Stochastic Modeling and Analysis of 3G Mobile Communication Systems

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Universität Dortmund

am Fachbereich Informatik

von

Axel Thümmler

Dortmund

2003

Axel Thümmler

Lehrstuhl IV - Rechnersysteme und Leistungsbewertung
Fachbereich Informatik
Universität Dortmund
D - 44227 Dortmund

Tag der mündlichen Prüfung: 31. Juli 2003

Dekan: Prof. Dr. Bernhard Steffen

Gutachter: Prof. Dr.-Ing. Christoph Lindemann, Prof. Dr.-Ing. Heinz Beilner

Abstract

Third-generation (3G) mobile communication systems are currently one of the key communication technologies in research and development due to the high market demand for advanced wireless communication. The current evolution is primarily characterized by a transition from circuit-switched voice-oriented networks to integrated multi-service all IP networks. To effectively design complex mobile communication systems, the design process should be accompanied by stochastic modeling and quantitative evaluation of different design alternatives. The most popular language for model specification used in industrial projects is the Unified Modeling Language (UML). Although conceived as a general-purpose modeling language, the current version of the UML does not contain building blocks for introducing stochastic timing into UML diagrams.

The first part of this thesis presents new results for numerical quantitative analysis of discrete-event stochastic systems specified in Petri net notation or as UML diagrams. An efficient algorithm for the state space generation out of an UML state diagram or activity diagram that allows quantitative analysis by means of the underlying stochastic process is presented. Furthermore, this thesis considers new methodological results for the effective numerical analysis of finite-state generalized semi-Markov processes with exponential and deterministic events by an embedded general state space Markov chain (GSSMC). Key contributions constitute (i) the observation that elements of the transition kernel of the GSSMC can always be computed by appropriate summation of transient state probabilities of continuous-time Markov chains and (ii) the derivation of conditions under which kernel elements are constant. To provide automated tool support, the presented algorithms are included in the software package DSPNexpress-NG available for download on the Web.

The support of multimedia services over wireless channels presents a number of technical challenges. One of the major challenges is to effectively utilize the scarce radio bandwidth in the access network by adaptive control of system parameters. The second part of this thesis is devoted to this topic. A Markov model representing the sharing of radio channels by circuit-switched connections and packet-switched sessions under a dynamic channel allocation scheme is evaluated. Closing the loop between network operation and network control, a framework for the adaptive quality of service management for 3G mobile networks is introduced. Building on this framework, a novel call admission control and bandwidth reservation scheme for the optimization of quality of service for mobile subscribers is presented. The performance of the solutions proposed in this thesis is investigated experimentally based on numerical quantitative analysis and discrete-event simulation.

Acknowledgement

At this place, I would like to thank all the people who contributed by their support substantially to the success of this thesis. I owe a special debt of gratitude to my thesis advisor Prof. Dr.-Ing. Christoph Lindemann for his support, guidance and ideas during the course of this work. As his first employee he introduced me with a lot of engagement into the field of numerical analysis of stochastic processes. Furthermore, he taught me to effectively develop research publications. I am also grateful to my second advisor Prof. Dr.-Ing. Heinz Beilner for many helpful comments on my thesis.

I thank all my colleagues of the Lehrstuhl IV for the friendly and motivating working atmosphere, in which this thesis could be developed. In particular, I would like to mention my colleague Marco Lohmann, with whom I jointly elaborated some of my publications in the field of mobile communications systems. Furthermore, his work together with Oliver Waldhorst during their time as student researchers should be acknowledged. Moreover, I would like to thank the former students David Proba, Jens Hänsel, and Joachim Przybilke for their active assistance in my research projects.

Last but not least I would like to thank my girlfriend Kerstin and my mother Charlotte for their motivating support, particularly in difficult times, and for enduring many weekends when I was busy with this thesis.

Danksagung

An dieser Stelle möchte ich mich bei den Menschen bedanken, die durch ihre Hilfe wesentlich zum Gelingen dieser Arbeit beigetragen haben. Mein besonderer Dank gilt meinem Doktorvater Prof. Dr.-Ing. Christoph Lindemann für seine Unterstützung und seine Leitung auf dem Weg zur Vollendung dieser Arbeit. Als sein erster Mitarbeiter hat er mich mit viel Engagement in den Bereich numerischer quantitativer Analyseverfahren für stochastische Prozesse eingeführt. Des weiteren hat er mir durch seine stets vorantreibende Art das effektive Erstellen wissenschaftlicher Publikationen beigebracht. Bei meinem Zweitgutachter Prof. Dr.-Ing. Heinz Beilner möchte ich mich für die vielen hilfreichen Anmerkungen zu meiner Dissertation bedanken.

Allen Mitarbeitern des Lehrstuhl IV danke ich für die freundliche und motivierende Atmosphäre, in der diese Arbeit entstehen konnte. Besonders zu erwähnen ist hier mein Arbeitskollege Marco Lohmann, mit dem ich einige meiner Publikationen im Bereich mobiler Kommunikationssysteme zusammen erarbeitet habe. Zu erwähnen sind auch seine Arbeiten zusammen mit Oliver Waldhorst während ihrer Zeit als Diplomanden am Lehrstuhl IV. Des weiteren möchte ich den ehemaligen Studenten David Proba, Jens Hänsel und Joachim Przybilke für ihre aktive Mithilfe in meinem Forschungsvorhaben danken.

Nicht zuletzt danke ich meiner Freundin Kerstin und meiner Mutter Charlotte für ihre motivierende Unterstützung, besonders in schwierigen Zeiten, und für das Erdulden der vielen Wochenenden, an denen ich mit dieser Arbeit beschäftigt war.

Contents

Chapter 1

Introduction

This introductory part gives a motivation for the research challenges addressed in this thesis. Furthermore, previous results related to the issues considered in this work are recapitulated. The research contributions of this thesis and an outline on what follows in the next chapters are given in the third section. The last section explicitly points out my individual contributions of joint publications, which are part of this thesis.

1.1 Motivation

Third-generation (3G) mobile communication systems are currently one of the key communication technologies in research and development due to the high market demand for advanced wireless communication. The rapid growth in traffic volume and increase in new services has begun to change the configuration and structure of wireless networks. Thus, future mobile communication systems will be distinguished by high integration of services, flexibility, and higher throughput. To support such features, the efficient use of radio spectrum and optimum management of radio resources will be essential.

Evolution as a high level context covers not only the technical evolution of network elements but also expansions to network architecture and services. Thus, we are concerned with challenges on different levels. From the point of view of network architecture the evolution is characterized by the convergence of fixed and mobile networks, that is, the wireless network serves as an access platform for the existing wired network, e.g. the Internet. Furthermore, decentralized network architecture and the specification of open interfaces between separate subsystems enables the distribution of intelligence throughout the network. Considering the wireless access technology, the most important technical innovation in 3G mobile networks is Wideband Code Division Multiple Access (WCDMA). The WCDMA technology provides a better utilization of the scarce radio resources and is more robust against channel interference. From the services point of view the evolution is primarily characterized by a transition from circuit-switched voice-oriented networks to integrated multi-service all IP networks. Thus, applications such as e-mail, Web browsing, and corporate local network access, as well as video conferencing, e-commerce, and multimedia can be supported over wireless data channels.

Since different parts of the worlds emphasize different issues, the global term 3G has regional synonyms: In the US and Japan, 3G often carries the name International Mobile Telephony 2000 (IMT-2000). In Europe, 3G has become Universal Mobile Telecommunications System (UMTS) following the ETSI perspective [HWB00]. The European industrial players have created the 3rd Generation Partnership Project (3GPP, [3GPP]) for the standardization of UMTS. While the standardization of 3G is still ongoing the discussion of technical issues beyond 3G has already started [MWIF01], [WWRF]. The vision for the future of wireless communication systems beyond 3G consists of a combination of several optimized access systems on a common IP-based medium access and core network platform [AH01].

However, the support of multimedia services over wireless channels still presents a number of technical challenges. The scarcity of radio resources and the variety of services with diverse quality of service (QoS) requirements are the key driving forces for emerging research on adaptive control of system parameters to match current traffic conditions, thus, closing the loop between network operation and network control. In particular, effective call admission control for wireless sessions with a prioritization of handover calls is considered to be an important research issue [PS01]. An admission controller decides to accept or reject a user's request based on the QoS demand of the user and the current network state. The purpose of the admission controller is to guarantee the QoS requirements of the user who requested admission while not violating the QoS of already admitted users. Even in the fixed network adaptive control of system parameters is an important issue when considering real-time or non real-time services other than traditional best-effort services, e.g. as proposed in the differentiated services architecture (DiffServ, [BBC+98]).

To effectively design the elements of complex mobile communication systems the design process should be accompanied by quantitative evaluation of different design alternatives. Such quantitative evaluation considers measures like response times, queue lengths, throughput, or loss probabilities and helps understanding system performance. The most popular language for model specification used in industrial projects is the Unified Modeling Language (UML, [OMG01a]). It is the proper successor to the object modeling languages of three previously leading object-oriented methods (Booch, OMT, and OOSE). The UML was invented by Booch, Rumbaugh, and Jacobson. In 1997, the UML was adopted as a standard by the Object Management Group. Although conceived as a general-purpose modeling language, the current version of the UML does not contain building blocks for introducing stochastic timing into UML diagrams. Besides the importance for mobile communication systems, quantitative analysis of UML diagrams is particularly relevant for the emerging research field of software performance engineering (see e.g., [Smi90], [SW98]) as well as for system engineering at large.

There exist a large number of software packages for quantitative analysis of systems specified as queueing networks, stochastic process algebras, or stochastic Petri nets. These automated tools free system analysts from the painstaking construction and solution of the underlying stochastic processes by hand and enable them to focus on the task of translating the dynamic behavior of the system into the model notation. Software packages for stochastic Petri nets include among many DSPNexpress [Lin98], APNN-Toolbox [BBK98], Möbius [DKS02], and SPNP [CMT89]. Such packages contain state-of-the-art quantitative analysis techniques and are widely distributed in academia. However, typically the recognition of such a package in industry is limited. Commercial UML design packages widely used in industry contain sophisticated user interfaces, but such packages either rely on outdated quantitative analysis methods or do not provide methods for quantitative system evaluation at all. To close the gap between commercial UML design tools and academic software packages for performance and dependability evaluation, UML system specifications enhanced by timing constraints must be transformed to a modeling language suitable for quantitative analysis or even better directly to their underlying stochastic processes.

UML state diagrams, queueing networks, stochastic process algebras, or stochastic Petri nets are formal modeling languages to represent the dynamic behavior of a discrete-event system. A discrete-event system (DES) is a discrete-state, event-driven system; that is, its state evolution depends entirely on the occurrence of asynchronous discrete events over time. Throughout this thesis, we are concerned with stochastic DES, i.e., the state of the system becomes a stochastic process due to the occurrence of events at random points in time. Thus, a probabilistic framework is required to describe the system behavior. The most general form of the stochastic process underlying a DES is a generalized semi-Markov process (GSMP), see e.g., Glasserman and Yao [GY92], Glynn [Gly89], Shedler [She93], and Whitt [Whi80]. Due to their generality, the analysis of GSMPs can be performed by discrete-event simulation only. Nevertheless, under the restriction that events occur after an associated activity with exponentially distributed delay (called *events with exponentially distributed delay* or simply *exponential events* throughout this thesis), efficient numerical methods for quantitative analysis exist. Even for the case if the DES contains additionally deterministic events (i.e., events that occur after an associated activity with deterministically distributed delay), efficient numerical algorithms exist under the restriction that deterministic events are not concurrently enabled. For the former type of DES the underlying GSMP reduces to a continuous-time Markov chain and for the latter one it reduces to a Markov regenerative stochastic process, respectively.

The tremendous increase of computer power in terms of processor speed and available main memory in recent years opens new possibilities for numerical analysis of DES. Today's modern off-the-shelf PCs have up to 2.5 GHertz clock rate and a main memory of 2 GBytes. As a consequence today's PCs are orders of magnitudes faster in numerical computations and can keep substantially more data in main memory than PCs or workstations several years ago. Thus, one major research challenge is to extend the class of numerically solvable stochastic processes. This can be done by utilizing new methodological results on GSMPs and by employing modern high-speed PCs for numerical solution. One part of this thesis addresses this issue.

1.2 Previous Work

This section focuses on previous work related to the issues addressed in this thesis. In particular, recent work on quantitative analysis of DES specified in the UML is considered. Furthermore, research results on numerical analysis of DES with exponential and deterministic events are recapitulated. These results mainly rely on the numerical steady-state and transient analysis of deterministic and stochastic Petri nets (DSPNs, [AC87]). Previous results on adaptive QoS management for 3G mobile networks are presented in the third subsection.

1.2.1 Quantitative Analysis of UML Specifications

The quantitative analysis of time-enhanced UML diagrams is an emerging area of research. As a first step in this direction, Douglass specified language extensions of the UML for specifying real-time constraints such as deadlines [Dou99]. Furthermore, there are activities of the OMG to extend the current version of the UML for modeling real-time applications. Therefore, the OMG sent out a Request for Proposal (RFP) that addresses the issue of schedulability, performance, and time [OMG99]. An initial response to the RFP that was recently adopted as a final specification was submitted by a group of OMG members consisting primarily of vendors of different kinds of real-time tools [OMG01b]. From the modeling point of view, the specification mainly addresses the issue of modeling general resources. A resource is viewed as a server for which the services can be qualified by one or more QoS characteristics (e.g., a response time). From the analysis point of view, the response introduces modeling approaches that are tailored to schedulability analysis and performance analysis. With schedulability analysis an execution order of different entities of the system is determined for optimizing criteria such as meet all hard deadlines or minimize the number of missed deadlines. The performance analysis model defines UML extensions for e.g. modeling workloads and performance values. It is demonstrated by a Web video application that is modeled with annotated activity diagrams. However, a detailed understanding how to effectively derive performance measures from UML diagrams is missing.

