monitor and control a single specific call from the remote end.
- *Domain Control* - With station domain control, the adjunct may initiate, terminate, monitor and control all calls related to a specific station. With ACD split domain control, the adjunct receives agent status reports for a specific split or skill.

IE identifier coding								hex	information element name
8	7	6	5	4	3	2	1		
0	0	0	0	0	0	0	1	\$01	ORIGINATING LINE INFORMATION
0	0	0	0	0	0	1	0	\$02	USER ENTERED CODE
0	0	0	0	0	0	1	1	\$03	RESOURCE STATUS
0	0	0	0	1	0	1	0	\$0a	TRUNK IDENTIFICATION
0	0	0	0	1	0	1	1	\$0b	TRUNK GROUP/TRUNK STATUS
0	0	0	1	0	1	1	1	\$17	OLD PARTY IDENTIFIER
0	0	0	1	1	0	1	1	\$1b	VERSION
0	0	0	1	1	1	0	0	\$1c	FACILITY
0	1	0	0	0	1	0	0	\$44	PARTY ID
0	1	0	0	0	1	1	0	\$46	COUNTER
0	1	0	0	0	1	1	1	\$47	SPECIFIC EVENT
0	1	0	0	1	0	0	0	\$48	FEATURE
0	1	0	0	1	0	0	1	\$49	DOMAIN
0	1	0	0	1	0	1	0	\$4a	CONFERENCE/TRANSFER OPTIONS
0	1	0	0	1	0	1	1	\$4b	CALL OPTIONS
0	1	0	0	1	1	0	1	\$4d	ITEM
0	1	0	0	1	1	1	0	\$4e	SERVICE CIRCUIT
0	1	0	0	1	1	1	1	\$4f	STATUS
0	1	0	1	0	0	0	1	\$51	RESOURCE IDENTIFIER
0	1	0	1	0	0	1	0	\$52	DATA ITEM
0	1	0	1	0	0	1	1	\$53	DATA BIT MAP
0	1	0	1	0	1	1	0	\$56	GENERIC BILLING
0	1	1	1	1	0	1	0	\$7a	MANAGEMENT INFORMATION
0	1	1	1	1	0	1	1	\$7b	LOOKAHEAD INTERFLOW
0	1	1	1	1	1	0	0	\$7c	UNIVERSAL CALL IDENTIFIER

Table C.4: ASAI Codeset 6 information elements

- *Notification* - The adjunct may request and cancel event reporting on certain calls. These event reports are sent by the switch for all new calls entering a specific ACD split or vector directory number (VDN) domain. VDNs provide logical entry points for calls to the switch. They are often related to certain applications in a call centre, e.g. support hotline, order management.
- *Routing* - The switch may request the destination for a specific call from the adjunct. Based on parameters delivered such as calling party, called party and collected digits the adjunct host derives a destination address and provides the result to the switch.
- *Request Features* - The adjunct may request the invocation of certain switch dependent features such as agent login, agent logout or call forward.
- *Value Queries* - The adjunct may request information about a specific switch resource. In contrast to domain control and notification, only a single result is provided.
- *Set Value* - The adjunct may request the switch to set certain switch related flags such as the message waiting indication of a telephone set.
- *Maintenance* - Switch and adjunct host may determine ASAI link states and recover from failures.

C.1.2 ASAI and UUS

As discussed in section 5.4.1, user-to-user signaling provides a powerful tool for transferring call centre related data over ISDN and QSig trunks. Fortunately, Avaya equipment allows the exchange of user-to-user information elements between adjunct hosts. In a multisite environment, the switches have to be interconnected using ISDN or QSig trunks, whereas the former are recommended for public use and the latter are reserved for private tie trunks. Naturally, public trunks are more restricted due to facts such as the availability of the user-to-user signaling supplementary service or the maximum length allowed for user-to-user information elements. In order to keep problems with UUS over ASAI to a minimum, the length of user-to-user data is not allowed to exceed 32 bytes.

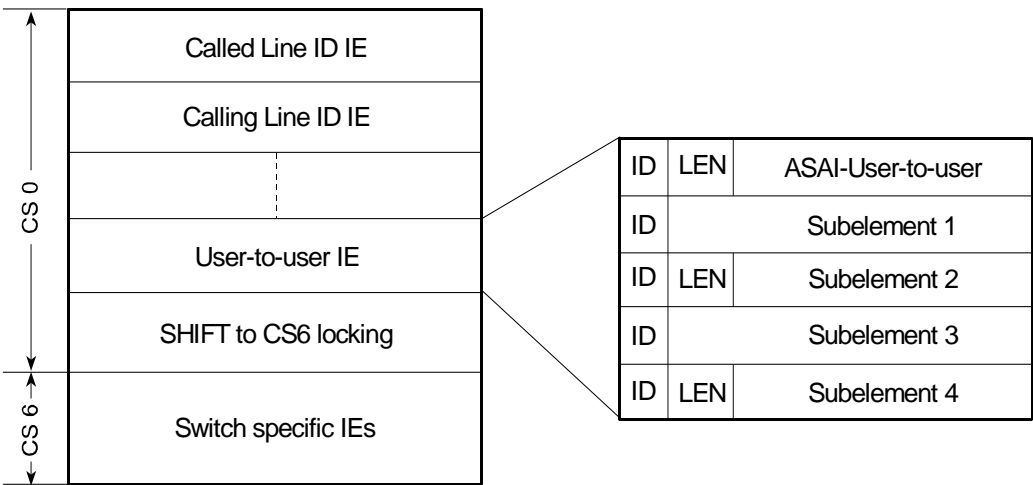


Figure C.2: Shared UUI treatment in Avaya telephone switches

In a multisite environment, the switch simply embeds the user-to-user data received from ASAI in a user-to-user information element to be sent in a SETUP or DISCONNECT message. Please note, that only UUS type 1 is supported. As the switches itself utilize UUS to transmit switch and call centre specific information such as collected digits, a proprietary mechanism for the shared use of UUS has been implemented. Each user-to-user information element now contains a certain number of *subelements* arranged according to a simple priority mechanism. Similar to information elements, each subelement is identified by a type. In case the subelement varies in length, a separate length indicator is included as well. Each subelement relates to a proprietary item such as queueing times and digits collected during queueing time. In addition to switch and call centre related subelements, an ASAI UUI field has been implemented. Within this subelement, any information sent by the adjunct host is carried over ISDN trunks with shared UUI treatment. Refer to figure C.2 for a graphical representation of a message fragment containing shared user-to-user information. Each subelement is assigned a priority, which helps to identify the less important items to be dropped in case, the length restrictions for user-to-user information elements are exceeded. Clearly, such a mechanism is only applicable between Avaya switches and has to be turned off, if platforms from other vendors are to be integrated in a multisite environment. In any case, all switches have to

support transparent UUS to enable adjunct hosts to transfer user specific data together with the call.

In terms of message length a less restrictive option than UUS is offered by Avaya for QSig networks. *Manufacturer specific information (MSI)* may be conveyed in envelopes across heterogenous switch environments. As foreign switches cannot process these envelopes, adjuncts can only be connected to the switches of a single family, namely Avaya. This places a major drawback to the entire solution. Whether to use UUS or MSI depends on the overall architecture of the voice network and the multisite call centre environment.

C.1.3 Sample ASAI message

A sample ASAI message is shown in table C.5. Please note the hierarchical structure of the message. Codeset 0 and codeset 6 information elements are embedded in the Q.931 section, which in turn is part of a single FACILITY information element. Please note the first SWITCH to codeset 6 information element. With bit 4 set to 0, codeset 6 has been locked for the entire message. At this point the message only carries a single FACILITY information element, as the other information has been embedded in that message. This sample message only serves an illustrative purpose, as some details such as bit codings have been omitted. For a detailed reference on the ASAI protocol refer to [12], information about ASAI call handling and feature interactions can be found in [13].

C.2 Call Processing Language

Although the *call processing language (CPL)* belongs to the family of CTI protocols, it significantly differs in architecture from the previous approach. CPL has been defined in RFC 2824 and RFC 3880 by the *Internet Engineering Task Force (IETF)*. It is an XML based scripting language, executed on a signaling server such as H.323 gatekeeper, SIP proxy, SIP registrar or SIP redirector. Each script is associated with a telephone number or URL and invoked, when a call establishment request arrives at the signaling server. Accordingly the process of user location is intercepted and its outcome under control of the script. This is similar to arming trigger points and invoking services in the intelligent network.

Notwithstanding CPL presents a common script language for H.323 and

message bytes	description
\$08	protocol discriminator 00001000 = \$08
\$02	0000, CRV length = 0010 = \$02
\$62	message type FACILITY
\$96	SWITCH to CS6 locking
\$1c \$38	FACILITY IE, length = \$38 (99)
\$91	Q.932 supplementary service
\$a1 \$35	INVOKE, length = \$35 (53)
\$02 \$01 \$02	INVOKE ID tag, length = \$01, id = \$02
\$02 \$01	OPERATION VALUE tag, length = \$01
\$95	OPERATION value = \$95 (EventReport)
\$40 \$58	Q.931 information elements tag, length = \$58 (88)
\$10 \$02 \$03 \$6b	CALL ID tag, length = \$02, id = \$036b (875)
\$6c \$05	CALLING PARTY IE, length = \$05
\$80	CALLING PARTY numbering plan = \$80 (unknown)
\$31 \$32 \$33 \$34	CALLING PARTY number = "1234"
\$70 \$05	CALLED PARTY IE, length = \$05
\$80	CALLING PARTY numbering plan = \$80 (unknown)
\$34 \$33 \$32 \$31	CALLED PARTY number = "4321"
\$7e \$05	USER-TO-USER IE, length = \$05
\$00	USER-TO-USER protocol discriminator = \$00 (User Specific)
\$54 \$45 \$53 \$54	USER-TO-USER information = "TEST"
\$96	SWITCH to CS6 locking
\$47 \$01	SPECIFIC EVENT IE, length = \$01
\$91	SPECIFIC event = \$91 (Incoming Call)
\$49 \$05	DOMAIN IE, length = \$05, domain = \$8c (VDN)
\$31 \$31 \$31 \$b1	DOMAIN info (VDN) = "1111"
\$7b \$08	LOOK-AHEAD-INTERFLOW IE, length = \$08
\$82	LOOK-AHEAD-INTERFLOW pri = 0, type = vector interflow
\$8f \$80 \$b8	LOOK-AHEAD-INTERFLOW timestamp = 15:00:56
\$54 \$45 \$53 \$54	LOOK-AHEAD-INTERFLOW VDN-name = "TEST"

Table C.5: Sample ASAI message

node type	description
switch	script flow control
location modifier	location management
signaling action	instructions related to signaling
non-signaling action	mail and log operations

Table C.6: List of CPL nodes

criteria	decisions are based on
address	address information of call request
string	string match
time	time and/or date
priority	priority of original call

Table C.7: Criteria for switch nodes

SIP related environments, it radically differs in mapping to the corresponding protocol elements. This is relatively straightforward for SIP, but much more difficult for H.323. Due to the binary nature of the latter the majority of protocol elements have to be converted. Furthermore special attention is necessary with respect to the information conveyed in H.323 *user-to-user information element (UUIE)*. These elements are extensions to the protocol suite, which have not been covered by Q.931. Of special interest to CPL are the user-to-user information elements `sourceAddress` and `language`.

Besides some ancilliary information used for execution each CPL script consists of language primitives called *nodes*. These nodes may possess one or more parameters and an output, both encoded as XML tags. Stating another node as output leads to a network of nodes, which allows for the execution of complex decisions. In table C.6 we have summarized the nodes known to CPL scripts [5]. In order to release the CPL developer from carrying the location of the call from one node to the other, it has been defined as global implicit variable, which can be accessed by location modifier instructions. As noted in the table, switch nodes are responsible for script flow control. Branching is possible based on the criteria listed in table C.7.