Recently, Cortellessa and Mirandola developed a framework for generating a performance model from parts of UML diagrams [CM00a], [CM00b]. Their proposed methodology makes use of UML use case diagrams, sequence diagrams, and deployment diagrams. They combined a set of sequence diagrams derived from different use cases to generate an execution graph. From the deployment diagram they derived an extended queueing network model that is parameterized by means of the execution graph. Nevertheless, this approach relies entirely on traditional approaches in software performance engineering introduced by Smith in 1990 [Smi90] and tailored to the UML in [SW98]. Petriu, Shousha, and Jalnapurkar developed a systematic approach to build a layered queueing network performance model from a UML description [PSJ00]. They demonstrated their approach by analyzing an existing telecommunication system. King and Pooley developed a methodology that considers the generation of a generalized stochastic Petri net from UML state diagrams embedded in collaboration diagrams [KP00]. The translation is obtained by associating to each state in the state diagram a place in the Petri net and to each transition in the state diagram a transition in the Petri net. Nevertheless, the performance analysis of UML state diagrams via the generation of a (canonical) Petri net results in an unnecessary overhead since the Petri net contains immediate transitions, that should be omitted in the quantitative analysis.

The need for methods and tools for quantitative analysis of performance of communication software is expressed in the *Workshop on Software and Performance (WOSP)*. This workshop is intended to bring together software engineers, developers, performance analysts, and modelers. The first workshop was held in 1998 in Santa Fe and subsequent workshop meetings followed on a two-year base in Ottawa and Rome.

1.2.2 Numerical Analysis of Discrete Event Systems

Previous work on the numerical analysis of discrete-event stochastic systems with exponential and deterministic events was mainly done in the context of deterministic and stochastic Petri nets (DSPN). For both stationary and transient analysis of DSPNs, approaches based on Markov renewal theory (see e.g. [AC87], [CKT94]) and on the method of supplementary variables (see e.g. [GL94], [GH99]) have been considered. Unfortunately, the practical applicability of the supplementary variables approach is severely limited because it requires, already in the restricted case, numerical solution of a system of partial differential equations. Under the restriction that in any marking of a DSPN at most one deterministic transition is enabled, a highly efficient numerical method for steady-state analysis has been introduced [Lin93] and implemented in the software package DSPNexpress [Lin98].

While steady-state analysis allows the evaluation of long run behavior of computer and telecommunication systems, a considerable number of important performance and dependability studies require the analysis of time-dependent behavior; i.e., transient analysis.

Previous work on transient analysis of DSPNs was always based on the restriction that deterministic transitions are not concurrently enabled. Choi, Kulkarni, and Trivedi observed that the marking process underlying a DSPN with this restriction is a Markov regenerative stochastic process [CKT93]. They introduced a numerical method for transient analysis of such DSPNs based on numerical inversion of Laplace-Stieltjes transforms. While this numerical method is certainly of theoretical interest, it is not suitable for transient analysis of large DSPNs. Recently, Ciardo and Li considered the approximate transient analysis of DSPNs with only a single deterministic transition that cannot get canceled [CL99].

Telek and Horváth recently studied the analysis of non-Markovian models with different preemption policies of non-exponential transitions. Using the supplementary variables approach, they developed state equations for transient analysis of Markov regenerative stochastic Petri nets in which timed transitions keep their remaining firing times if their firing process gets preempted and subsequently resumed (denoted by *preemptive resume, prs*) instead of discarding them and restarting the firing process (denoted by *preemptive repeat different, prd*) [TH98]. In a recent paper they considered the time domain analysis of non-Markovian stochastic Petri nets with *preemptive repeat identical (pri)* type non-exponential transitions, i.e., in case of preemption and subsequent resumption of a non-exponential transition the firing process is restarted with the previously sampled random firing time [TH02]. The analysis of non-Markovian models under the assumption that only one non-exponential transition can be enabled at a time was also studied by Grassmann [Gra82] and by de Souza e Silva, Gail, and Muntz [SGM95]. Their approach is based on a discrete-time Markov chain embedded at starting or completion times of non-exponential events.

Considering finite-state GSMPs with exponential and deterministic events, Lindemann and Shedler introduced the first cost-effective numerical method for the analysis of such processes. Their approach is based on a general state space Markov chain (GSSMC) embedded at equidistant time points of the continuous-time GSMP [LS96]. This numerical approach constitutes of two main steps: the derivation of the transition kernel and the solution of a system of multidimensional Fredholm integral equations. Efficient numerical solvers for these integral equations, which constitute the time-dependent and stationary equations of the GSMP, have also been presented in [LT99] and [LS96], respectively.

1.2.3 QoS Management for 3G Mobile Networks

One new innovation in 3G mobile networks constitutes the introduction of a packet-switched bearer service, called the *General Packet Radio Service (GPRS)* [ETSI99]. By adding GPRS functionality to the existing GSM network, operators can give their subscribers resource-efficient wireless access to external Internet protocol-based networks, such as the Internet and corporate intranets. As impressively demonstrated by the Internet, packet-switched networks make more efficient use of the resources for bursty data applications and provide more flexibility in general.

To evaluate the performance of GPRS, several simulation studies were conducted. Early simulation studies for GPRS have been reported in [BW97], [CG97]. Meyer evaluated the performance of TCP over GPRS under several carrier to interference conditions and data coding schemes [KMM00], [Mey99]. Malomsoky, Nádas, Tóth, and Zarándy developed a simulator for dimensioning GSM networks with GPRS [MNT+00]. Stuckmann, and Müller developed a system simulator for GPRS and studied the correlation of GSM and GPRS users for fixed and on-demand channel allocation techniques [SM00]. Several analytical models based on continuous-time Markov chains have been introduced for studying performance issues in GSM networks. Ajmone Marsan, Marano, Mastroianni, and Meo evaluated the impact of reserving channels for data and multimedia services on the performance in a circuit-switched GSM network [AMM+00]. Ajmone Marsan, De Carolis, Leonardi, Lo Cigno, and Meo developed an approximate analytical model for evaluating the performance of dual-band GSM networks [AM00]. Boucherie and Litjens developed a Markov model for analyzing the performance of GPRS under a given GSM call characteristic [BL00]. Recently, Ermel, Begain, Müller, Schüler, and Schweigel developed a Markov model for deriving blocking probabilities and average data rates for GPRS in GSM networks [BEM+00]. In none of these previous works, the question how many packet data channels should be allocated for GPRS for a given amount of traffic in order to guarantee appropriate quality of service has been investigated.

The QoS concept and architecture for UMTS networks has been specified by the 3GPP in [3GPP01a]. There, the terms of the UMTS management and control functions (e.g., admission controller and resource manager) are defined and their functionality is roughly outlined. However, a detailed technical understanding how network management should effectively be performed for 3G networks is subject to current industrial and academic research. In a visionary paper, Schwartz posed the engineering challenges in network management and control occurring from the introduction of multimedia services for wireless networks [Sch95]. Due to the scarce and costly radio frequencies for 3G mobile networks, adaptive resource management constitutes an important design issue for such networks. Das, Jayaram, Kakani, and Sen proposed a framework for QoS provisioning for multimedia services in 3G wireless access networks [DJK+00]. They developed an integrated framework by combining various approaches to call admission control, channel reservation, bandwidth degradation, and bandwidth compaction.

In [LLT02], we introduced a framework for the adaptive control of UMTS networks, which utilizes online monitoring of QoS measures (e.g., handover failure and call blocking probabilities) in order to adjust system parameters of the admission controller and the packet scheduler. The presented approach is based on a look-up table called the Performance

Management Information Base (P-MIB). Entries of the P-MIB are determined using extensive offline simulation experiments. In fact, many simulation runs have to be conducted to determine the optimal parameter configuration for the considered scenarios. Given the entries of the P-MIB, we showed how to improve QoS for mobile users by periodically adjusting system parameters. However, the practical applicability of this approach is limited if the P-MIB comprises many entries (i.e., many scenarios have to be considered) because of the high computational effort for determining these entries by many simulation experiments.

1.3 Summary of Contributions in this Thesis

This thesis aims to combine the different research issues of modeling and analysis of DES with research topics arising in 3G mobile networks. First of all, Chapter 2 gives a brief introduction to the basic concepts of mobile communication systems of the third generation. In particular, the principles of cellular radio communications, the UMTS network architecture, and the different QoS classes defined for UMTS are introduced.

From the modeling point of view, Chapter 3 presents an approach for the automatic generation of a performance evaluation model from system specifications described through UML state diagrams or activity diagrams. To enable quantitative evaluation of UML system specifications, the building blocks of the UML must be enhanced for specifying deterministic and stochastic delays and a well-defined mapping of UML diagrams onto their underlying DES (i.e., the underlying stochastic process) must be introduced. The contribution presented in Chapter 3 is twofold: First, extensions to UML state diagrams and activity diagrams to allow the association of events with exponentially distributed and deterministic delays are proposed. Subsequently, it is shown how to map these time-enhanced UML system specifications onto the underlying stochastic process. A particular stochastic process, the generalized semi-Markov process, is identified as the appropriate vehicle on which quantitative analysis is performed. Second, an efficient algorithm for the direct and automated state space generation out of these UML diagrams that removes vanishing states, i.e. states with only immediate transitions enabled, and considers branching probabilities within state diagrams is presented. Furthermore, a performance evaluation framework that allows a system designer to predict performance measures at several steps in the design process by the concept of nested states provided in state diagrams is introduced.

The approach presented in Chapter 3 is implemented in the new version of DSPNexpress [Lin98], called *DSPNexpress-Next-Generation (DSPNexpress-NG)* [DSPN], that provides tool support for the automated quantitative analysis of DES underlying UML diagrams and Petri nets. DSPNexpress-NG contains filters to the commercial UML design packages Rhapsody™ [Rhap] and Together™ [Toge]. The linkage of the DSPNexpress software to commercial UML design packages effectively supports the design process because these tools contain sophisticated user interfaces for user-friendly model specification that are widely used in industry.

Considering the analysis of DES, Chapter 4 builds on previous results presented in [LS96] for the numerical analysis of GSMPs with exponential and deterministic events. Recall that this previous approach is based on a GSSMC embedded at equidistant time points nD (n=1,2,...) of the continuous-time GSMP. To make this GSSMC approach effectively applicable in performance and dependability modeling projects at large, the remaining open problem constitutes the algorithmic generation of the simplest form of the transition kernel of this GSSMC given the building blocks of the GSMP. The transition kernel of the GSSMC specifies one-step jump probabilities from a given state s at instant of time nD to all reachable new states s' at instant of time (n+1)D. In general, elements of the transition kernel of a GSSMC are functions of clock readings associated with the old state s and intervals of clock readings associated with the new state s'. Two theorems that provide the foundation for such an effective algorithmic generation of the transition kernel are presented. Key contributions constitute (i) the observation that kernel elements can always be computed by summation of transient state probabilities of continuous-time Markov chains (Theorem 4.9) and (ii) the derivation of conditions on the building blocks of the GSMP under which kernel elements are constant; i.e., are not functions of clock readings (Theorem 4.11). The exploitation of constant kernel elements is the key driver for a fast solution methodology and additionally reduces memory requirements significantly. The impact of these theorems is illustrated by describing the algorithmic generation of the transition kernel and the exploitation of constant kernel elements.

Throughout Chapter 4, the presented methodology is described with a running example of an M/D/2/K queue. To illustrate the practical applicability of the methodological results, an MMPP/D/2/K queueing system and an UML state diagram representing a single cell in a cellular network are considered as examples. Performance curves plotting computational effort for the derivation of the transition kernel and for the time-dependent and stationary solution of the considered examples are presented. The presented numerical analysis algorithm is implemented in the open-source software package DSPNexpress-NG [DSPN]. Software download of DSPNexpress-NG and an online demonstration of the GSMP analysis algorithm are provided on the Web [DSPN].

Applying stochastic modeling with UML diagrams and numerical quantitative analysis, Chapter 5 presents an efficient and accurate model for the radio interface of the General Packet Radio Service (GPRS) in a GSM network. The GPRS model represents the sharing of radio channels by circuit-switched GSM connections and packet-switched GPRS sessions under a dynamic channel allocation scheme. A fixed number of physical channels are assumed to be permanently reserved for GPRS sessions and the remaining channels are assumed to be shared by GSM and GPRS connections. The model is utilized for investigating how many packet data channels should be allocated for GPRS for a given amount of traffic in order to guarantee appropriate quality of service. Performance curves for average carried data traffic, packet loss probability, throughput per user, and queueing delay for different network configurations and traffic parameters are presented. The model is presented in time-enhanced UML state diagram notation as proposed in Chapter 3.

In contrast to previous work, the GPRS model explicitly represents the mobility of users by taking into account arrivals of new GSM and GPRS users as well as handovers from neighboring cells. Furthermore, the traffic model defined by the 3rd Generation Partnership Project (3GPP) in [ETSI98] that can be effectively represented by an interrupted Poisson process, i.e., an on-off source, is employed. A cluster comprising seven hexagonal cells in an integrated GSM/GPRS network, serving circuit-switched voice and packet-switched data sessions is considered. To allow the effective employment of numerical solution methods, the GPRS model represents just one cell (i.e., the mid cell) and employs the procedure for balancing incoming and outgoing handover rates introduced in [ACL+99]. To validate this simplification, a comparison of the results of the GPRS model with a detailed simulator implemented using the simulation library CSIM [CSIM] is provided. The simulator represents the entire cell cluster on the network level. Furthermore, an implementation of the TCP flow control mechanism is included in the simulator. This validation shows that almost all performance curves derived from the UML model of GPRS lie in the confidence intervals of the corresponding curve derived from the simulator. Using the presented GPRS model sensitive performance measures can be computed on a modern PC within few minutes of CPU solution time. Note that even with simulation runs in the order of hours proper estimates for measures such as very low packet loss probabilities cannot be derived using discrete-event simulation because the large width of confidence intervals makes the results meaningless.