At the time of writing CPL still has been an item of active research. However, the core functionality has been in place for years and can be considered as stable technology. As to be expected from the complexity arising for H.323 the major part of research focuses on SIP related entities. Items

of discussion have been a standardized secure script upload mechanism and inclusion of context awareness. For more information refer to the article [5] by E.Z. Eslami.

C.3 Edge of CTI

In the introduction to this appendix we have mentioned two concepts, which reside on the boundary of our CTI definition. These were the *intelligent network (IN)* and an API called SIP servlets. The former is well documented in a series of standards issued by the ITU. The corresponding standardization efforts have been carried out in the last millenium with application to graphical user interfaces (GUI) in mind. We do not delve into any details here and refer to the book [6] by I. Faynberg, L.R. Gabuzda, M.P. Kaplan and N.J. Shah. However, no programming API has been defined by the ITU. Among others one of the most popular is *Java for integrated networks (JAIN)*. As suggested by its name, JAIN is based on the Java programming language. In the context of JAIN a series of concepts and protocols have been specified, which allow for implementation of a Java based *service logic execution environment (SLEE)*. Doing so will replace a significant portion of more or less proprietary IN systems by their Java counterparts. This also leads to the use of open interfaces and APIs not directly related to telephony and communication. Examples include *Java database connection (JDBC)* and *remote method invocation (RMI)*. Although these efforts led to an extremely well defined and powerful environment for creation of public network services, some have considered the framework of JAIN as too complex. This led to vendor specific protocol implementations such as the switch API by Telesoft Technologies. By encoding *transaction capabilities application part (TCAP)* and *intelligent network application part (INAP)* messages in XML and conveying them across an IP network a leightweight alternative to the traditional IN protocol stack has been defined. No API is necessary, because comfortable XML libraries already exist and may be downloaded from the internet at no cost.

Turning attention to SIP servlets we again face the Java programming language. As suggested by the name, SIP servlets are based on servlet technology similar to HTTP servlets [9]. Like IN, it is an exteme case of CTI, as only an API without related CTI protocol has been specified. Instead it directly relates to the underlying signaling protocol. Accordingly SIP servlets

are better seen as a mean to implement services on top of SIP. So we are rather close to an API for programming the SIP protocol stack. Another indication for that is given by the tight coupling between SIP servlets and the application roles defined for SIP, which are UAC, UAS, proxy and back-to-back UA.

From a programming perspective, the SIP servlet API invokes appropriate methods on arrival of a SIP message or response. Accordingly there are two methods available, which are `doRequest` and `doResponse`. Please note, that method belongs to object oriented jargon and is best described as function related to an object. To simplify development further, the SIP servlet framework also provides standard implementations for `doRequest` and `doResponse`. These allow for more granularity with respect to the messages processed. Accordingly we do find methods such as `doInvite`, `doAck`, `doPublish`, `doErrorResponse` and `doRedirectResponse` for each SIP request or response. Obviously these methods are sufficient for passive purposes. When a new request is to be invoked, the factory interface `SipFactory` provides means for creation of the necessary element. Furthermore the SIP servlet framework defines utility classes, which implement features such as a timer service.

SIP servlets are a powerful technology, which perfectly fit into the concept of IMS. The natural convergence with HTTP servlets eases development of converged applications such as click-to-dial. On the other hand, the SIP servlets API missed out thread safety, which leads to complications in the interaction of multiple services. Furthermore state changes have to be maintained explicitly by the developer, i.e. there is no concept such as the *basic call state model (BCSM)* of IN. However, it is expected, that the former shortcoming will be eliminated in the near future. For more details on SIP servlets please refer to the book [3] by C. Boulton and K. Gronowski.

Appendix D

System Administration Examples

In discussing call centre interconnection, communication protocols and CTI our presentation adopted a system independent viewpoint. This has been achieved by closely following the existing reference standards. However, in this appendix we do exactly the opposite by delving deeply into the details of an operating call centre architecture. For our purposes we have chosen the Lucent Technologies' Definity G3 series of telephone switches, because they exhibit a versatile portfolio of parameters most suitable to our needs. We have chosen releases between R6 and R9, so the entire system architecture is TDM based, i.e. we are talking about traditional telephone switches. More recently the Definity series of telephone switches has been replaced by distributed communication platforms, but most administrative details can still be found in this new architecture. Please note, that all components are now intellectual property of Avaya. For further reference, please consult their supporting website for system related documentation.

All examples have been prepared from screen forms provided by the system terminal. The most important entries have been annotated in the examples or mentioned in the description of the corresponding screen form. We encourage the reader to scroll back to the relevant chapter and identify the related theoretical or standard based items. We hope, that our highlighting provides a guideline to the jungle of administration, as a vast amount of system specific entries have to remain unexplained.

D.1 Interconnections

In this section we will proceed from bottom up. Starting from lower layers, i.e. trunk setup and signaling channel configuration, we will end up with issues of addressing and telephone routing. For the latter the interrelation between trunk access and number plan will be investigated by presenting further details of *automatic route selection (ARS)*, *automatic alternate routing (AAR)* and the *uniform dial plan (UDP)*.

D.1.1 Public ISDN access

Interface capabilities are defined using the DS1 administration form. These include interface protocol (ETSI, national ISDN, QSig), line companding (e.g. A-Law, My-Law) as well as the line coding (e.g. hdb3). The properties of the ISDN signaling channel are specified in the signaling group administration form. It is possible to redefine the channel number for the signaling channel and the supplementary service protocol type. In order not to overload this section, only trunk group forms are presented in this subsection.

```

display trunk-group 1                                     Page 1 of 19  SPE A
                                     TRUNK GROUP

Group Number: 1                      Group Type: isdn      CDR Reports: y
Group Name: ISDN PRI                  COR: 1              TN: 1      TAC: 89
Direction: two-way                   Outgoing Display? n
Dial Access? y                       Busy Threshold: 255   Night Service:
Queue Length: 0
Service Type: public-ntwrk           Auth Code? n      TestCall ITC: rest
Far End Test Line No:

TestCall BCC: 4
TRUNK PARAMETERS
  Codeset to Send Display: 0
  Max Message Size to Send: 260
  Supplementary Service Protocol: c
  Digit Treatment: insertion
  Trunk Hunt: cyclical
  Connected to Toll? n
  Calling Number - Delete:
  Bit Rate: 9600
  Disconnect Supervision - In? y Out? y
  Answer Supervision Timeout: 0

  Define for ETSI - ISDN
  public network connections
  Codeset parameters
  Codeset to Send TCM,Lookahead: 6
  Charge Advice: automatic
  Digit Handling (in/out): overlap/overlap
  Digits: *5
  Define digit treatment for
  incoming calls
  STT Loss: normal
  DTT to DCO Loss: normal
  Synchronization: sync
  Duplex: full
  
```

Trunk setup for the Definity PBX system - Note the supplementary service protocol settings (c = ETSI) and codeset parameters.

```

display trunk-group 1                                     Page 2 of 19  SPE A
TRUNK FEATURES
  ACA Assignment? y          Measured: both          Wideband Support? n
Long Holding Time(hours): 1          Maintenance Tests? y
Short Holding Time(sec): 10      Data Restriction? n      NCA-TSC Trunk Member: 1
Short Holding Threshold: 15      Send Name: n          Send Calling Number: y
Used for DCS? n                  Send Connected Number: n
Suppress # Outpulsing? y      Numbering Format: unknown
Outgoing Channel ID Encoding: preferred      UUI IE Treatment: service-provider
  Charge Conversion: 1
  Decimal Point: none
  Currency Symbol:
  Charge Type: units
                                     Handle user-to-user transparently
                                     ^
Send UCID? y ← Send additional switch specific
Send Codeset 6/7 LAI IE? y ← information

```

Trunk setup for the Definity PBX system - Note the parameters for name/number identification services and the treatment option for user-to-user signaling

```

display trunk-group 1                                     Page 3 of 19  SPE A
TRUNK GROUP
Administered Members (min/max): 1/30
Total Administered Members: 30
GROUP MEMBER ASSIGNMENTS
  Port   Code Sfx Name      Night      Sig Grp
1: 01C1901 TN464 F          Night      1
2: 01C1902 TN464 F          Night      1
3: 01C1903 TN464 F          Night      1
4: 01C1904 TN464 F          Night      1
5: 01C1905 TN464 F          Night      1
6: 01C1906 TN464 F          Night      1
7: 01C1907 TN464 F          Night      1
8: 01C1908 TN464 F          Night      1
9: 01C1909 TN464 F          Night      1
10: 01C1910 TN464 F         Night      1
11: 01C1911 TN464 F         Night      1
12: 01C1912 TN464 F         Night      1
13: 01C1913 TN464 F         Night      1
14: 01C1914 TN464 F         Night      1
15: 01C1915 TN464 F         Night      1

```

Trunk setup for the Definity PBX system - This page is used to assign single channels to a trunk group. Note the association of a signaling group to each channel

D.1.2 Private Networking

Similar to public network access, the administration forms for DS1 interface, signaling group and trunk group have to be filled in. To illustrate the differences between private ISDN trunks and QSig trunks, all relevant forms will be shown twice. In some respect private networking administration is more difficult than configuring a trunk for public network access. First, both sides need to be aligned. In case of QSig, the relevant entries are found on the DS1 form. If supplementary service support is desired, an appropriate protocol has to be chosen. An adapted US ISDN protocol exists for such purposes, but in heterogenous environments QSig is the right choice. Last but not least additional switch or call centre specific links may be established based on the *distributed communications system (DCS)*, which is implemented only with AT&T, Lucent Technologies and Avaya products.

ISDN trunk setup

```

display ds1 01a16                                     SPE R
                                                    DS1 CIRCUIT PACK

Location: 01A16                                     Name:
Bit Rate: 2.048                                     Line Coding: hdb3

Signaling Mode: isdn-pri
Connect: pbx

Interface Companding: alaw
Idle Code: 01010100

Country Protocol: etsi
Protocol Version: a
CRC? y

Interface: network
Country Protocol: etsi
Protocol Version: a
CRC? y

DCP/Analog Bearer Capability: 3.1kHz

Slip Detection? y                                Near-end CSU Type: other
  
```

Define ISDN tie trunk
with ETSI signaling
(side a)

DS1 interface paremeters are specified here - Note the interface and country settings for ISDN

```

display signaling-group 1                               Page 1 of 5  SPE R
                                SIGNALING GROUP

Group Number: 1
Associated Signaling? y      Max number of NCA TSC: 0
Primary D-Channel: 01A1616  Max number of CA TSC: 0
                                Trunk Group for NCA TSC:
Trunk Group for Channel Selection: 1
Supplementary Service Protocol: a

Define D-channel protocol ETSI-ISDN signaling
Declare channel 16 as the signaling channel

```

The signaling group defines the ISDN signaling channel

```

display trunk-group 1                               Page 1 of 10  SPE R
                                TRUNK GROUP

Group Number: 1      Group Type: isdn      CDR Reports: y
Group Name: Tie Trunk      COR: 1      TN: 1      TAC: 19
Direction: two-way      Outgoing Display? n
Dial Access? y      Busy Threshold: 99      Night Service:
Queue Length: 0
Service Type: tie      Auth Code? n      TestCall ITC: rest
Far End Test Line No:

TestCall BCC: 4
TRUNK PARAMETERS
Codeset to Send Display: 0
Max Message Size to Send: 260
Supplementary Service Protocol: c

define ETSI-ISDN tie trunk
Codeset parameters
Codeset to Send TCM, Lookahead: 6
Charge Advice: none
Digit Handling (in/out): enbloc/enbloc

Trunk Hunt: cyclical
Connected to Toll? n      STT Loss: normal      DTT to DCO Loss: normal
Calling Number - Delete:      Insert:      Numbering Format:
Bit Rate: 1200      Synchronization: async      Duplex: full
Disconnect Supervision - In? y Out? y
Answer Supervision Timeout: 0

```

Trunk setup for the Definity PBX system - In opposite to public trunk access a different service type is specified

```

display trunk-group 1                                     Page 2 of 10  SPE R
TRUNK FEATURES
  ACA Assignment? n                                     Measured: internal  Wideband Support? n
                                                         Internal Alert? n    Maintenance Tests? y
                                                         Data Restriction? n  NCA-TSC Trunk Member:
                                                         Send Name: n         Send Calling Number: y
                                                         Send Connected Number: y
  Used for DCS? n
  Suppress # Outpulsing? y                               Numbering Format: public
  Outgoing Channel ID Encoding: exclusive                 UUI IE Treatment: service-provider

  Send UCID? n ← Send additional switch
  Send Codeset 6/7 LAI IE? y ← specific information

```

Trunk setup for the Definity PBX system - Note the settings for supplementary services

QSig trunk setup

```

change ds1 01a16                                         Page 1 of 1  SPE R
DS1 CIRCUIT PACK

Location: 01A16                                           Name: QSig Trunk
Bit Rate: 2.048                                           Line Coding: hdb3

Signaling Mode: isdn-pri                                  Interface: peer-master
Connect: pbx                                               Peer Protocol: Q-SIG
                                                         Side: a
                                                         CRC? y
                                                         Channel Numbering: timeslot
                                                         DCP/Analog Bearer Capability: 3.1kHz

Interface Companding: alaw                                Define QSig master (side a)
Idle Code: 01010100
                                                         European (ETSI) setting

Slip Detection? y                                         Near-end CSU Type: other

```

DS1 interface settings for QSig - Side A (master)

```

change ds1 01a16                               Page 1 of 1  SPE R
DS1 CIRCUIT PACK

Location: 01A16                                Name: QSig Trunk
Bit Rate: 2.048                                Line Coding: hdb3

Signaling Mode: isdn-pri                        Interface: peer-slave
Connect: pbx                                    Peer Protocol: Q-SIG
                                                Side: b
                                                CRC? y
Interface Companding: alaw                      Channel Numbering: timeslot
Idle Code: 01010100                            DCP/Analog Bearer Capability: 3.1kHz
                                                Define QSig slave
                                                (side b)

Slip Detection? y                               Near-end CSU Type: other

```

DS1 interface settings for QSig - Side B (slave)

```

change signaling-group 1                         Page 1 of 5  SPE R
SIGNALING GROUP

Group Number: 1
Associated Signaling? y                         Max number of NCA TSC: 0
Primary D-Channel: 01A1616                     Max number of CA TSC: 0
Trunk Group for Channel Selection: 1            Trunk Group for NCA TSC:
Supplementary Service Protocol: b
                                                No temporary signaling
                                                connections

Define D-channel protocol
for QSig signaling
Declare channel 16 as
the signaling channel

```

The signaling group defines the QSig signaling channel

```

change trunk-group 1                                     Page 1 of 10  SPE A
                                     TRUNK GROUP

Group Number: 1                      Group Type: isdn          CDR Reports: y
  Group Name: QSig Tie Trunk          COR: 1              TN: 1          TAC: 19
    Direction: two-way          Outgoing Display? n
    Dial Access? y              Busy Threshold: 99          Night Service:       
    Queue Length: 0
    Service Type: tie          Auth Code? n          TestCall ITC: rest
    TestCall BCC: 4          Far End Test Line No:       
TRUNK PARAMETERS
  Codeset to Send Display: 6          Codeset parameters
  Max Message Size to Send: 260      Codeset to Send TCM, Lookahead: 6
  Supplementary Service Protocol: b  Charge Advice: none
  Digit Handling (in/out): enbloc/enbloc
  Define for QSig tie trunk
  Trunk Hunt: cyclical
  Connected to Toll? n          STT Loss: normal          DTT to DCO Loss: normal
  Calling Number - Delete:        Insert:                 Numbering Format:       
  Bit Rate: 1200          Synchronization: async          Duplex: full
  Disconnect Supervision - In? y Out? y
  Answer Supervision Timeout: 0

```

Trunk setup for QSig private networking - Note the different country settings

```

change trunk-group 1                                     Page 2 of 10  SPE A
TRUNK FEATURES
  ACA Assignment? n          Measured: internal          Wideband Support? n
                                Internal Alert? n          Maintenance Tests? y
                                Data Restriction? n          NCA-TSC Trunk Member:       
                                Send Name: n          Send Calling Number: y
                                Send Connected Number: y
  Used for DCS? n          Suppress # Outpulsing? y          Numbering Format: public
  Outgoing Channel ID Encoding: preferred          UUI IE Treatment: service-provider
  Send UCID? n          Send Codeset 6/7 LAI IE? y
  Send additional switch specific information
  No proprietary DCS private network connection
  Handle user-to-user info transparently

```

Trunk setup for the Definity PBX system - DCS is definitely disabled

D.1.3 Call Routing with AAR, ARS and UDP

The *automatic route selection (ARS)* table allows for partitioning of outgoing calls. It can also be used to route calls to remote locations over the public network, if no private network is available. With ARS administered properly, calls are routed based on the digits dialed and calling privileges. The ARS digit analysis table is used to determine the route pattern from dialed digits. Each route pattern specifies one or more trunks, which are selected according to the privileges defined by the *class of restriction (COR)* and the *facility restriction level (FRL)*. The class of restriction may be assigned to trunks and stations, whereas the facility restriction level is directly related with the class of restriction.

The *automatic alternate routing (AAR)* table allows for partitioning of calls on interconnection trunks. Similar to the ARS table, privileges may be assigned using CORs and FRLs. AAR codes also build the base for the uniform dial plan. The *uniform dial plan (UDP)* provides a common 4- or 5-digit dial plan, that can be shared among a group of switches thus enabling each location to see a unique systemwide internal numbering plan. UDP calls between locations are routed by converting the dialed number into a private networking number with 7 digits. This number consists of a 3 digit location code and the last 4 digits of the extension number. Based on the type of conversion specified in the UDP table, the first 3 digits are interpreted by either using the AAR digit analysis table or the *extended number portability (ENP) table*. The latter utilizes node number routing instead of routing patterns. Details about the UDP codes, number conversion and routing are shown in figure D.1.

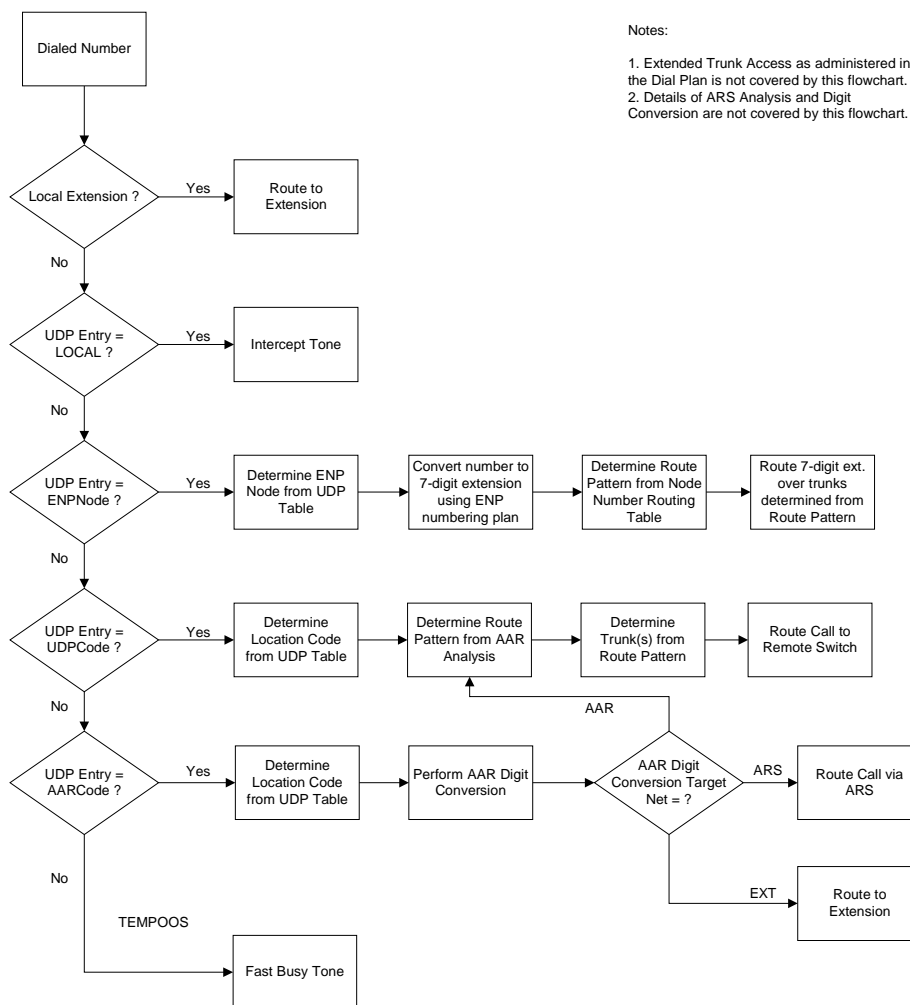


Figure D.1: UDP extension analysis

ARS digit analysis table - Note the association to the route pattern

A typical route pattern used for ARS entries - The route builds the link between trunks and digit treatment. Note the FRL to put restrictions on caller access

AAR digit analysis table - Note the association to the route pattern.

A typical route pattern used for AAR entries - The route builds the link between trunks and digit treatment. Note the digit treatment to recreate the original extension

display udp 23									
Page 1 of 2 SPE A									
UNIFORM DIALING PLAN									
Ext Codes: 23ddx									
Ext Code: 2xxxx		Type:		<div>UDP codes may be set for single extensions or group of extensions</div>					
Ext Code: 23xxx		Type:		UDPCode 802					
dd	Type	dd	Type	dd	Type	dd	Type	dd	Type
0x:		1x:		2x:		3x:		4x:	
00:		10:		20:		30:		40:	
01:		11:		21:		31:		41:	
02:		12:		22:		32:		42:	
03:		13:		23:		33:		43:	
04:		14:		24:		34:		44:	
05:		15:		25:		35:		45:	
06:		16:		26:		36:		46:	
07:		17:		27:		37:		47:	
08:		18:		28:		38:		48:	
09:		19:		29:		39:		49:	

Locations are entered in the UDP table - these entries are interpreted by the AAR or ENP tables

D.1.4 Caller Privileges

In the preceding sections, caller privileges have been mentioned as a mechanism to allow or deny calls on a selective base. Three types of access lists may be associated with each class of restriction, namely the *restricted call list (RCL)*, the *unrestricted call list (UCL)* and the *toll list (TOLL)*. Access control can now be established by entering a number into one of these lists. By adding a number to the RCL or toll list, calls to these numbers may be explicitly forbidden, whereas numbers on the toll list may be explicitly allowed by entering them into the UCL. Besides these parameters, COR exclusions and FRLs still apply. The entire decision graph is shown in figure D.2.