Chapter 6 introduces the concept of adaptive QoS management of 3G mobile networks in order to improve QoS for mobile subscribers. The introduced approach constantly monitors QoS measures such as packet loss probability and handover failure probability during operation of the network. Based on the values of the QoS measures just observed, system parameters of an admission controller are adjusted by an adaptive QoS management entity. Thus, the adaptive control framework closes the loop between network operation and network control. System parameters to be controlled during network operation comprise a threshold value of the access queue for admission of non real-time traffic, and watermarks specifying bandwidth portions of the overall available bandwidth for data, voice, and handover traffic. As a main new result, the introduced approach is based on a mathematical framework for the proposed update schemes rather than a look-up table. As a consequence, the adaptive control mechanism can be adjusted in an intuitive way and optimal system parameter configurations can efficiently be determined. Performance curves derived by simulation evidently illustrate the gain of the approach for adaptive QoS management. In fact, for UMTS networks, simulation results show that handover failure probability can be improved by more than one order of magnitude. Moreover, packet loss probability can be effectively regulated to a predefined level.

Chapter 7 summarizes the results of this thesis. Furthermore, research issues that have been left open are discussed and directions for future research are given.

1.4 Individual Contributions of Joint Publications

In the following, my individual contributions of joint publications, which are part of this thesis, are summarized. Prof. Dr.-Ing. Christoph Lindemann supervised the research work and the preparation of the published manuscripts. Furthermore, he contributed to the basic ideas underlying this thesis and helped identifying the fundamental research challenges.

The direct mapping of time-enhanced UML diagrams onto GSMPs (see Chapter 3) was developed by myself (see [LTK+00] and [LTK+02]). The implementation of the corresponding algorithms and the integration into DSPNexpress-NG was supported by the diploma thesis of David Proba [Pro03]. Alexander Klemm, Marco Lohmann, and Oliver Waldhorst supported this work during their time as student researchers. Alexander Klemm implemented in his diploma thesis a first software prototype for mapping UML diagrams, designed with the tool Statemate™, onto canonical Petri nets [Kle99]. Marco Lohmann and Oliver Waldhorst implemented in their diploma theses software prototypes of the algorithms for generating the transition kernel of a GSSMC [Wal00] and for numerical solution of the system of integral equations [Loh00] (see Chapter 4).

The results of Chapter 4 build on previous results of Lindemann and Shedler [LS96], [Lin98]. They proposed an approach for steady-state analysis of concurrent DSPNs by an embedded GSSMC. I developed new results on properties of the transition kernel of the GSSMC (Theorem 4.9 and Theorem 4.11), that constitute the building blocks for an efficient algorithm for kernel generation. Furthermore, I determined the system of Fredholm integral equations for transient analysis [LT99]. Without these new results a practical implementation of the GSMP analysis methodology would not have been possible. An extract of Chapter 4 with newly derived results on properties of the transition kernel is submitted for publication [LT03b].

The GPRS model presented in Chapter 5 was developed, implemented, and analyzed by myself. The implementation of the simulator for validating the analytical GPRS model was supported by the diploma thesis of Joachim Przybilke [Prz00]. In addition to the published results [LT01a], [LT01b], and [LT03a] this thesis contains a representation of the GPRS model with UML state diagrams, which can be analyzed with the methods presented in Chapter 3 and 4.

The contributions presented in Chapter 6 are results from combined research work together with Marco Lohmann (see [LLT02], [LLT03a], and [LLT03b]). This thesis concentrates on the adaptive management of quality of service for mobile subscribers by adjusting system parameters on session level, i.e., admission control. In contrast, Marco Lohmann investigated the adaptive revenue management for improving provider revenue by adjusting queueing weights of a packet scheduler, that is, he considered management of system parameters on packet level [Loh04]. Furthermore, Marco Lohmann developed a thorough approach to traffic modeling for 3G mobile networks, which is also part of the joint publications but not part of this thesis. The implementation of the UMTS simulator presented in Chapter 6 was supported by the diploma thesis of Jens Hänsel [Hän01]. The results presented in [LLT03b] are submitted for publication and are currently under review.

Publications

C. Lindemann and A. Thümmler, Numerical Analysis of Generalized Semi-Markov Processes, *(submitted for publication)*, 2003.

C. Lindemann, M. Lohmann, and A. Thümmler, Adaptive Call Admission Control for QoS/Revenue Optimization in CDMA Cellular Networks, *(submitted for publication)*, 2003.

C. Lindemann, M. Lohmann, and A. Thümmler, A Unified Approach for Improving QoS and Provider Revenue in 3G Mobile Networks, *Mobile Networks and Applications* **8**, 209-221, 2003.

C. Lindemann and A. Thümmler, Performance Analysis of the General Packet Radio Service, *Computer Networks* **41**, 1-17, 2003.

C. Lindemann, A. Thümmler, A. Klemm, M. Lohmann, and O. Waldhorst, Performance Analysis of Time-enhanced UML Diagrams Based on Stochastic Processes, *Proc. 3rd Int. Workshop on Software and Performance (WOSP), Rome, Italy*, 25-34, 2002.

C. Lindemann, M. Lohmann, and A. Thümmler, Adaptive Performance Management for UMTS Networks, *Computer Networks* **38**, 477-496, 2002.

C. Lindemann and A. Thümmler, Performance Analysis of the General Packet Radio Service, *Proc. 21st Int. Conference on Distributed Computing Systems (ICDCS), Phoenix, Arizona*, 673-680, 2001.

C. Lindemann and A. Thümmler, Evaluating the GPRS Radio Interface for Different Quality of Service Profiles, in: U. Killat, W. Lamerdorf (Eds.), *Informatik aktuell: 12th GI/ITG Fachtagung Kommunikation in Verteilten Systemen (KiVS), Hamburg, Germany*, 291-301, Springer, 2001.

C. Lindemann, A. Thümmler, A. Klemm, M. Lohmann, and O. Waldhorst, Quantitative System Evaluation with DSPNexpress 2000, *Proc. 2nd Int. Workshop on Software and Performance (WOSP), Ottawa, Canada*, 12-17, 2000.

C. Lindemann and A. Thümmler, Transient Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions, *Performance Evaluation* **36&37**, 35-54, 1999.

C. Lindemann, A. Reuys, and A. Thümmler, The DSPNexpress 2.000 Performance and Dependability Modeling Environment, *Proc 29th Int. Symposium on Fault-Tolerant Computing Systems (FTCS), Madison, Wisconsin*, 228-231, 1999.

Chapter 2

Mobile Communication Systems of the Third Generation

This chapter is intended to provide the basic concepts and notations for 3G mobile communication systems. In particular, the cellular concept, which resolves the basic problems of radio systems in terms of radio system capacity constraints, is introduced. Furthermore, the UMTS access network and core network architecture as well as the UMTS quality of service profiles as specified by the 3rd Generation Partnership Project [3GPP] are considered. These basics are the background for the proposed solutions and performance studies presented in the subsequent chapters of this thesis.

2.1 Cellular Radio Communication Principles

A wireless access network provides *wireless terminals* within the service area with the possibility to reach information as well as other terminals. Ideally, the access network is an interface to global telephone and data communication networks. Communication can be initiated either from the wireless terminal or from within the fixed network. A terminal can then forward or retrieve information streams such as speech, sound, video, text, programs, or combinations thereof. High level applications can use the information flow to provide, for example, a wide range of information and transaction based services.

Service everywhere within a large geographical area can be achieved by covering the area with many *base stations*. Since radio signals are attenuated with distance, there is a limited area within which it is possible for mobile terminals to communicate with a specific base station. The maximum geographical area served by a base station is denoted a *cell*. Cells are usually designed to overlap so that an ongoing connection can be transferred from one base station to the next when terminals move. Cells also need to overlap due to the fact that the electromagnetic waves from an antenna do not spread out uniformly, but tend to be reflected, scattered, or shadowed by physical objects in the base station's surroundings. Maintaining the traffic connection with a moving terminal is made possible with the help of the *handover*

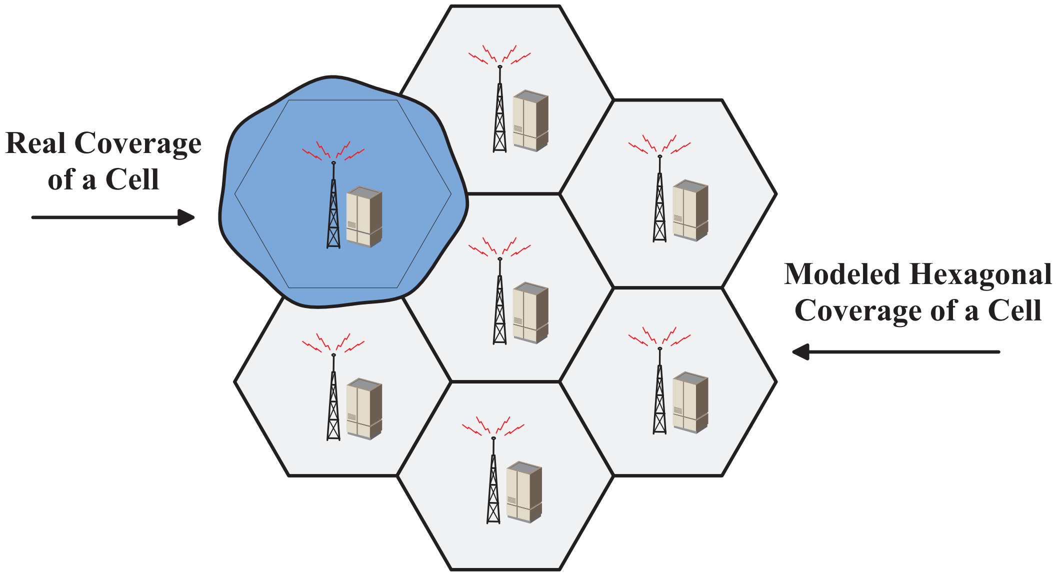


Figure 2.1. A cluster of cells in a cellular network

procedure, i.e., when the terminal moves from the coverage area of one cell to another, a new connection with the target cell has to be set up and the connection with the old cell may be released. Figure 2.1 illustrates a cluster of seven cells in a cellular network.

The primary radio resource is a predefined frequency band of the radio spectrum. This frequency band is usually partitioned into two sub-bands corresponding to *uplink* (i.e., the link from the wireless terminal to the base station) and *downlink* (i.e., the link from the base station to the wireless terminal) connections. Furthermore, each sub-band is divided into many equally spaced *carrier frequencies*, allowing several users to simultaneously utilize the frequency band. This concept is called *frequency division multiplexing (FDM)*. A more fine-grained partitioning of a frequency band is possible either by imposing a frame structure with several time slots (*time division multiplexing, TDM*), or by utilizing orthogonal codes (*code division multiplexing, CDM*). The acronyms FDMA (frequency division multiple access), TDMA (time division multiple access) and CDMA (code division multiple access) are used to categorize systems based on these multiple access principles. For FDMA systems, a single carrier frequency is denoted a *channel*. For TDMA systems, the term usually refers to a single periodical recurring time slot on a carrier frequency. Thus, a TDMA system allows a number of users to access a single frequency channel. Unlike in TDMA and FDMA systems, all simultaneously active connections can occupy the same bandwidth at the same time in CDMA systems. Every connection is assigned a code used for cell, channel, and user separation. Since every user uses the same frequency band simultaneously, there is no time slots or frequency allocation in the same sense as in TDMA and FDMA based system, respectively. Thus, for CDMA systems a combination of code sequence and frequency band define a channel.

The cellular concept increases the radio system capacity especially when utilizing with *frequency reuse*, i.e., in each cell of a cluster a different frequency band is used and in neighboring clusters the same frequency bands can be reused. The smaller the cells, the more efficiently the radio spectrum is used but the cost of the system increases at the same time because more base stations are needed. Another important issue when dealing with cellular networks is interference. The basic reason behind interference is that there are many simultaneous radio connections to the base station. Depending upon the source of interference, it can be classified as intra-cell interference, inter-cell interference, and interference due to thermal noise. In FDMA/TDMA based systems inter-cell interference is the key issue to deal with whereas in CDMA based systems intra-cell interference is the most crucial type of interference.

2.2 UMTS Network Architecture

The European standard for third generation mobile communication systems is the *Universal Mobile Telecommunication Systems (UMTS)*, which is still under standardization by the *3rd Generation Partnership Project (3GPP)*. The 3GPP originally decided to prepare specifications on a yearly basis, with the first specification release being Release 99 (3GPP R99). This first specification set has a relatively strong “GSM presence”. From the UMTS point of view the GSM presence is very important since, first, the UMTS network must be backward compatible with the existing GSM networks and second, the GSM and UMTS networks must be able to inter-operate together. Future specification activities were scheduled into two new specification releases 3GPP R4 and 3GPP R5 (the 3GPP R99 is sometimes called 3GPP R3). The 3GPP R4 defines major changes in UMTS core network circuit-switched side and those are related to the separation of user data flows and their control mechanisms. The 3GPP R5 aims to introduce a UMTS network where the transport network utilizes IP networking as much as possible. This goal is called the “All IP” network and contributes to the evolution of mobile communication systems of the fourth generation (4G) [DP00], [DF01].

The UMTS network architecture standardized by the 3GPP R99 is presented in Figure 2.2. The release distinguishes the UMTS access network, i.e., the *UMTS Terrestrial Radio Access Network (UTRAN)* [3GPP01b], and the UMTS core network [3GPP00]. The UTRAN consists of a set of *Radio Network Controllers (RNC)* that are connected to the core network through the open interface *Iu*. A RNC is connected to a set of Node B elements, referred to as *Base Stations (BS)* in this thesis, each of which can serve one or several cells (depending on omni-directional or sector cells). The RNC is responsible for control of the connected BS, i.e., transceiver stations, and the radio link *Uu* to *the User Equipment (UE)*. Inside the UTRAN, several radio network controllers can be interconnected together to support smooth handover for mobile stations leaving the area covered by the serving RNC and entering the area of a drift RNC.