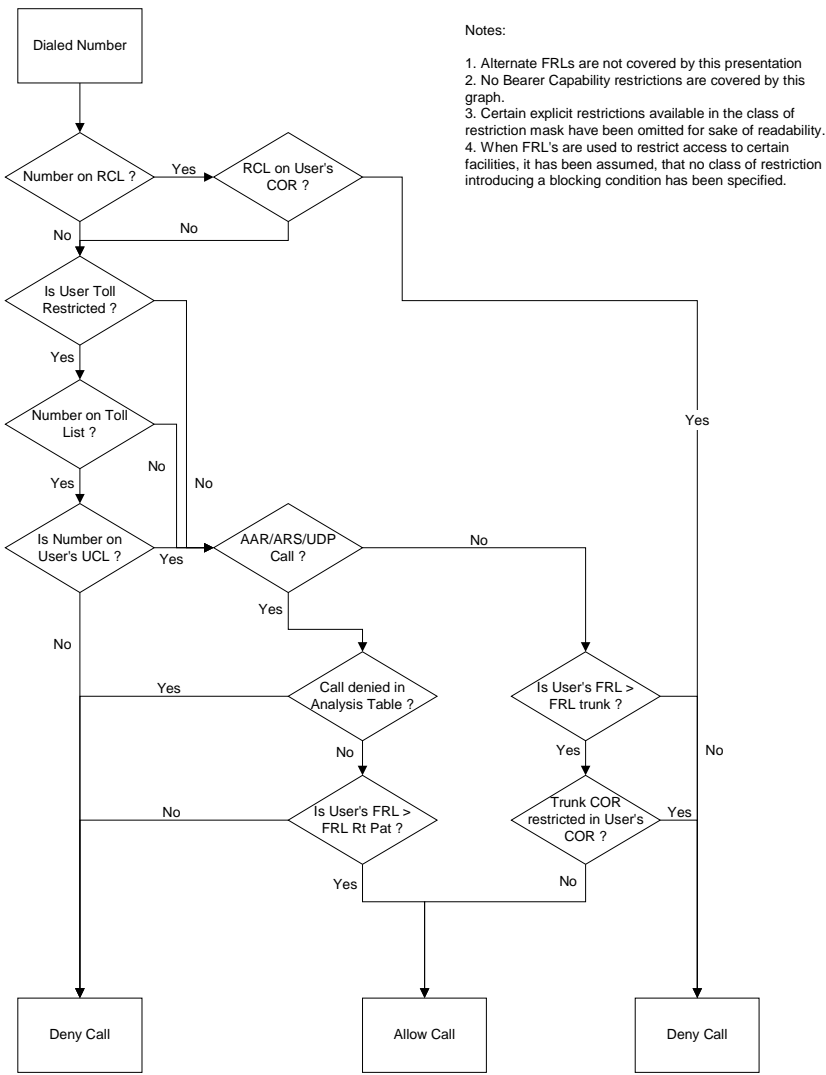


Figure D.2: Caller privileges

```

display cor 11                                     Page 1 of 3 SPE R
                                CLASS OF RESTRICTION

COR Number: 11
COR Description: restr. internat.

FRL: 1
Can Be Service Observed? n
Can Be A Service Observer? n
Time of Day Chart: 1
Priority Queuing? n
Restriction Override: all
Restricted Call List? y
Unrestricted Call List:
Access to MCT? y
Category For MFC ANI: 7
Send ANI for MFE? n
MF ANI Prefix:
Hear System Music on Hold? y

APLT? y
Calling Party Restriction: all-toll
Called Party Restriction: none
Forced Entry of Account Codes? n
Direct Agent Calling? n
Facility Access Trunk Test? n
Can Change Coverage? n
Fully Restricted Service? n
Hear VDN of Origin Annc.? n
Add/Remove Agent Skills? n
Automatic Charge Display? n
PASTE (Display PBX Data on Phone)? n
Can Be Picked Up By Directed Call Pickup? n
Can Use Directed Call Pickup? n
  
```

Via class of restriction, RCL, UCLs and toll lists are associated to station, trunks and other resources

Class of restriction - Note the entries for calling party restriction, RCL and UCL

```

display toll 0                                     SPE R
                                TOLL ANALYSIS
                                Percent Full: 1

Dialled String  Total  Min Max  RCL  Toll List  CDR  FEAC  <--Unrestricted Call List-->
                                1  2  3  4  5  6  7  8  9 10
00              8   25   x
0450            8   25   x
0457            8   25   x
0458            8   25   x
0459            8   25   x
0900            8   25   x
  
```

Entries are either associated to the restricted call list, the unrestricted call lists or the toll list

Using that form entries may be added to RCL, UCL and toll list - Note the flexibility in adding groups of numbers

D.2 CTI Administration

In the following we will only present the switch specific entries related to CTI administration. Although it is necessary to align telephone switch and attached adjunct system, we have omitted the configuration of the latter. Adding another product would adversely affect readability and is beyond the scope of this appendix. From an architectural viewpoint an intelligent communication device has been added to the switch. This so called *LAN gateway* serves as an interface in the communication between PBX and adjunct system. Featuring an Ethernet connector, it relies on TCP/IP to convey the ASAI protocol messages across the network.

D.2.1 Configuration of the LAN Gateway

The following figures show a 9-step process to setup the LAN gateway for adjunct operation. Major tasks include the assignment of the network parameters such as IP address, IP netmask and default router as well as the client access permissions. Please note, that no switch specific parameters are defined during the LAN gateway configuration process.

Release 1.0, Issue 2.0

Main Menu

- 1. Login/Password Administration
- 2. TCP/IP Administration...
- 3. Brouter Administration
- 4. Maintenance...
- 5. Port Status/Control
- 6. Exit

Step 1: Press "2" to enter TCP/IP administration

Step 8: Press "3" to enter Brouter administration

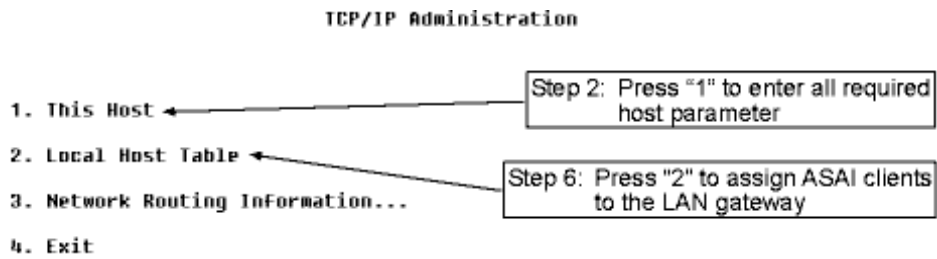
view/add/delete user logins or modify passwords

EXIT

REFRESH

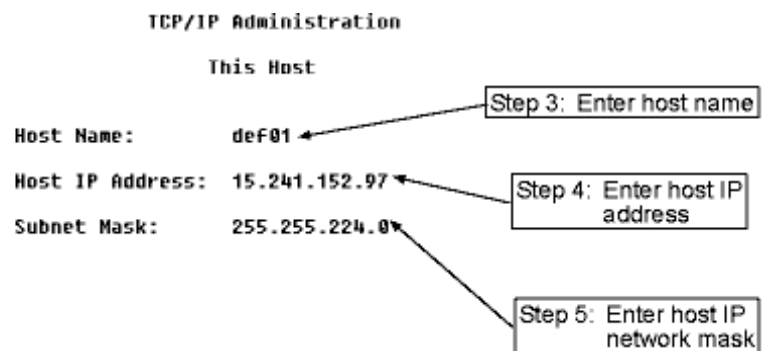
ENTER

HELP



routing daemon, default gateway, and routing table

EXIT **REFRESH** **ENTER** **HELP**



specify the host name of this machine

CANCEL **REFRESH** **ENTER** **CLEAR** **HELP**

TCP/IP Administration
Local Host Table

Page
1 of 2

IP Address	Host Name	Aliases
15.241.152.92	cti04	
15.241.152.94	cti05	
15.241.152.96	cti06	

Step 7: Add all ASAI clients permitted to access the LAN gateway with host name and IP address

Press DELETE to delete this entry or ADD to add a new entry

CANCEL **REFRESH** **ADD** **DELETE**

HELP **NXTPG** **PRUPG**

Router Administration

Port	Client Name or IP Address	Client Link	Client Status
1	cti04	1	in use

Step 9: Add client links (these also refer to the port used when defining an ASAI station in Definity) for all attached ASAI clients

Press DELETE to delete this entry or ADD to add a new entry

CANCEL **REFRESH** **ADD** **DELETE**

HELP **UPDATE**

Port Status/Control

Port	DEFINITY Port State	TCP/IP Connection State	Brouter Service State	Messages to DEFINITY	Messages From DEFINITY	Message Period (minutes)
1	CONNECTED	ESTABLISHED	in service	15	44	30

After having started the ASAI client, the current communication status may be accessed by using item "5" port/status control from the main menu

Press STATE, DROP, or MSGPER to effect this entry

CANCEL	REFRESH		DROP	HELP	STATE	UPDATE	MSGPER
--------	---------	--	------	------	-------	--------	--------

D.2.2 ASAI Station Administration

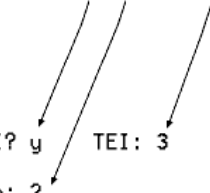
From a hardware point of view, the LAN gateway interfaces to the backplane of the PBX system. The Definity PBX interacts with an adjunct attached to a local area network via the LAN gateway. As a PBX can only deal with telephony resources such as trunks, stations, tone detectors and generators, ASAI channels have to be administered as stations on the switch. Up to 8 ASAI communication channels are possible per LAN gateway. Thus from a PBX perspective messages are sent to an ASAI station, which are transmitted over the local area network to an adjunct system. The following figures present corresponding station settings.

```
display station 15000                                     Page 1 of 2  SPE A
STATION
Extension: 15000                                           BCC: 0
Type: ASAI                                                TN: 1
Port: 01A0701                                           COR: 1
Name: Local ASAI Link 1                                  COS: 1
Hunt-to Station:

STATION OPTIONS

BRI OPTIONS
  XID? n
  MIM Support? n
  Fixed TEI? y
  TEI: 3
  CRU Length: 2
```

Define terminal endpoint parameters and CRV length according to the requirements as desired by the adjunct



Station configuration - Note the terminal endpoint identifier settings

```
display station 15000                                     Page 2 of 2  SPE A
STATION
FEATURE OPTIONS
  Event Minimization? n

  Per Station CPN - Send Calling Number?
  Audible Message Waiting? n
```

Station configuration - Use event minimization only under heavy load conditions

D.2.3 Shared UUS Configuration

As already mentioned above, the Definity PBX itself utilizes UUS to transmit switch and call centre specific information between locations. PBX related data and user-to-user information received from an adjunct are packed into a single information element to be sent to the remote switch. The remote switch extracts the user-to-user information to be passed to an attached adjunct via the ASAI protocol. The following screens show the differences in trunk group administration to set up this type of shared operation. Please note, that shared UUI IE treatment is completely transparent to the attached adjunct.

```

change trunk-group 1                                     Page 2 of 11  SPE R
TRUNK FEATURES
  ACA Assignment? n                                     Measured: internal  Wideband Support? n
                                     Internal Alert? n      Maintenance Tests? y
  Data Restriction? n                                     Send Name: n        Send Calling Number: y
                                     Send Connected Number: y
  Used for DCS? n
  Suppress # Outpulsing? y   Numbering Format: public
  Outgoing Channel ID Encoding: exclusive   UUI IE Treatment: shared
                                               Maximum Size of UUI IE Contents: 128
                                               Utilize UUI to transport switch
                                               specific information
  Send UCID? n
  Send Codeset 6/7 LAI IE? y
  BSR Reply-best DISC Cause Value: 31
  Send additional switch
  specific information

```

WARNING: Setting both Shared UUI and Send LAI IE could exceed Message limits

Trunk setup for shared UUI transmission - When the UUI IE treatment option is set to shared another screen becomes available

```
change trunk-group 1
```

```
Page 4 of 11 SPE R
```

```
SHARED UUI FEATURE PRIORITIES
```

```
ASAI: 1 ← UUI as provided by an adjunct
```

```
Universal Call ID (UCID): 2 ← systemwide call identification
```

```
MULTI SITE ROUTING (MSR)
```

```
In-UDN Time: 3
```

```
UDN Name: 4
```

```
Collected Digits: 5
```

```
Other LAI Information: 6
```

Trunk setup for shared UUI transmission - Note the arrangement of subelements

Appendix E

List of Acronyms

Many abbreviations and acronyms arise in the context of queueing theory, call centre engineering, telecommunication technology and data networking. Although almost all of them have been defined throughout the text, we have decided to assemble a list of acronyms. We hope, that such an undertaking helps to avoid misunderstandings and intensive page browsing.