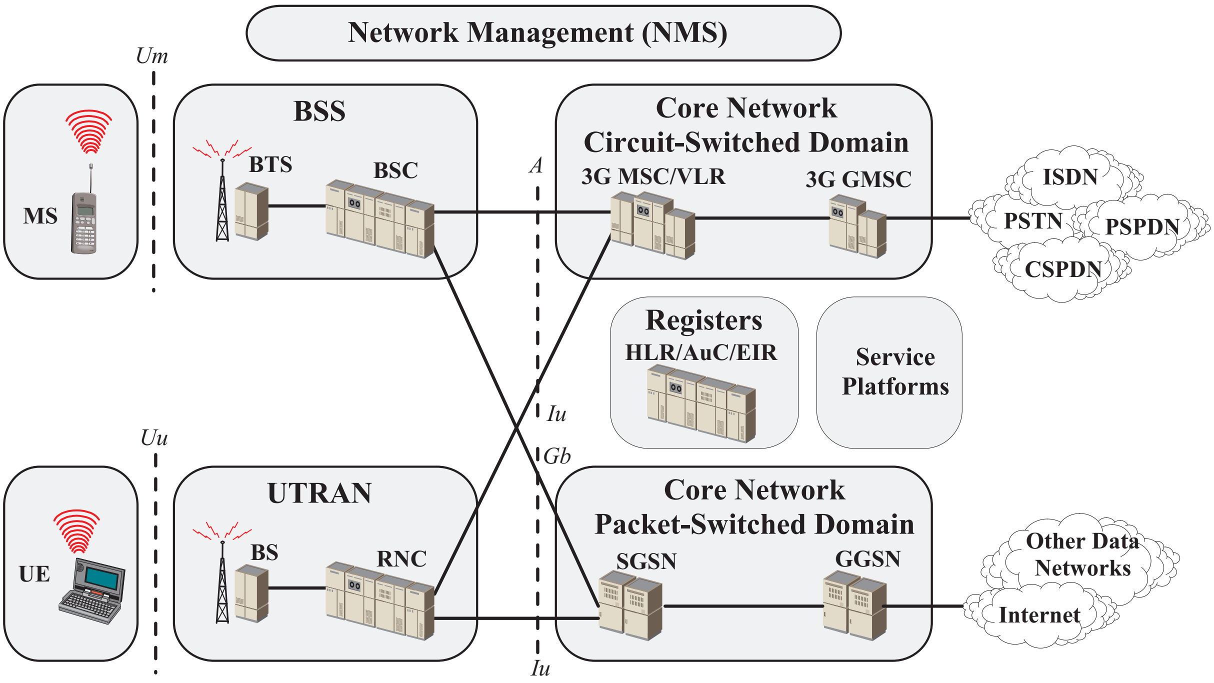


Figure 2.2. Architecture of UMTS Access- and Core Network

3G introduces the new radio access method Wideband CDMA (WCDMA). Since WCDMA and the corresponding radio access equipment are not compatible with GSM equipment and interoperability between existing GSM and UMTS should be provided, traditional GSM network elements are still required (even with some changes). In GSM systems the radio access network is called the *Base Station Subsystem (BSS)*, which is connected to the core network through the interfaces *A* and *Iu*. The counterpart of the Node B element is the *Base Transceiver Station (BTS)* which is connected to the *Mobile Stations (MS)* via the radio link *Um*. A GSM *Base Station Controller (BSC)* is responsible for similar functionality as a RNC.

The UMTS Core Network (CN) is the basic platform for all communication services provided to the UMTS subscriber. The basic communication services include switching of circuit-switched calls and routing of data packets from packet-switched connections. Both of these traffic types require some specific arrangements and this is why the core network functionality is further divided into two domains, packet-switched and circuit-switched domains. The circuit-switched domain has two basic network elements, which can be physically combined. These elements are the *Mobile Switching Centre/Visitors Location Register (MSC/VLR)* and the *Gateway Mobile Switching Centre (GMSC)*. The MSC/VLR element is responsible for circuit-switched connection management activities, mobility management related issues like location update, location registration, paging, and security activities. The GMSC element takes care of the incoming/outgoing connections to/from other networks, like the *Public Switched Telephone Network (PSTN)*, the *Integrated Services Digital Network (ISDN)*, or *Circuit-Switched* and *Packet-Switched Public Data Networks (CSPDN/PSPDN)*, respectively.

The packet-switched domain of the core network essentially represents the functionality introduced by the *General Packet Radio Service (GPRS)* in GSM Phase 2+ [ETSI99]. Two new node types, *Serving GPRS Support Node (SGSN)* and *Gateway GPRS Support Node (GGSN)*, are introduced to handle packet-switched data. The GGSN is the gateway node between an external packet-switched data network (e.g. the Internet) and the core network. In case of an external IP network, the GGSN is seen as an ordinary router serving all addresses that were static or temporarily assigned to the mobile stations. Its task is to assign the correct SGSN for a mobile station depending on the location of the mobile station. The SGSN connects the core network and the radio access network, and switches the packets to the correct BSC via the *Gb* interface or the correct RNC via the *Iu* interface, respectively.

In Figure 2.2 the part named “Registers” contains *Home Location Register (HLR), Authentication Centre (AuC),* and *Equipment Identity Register (EIR)*. This part of the core network contains the addressing and identity information for both the circuit-switched and packet-switched domains. The HLR contains permanent data of the subscribers, the AuC is a database generating authentication vectors for security activities, and the EIR maintains identification information related to the user equipment hardware. In addition to these registers, the VLR database contains temporary copies of the active subscribes, which have performed location update in the area covered by the corresponding MSC.

From the service point of view several service platforms are built on the basic UMTS architecture. This differentiation between service platforms and UMTS network elements creates more commercial potential and openness in the market place. Service platforms are for example the value added services known from GSM, WAP offering a Web browser for the end user, positioning services, or the service platform CAMEL, which enables customized services for individual subscribers, e.g., a virtual home environment. For maintenance and operation of the UMTS access network and core network the *Network Management Subsystem (NMS)* is required. It performs tasks like fault control by monitoring the allocated network resources, reconfiguration of network elements, or collection of performance statistics.

2.3 UMTS Quality of Service Architecture

From the services point of view the evolution from first generation analog mobile networks over second generation to 3G mobile communication systems is an evolution from single-service technology limited networks to multi-service networks that support a variety of applications with different traffic characteristics. Since different traffic types, e.g., real-time or non real-time traffic, require different aspects of quality of service (QoS) a demand for defining different QoS classes arises.

	Guaranteed Services		**Best-Effort Services**	
	Conversational class	**Streaming class**	**Interactive class**	**Background class**
Bandwidth requirements	guaranteed bit rate	guaranteed bit rate	high priority	low priority
Delay	stringent delay bound (< 100 msec.)	not bounded	not bounded	not bounded
Delay jitter	limited	limited	not limited	not limited
Application examples	Real-time traffic flows		Non real-time traffic flows	
	telephony speech, video conferencing	video-on-demand, HDTV, e-newspaper	Web browsing	background download FTP, e-mail

Table 2.3. Characterization of UMTS traffic

The QoS architecture specified for UMTS by 3GPP [3GPP01a] distinguishes between four QoS classes: *conversational class*, *streaming class*, *interactive class*, and *background class*. The main distinguishing factor between these QoS classes lies in the delay sensitivity of the traffic. The conversational class is meant for traffic that is very delay sensitive while the background class is the least delay sensitive traffic class. Conversational and streaming classes are mainly intended to be used to carry real-time traffic flows. A conversational real-time traffic stream is characterized by requiring low transfer delay and small delay-jitter because of the conversational nature of the stream. The maximum transfer delay is given by the human perception of video and audio conversation (e.g., video conferencing or voice over IP). The streaming traffic class consists of one-way real-time traffic streams, e.g., viewing video clips or audio clips. The limit for acceptable transfer delay and delay-jitter is not as stringent as in the conversational class. Delay-jitter can be reduced by a time alignment function at the receiver that buffers data packets temporarily.

Interactive class and background class are mainly meant to be used by traditional Internet applications like WWW, e-mail, and FTP. The main difference between the interactive and the background class is that interactive class is mainly used by applications as e.g., interactive Web browsing, while background class is meant for e.g., background download of e-mails or background file downloading. Traffic in the interactive class has higher priority than background class traffic. Thus, background applications use transmission resources only when interactive applications do not need them. This is very important in a wireless environment where considerably less bandwidth capacity is available than in core networks. Table 2.3 summarizes the characteristics of the QoS classes for UMTS.

Chapter 3

Modeling Discrete Event Stochastic Systems

This chapter is devoted to the modeling of discrete-event stochastic systems. In particular, deterministic and stochastic Petri nets (DSPN) and UML state diagrams are introduced as modeling languages. DSPNs have been proved to be an appropriate vehicle for quantitative analysis, whereas the analysis of time-enhanced UML diagrams is an emerging area of research [OMG99], [OMG01b]. Section 3.1 and Section 3.2 recapitulate both modeling formalisms and introduces a framework for quantitative analysis. Section 3.3 proposes extensions to UML state diagrams and activity diagrams to allow the specification of stochastic timing. An algorithm for the automated derivation of quantitative results out of an UML diagram is presented. Section 3.4 introduces DSPNexpress-NG, which provides automated tool support for quantitative analysis of both DSPNs and UML diagrams.

3.1 Deterministic and Stochastic Petri Nets

Petri nets were originally introduced by C.A. Petri in 1962. Formally, a Petri net is a directed bipartite graph, i.e., a graph with two disjoint types of nodes: *places* and *transitions*. A directed arc connecting a place (transition) to a transition (place) is called an *input* (res. *output*) *arc* of the transition. A positive integer called *multiplicity* can be associated with each arc. Places connected to a transition by input arcs are called the *input places* of this transition, and those connected by means of output arcs are called its *output places*. Each place may contain zero or more tokens in a *marking*. A marking represents the state of the model at a particular instant. This concept is central to Petri nets. A transition is said to be *enabled*, if all of its input places contain at least as many tokens as the multiplicity of the corresponding input arc. A transition may *fire* when it is enabled, and on firing, a number of tokens equal to the multiplicity of the input arc are removed from each of the input places, and a number of tokens equal to the multiplicity of the output arc are added in each of the output places. The sequencing of firing is an important issue in Petri nets. If two transitions are enabled in a marking, they cannot fire “at the same time”: a choice must be made concerning which one to fire first, the other can only fire after that, if it is still enabled.

The firing of a transition may transform a Petri net from one marking into another. With respect to a given initial marking μ_0, the *reachability set* is defined as the set of all markings reachable through any possible firing sequences of transitions, starting from the initial marking. The evolution of a Petri net can be completely described by its *reachability graph*, in which each marking in the reachability set is a node in the graph, while the arcs describe the possible marking-to-marking transitions. Arcs are labeled with the name of the transition whose firing caused the associated changes in the marking.

Stochastic Petri nets are obtained by associating stochastic and timing information to Petri nets. We do this by attaching *firing time* to each transition, representing the time that must elapse from the instant that the transition is enabled until the instant it actually fires. Throughout this thesis, it is assumed that among all enabled timed transitions in a stochastic Petri net the one with the minimum remaining firing time determines the next marking change. Furthermore, after a marking change each timed transition newly enabled samples a firing time from its firing delay distribution and each timed transition, which has already been enabled in the previous marking and is still enabled in the current marking, keeps its remaining firing time. The sampled firing time of transitions, which are disabled by the marking change, is lost. This stochastic behavior corresponds to the execution policy *race with enabling memory* as defined in [ABB+89]. Another important execution policy is *race with age memory*, where firing times of transitions are not lost if they are disabled by a marking change but resumed in the first marking that enables them again. As previously introduced in Section 1.2.2 similar execution policies are also defined by Telek et al. (see e.g., [TH98]). Nevertheless, a detailed discussion of these policies is out of the scope of this thesis.

In deterministic and stochastic Petri Nets (DSPN, [AC87]) three types of transitions exist: immediate transitions drawn as thin bars fire without delay, exponential transitions drawn as empty bars fire after an exponentially distributed delay whereas deterministic transitions drawn as black bars fire after a constant delay. If both an immediate transition and a timed transition are enabled at the same instant, the immediate transition fires first. If several immediate transitions compete for firing, *firing probabilities*, usually specified as *weights* to be normalized, should be specified to resolve these conflicts. Other extensions in DSPN include inhibitor arcs and transition priorities. *Inhibitor arcs* are drawn with small hollow circles instead of arrows at their terminating end. A transition with an inhibitor arc is enabled if the number of tokens in the input place of the inhibitor arc is less than the multiplicity of the arc. *Transition priorities* are defined by assigning an integer priority level to each transition, which adds the constraint that a transition may be enabled in a marking only if no higher priority transition is enabled. Several further extensions such as marking-dependent firing rates, marking-dependent arc multiplicities, making-dependent transition weights or infinite-server firing semantics are defined in the literature (see e.g. [Lin98]). These extensions allow a more compact representation of several system features in a DSPN, although they do not extend their modeling power. Note that the formalism introduced for

DSPN include the definition of a generalized stochastic Petri net (GSPN, [ABC84]), in which the firing delays of timed transitions are only allowed to be exponentially distributed.

A marking of a DSPN is called *vanishing* if at least one immediate transition is enabled in the marking and *tangible* otherwise. Furthermore, the *extended reachability graph* is distinguished from the *tangible reachability graph*. The former comprises all markings of the reachability set, i.e., tangible and vanishing markings, whereas the latter comprises only the tangible markings. It is assumed that the tangible reachability graph of the DSPN comprises a finite number of tangible markings. Numerical analysis of DSPNs proceeds by computing transient or stationary distributions for its underlying continuous-time stochastic process $\{S(t): t \geq 0\}$. The process $\{S(t): t \geq 0\}$ has a discrete state space (i.e., the tangible markings of the DSPN) and is denoted as the *marking process* of the DSPN. The analysis procedure can be decomposed into four steps:

(1) Generating the extended reachability graph with the markings of the reachability set as nodes and some stochastic information attached to the arcs. Thus, all markings are related to each other with stochastic information.

(2) Eliminating the vanishing markings and the corresponding transitions from the extended reachability graph. The resulting graph only contains the tangible markings, i.e., is the tangible reachability graph. This task can be performed with a well known algorithms proposed by Balbo et al. [BCF+87].

(3) Determine the steady-state or transient *marking probabilities* from numerical analysis of the underlying marking process.

(4) Combining the marking probabilities to quantitative performance measures, such as mean queue length, loss probabilities or throughput values.

Without any restrictions on the usage of deterministic transitions, the marking process of a DSPN can be represented as a generalized semi-Markov process (GSMP) with exponential and deterministic events (an exact definition of a GSMP is provided in Section 4.2). A GSMP describes the evolution of the stochastic behavior of a discrete-event stochastic system over time. Although a GSMP constitutes a very general stochastic process, a rich body of theoretical results on monotonicity, regeneration, and continuity is available [She93], [Whi80]. In general, the analysis of a GSMP can be performed by discrete-event simulation only. For finite-state GSMPs with exponential and deterministic events, a cost-effective numerical method is introduced in Chapter 4.

As an example of a DSPN an MMPP/D/2/K queueing system with first-come first-served (FCFS) scheduling discipline is considered. Customers arrive according to a two-state Markov-modulated Poisson process (MMPP, [FM93]). Opposed to an ordinary Poisson process, the MMPP can capture important correlations between the interarrival times if the

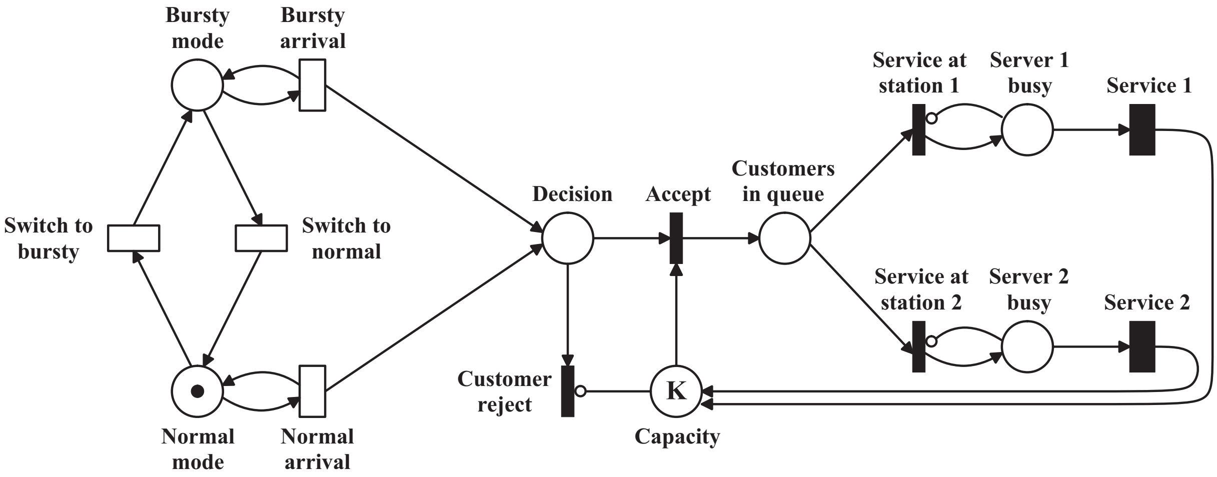


Figure 3.1. DSPN of the MMPP/D/2/K queue

parameters of the MMPP are determined by appropriate fitting procedures to match a given traffic trace [Ryd96]. The MMPP has been extensively employed for modeling traffic processes with time-varying arrival rate, i.e. bursty traffic. Furthermore, it is shown to be a quite accurate model for Internet traffic, which usually shows self-similarity among different time scales [KLL03]. The two states of the MMPP represent bursty mode and normal mode, i.e., non-bursty mode, of the arrival process. They are represented by the places *Bursty mode* and *Normal mode*, respectively. On firing of transition *Switch to bursty* (*Switch to normal*) the MMPP changes its state from normal mode to bursty mode (bursty mode to normal mode). Customer arrivals in bursty mode are represented with transition *Bursty arrival* and the arrival of a customer in normal mode is modeled with transition *Normal arrival*. The K tokens residing in place *Capacity* in the initial marking represent the finite number of buffers of the queueing system. Tokens contained in the places *Customers in queue* represent customers waiting in the queue. Tokens contained in the places *Server 1 busy* and *Server 2 busy* represent customers currently being served. The constant service requirements are modeled by the deterministic transitions *Service 1* and *Service 2*. If the immediate transitions *Service at station 1* and *Service at station 2* both have associated firing weights one, then arriving customers to an empty system join each server with equal probability. The number of tangible markings of this DSPN is given by 2·(K+2).

3.2 Unified Modeling Language

The UML [BJR99], [OMG01a] provides different views of a model that are represented by graphical diagrams. These diagrams include *use case diagrams*, *class diagrams*, *behavior diagrams*, and *implementation diagrams*. Behavior diagrams include *state diagrams*, *activity diagrams*, and interaction diagrams like *sequence diagrams* and *collaboration diagrams*. It is the freedom of the designer to choose the types of UML diagrams best suited for the intended representation of a system. In this thesis, state diagrams and activity diagrams are chosen as building blocks for enabling quantitative system analysis with the UML. The diagrams are presented using the notation from [Dou99].

3.2.1 State Diagrams and Activity Diagrams

State diagrams of the UML provide a simple but formal means of modeling the complex event-driven system behavior. All semantics necessary to express behavior (i.e., states, historical properties, transitions, and compound transitional connectors) are available on the state diagram palette. The semantics and notation of state diagrams, also known as statecharts, are substantially those of Harel's statecharts (see e.g., [Har87]) with modifications to make them object-oriented. A state diagram represents a state machine; a state being a condition during the life of an object or an interaction during which it satisfies some condition, performs some action, or waits for some event.

Figure 3.2 provides an example of the state diagram notations considered in this thesis. *States* are shown as named rectangles with rounded corners and represent a possible situation for an object. The initial state of an object is labeled with a *default connector* that is represented by a short arc starting from a small filled circle. States can be ordered hierarchically and/or concurrently. Concurrent states are separated by dashed lines. These states are also called *and-states* because they are independent states that may be concurrently active with other states. In Figure 3.2 the states *B3* and *B4* are and-states. The hierarchy of states is represented by the nesting of states within states. The outer enclosing state is called *superstate* and the inner states are called *substates*. In Figure 3.2 state *A* is a superstate of states *B1*, *B2*, *B3*, and *B4*. Furthermore, states *C1* and *C2* are substates of state *B2*, states *C3*

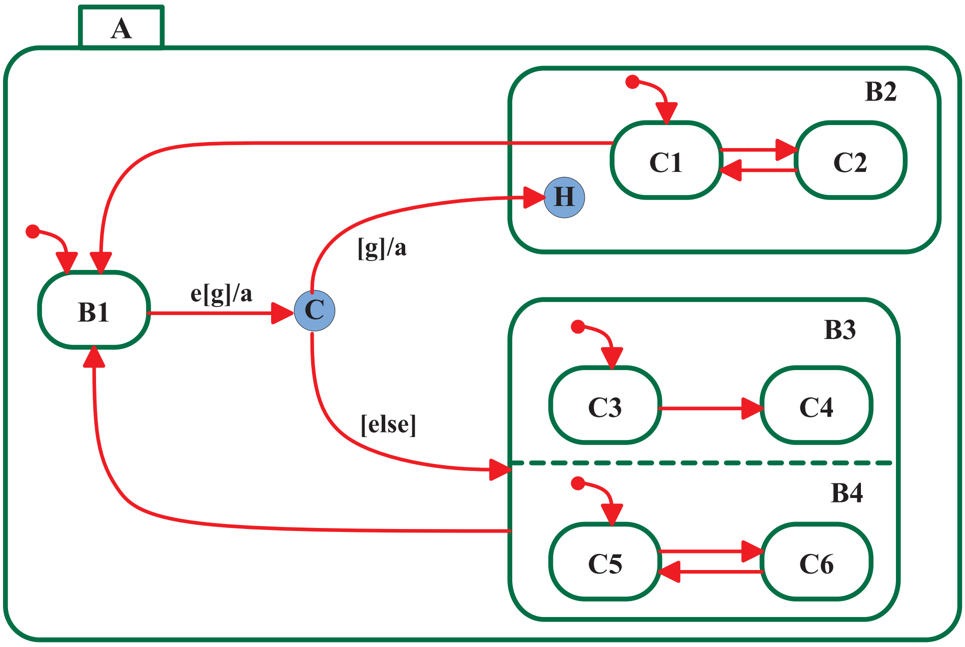


Figure 3.2. State diagram notation

and *C4* are substates of state *B3*, and states *C5* and *C6* are substates of state *B4*. The substate that is visited by entering the corresponding superstate is labeled with the default connector, e.g., states *B1*, *C1*, *C3*, and *C5* in Figure 3.2. Furthermore, *pseudostates* are defined in UML state diagrams. The *conditional pseudostate* allows one of a set of branching transitions to be selected based on some guarding condition. A conditional pseudostate is indicated by a circled C (see Figure 3.2). The *history pseudostate* indicates that when entering a superstate the initial or default state is that last active substate of the superstate (see state *B2* in Figure 3.2). The UML identifies two kinds of history - shallow and deep. Shallow history means that the last active substate is the active default, but if that substate is further decomposed into sub-substates, no knowledge is retained of that nested history. Deep history means that history is remembered to all levels of nesting. Shallow history is indicated by a circled H and deep history by a circled H*.

Transitions are shown as directed arcs between states. Transitions are labeled with a *trigger event*, a *guard*, and one or several *actions*. A proper definition of a transition must contain either one of these three building blocks. The two others are optional. Actions are considered to be processes that occur quickly and are not interruptible. A guard is a logical condition that will return only “true” or “false”. If the trigger event occurs and the guard resolves to “true” the actions are executed and the corresponding state change is performed. Thus, arcs of a transition are labeled with the following notation:

```
event[guard]/action1;action2;action3...
```

In Figure 3.2, the arc starting from state *B1* is labeled with event `e`, guard `g`, and action `a`. The arc that is labeled with `[g]/a` and which starts from the conditional pseudostate is chosen if guard `g` resolves to “true”. If so, action `a` is executed. Otherwise, the arc labeled with `[else]` is taken and no action is executed.

Activity diagrams of the UML are used to model sequence and parallelism of activities. An activity diagram is a special case of a state diagram in which all (or at least most) of the states are action states (i.e., activities) and in which all (or at least most) of the transitions are triggered by completion of the actions in the source states. The purpose of this diagram is to focus on the flows driven by internal processing (as opposed to external events). Activity diagrams are used in situations where all or most of the events represent the completion of internally generated actions (that is a procedural flow of control). State diagrams, on the other hand, are used in situations where asynchronous events predominate. An example of a state diagram is presented in Section 3.3.

3.2.2 Framework for Quantitative Analysis of UML Diagrams

In order to accompany the design process of software and hardware systems with performance evaluation in different design stages, a framework for quantitative analysis of

UML diagrams is needed. Figure 3.3 presents the proposed framework for deriving performance measures for UML diagrams by analysis of their underlying stochastic process. As illustrated, the derivation of performance measures is divided into four main steps. In the first step, a state diagram or activity diagram has to be specified with an UML design tool like Rhapsody™ [Rhap] or Together™ [Toge]. With these tools output files can be generated with a textual representation of the UML diagram (e.g., the "`<ModelName>.dfState`"-file of Together™). For timed events an extra specification file is needed to include the expected waiting delays and information about the delay distribution, i.e., deterministic or exponentially distributed delay.

After adding timing specifications to the UML diagram, the derivation of the state transition graph has to be performed; see step (2) in Figure 3.3. Therefore, the state space of the UML diagram has to be explored as described in Section 3.3. Note that the state transition graph is a formal representation of the discrete-event stochastic system specified in the UML or even other specification languages like deterministic and stochastic Petri nets as previously introduced. Its generation is a prerequisite for the quantitative analysis of the underlying GSMP. The core of the quantitative evaluation process constitutes the numerical solution of the underlying GSMP; see step (3) in Figure 3.3. For a detailed mathematical treatment of the different stochastic processes underlying discrete-event stochastic systems see [Lin98]. Alternatively, the transient or steady-state solution of the GSMP can be derived by discrete-event simulation e.g., as provided by the commercial simulation library CSIM [CSIM]. The results are stored in a data structure comprising state probabilities for the state transition graph.