3GPP	...	third generation partnership project
AAA	...	authentication and accounting center
AACW	...	average after call work time
AAL	...	ATM adaption layer
ABM	...	asynchronous balanced mode
ACD	...	automatic call distributor
ACF	...	admission confirm
ACW	...	after call work
ADA	...	average delay to abandon
ADH	...	average delay to handle
ADPCM	...	adaptive pulse code modulation
AE	...	average excess
AIT	...	average inbound time
ANF	...	additional network feature
ANI	...	automatic number identification
AOC	...	advice of charge
AOT	...	average outbound time
ARJ	...	admission reject

ARP	...	address resolution protocol
ARQ	...	admission request
AS	...	application server
ASA	...	average speed of answer
ASAI	...	adjunct switch application interface
ASAP	...	aggregate server access protocol
AST	...	average service time
ATM	...	asynchronous transfer mode
ATT	...	average talk time
AUX	...	auxiliary state
AWT	...	acceptable waiting time
AWT	...	average after call work time
AXT	...	average auxiliary time
B-ISDN	...	broadband ISDN
B-PSDN	...	broadband private integrated services network
BAS	...	bit-rate allocation signal
BCSM	...	basic call state model
BE	...	border element
BMAP	...	batch markov arrival processes
CA	...	channel allocation
CAS	...	channel associated signaling
CC	...	call centre
CC	...	contributing source count
CCS	...	centum call seconds
CDF	...	cumulative distribution function
CDR	...	call detail record
CE	...	carrier ethernet
CELP	...	codebook excitation linear prediction
CEPT	...	european conference of posts and telecommunications administrations
CIF	...	common intermediate format
CLP	...	cell loss priority
CPL	...	call processing language
CPU	...	central processing unit
CRC	...	cyclic redundancy check
CRM	...	customer relationship management
CRV	...	call reference number
CS1	...	capability set 1

CS2	...	capability set 2
CSA	...	callpath services architecture
CSRC	...	list of contributing sources
CSTA	...	computer support telecommunications applications
CTI	...	computer telephone integration
CTN	...	corporate telecommunication network
DCE	...	data communications equipment
DCT	...	discrete cosine transformation
DECT	...	digital european cordless telephone
DHCP	...	dynamic host configuration protocol
DID	...	direct inward dialing
DIFFSERV	...	differentiated services
DM	...	disconnect mode
DNIS	...	dialed number identification service
DNS	...	domain name service
DSCP	...	differentiated services code point
DSI	...	digital speech interpolation
DSL	...	digital subscriber line
DSS1	...	digital subscriber system no. 1
DSS2	...	digital subscriber system no. 2
DTE	...	data terminal equipment
EA0	...	extension address bit 0
EA1	...	extension address bit 1
ECMA	...	european computer manufacturer association
EFM	...	ethernet in the first mile
EM	...	expectation maximization
EMAIL	...	electronic mail
ENUM	...	electronic numbering
ETSI	...	european telecommunication standards institute
EWI	...	expected wait time
FAS	...	frame alignment signal
FCFS	...	first come first serve
FCS	...	frame check sequence
FRMR	...	frame reject
FTP	...	file transfer protocol
GCF	...	gatekeeper confirm
GEF	...	generic extensibility framework
GFC	...	generic flow control

GMPLS	...	generalized multiprotocol label switching
GOB	...	group of blocks
GRQ	...	gatekeeper request
GSM	...	group speciale mobile
GUI	...	graphical user interface
HDLC	...	high level data link control
HEC	...	header error correction
HOL	...	head of the line
HTML	...	hypertext markup language
HTTP	...	hypertext transfer protocol
IA5	...	international alphabet no. 5
IAM	...	initial address message
IANA	...	internet assigned numbers authority
IAX	...	inter-Asterisk exchange protocol
IETF	...	internet engineering task force
ILR	...	increasing likelihood ration
IMS	...	IP multimedia subsystem
IN	...	intelligent network
INAP	...	intelligent network application part
IP	...	internet protocol
IPP	...	interrupted Poisson process
IQ	...	information query
ISDN	...	integrated services digital network
ISO	...	international standards organization
ISUP	...	ISDN user part
ITC	...	international teletraffic congress
ITU	...	international telecommunication union
IVR	...	interactive voice response system
JAIN	...	Java for integrated networks
JDBC	...	Java database connection
JID	...	Jabber ID
JPEG	...	joint photographic experts group
JSBW	...	join the queue with shortest actual wait
JSEW	...	join the queue with shortest expected wait
JSQ	...	join the shortest queue
JTAPI	...	Java telephony application interface
KPI	...	key performance indicator
LAN	...	local area network

LAPD	...	link access protocol for the d-channel
LAPF	...	link access protocol F
LCC	...	lost calls cleared
LCD	...	liquid crystal display
LCFS	...	last come first serve
LCH	...	lost calls held
LDAP	...	lightweight directory access protocol
LE	...	local exchange
LED	...	light emitting diode
LT	...	line termination
MAC	...	medium access control
MBONE	...	internet's multicast backbone
MC	...	multipoint controller
MCU	...	multipoint control unit
MEDA	...	mixed Erlang distributions for approximation
MEFIT	...	mixture of Erlang distribution fitting
MG	...	media gateway
MGC	...	media gateway controller
MGCP	...	media gateway control protocol
MIS	...	management information system
MLE	...	maximum likelihood estimation
MMPP	...	Markov modulated Poisson process
MP	...	multipoint processor
MPEG	...	moving pictures expert group
MPLS	...	multiprotocol label switching
MSI	...	manufacturer specific information
MSN	...	multiple subscriber number
MTP	...	message transfer part
MUC	...	multi-user chat
NAT	...	native address translation
NCA	...	number of calls abandoned
NCH	...	number of calls handled
NCO	...	number of calls offered
NCW	...	number of calls waiting
NGN	...	next generation network
NNI	...	network-network interface
NSCA	...	number of successful call attempts
NT	...	network termination

OA&M	...	operations, administration and maintenance
OCW	...	oldest call waiting
OS	...	operating system
PAT	...	port address translation
PBX	...	private branch exchange
PCM	...	pulse code modulation
PDA	...	personal digital assistant
PDF	...	probability density function
PINX	...	private integrated services network exchange
PMBS	...	ISDN packet mode bearer service
PNNI	...	private network-network interface
POTS	...	plain ordinary telephone system
PS	...	processor sharing
PSTN	...	public switched telephone network
PT	...	payload type
QCIF	...	quarter common intermediate format
QoS	...	quality of service
RADIUS	...	remote dial in user service
RARP	...	reverse address resolution protocol
RAS	...	registration, admission and status
RCF	...	registration confirm
REJ	...	reject
REL	...	release
RLT	...	rational Laplace transformation
RMI	...	remote method invocation
RNR	...	receive not ready
RR	...	receive ready
RRQ	...	registration request
RSS	...	random selection for service
RSVP	...	resource reservation protocol
RTCP	...	real time control protocol
RTP	...	real time transport protocol
RTSP	...	real time streaming protocol
SAAL	...	signaling adaption layer
SABME	...	asynchronous balanced mode extended
SACK	...	selective acknowledge
SAPI	...	service access point identifier
SCAI	...	switch computer application interface

SCP	...	service control point
SCTP	...	stream control transport protocol
SDH	...	synchronous digital hierarchy
SDP	...	session description protocol
SET	...	simple endpoint type
SG	...	signaling gateway
SIP	...	session initiation protocol
SL	...	service level
SLA	...	service level agreement
SLEE	...	service logic execution environment
SMDR	...	standard message detail record
SMP	...	system management platform
SMTP	...	simple mail transfer protocol
SNMP	...	simple network management protocol
SPT	...	shortest processing time first
SRPT	...	shortest remaining processing time first
SS7	...	common channel signaling system no. 7
SSRC	...	synchronization source
STUN	...	simple traversal of UDP through NAT
TA	...	terminal adapter
TACW	...	timed after call work
TAPI	...	telephony application programming interface
TASI	...	time assignment speech interpolation
TCAP	...	transaction capabilities application part
TCP	...	transmission control protocol
TDM	...	time division multiplexer
TE	...	terminal equipment
TEI	...	terminal endpoint identifier
TLS	...	transport layer security
TMS	...	time multiplex switching
TOS	...	type of service
TSAPI	...	telephony services application programming interface
TSI	...	time slot interchange
TURN	...	traversal using relay NAT
UA	...	unnumbered acknowledge
UA	...	user agent
UAC	...	user agent client
UAS	...	user agent server

UDP	...	user datagram protocol
UMVU	...	uniform minimum variance (unbiased)
UNI	...	user-network interface
URI	...	unified resource identifier
USBS	...	user signaling bearer service
USR	...	user information messages
UUIE	...	user-to-user information element
UUS	...	user-to-user signaling
VAD	...	voice activity detection
VC	...	virtual circuit
VCI	...	virtual channel identification
VDN	...	vector directory number
VLAN	...	virtual LAN
VPI	...	virtual path identifier
WWW	...	world wide web
XEP	...	XMPP extension protocol
XID	...	exchange identification
XML	...	extensive markup language
XMPP	...	extensible messaging and presence protocol

Bibliography

- [1] *Microsoft Windows NT Server Networking Guide* - Microsoft Press 1996
- [2] *Inter-Asterisk Exchange (IAX)* - M. Boucadair - Wiley 2009
- [3] *Understanding SIP Servlets 1.1* - C. Boulton, K. Gronowski - Artech House 2009
- [4] *Signalisierung zwischen privaten Vermittlungsstellen* - Christian Dombacher - Vortrag Universität Wien 1996
- [5] *Programming IP Telephony Services with the Call Processing Language* - M.Z. Eslami - School of Information and Communication Technology KTH Stockholm 2007
- [6] *Intelligent Network Standards, Their Application to Services* - I. Faynberg, L.R. Gabuzda, M.P. Kaplan, N.J. Shah - McGraw Hill 1997
- [7] *Integrated Broadband Networks* - Rainer Händel, Manfred Huber - Addison Wesley 1991
- [8] *Integrated Services Digital Networks* - Hermann Helgert - Addison Wesley 1991
- [9] *Java Servlet Programming* - J. Hunter, W. Crawford - O'Reilly 2001
- [10] *Understanding the Session Initiation Protocol* - A.B. Johnston - Artech House 2004
- [11] *CentreVu Call Management System Release 3 Version 6 CentreVu Advocate User Guide* - Lucent Technologies 1998

- [12] *Call Visor ASAI Protocol Reference* - Lucent Technologies 1998
- [13] *Call Visor ASAI Technical Reference* - Lucent Technologies 1998
- [14] *QSig - Handbook for Communication Managers* - IPNS Forum 1995
- [15] *XMPP, The Definitive Guide* - P. Saint-Andra, K. Smith, R. Tronçon - O'Reilly 2009
- [16] *Videoconferencing and Videotelephony* - Richard Schaphorst - Artech House 1996
- [17] *Technik der Netze* - Gerd Siegmund - Hüthig 1999
- [18] *Internet Communications using SIP* - H. Sinnreich, A.B. Johnston - Wiley 2001
- [19] *Packet Switching Evolution from Narrowband to Broadband ISDN* - Michael Smouts - Artech House 1991
- [20] *The Mathematics of Internet Congestion Control* - Birkhäuser Verlag 2005
- [21] *Stream Control Transport Protocol, A Reference Guide* - R.R. Stewart, Q. Xie - Addison Wesley 2002
- [22] *Supporting Service Level Agreements on IP Networks* - Dinesh Verma - Macmillan Publishing 1999
- [23] The Impact of Retrials on Call Center Performance - S. Aguir, F. Karaesmen, O. Zeynep Aksin, F. Chauvet - published in *OR Spectrum* 26 (2004)
- [24] *Teletraffic Theory and Applications* - H. Akimaru, K. Kawashima - Springer Verlag 1999
- [25] An $M/PH/k$ retrial queue with finite number of sources - A.S. Alfa, K.P.S. Isotupa - published in *Computers and Operations Research* 31 (2004), pages 1455-1464
- [26] *Probability, Statistics and Queueing Theory* - A.O. Allen - Academic Press 1978