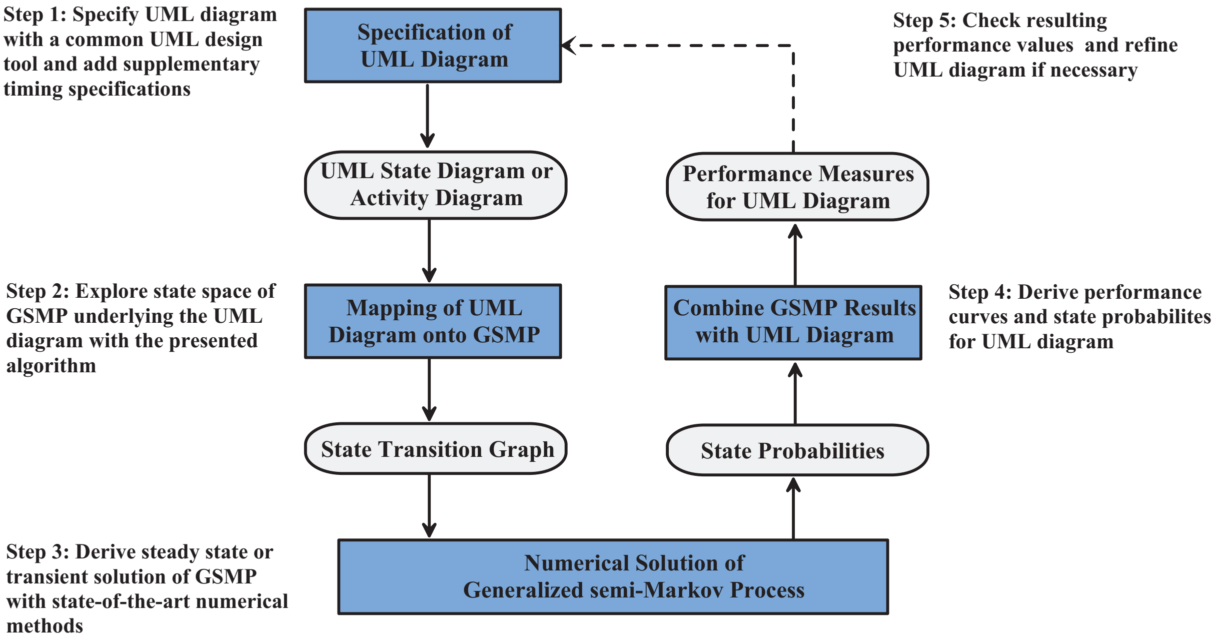

Step 1: Specify UML diagram with a common UML design tool and add supplementary timing specifications
Specification of UML Diagram
Step 5: Check resulting performance values and refine UML diagram if necessary
UML State Diagram or Activity Diagram
Performance Measures for UML Diagram
Step 2: Explore state space of GSMP underlying the UML diagram with the presented algorithm
Mapping of UML Diagram onto GSMP
Combine GSMP Results with UML Diagram
Step 4: Derive performance curves and state probabilites for UML diagram
State Transition Graph
State Probabilities
Step 3: Derive steady state or transient solution of GSMP with state-of-the-art numerical methods
Numerical Solution of Generalized semi-Markov Process

Figure 3.3. Derivation of quantitative performance measures out of UML diagrams

The main task of step (4) constitutes the computation of performance measures like throughput, loss probabilities, mean response time of a resource etc. given the previously computed state probabilities of the underlying GSMP. Therefore, the probabilities have to be combined to obtain performance measures of interest or just the probability of being in a particular state of the state diagram. One approach to experimental design is to derive performance curves for a certain performance measure by varying the value of one delay parameter of an UML diagram while the other parameters are kept fixed. These performance curves give system engineers significant insight into the system dynamics before implementing the system. The framework allows system engineers to check system performance in early design stages and to refine their model if necessary as shown in step (5) in Figure 3.3. As described in Section 3.3 the system performance can be evaluated at several steps in the development lifecycle using the concept of nesting substates into superstates available at the states diagrams palette. Therefore, it is possible to obtain rough quantitative values in early design stages as well as more accurate performance indices in a final design phase.

3.3 Mapping of Time-enhanced UML Diagrams onto Stochastic Processes

In order to introduce timing in a UML state diagram or activity diagram, trigger events are associated with a certain delay, i.e., a *timed event* occurs after an activity with a given (stochastic) delay. Throughout this thesis these delays are assumed to be deterministic or exponentially distributed although other delay distributions are possible from the modeling point of view. Nevertheless, in Chapter 4 an algorithm for the analysis of DES with exponential and deterministic events is developed and therefore only this restricted case is considered here. State transitions that are triggered by timed events are called *timed transitions*. Transitions that occur immediately, i.e., only a guard must be evaluated and/or an action executed are called *immediate transitions*. To represent timed events in a state diagram or activity diagram, new syntactical expressions that can be directly derived from the performance value definition *PAperfValue* introduced in [OMG01b] are defined. The expression `EXP_<id>` defines an event that triggers a state transition after an exponentially distributed delay characterized by the identifier `<id>`. The expression `DET_<id>` defines an event that triggers a state transition after a deterministic delay. The corresponding identifier `<id>` must be further specified in a supplementary file. For each identifier this file contains the type of the event and its firing rate, that is λ in the case of an exponential distribution with parameter λ. Furthermore, randomness is introduced in the sequential flow of activities. That is, weights $w_1, ..., w_n$ can be associated with immediate transitions $t_1, ..., t_n$. If more than one guard of the transitions resolves to “true” (i.e., more than one transition is enabled) the weights of these transitions are used to compute *branching probabilities* by the following normalization formula:

$$p(t_i) = \frac{w_i}{\sum_{j:t_j \text{ enabled}} w_j} \qquad (3.1)$$

With this definition the next state is chosen from a discrete probability distribution among the enabled transitions similar to the firing probabilities previously defined for DSPN. The optional expression `IMM_<id>` explicitly represents an immediate transition whose weights must be specified in the supplementary file. *Transition priorities* are defined in the same way as for DSPNs, i.e., a transition is enabled only if no higher priority transition is enabled. Per definition, timed transitions have priority zero and immediate transitions have priority of at least one. Priorities of immediate transitions can be defined in the supplementary specification file.

The syntactical representation of a timed transition is the same as for a transition in a standard UML state diagram without time-enhancement. However, the semantics is somewhat different: If the guard resolves to “true” the timed event is scheduled and an associated clock that indicates the time until the event occurs is set according to its clock setting distribution. Unless the guard changes its value from “true” to “false” until the time the event occurs, the actions are executed and the corresponding state change is performed. Otherwise the event is canceled and the timed transition gets disabled. Note that this semantics corresponds to the behavior of a GSMP as defined in Section 4.2.

To provide an example, Figure 3.4 shows the state diagram of a queueing system with a customer arrival process that is controlled by a Markov-modulated Poisson process as

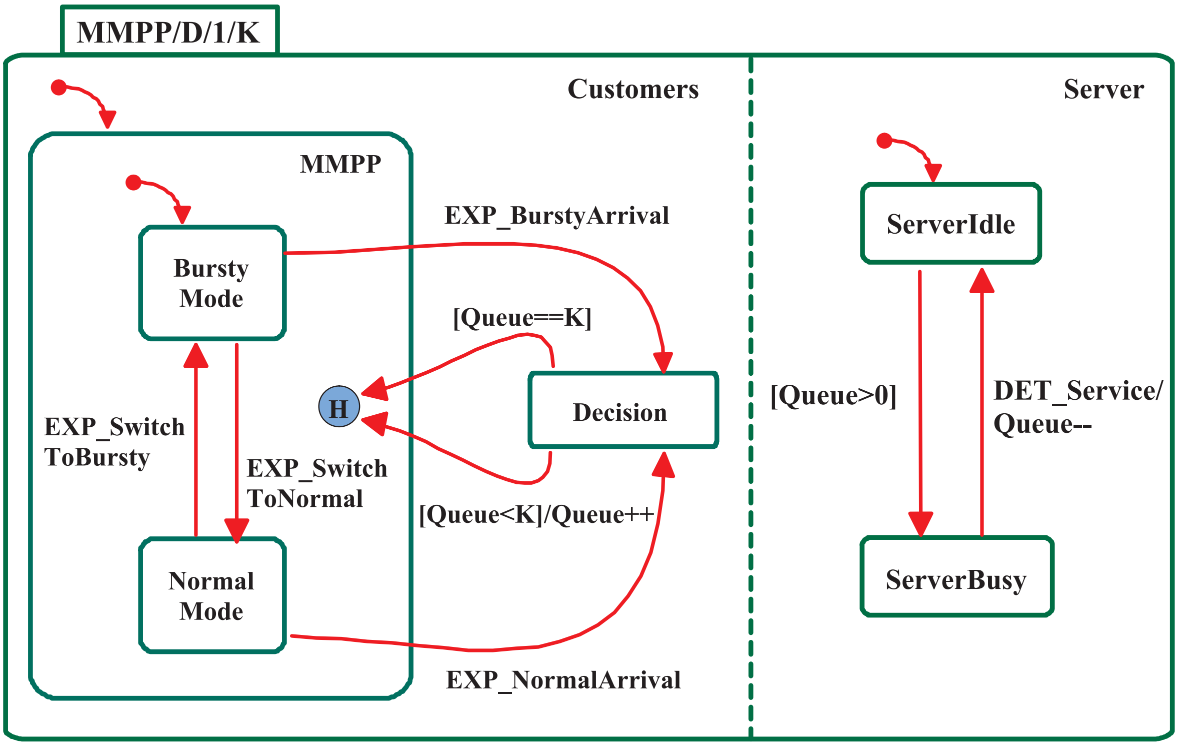


Figure 3.4. UML state diagram of an MMPP/D/1/K queue

previously introduced in Section 3.1. For ease of graphical representation, only a single-server queueing system with deterministic service time and finite buffer of size K, i.e., an MMPP/D/1/K queue, is considered. The scheduling discipline is assumed to be first-come first-served. However, other scheduling disciplines such as last-come first-served or processor sharing can be represented by the same model; in fact, quantitative measures such as throughput or loss probability are the same. In particular, the model is represented by two concurrent superstates *Customers* and *Server*. These superstates communicate with each other through the variable *Queue*, which contains the current number of customers in the system. The superstate *Customers* contains substates *Decision* and *MMPP*. The MMPP is parameterized by two states representing bursty mode and normal mode of customer arrivals. Thus, superstate *MMPP* contains substates *BurstyMode* and *NormalMode*. Since the arrival process, i.e., the MMPP, continuous in the same state after a customer arrival, a history pseudostate is used to store the previous state of the MMPP. The state *Server* is represented by the substates *ServerIdle* and *ServerBusy*. Note that state *Decision* could have also been modeled by a conditional pseudostate, since only a guard is evaluated and possibly an action is executed. Exponential events *BurstyArrival* and *NormalArrival* correspond to the arrival process of customers in bursty mode and normal mode. Exponential events *SwitchToBursty* and *SwitchToNormal* correspond to state changes of the MMPP. The service time is modeled with deterministic event *Service*.

We view a state diagram or an activity diagram of the UML as a discrete-state, event-driven system. That is, its state evolution depends entirely on the occurrence of asynchronous discrete events over time. For ease of exposition the methodology is described for quantitative analysis of state diagrams only. Activity diagrams can be treated as a special case of state diagrams. The key idea of the state space exploration is to map a “configuration” of the UML state diagram onto an appropriate state of the underlying stochastic process. Furthermore, a transition in the state diagram is mapped onto a state change of the underlying process. Therefore the stochastic process can be completely represented by the *state transition graph*, a directed graph with labeled arcs.

In the following, the derivation of the state transition graph via the exploration of the transition system is described. The *transition system* consists of all possible configurations of the corresponding UML diagram. A *configuration* of a state diagram is a snapshot of its execution. One can view a configuration as the information that is needed to completely restore the “state” of the system. A configuration consists of the active substates of all concurrent states of the system, the history information and the setting of all variables. Let $s_1, ..., s_n$ be the states of the state diagram, $h_1, ..., h_r$ the history pseudostates, and $v_1, ..., v_m$ the variables corresponding to the state diagram. Formally, a configuration C is represented by a tuple (S,H,V) comprising a set S that contains all active states (i.e., one substate of each concurrent state), a mapping H of each history pseudostate h_j onto a set H_j comprising the stored history states, and a mapping V of each variable onto an appropriate value. In the case of shallow history the sets H_j contain each only one state, namely the substate that has to be restored when entering the history pseudostate h_j. In case of deep history further substates have to be stored if even one substate is a concurrent state.

The configurations of the transition system are connected by directed arcs that represent a change of a configuration in the state diagram. Two causes of a change of a configuration in the transition system are distinguished. First, a change of a configuration can be triggered by a timed transition with events that have exponentially distributed or deterministic delays. Furthermore, a change of a configuration can occur without delay. That is due to the evaluation of a guard or the execution of an action. As in generalized stochastic Petri nets, immediate transitions are defined to have priority over timed transitions. That is if the guard of an immediate transition t_i is “true”, this transition triggers the configuration change with (branching) probability $p(t_i)$ (see Eq. (3.1)).

Similar to stochastic Petri nets, configurations in the transition system can be classified as *tangible* or *vanishing* configurations. A tangible configuration is a configuration in which only timed transitions are enabled. A vanishing configuration is a configuration in which one or more immediate transitions are enabled. For the quantitative analysis of UML state diagrams, only tangible configurations of the underlying transition system need to be considered. This is because the time being in a vanishing configuration is equal to zero, since immediate transitions occur without delay. Thus, only the tangible configurations constitute states of the stochastic process for which the quantitative analysis is performed. All vanishing configurations in a transition system have to be removed to obtain the state transition graph that comprises only tangible configurations and directed arcs between certain tangible configurations of the transition system.

Figure 3.5 presents a pseudo-code algorithm for the generation of the state transition graph. The initial tangible configuration can be calculated with a top-down activation of the default states of the state diagram beginning with the top-level state. Starting with the initial tangible configuration the algorithm directly derives the state transition graph without explicitly generating the transition system. The key procedure of the algorithm is performed in step (9) and (10). The configuration c_{succ} in step (9) is derived using the history information H stored in configuration $c = (S,H,V)$ and the execution of the actions of transition t that may effect the variable setting V. Note that configuration c_{succ} may be a vanishing configuration that should not be inserted in the state transition graph. Therefore, the set of tangible configurations that can be reached through c_{succ} has to be determined recursively with graph analysis methods based on a depth-first-search starting at c_{succ} (see step (10) of Figure 3.5). Note that an effective method for this task can be borrowed from reachability analysis for generalized stochastic Petri nets [ABC+95].

```
Import state diagram from UML design tool and add timing specifications
Derive initial tangible configuration $c_{init}$
Label $c_{init}$ as NEW and insert $c_{init}$ into the empty transition graph G
**FOR EACH** configuration c = (S,H,V) labeled as NEW in G **DO**
    Label c as VISITED
    **FOR EACH** state $s \in S$ **DO**
        **FOR EACH** transition t originating from state s **DO**
            **IF** guard of t resolves to TRUE in configuration c **THEN DO**
                Derive configuration $c_{succ}$ reached through transition t
                Derive set of tangible configurations C reachable from $c_{succ}$
                **FOR EACH** configuration $c_{tang}$ in C **DO**
                    Label $c_{tang}$ as NEW and insert $c_{tang}$ into G
                    Insert arc $c \rightarrow c_{tang}$ labeled with delay distribution and branching probability into G
                **OD**
            **OD**
        **OD**
    **OD**
**OD**
```

Figure 3.5. Algorithm for generation of state transition graph

Besides the state space exploration and the effective removal of vanishing configurations, quantitative analysis of UML specifications requires a mapping of the state transition graph onto an appropriate stochastic process for which simulation-based and/or analytical numerical analysis methods are known. This mapping is the key for the quantitative analysis of UML state diagrams and activity diagrams. In fact, every tangible configuration of the state diagram maps onto a state of the corresponding state space of the underlying stochastic process and every configuration change in the state transition graph corresponds to a state change in the stochastic process. The most general form of the stochastic process underlying a state diagram is the generalized semi-Markov process. Thus, the well-defined derivation of the state transition graph of the GSMP is the key for quantitative analysis of UML system specifications.

Considering the example presented in Figure 3.4, each configuration of the MMPP/D/1/K queue can be represented by a tupel comprising the active states, the history information, and the value of the variable *Queue* (i.e., 0,1,...,K customers in the system). The set of states S of a configuration contains one substate of each of the concurrent superstates *Customers* and *Server*, i.e., *NormalMode (N)*, *BurstyMode (B)*, or *Decision (D)*, of superstate *Customers* and *ServerIdle (I)* or *ServerBusy (B)* of superstate *Server*. The history information is simply N or B corresponding to the *NormalMode* and *BurstyMode* substate, respectively. The transition system of the MMPP/D/1/K queue is presented in Figure 3.6. Tangible configurations are drawn as white circles and are associated with state numbers s_{00}, s_{01}, .., s_{K1}. Vanishing configurations are drawn as dashed circles. Recall that directed arcs between configurations

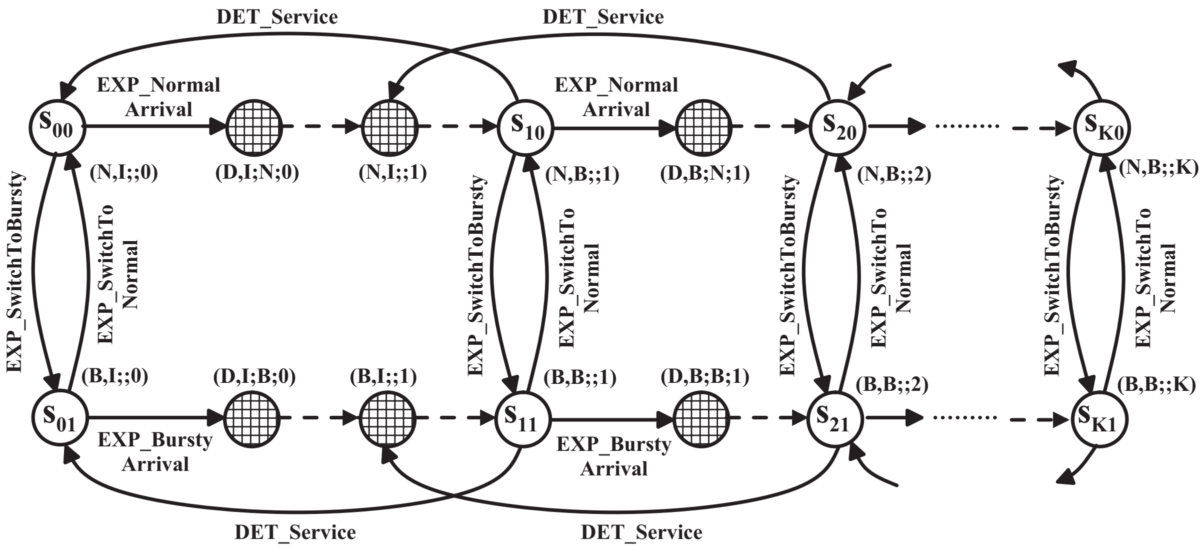


Figure 3.6. Transition system of the UML state diagram of an MMPP/D/1/K queue

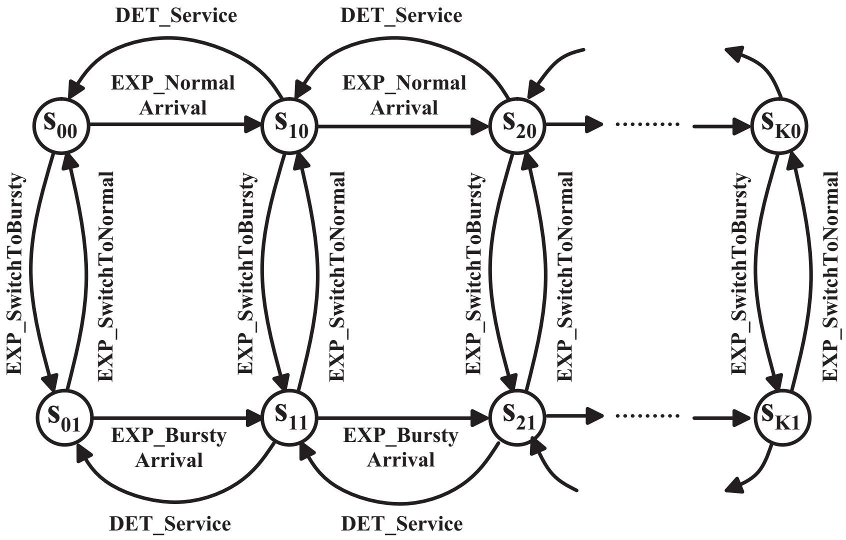


Figure 3.7. State transition graph of the GSMP underlying the UML state diagram of an MMPP/D/1/K queue

represent feasible state transitions. Immediate state transitions are drawn with dashed arcs and timed transitions are shown as continuous arcs. Below a state number the corresponding configuration is written in the following notation:

(customers substate, server substate; history state; queue length)

Note that the transition system is not derived explicitly in the algorithm presented in Figure 3.5. Instead the state transition graph is derived directly by ignoring the vanishing configurations when exploring the state space (step (10) of Figure 3.5). The state transition graph of the GSMP corresponding to the MMPP/D/1/K queue is shown in Figure 3.7. Putting it all together, the state transition graph of the GSMP consists of 2·(K+1) tangible configurations representing 0,1,2,...,K customers in the system with the MMPP residing in normal or bursty mode. That is, configurations s_{i0} correspond to i customers in the system with the MMPP residing in normal mode and configurations s_{i1} correspond to i customers in the system with the MMPP residing in bursty mode.

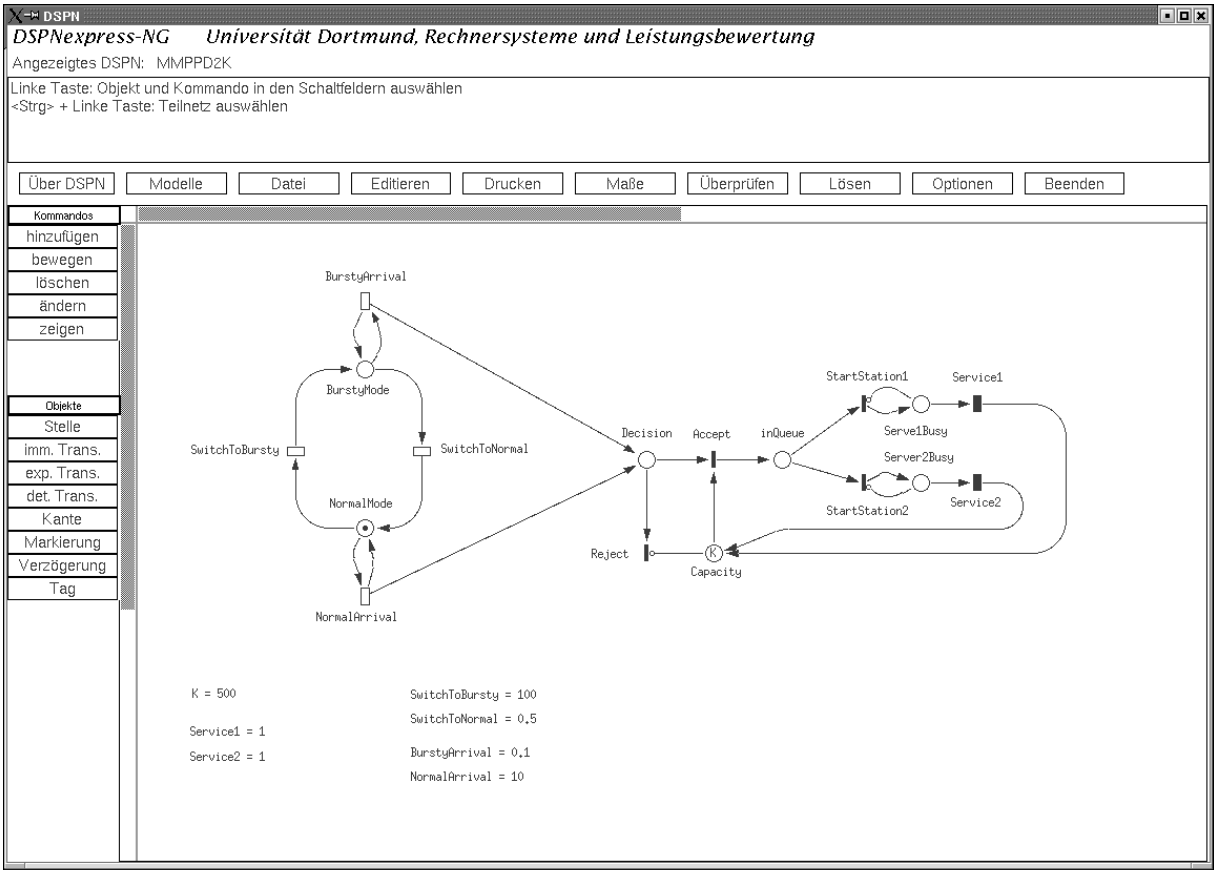
It should be pointed out that the framework for generating a performance model, i.e., the state transition graph, and subsequently deriving performance curves out of a UML state diagram or activity diagram can be applied in several steps in the system or software development lifecycle. In order to get insight in the quantitative system dynamics in early design stages a hierarchical modeling of the system with state diagrams is suggested. That is, in a first step the system designer has to identify the “main” states of the system. That can be done for example by using a set of sequence diagrams derived from different use cases. Furthermore, a rough design of the system reduces the state space size significantly and therefore performance values can be obtained very quickly. In later design stages the main states should be modeled in more detail using the concept of nesting substates into superstates. Thus, it is possible to obtain rough performance values in early design stages as well as more accurate performance indices in a subsequent design phase.

3.4 Automated Quantitative Analysis with DSPNexpress-NG

In order to automate the performance evaluation process the described methodology is implemented in the new release of DSPNexpress called *DSPNexpress-Next-Generation (DSPNexpress-NG)* [DSPN]. The previous version of DSPNexpress, DSPNexpress 1.5, is known for its highly efficient numerical method for steady-state analysis of deterministic and stochastic Petri nets without concurrent deterministic transitions [Lin95b], [Lin98]. Furthermore, DSPNexpress 1.5 contained already a graphical user interface running under X11 allowing easy model specification, modification, graphical animation, as well as automate quantitative analysis.

The main goal of DSPNexpress-NG constitutes the availability of an open interface for utilizing the numerical solvers for GSMPs for the quantitative evaluation of systems specified in modeling formalisms other than just deterministic and stochastic Petri nets (DSPNs). In fact, the main research contribution of DSPNexpress-NG constitute the implementation of an efficient numerical method for transient and steady-state analysis of GSMPs with exponential and deterministic events, which is further described in Chapter 4. Open-source software download of DSPNexpress-NG is provided on the Web so that researchers are free to develop new interfaces to other modeling formalisms or include their own numerical or simulation based analysis methods [DSPN].

Currently, DSPNexpress-NG can perform quantitative analysis for DSPNs and UML specifications, i.e., state diagrams and activity diagrams. DSPNs can be edited directly with the Petri net editor of DSPNexpress-NG. To illustrate the features of this graphical interface,



DSPNexpress-NG Universität Dortmund, Rechnersysteme und Leistungsbewertung

Angezeigtes DSPN: MMPPD2K

Linke Taste: Objekt und Kommando in den Schaltfeldern auswählen
<Strg> + Linke Taste: Teilnetz auswählen