- [27] *Probability, Statistics and Queueing Theory* - A.O. Allen - Academic Press 1990
- [28] On the stability of retrial queues - E. Altmann, A.A. Borovkov - published in *Queueing Systems* 26 (1997), pages 343-363
- [29] *Retrial Queueing Systems, A Computational Approach* - J.R. Artalejo, A. Gomez-Corral - Springer Verlag 2008
- [30] *Applied Probability and Queues* - S. Asmussen, Springer 2003
- [31] Phase-Type Distributions and Related Point Processes: Fitting and Recent Advances - S. Asmussen - published in *Matrix-Analytic Methods in Stochastic Models* (1997), pages 137-149
- [32] Renewal Theory and Queueing Algorithms for Matrix-Exponential Distributions - S. Asmussen, M. Bladt - published in *Matrix-Analytic Methods in Stochastic Models* (1997), pages 313-341
- [33] *AT&T Integrated Access and Cross-Connect System Product Description* - AT&T 1992
- [34] On Queues with Impatient Customers - F. Bacelli, G. Hebuterne - published in *Performance* (1981), pages 159-179
- [35] *Integrierte Unternehmensnetze* - Anatol Badach - Hüthig Verlag 1997
- [36] *Statistical Inference for Stochastic Processes* - I. Basawa, P. Rao - Academic Press 1980
- [37] *Principles of Telecommunication Traffic Engineering 3rd Edition* - D. Bear - IEE Communications 1988
- [38] *Ethernet in the First Mile* - M. Beck - McGraw Hill 2005
- [39] *Digital Telephony* - J.C. Bellamy - Wiley 1982
- [40] *The general distributional Little's law and its applications* - D. Bertsimas, D. Nakazato - March 1991
- [41] *A Queueing Model for Call Blending in Call Centers* - S. Bhulai, G. Koole - Vrije Universiteit Amsterdam 2000

- [42] *OSI in der Anwendungsebene* - Ulf Beyschlag - Datacom Verlag 1988
- [43] *Statistical Inference for Markov Processes* - P. Billingsley - University of Chicago Press 1961
- [44] *Numerical Methods for Structured Markov Chains* - D.A. Bini, G. Latouche, B. Meini - Oxford University Press 2005
- [45] *Queueing Theory* - P.P. Bocharov, C. D'Apice, A.V. Pechinkin, S. Salerno - VSP Brill Publishers 2004
- [46] *ISDN - Digitale Netze für Sprach-, Text-, Daten-, Video- und Multi-mediakommunikation, 4. Auflage* - Peter Bocker - Springer Verlag 1997
- [47] *CCITT-Empfehlungen der V-Serie und der X-Serie, Band 1 Datenpaketvermittlung, Internationale Standards* - Jürgen Böhm - R.v.Decker Verlag 1981
- [48] *Dimensioning Large Call Centres* - S. Borst, A. Mandelbaum, M. Reimann - CWI Amsterdam, Technion Haifa and Bell Labs 2002
- [49] *Stationary Stochastic Models* - A. Brandt, B. Lisek, P. Franken - Akademie Verlag 1990
- [50] On the $M(n)/M(m)/s$ Queue with impatient Calls - A. Brandt, M.Brandt - published in *Performance Evaluation* 35, pages 1-18
- [51] *Markov Chains* - Pierre Brémaud - Springer Verlag 1999
- [52] *An Introduction to Queueing Theory* - L. Breuer, D. Baum - Springer Verlag 2005
- [53] Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective - L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, L. Zhao - published in *Journal of the American Statistical Association* 100 (2005), pages 36-50
- [54] *TCP/IP Blueprints* - Robin Burk, Martin Bligh, Thomas Lee, et al. - SAMS Publishing 1997
- [55] *Classpad 330 User's Guide* - Casio 2009

- [56] *Ethernet-LANs* - Peter Chylla, Heinz Gerd Hegering - Datacom Verlag 1987
- [57] *Superposition of Point Processes* - E. Cinlar 1972 - published in *Stationary Point Processes* from Wiley (1972), pages 549-606
- [58] *Introduction to Stochastic Processes* - E. Cinlar - Prentice Hall 1975
- [59] *Introduction to Queueing Theory* - R.B. Cooper - Macmillan Publishing 1972
- [60] *Renewal Theory* - D.R. Cox - Methuen 1962
- [61] *The Theory of Stochastic Processes* - D.R. Cox, H.D. Miller - Methuen 1965
- [62] *Queueing Theory for Telecommunications* - J.N. Daigle - Addison Wesley 1992
- [63] *Queueing Theory with Applications to Packet Telecommunication* - J.N. Daigle - Springer Verlag 2005
- [64] *Stochastic Processes* - J. L. Doob - Wiley 1953
- [65] *IP Telephony* - Bill Douskalis - Prentice Hall 2000
- [66] *Putting VoIP to Work* - Bill Douskalis - Prentice Hall 2002
- [67] *ATM, Theory and Applications* - David E. McDysan, Darren L. Spohn - McGraw Hill 1994
- [68] ECMA Technical Report TR/87, *Using CSTA for SIP Phone User Agents (uaCSTA)* - ECMA International 2004
- [69] ECMA Standard 248, *Web Services Description Language (WSDL) for CSTA Phase III 4th Edition* - ECMA International 2009
- [70] ECMA Standard 323, *XML Protocol for Computer Supported Telecommunications Applications (CSTA) Phase III 5th Edition* - ECMA International 2009
- [71] *Sample-Path Analysis Of Queueing Systems* - M. El-Taha, S. Stidham - Kluwer Academic Publishers 1999

- [72] Fractal Queueing Models - A. Erramilli, O. Narayan, W. Willinger - published in *Frontiers in Queueing* by CRC Press 1997, pages 245-269
- [73] *Characterization of Matrix-Exponential Distributions* - M.W. Fackrell - 2003
- [74] *Retrial Queues* - G.I. Falin, J.G.C. Templeton - Chapman & Hall 1997
- [75] On a System with Impatience and Repeated Calls - G. Fayolle, M.A. Brun - published in *Queueing Theory and its Applications* by North-Holland (1988), pages 283-305
- [76] *TCP/IP Architecture, Protocols and Implementation* - Sidnie Feit - McGraw Hill 1993
- [77] *Theorie Statistischer Schätzung* - Klaus Felsenstein - Skriptum der TU Wien 2006
- [78] Prior Distributions on Spaces of Probability Measures - T.S. Ferguson - published in *Annals of Statistics 2* (1974), pages 615-629
- [79] *Telecommunications Switching, Traffic and Networks* - J.E. Flood - Prentice Hall 1995
- [80] Approximating multi-skill blocking systems by HyperExponential Decomposition - G.J. Franx, G. Koole, A. Pot - Vrije Universiteit Amsterdam 2005
- [81] *Matrizenrechnung I* - F.R. Gantmacher - VEB Verlag 1958
- [82] *Matrizenrechnung II* - F.R. Gantmacher - VEB Verlag 1959
- [83] *Transform Techniques for Probability Modeling* - W.C. Giffin - Academic Press 1975
- [84] *TCP/IP Protokolle, Projektplanung, Realisierung* - Gerhard M. Glaser, Mathias Hein, Johannes Vogl - Datacom Verlag 1990
- [85] *Handbuch der Bedienungstheorie 2* - Autorenkollektiv - Akademie Verlag 1984
- [86] *Introduction to Queueing Theory, 2nd Edition* - B.V. Gnedenko, I.N. Kovalenko - Birkhäuser Verlag 1989

- [87] *Token Ring 2. Auflage* - Hans Georg Göhring, Franz Joachim Kauffels - Datacom Verlag 1993
- [88] *Theorie Stochastischer Prozesse* - Karl Grill - TU Wien 2007
- [89] *Computer/Telecom Integration* - Arkady Grinberg - McGraw Hill 1992
- [90] *Fundamentals of Queueing Theory* - D. Gross, C. M. Harris - Wiley 1985
- [91] Queueing Model with State Dependent Balking and Reneging: Its Complementary and Equivalence - Surenda M. Gupta - published in *Performance Evaluation Review Vol. 22, No. 2-4*
- [92] *Performance Modelling of Communication Networks and Computer Architectures* - Peter G. Harrison, Naresh M. Patel - Addison Wesley 1993
- [93] *To Queue or Not To Queue, Equilibrium Behavior in Queueing Systems* - R. Hassin, M. Haviv - Kluwer Academic Publishers 2003
- [94] *Switching-Technologie in lokalen Netzen* - Mathias Hein - Thomson Publishing 1996
- [95] *Applied and Computational Complex Analysis Vol. 2* - P. Henrici - Wiley 1977
- [96] *Queueing Tables and Graphs* - F.S. Hillier, O.S. Yu - North Holland 1981
- [97] The $G/M/m$ Queue with Finite Waiting Room - P. Hokstad - published in *Journal of Applied Probability* (1975), pages 779-792
- [98] *Matrix Analysis* - R.A. Horn, C.R. Johnson - Cambridge University Press 1985
- [99] *Topics in Matrix Analysis* - R.A. Horn, C.R. Johnson - Cambridge University Press 1991
- [100] PhFit: A General Phase-Type Fitting Tool - A. Horvath, M. Telek - Technical University of Budapest 2001

- [101] *Switching and Traffic Theory for Integrated Broadband Networks* - Joseph Y. Hui - Kluwer Academic Publishers 1990
- [102] *Basic Stochastic Processes* - R. Iranpour, F. Chagon - Macmillan Publishing 1988
- [103] *Fitting Phase-Type Distributions to Data from a Telephone Call Center* - E. Ishay - Technion Haifa 2002
- [104] *General Tariff Principles, Charging and Accounting in International Telecommunications Services, Series D Recommendations* - CCITT 1989
- [105] *Telephone Network and ISDN Quality of Service, Network Management and Traffic Engineering, Recommendation E.500* - CCITT 1992
- [106] *Priority Queues* - N.K. Jaiswal - Academic Press 1968
- [107] Stochastic Modeling of Traffic Processes - D.L. Jagerman, B. Melamed, W. Willinger - published in *Frontiers in Queueing* by CRC Press 1997, pages 271-320
- [108] *Continuous Univariate Distributions Volume 1* - N.L. Johnson, S. Kotz, N. Balakrishnan - Wiley 1994
- [109] *Continued Fractions* - W.B. Jones, W.J. Thron - Cambridge University Press 1984
- [110] Managing uncertainty in call centers using Poisson mixtures - G. Jongbloed, G. Koole - published in *Applied Stochastic Models in Business and Industry* 17 (2001), pages 307-318
- [111] Real-Time Scheduling Policies for Multiclass Call Centers with Impatient Customers - O. Jouini, A. Pot, G. Koole, Y. Dallery - Ecole Centrale Paris and Vrije Universiteit Amsterdam (2008)
- [112] *Java Telephony API V1.2 Specification* - Sun Microsystems 1998
- [113] *The Java Telephony API, An Overview, White Paper* - Sun Microsystems 1997

- [114] Mathematical Methods in Queueing Theory - V.V. Kalashnikov - Kluwer Academic Publishers 1994
- [115] *A First Course in Stochastic Processes* - S. Karlin, H. Taylor - Academic Press 1975
- [116] *A Second Course in Stochastic Processes* - S. Karlin, H. Taylor - Academic Press 1981
- [117] *3G Mobile Networks* - S. Kasera, N. Narang - McGraw Hill 2005
- [118] *Delivering Carrier Ethernet: Extending Ethernet Beyond the LAN* - A. Kasim et al - McGraw Hill 2007
- [119] *Denumerable Markov Chains* - J. Kemeny, J. Snell, A. Knapp - Springer Verlag 1976
- [120] *Rechnernetze nach OSI* - H. Kerner - Addison Wesley 1992
- [121] Heuristic approximation for the mean waiting time in the $GI/G/s$ queue - T. Kimura - published in *Economic Journal of Hokkaido University* (1987), pages 87-98
- [122] Approximations for the waiting time in the $GI/G/s$ queue - T. Kimura - published in *Journal of the Operations Research* 34 (1991), pages 173-186
- [123] *Queueing Systems: Theory* - L. Kleinrock - Wiley 1975
- [124] *Queueing Systems: Computer Applications* - L. Kleinrock - Wiley 1976
- [125] On the Modification of Rouché's Theorem for Queueing Theory Problems - V. Klimenok - published in *Queueing Systems* 38 (2001)
- [126] A Simple Proof of the Optimality of a Threshold Policy in a Two-Server Queueing System - G. Koole - INRIA Sophia Antipolis (1991)
- [127] Minimizing response times and queue lengths in systems of parallel queues - G. Koole, P.D. Sparaggis, D. Towsley - published in *Journal of Applied Probability* 36 (1999), pages 1185-1193

- [128] Exponential Approximation of Multi-Skill Call Centers Architecture - G. Koole, J. Talim - published in *Proceedings of QNETs* (2000), pages 23/1-10
- [129] Queueing Models of Call Centers - G. Koole, A. Mandelbaum - Vrije Universiteit Amsterdam and Technion Haifa 2001
- [130] Redefining the service level in call centers - G. Koole - Dept. of Mathematics, Vrije Universiteit Amsterdam 2005
- [131] *Optimization of Business Processes: An Introduction to Applied Stochastic Modeling* - G. Koole - Vrije Universiteit Amsterdam 2008
- [132] *Stochastic Theory of Service Systems* - L. Kosten - Pergamon Press 1973
- [133] Estimating Parameters of Cox Distributions - M. Kramer - published in *Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen* (1993)
- [134] Retrial Queues Revisited - V.G. Kulkarni, H.M. Liang - published in *Frontiers in Queueing* by CRC Press 1997, pages 19-34
- [135] Balking and Reneging in $M/G/s$ Systems: Exakt Analysis and Approximations - L. Liu, V.G. Kulkarni 2006
- [136] *IP Telephony with H.323* - V. Kumar, M. Kopi, S. Sengodan - Wiley 2001
- [137] Parameter Approximations for Phase-Type Distributions - A. Lang, J.L. Arthur - published in *Matrix-Analytic Methods in Stochastic Models* (1997), pages 151-206
- [138] *Introduction to Matrix Analytic Methods in Stochastic Modeling* - G. Latouche, V. Ramaswami - SIAM 1999
- [139] *Statistical Models and Methods for Lifetime Data* - J.F. Lawless - Wiley 2003
- [140] *Theory of Point Estimation* - E.L. Lehmann, G. Casella - Springer Verlag 1998

- [141] *Statistical Theory* - B.W. Lindgren - Macmillan Publishing 1968
- [142] *Queueing Theory, A Linear Algebraic Approach* - L. Lipsky - Macmillan Publishing 1992
- [143] Empirical Analysis of a Call Center - A. Mandelbaum, S. Zeltyn - Technion (Israel) 2005
- [144] The Palm/Erlang-A Queue, with Applications to Call Centers - A. Mandelbaum, S. Zeltyn - Technion (Israel) 2005
- [145] *Bayesian Decision Problems and Markov Chains* - J.J. Martin - Krieger Publishing 1975
- [146] *Basic Traffic Analysis* - Roberta Martine - AT&T Bell Labs 1994
- [147] Large Deviations for Join the Shorter Queue - D.R. McDonald, S.R.E. Turner - published in *Analysis of Communication Networks* by AMS (2000), pages 109-133
- [148] *The EM Algorithm and Extensions* - G.J. McLachlan, T. Krishnan - Wiley 2008
- [149] *Telecommunications Technology Handbook* - Daniel Minoli - Artech House 1991
- [150] Analytic Modeling with Matrix Exponential Distributions - K. Mitchell, J. Place, A. van de Liefvoort - published in *WMC'96-ME Distributions (1995)*
- [151] *Queues, Inventories and Maintenance* - P.M. Morse - Wiley 1958
- [152] *Token Ring Troubleshooting* - Dan Nassar - New Riders Publishing 1992
- [153] *Matrix-Geometric Solutions in Stochastic Models* - M.F. Neuts - Dover 1994
- [154] *Structured Stochastic Matrices of $M/G/1$ Type and their Applications* - M.F. Neuts - Marcel Dekker 1989
- [155] *Fast Packet Switching for Integrated Services* - Peter Newman - Doctoral Dissertation University of Cambridge 1988

- [156] A New Stage Method Getting Arbitrary Coefficient of Variation by Two Stages - S. Nojo, H. Watanabe - published in *Transactions of the IEICE E70* (1987), pages 33-36
- [157] New Approximation for General Distributions by n -stage Markov Processes based on Continued Fractions - S. Nojo, G. Kimura - published in *Teletraffic Science for New Cost-Effective Systems ITC-12* by Elsevier (1989), pages 1337-1343
- [158] *The EMPHT-Programme* - M. Olsson - Chalmers University of Technology & Göteborg University 1998
- [159] Necessary and Sufficient Conditions for Representing General Distributions by Coxians - T. Osogami, M. Harchol-Balter - published in *Tools 2003, LCNS 2704*, pages 182-199
- [160] A Closed-Form Solution for Mapping General Distributions to Minimal PH Distributions - T. Osogami, M. Harchol-Balter - published in *Tools 2003, LCNS 2704*, pages 182-199
- [161] *Queueing Theory in OR* - E. Page - Crane Russak & Company 1972
- [162] Contributions to the Theory on Delay Systems - C. Palm - published in *Tele* (1957), pages 37-67
- [163] *Self-Similar Network Traffic and Performance Evaluation* - K. Park, W. Willinger - Wiley 2000
- [164] *Gigabit Networking* - Graig Partridge - Addison Wesley 1994
- [165] *Switching Theory, Architecture and Performance in Broadband ATM Networks* - Achille Pattavina - Wiley 1998
- [166] *Interconnections, Bridges and Routers* - Radia Perlman - Addison Wesley 1993
- [167] *Designing a Call Center with an IVR* - K. Polina - Technion Haifa 2006
- [168] *Queues and Inventories, A Study of their Stochastic Processes* - N.U. Prabhu - Wiley 1965

- [169] *Asynchronous Transfer Mode*, Die Lösung für Breitband ISDN - Martin de Prycker - Prentice Hall 1994
- [170] *Stochastic Inequalities for Queues* - A.A.N. Ridder - Offsetdrukkerij Pasmans BV's-Gravenhage 1987
- [171] *Stochastic Service Systems* - J. Riordan - Wiley 1962
- [172] *Stochastic Networks and Queues* - Philippe Robert - Springer Verlag 2003
- [173] *The Bayesian Choice* - C.P. Robert - Springer Verlag 2007
- [174] *Diffusions, Markov Processes and Martingales Volume 1+2* - L. Rogers, D. Williams - Cambridge University Press 2000
- [175] *Applied Probability Models With Optimization Applications* - Sheldon M. Ross - Holden Day 1970
- [176] *Principles of Mathematical Analysis* - W. Rudin - McGraw Hill 1976
- [177] *Signaling System #7* - T. Russell - McGraw Hill 1995
- [178] *Session Initiation Protocol (SIP)* - T. Russel - McGraw Hill 2008
- [179] *Elements of Queueing Theory with Applications* - T.L. Saaty - McGraw Hill 1961
- [180] *Warteschlangen* - R. Schassberger - Springer Verlag 1973
- [181] Approximation von empirischen Verteilungsfunktionen mit Erlangmischverteilungen und Coxverteilungen - L. Schmickler - published in *Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen* (1987), pages 118-133
- [182] Erweiterung des Verfahrens MEDA zur analitischen Beschreibung empirischer Verteilungsfunktionen - L. Schmickler - published in *Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen* (1989), pages 175-189
- [183] *Matrix Analysis for Statistics* - J.R. Schott - Wiley 1997

- [184] *Mediation in a Multi-Service IP Network* - Limor Schweitzer, Matthew Lucas - white paper published in the internet 1999
- [185] *Self-Similar Processes in Telecommunications* - O.I. Sheluhin, S.M. Smolskiy, A.V. Osin - Wiley 2007
- [186] *Modellierung von Warteschlangensystemen mit dynamisch erzeugten Markov-Ketten* - M. Sommereder - Diplomarbeit TU Wien 2007
- [187] *Special Functions in Queueing Theory* - H.M. Srivastava, B.R.K. Kashyap - Academic Press 1982
- [188] Bilingual Call Centres - D.A. Stanford, W.K. Grassmann - published in *Analysis of Communication Networks* by AMS (2000), pages 31-47
- [189] *Complex Analysis* - I. Stewart, D. Tall - Cambridge University Press 1983
- [190] *Introduction to the Numerical Solution of Markov Chains* - William J. Stewart - Princeton University Press 1994
- [191] *Performance Analysis and Optimization of Inbound Call Centers* - R. Stollitz - Springer Verlag 2003
- [192] *Das OSI-Referenzmodell* - Klaus H. Stöttinger - Datacom Verlag 1989
- [193] *X.25 Datenpaketvermittlung* - Klaus Stöttinger - Datacom Verlag 1995
- [194] Queueing with Balking and Reneging in $M/G/1$ Systems - S. Subba Rao - published in *Metrika* 12 (1967)
- [195] Balking and Reneging in $M/G/1$ Systems with Post-Ponable Interruptions - S. Subba Rao - published in *Metrika* 14 (1969)
- [196] *Introduction to the Theory of Queues* - L. Takacs - Oxford University Press 1962
- [197] *Queueing Analysis Volume 1 (Vacation and Priority Systems)* - H. Takagi - North Holland 1991
- [198] *Queueing Analysis Volume 2 (Finite Systems)* - H. Takagi - North Holland 1993

- [199] *IxP Telephony with TAPI 3.0 White Paper* - Microsoft 1998
- [200] *Analytische Leistungsbewertung verteilter Systeme* - Phuoc Tran Gia - Springer Verlag 1996
- [201] *A First Course in Stochastic Models* - H. Tijms - Wiley 2003
- [202] New and Old Results for the $M/D/c$ Queue - H. Tijms - published in *International Journal of Electronics and Communication* 60 (2006), pages 125-130
- [203] Large Deviations for Join the Shorter Queue - S.R.E. Turner - published in *Analysis of Communication Networks* by AMS (2000), pages 95-108
- [204] *Algorithms and Approximations for Queueing Systems* - M.H. van Hoorn - Mathematisch Centrum 1984
- [205] Call Packing Bounds for Overflow Queues - N.M. van Dijk, E. van der Sluis - University of Amsterdam 2004
- [206] *Versit Archives CD-ROM* - ECTF 1996
- [207] *The Solution of Quasi Birth and Death Processes arising from Multiple Access Computer Systems* - V. Wallace - SEL Technical Report (1969) No. 35
- [208] *Mobilfunknetze und ihre Protokolle* - B. Walke - Teubner Verlag Stuttgart 1998
- [209] *Communication Networks: A First Course* - Jean Walrand - Aksen Associates 1991
- [210] *Internet QoS* - Zheng Wang - Morgan Kaufmann Publishers 2001
- [211] A Diffusion Approximation for a $GI/GI/1$ Queue with Balking and Reneging - A. Ward, P. Glynn - published in *Queueing Systems* 50 (2005), pages 371-400
- [212] Approximating a Point Process by a Renewal Process, I: Two Basic Methods - W. Whitt - published in *Queueing Systems* 30 (1982), pages 125-147

- [213] Approximations for the $GI/G/m$ queue - W. Whitt 1993 - published in *Production and Operations Management* 2 (1993), pages 114-161
- [214] Improving Service by Informing Customers about Anticipated Delays - W. Whitt 1998 - published in *Management Science* 45 (1999), pages 192-207
- [215] *Stochastic Modeling and the Theory of Queues* - R.W. Wolff - Prentice Hall 1989
- [216] Call Centers with Impatient Customers: Many-Server Asymptotics of the $M/M/n + G$ Queue - A. Mandelbaum, S. Zeltyn 2005 - published in *Queueing Systems* 51 (2005), pages 361-402
- [217] *Birth and Death Processes and Markov Chains* - W. Zikun, Y. Yiangqun - Springer Verlag 1992

Index

- acceptable waiting time, 372
- adjuncts, 55
 - call accounting, 55
 - reader boards, 57
- after call work, 16
 - estimation, 321
 - model with, 355
 - performance indicator, 27
 - timed, 349, 362
- agent, 15
 - selection, 14
 - skill, 14, 334
- algorithm
 - EM, 213, 218
 - EMPHT, 215
 - Evolutionary, 219
 - MEDA, 201
 - MEFIT, 201
 - PhFit, 219
- arrival process, 61, 319
- asynchronous transfer mode, 22, 24, 25, 40, 42, 53, 244, 258
- automatic call distributor, 29
 - adjunct, 32
 - integrated, 34
- Bayesian analysis, 216
- binomial coefficient, 106
- birth-death process, 90
 - average system size, 91
 - average time in system, 91
- infinitesimal generator, 108
- positive recurrence, 108
- recurrence, 108
- steady state solution, 91
- throughput, 91
- transition probabilities, 108
- call
 - allocation, 221
 - distribution, 13, 328
 - priorities, 332
 - routing, 13
 - selection, 15
- call centre
 - analytic modeling, 311
 - applications, 10
 - architecture, 29
 - CTI-enabled, 9, 11, 16
 - distributed, 221
 - interconnection, 221
 - internet, 9
 - key performance indicator, 27
 - multimedia, 302
 - signaling, 308
 - network interconnection, 304
 - perspective, 312
 - pre-routing, 295
 - public network solutions, 375
 - signaling between, 299
 - virtual, 222, 295
- call management, 11

- blending, 12, 336
 - inbound, 11
 - outbound, 12
- callflow, 325
- Chapman Kolmogorov equations, 384
- circuit switching, 19
 - wideband, 22
- connection oriented, 24, 25, 244, 246, 256, 262, 263
- connectionless, 25, 247, 256, 260, 264, 276, 301
- CTI, 9
 - application interface, 32, 54, 295, 299, 303, 407
 - ASAI, 33, 408
 - CPL, 33
 - CSA, 33
 - CSTA, 33
 - JTAPI, 33
 - Meridian Link, 33
 - SCAI, 32
 - TAPI, 33
 - TSAPI, 33
 - server, 29, 67, 298
- data analysis, 195
- digital cross connect, 26
- direct sum, 192
- distributed communication platform, 43
- distribution, 63
 - approximation, 72
 - conjugate prior, 216
 - Cox, 202, 208
 - Dirichlet, 218
 - Erlang, 69, 161
 - Erlang-Cox, 202
 - exponential, 63
 - generalized Erlang, 72
 - hyper-Erlang, 71, 72, 163
 - hyperexponential, 69, 162, 208
 - marginal, 216
 - matrix beta, 218
 - matrix beta-1, 218
 - matrix exponential, 155
 - phase type, 71, 155
 - Poisson, 64
 - posterior, 197, 216
 - prior, 197
- Engset model, 107
 - Engset distribution, 107
- Erlang A, 100
- Erlang B, 96
 - heavy traffic approximation, 98
 - recursion formula, 97
- estimator
 - Bayes, 197
 - consistent, 198
 - unbiased, 198
- fixed node approximation, 207
 - two stage, 209
- generalized server occupancy, 145
- hyperparameter, 216
- incomplete gamma function, 102
- interactive voice response, 53
- Kronecker function, 130
- Kronecker product, 192
- Lindley Integral Equation, 151
- Little's law, 77
 - for distributions, 78
- Markov chain, 384

- aperiodic, 386
- continuous time, 388
- definition, 384
- discrete time, 384
- embedded, 388
- ergodic, 387
- homogenous, 384
- infinitesimal generator, 389
- irreducible, 385
- Kolmogorov's backward equations, 390
- Kolmogorov's forward equations, 390
- recurrence, 386
- skip-free, 107
- state analysis, 345
- stationary probabilities, 387, 391
- transition probabilities, 389
- transition semigroup, 389
 - conservative, 389
 - stable, 389
- Markov process, 383
 - definition, 383
 - homogenous, 383
 - transition probabilities, 383
- memoryless, 64
- method of moments
 - Bayes, 198
- modulation, 38
 - adaptive pulse code, 20
 - pulse code, 20, 38
- moment
 - normalized, 203
 - sample, 198
 - theoretical, 198
- multiplexing, 22
 - call, 245
 - in TCP, 263
 - in UDP, 265
 - layer 1, 230, 232
 - space division, 20, 21, 39, 40, 51
 - statistical, 22, 25, 244
 - time division, 20
- packet switching, 22
- PASTA, 109
- Pollaczek-Khintchine Formula, 129
- public network, 375
- quality of service, 14, 15, 22, 24, 25, 45, 51, 223, 244, 259, 262, 266, 277, 278
 - feedback, 266
 - policy server, 56
- quasi birth-death process, 185
 - homogenous, 186
 - infinitesimal generator, 186, 188
 - inhomogenous, 188
 - linear level reduction, 189
 - quadratic matrix equation, 187
- queueing discipline, 62, 328
- queueing models
 - D/M/1, 148
 - D/M/c, 153
 - finite M/G/1, 134
 - average queue size, 134
 - average queueing time, 134
 - average system size, 134
 - average time in system, 134
 - mean arrival rate, 134
 - finite M/M/c, 105
 - average queue size, 106
 - average queueing time, 107
 - average system size, 106
 - average time in system, 107

- mean arrival rate, 106
 - steady state distribution, 105
- finite M/M/c/c, 107
 - loss probability, 107
 - steady state distribution, 107
- G/G/1, 150
 - waiting time distribution, 152
- G/G/1+G, 153
 - waiting time distribution, 154
- G/G/c, 152
 - average queueing time, 152
- G/M/1, 146
 - average queue size, 147
 - average queueing time, 147
 - average system size, 147
 - average time in system, 147
 - steady state distribution, 146
- G/M/c, 143
 - arrival steady state distribution, 144
 - average queue size, 146
 - average queueing time, 145
 - average system size, 146
 - average time in system, 146
 - conditional queue length distribution, 146
 - probability of delay, 146
- G/M/c/K, 148
 - average queue size, 149
 - average queueing time, 149
 - average system size, 149
 - average time in system, 149
 - effective arrival rate, 149
 - loss probability, 149
 - steady state distribution, 149
- M/D/1, 130
 - average system size, 131
 - steady state distribution, 131
- M/D/c
 - average system size, 136
 - steady state distribution, 135
- M/G/1, 124
 - average queue size, 129
 - average system size, 127, 129
 - average time in system, 129
 - average waiting time, 129
 - steady state distribution, 127
- M/G/1+G, 138
- M/G/1/1 with retrials, 139
 - average queue size, 141
 - average queueing time, 141
 - average system size, 141
 - average time in system, 141
 - steady state distribution, 141
- M/G/1/K, 132
 - average queue size, 133
 - average queueing time, 133
 - average system size, 133
 - average time in system, 133
 - effective arrival rate, 133
 - steady state distribution, 133
- M/G/c, 135
 - average queue size, 137
 - average system size, 135
 - steady state distribution, 136
- M/G/c/c, 97
- M/G/c/c with retrials, 142
 - average system size, 142
- M/G/c/K, 137
 - effective arrival rate, 137
 - loss probability, 137
 - steady state distribution, 137
- M/M/1, 81, 129, 147
 - average queue size, 85
 - average queueing time, 85
 - average system size, 84, 130

- average time in system, 85
- steady state distribution, 84, 130
- M/M/1/1 with retrials, 118
 - average queue size, 120
 - average queueing time, 120
 - average system size, 120
 - average time in system, 120
 - conditional average wait, 121
 - steady state distribution, 119
- M/M/1/K, 87
 - average queue size, 89
 - average queueing time, 89
 - average system size, 88
 - average time in system, 89
 - steady state distribution, 88
- M/M/c, 92
 - average queue size, 93
 - average queueing time, 94
 - average system size, 94
 - average time in system, 94
 - probability of delay, 93
 - steady state distribution, 92
- M/M/c+G, 111
 - average queue size, 115
 - average queueing time, 115
 - average system size, 115
 - average time in system, 115
 - loss probability, 114
 - steady state distribution, 113
- M/M/c+G with balking, 113
 - average queueing time, 115
 - steady state distribution, 114
- M/M/c+M, 100
 - average queue size, 104
 - average queueing time, 104
 - average system size, 104
 - average time in system, 104
 - loss probability, 103
 - Palm's solution, 102
 - probability of delay, 102
 - steady state distribution, 101
- M/M/c+M with balking, 100
 - steady state distribution, 100
- M/M/c/c, 96
- M/M/c/c with retrials, 121
 - average queue size, 122, 123
 - average queueing time, 123
 - average system size, 123
 - average time in system, 123
 - blocking probability, 123
- M/M/c/K, 95
 - average queue size, 97
 - average queueing time, 97
 - average system size, 97
 - average time in system, 97
 - probability of delay, 96
 - steady state distribution, 95
- M/ME/1, 168
 - average queueing time, 169
 - average system size, 169
 - average time in system, 169
 - steady state distribution, 168
 - waiting time distribution, 169
- M/ME/1/N, 164
 - average queue size, 168
 - average queueing time, 168
 - average system size, 168
 - average time in system, 168
 - mean service time, 168
 - steady state distribution, 166
- M/ME/1/K, 165
- M/ME/c, 179
 - maximum utilization, 179
 - steady state distribution, 181
- M/ME/c/N, 173

- average queue size, 179
- average queueing time, 179
- average system size, 179
- average time in system, 179
- steady state distribution, 178
- M/PH/1
 - infinitesimal generator, 172
- M/PH/c/c with retrials, 191
 - infinitesimal generator, 191
- ME/M/1, 170
 - steady state probabilities, 171
- ME/M/1//N, 169
- ME/M/c, 183
 - steady state distribution, 184
- ME/M/c//N, 181
 - steady state distribution, 182
- PH/M/1
 - infinitesimal generator, 172
- PH/M/c
 - infinitesimal generator, 184
- state dependent M/M/c+G, 116
 - average queueing time, 117
 - average queueing time if lost, 117
 - average queueing time if served, 117
 - effective arrival rate, 116
 - loss probability, 116
 - probability of delay, 116
 - steady state distribution, 116
- queueing theory, 59
 - statistical distributions, 63
- rate conservation law, 110
- Rouche's theorem, 111
 - Extended Version, 111
- service process, 61, 62, 320
- signaling
 - for multimedia call centres, 308
 - user-to-user, 299
- signaling systems, 222
 - broadband ISDN, 243
 - H.323, 258
 - IP based, 258
 - ISDN, 226
 - multimedia options, 250
 - QSig, 253
 - traditional, 225
- softswitch, 43
- split group, 13
- stochastic chain, 381
- stochastic matrix, 389
- stochastic process, 381
 - definition, 381
 - stationary, 382
- sufficient statistic, 198
- supervisor, 18
 - service level, 18
- switch architecture, 35
- switching systems
 - architecture, 39
 - evolution, 36
 - functionality, 35
 - network analysis, 47
- traffic
 - assessment, 314
 - data sources, 314
 - inbound, 11
 - intensity, 75
 - multimedia, 25, 264
 - outbound, 12
 - overflow, 341
 - processes, 316
 - self similar, 317

- specification (RSVP), 262
- types of (ATM), 247

virtual circuit, 23, 24

weak convergence, 72

work conservation, 89

workforce management, 61, 371