K = 500

Service1 = 1

Service2 = 1

SwitchToBursty = 100

SwitchToNormal = 0.5

BurstyArrival = 0.1

NormalArrival = 10

Figure 3.8. The graphical user interface of DSPNexpress-NG

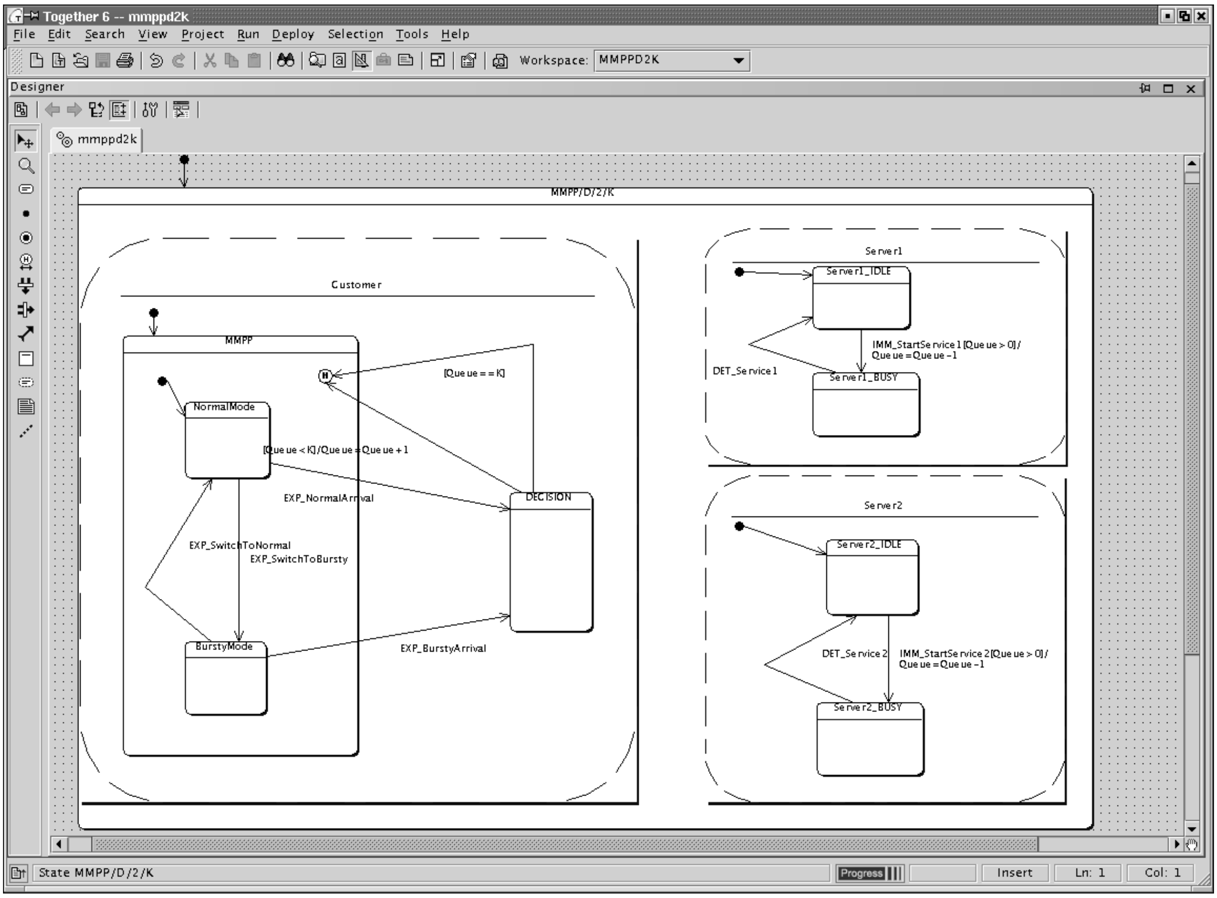


Figure 3.9. Modeling the MMPP/D/2/K queue with Together™

consider the snapshot shown in Figure 3.8. A DSPN of an MMPP/D/2/K queue as introduced in Section 3.1 is displayed. At any time, DSPNexpress-NG provides online help messages displayed in the third line of the interface. The command line and the object line are located on the left side of the interface. The buttons are located in a vertical line between the online help line and the working area. The working area constitutes the remaining big rectangle, which contains the graphical representation of the DSPN. A detailed description of the features of the graphical interface is given in Chapter 10 of [Lin98].

In order to utilize the numerical solvers of DSPNexpress-NG for quantitative analysis of UML state diagrams and activity diagrams, these diagrams must be imported through the *UML2DES Converter*, provided by DSPNexpress-NG. The task of the UML2DES Converter is to derive the state transition graph out of the UML diagram according to the algorithm presented in Figure 3.5. To perform this task, a supplementary file containing timing specifications for associating exponentially distributed or deterministic delays with events is required by the UML2DES Converter. DSPNexpress-NG allows the user-friendly specification of performance studies (i.e., what/if studies). For DSPNs performance measures of interest can be defined in the graphical interface. Performance measures corresponding to UML diagrams can be defined in the supplementary specification file. A *report generator* is responsible to evaluate the specified performance measures by combining the steady-state or transient state probabilities.

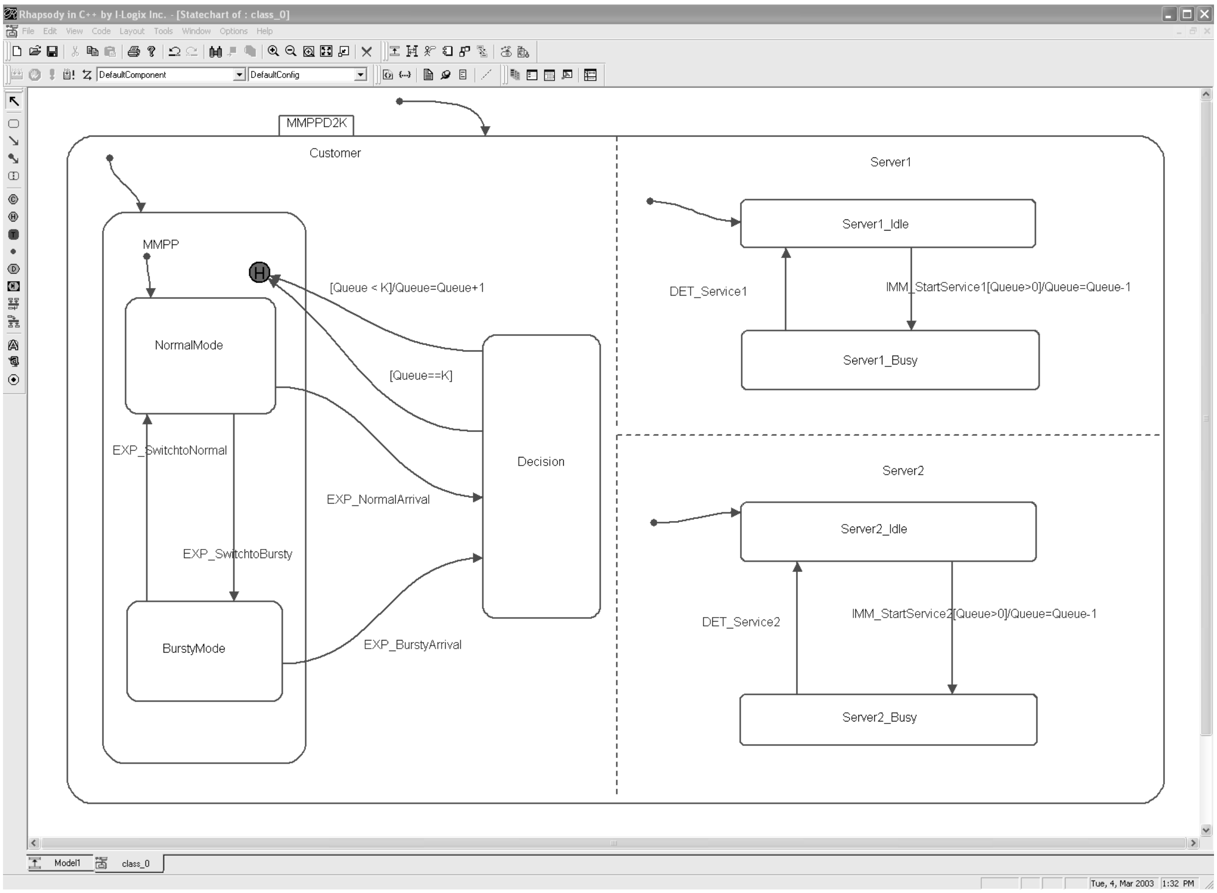


Figure 3.10 Modeling the MMPP/D/2/K queue with Rhapsody™

Figures 3.9 and 3.10 show a snapshot of Together™ version 6.0.1 and Rhapsody™ version 4.0.1 with an UML state diagram of the MMPP/D/2/K queue in the working area. The corresponding timing specification file is presented in Figure 3.11. Besides the definition of average rates for exponential and deterministic events, weights and priorities for immediate transitions *StartService1* and *StartService2* are considered. Since both immediate transitions have the same associated firing weight and the same priority, an arriving customer to an empty system joins each server with equal branching probability 1/2. Note that similar to the definition of marking-dependent firing rates and marking-dependent transition weights in a DSPN, the definition of firing rates and transition weights that depend on a particular state or a particular setting of a variable are supported. The timing specification file presented in Figure 3.11 additionally contains the definition of several performance measures such as mean queue length, customers reject probability, or queuing delay. Similar to DSPN models, the building blocks for defining a performance measure are mean value of a variable (i.e., `E{<variable>}`), throughput of a timed transition (i.e., `X{<transition>}`), the probability of being in a particular state (i.e., `P{<state>}`), or the probability of a particular variable setting. Arithmetic operations (i.e., “+”, “-”, “*”, “/”, and `MIN{·}`) and logical combinations (i.e., `AND` and `OR`) are also supported in the definition of performance measures for UML state diagrams.

```
#----------------------------------------------------------------------------------------
#Specification of variables and constants
#TYPE         IDENTIFIER                   VALUE
#----------------------------------------------------------------------------------------
VAR           Queue                        0;
CONST         K                            500;

#----------------------------------------------------------------------------------------
#Specification of exponential and deterministic events
#TYPE         IDENTIFIER                   RATE
#----------------------------------------------------------------------------------------
EXP           NormalArrival                0.1;
EXP           BurstyArrival                10;
EXP           SwitchToNormal               2.0;
EXP           SwitchToBursty               0.01;
DET           Service1                     1.0;
DET           Service2                     1.0;

#----------------------------------------------------------------------------------------
#Specification of immediate transitions
#TYPE         IDENTIFIER                   WEIGHT          PRIORITY
#----------------------------------------------------------------------------------------
IMM           StartService1                1;              1;
IMM           StartService2                1;              1;

#----------------------------------------------------------------------------------------
#Specification of performance measures
#TYPE         NAME                         DEFINITION
#----------------------------------------------------------------------------------------
MEASURE       MeanQueueLength              E{Queue};
MEASURE       AverageArrivalRate           X{BurstyArrival} + X{NormalArrival};
MEASURE       EffectiveArrivalRate         BurstyArrival*P{BurstyMode AND Queue < K}
                                            + NormalArrival*P{NormalMode AND Queue < K};
MEASURE       CustomerRejectProbability    1 - EffectiveArrivalRate/AverageArrivalRate;
MEASURE       SystemFullProbability        P{Queue == K};
MEASURE       QueueingDelay                MeanQueueLength/EffectiveArrivalRate;
```

Figure 3.11. Timing specification file for UML state diagram of MMPP/D/2/K queue

3.5 Summary

In this chapter, extensions to UML state diagrams and activity diagrams to allow the association of events with exponentially distributed and deterministic delays are proposed. A particular stochastic process, the generalized semi-Markov process, is identified as the appropriate vehicle on which quantitative analysis is performed. Furthermore, a framework for the automated quantitative analysis of UML diagrams enhanced with deterministic and stochastic delays is introduced. The main contribution of this chapter is the development of an algorithm for the automated derivation of the state space underlying a UML state diagram or activity diagram that additionally deals with history pseudostates, configuration-dependent firing rates of exponential events, and configuration-dependent branching probabilities of immediate transitions. Automated tool support for quantitative analysis of DSPNs and UML diagrams is provided by DSPNexpress-NG [DSPN].

Chapter 4

Analysis of Discrete Event Stochastic Systems

This chapter considers the numerical quantitative analysis of discrete-event stochastic systems. In particular, a new numerical method for transient and stationary analysis of generalized semi-Markov processes with exponential and possibly concurrent deterministic clock setting distributions is introduced. The presented methodology can be utilized to derive quantitative performance measures for a model specified in Petri net notation, as UML state diagram or activity diagram, or even other specification languages for discrete-event stochastic systems. To make the text self-content, the basic definition of a Markov process and the analysis of continuous-time Markov chains are briefly recapitulated in Section 4.1. For a rigorous presentation of Markov processes and their numerical analysis we refer to the standard literature, e.g., [Cas93], [Nel95], or [Tri02]. Section 4.2 introduces the notion of a generalized semi-Markov process. Section 4.3 considers the generation of the transition kernel of the embedded general state space Markov chain (GSSMC) and presents two theorems on properties of the GSSMC, which constitute the main results of this chapter. Section 4.4 introduces the system of integral equations for transient and stationary analysis of the GSSMC. In Section 4.5, the algorithmic generation of the transition kernel and the exploitation of constant kernel elements as well as the numerical solution of the system of integral equations are illustrated. To show the practical applicability of the methodological results, Section 4.6 considers an MMPP/D/2/K queue and an UML model representing a single cell in a cellular network as examples.

4.1 Continuous-Time Markov Chains

The previous chapter ends with the mapping of a model specification onto the underlying stochastic process. Thus, we start by defining a stochastic process.

Definition 4.1 (Stochastic process): A *stochastic process* is a family of random variables $\{X(t): t \in T\}$ defined over a given probability space, indexed with parameter t, where t varies over the index set T.

It is typical to think of t as time and T as a set of points in time. The values of the random variables X(t) are called *states* and the set of all possible states defines the *state space* S. If the state space of a stochastic process is discrete, i.e., a finite or countable infinite set, then it is called a discrete-state process, often referred to as a “chain”. In this case, the state space is assumed to be S = {s_1, s_2, ...} or, when there is no ambiguity, states are simply referred by their index, i.e., S = {1, 2, ...}. Alternatively, if the state space is continuous, then we have a continuous-state process. Similarly, if the index set T is discrete, then we have a discrete-time process; otherwise we have a continuous-time process. In general, the random variables {X(t)} of a stochastic process can depend on each other. A special class of stochastic processes are Markov processes, which allow a limited amount of dependence between the random variables.

Definition 4.2 (Markov process): A stochastic process {X(t): t $\in$ T} with state space S is called a *Markov process* if for any sequence of time points $t_0 < t_1 < t_2 < ... < t_n < t$, the conditional distribution of X(t) for given values of $X(t_0)$, $X(t_1)$, ..., $X(t_n)$ depends only on $X(t_n)$, i.e.,

$$P\left[X(t) \le x \middle| X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \ldots, X(t_0) = x_0\right] = P\left[X(t) \le x \middle| X(t_n) = x_n\right] \quad (4.1)$$

for all t, t_n, t_{n-1},...,t_0 $\in$ T with x, x_n, x_{n-1},..., x_0 $\in$ S.

To interpret (4.1), we think of t_n as being the present time. Eq. (4.1) states that the evolution of a Markov process at a future time t, conditioned on its present and past values, depends only on its present value. Thus, the present value $X(t_n)$ contains all the information about the past evolution of the process that is needed to determine the future distribution of the process. The condition (4.1) is denoted the *Markov property* or *memoryless property*. In many problems of interest, the conditional distribution function (4.1) depends only on the time difference $t - t_n$, i.e., is invariant with respect to the time origin t_n:

$$P\left[X(t) \le x \middle| X(t_n) = x_n\right] = P\left[X(t - t_n) \le x \middle| X(0) = x_n\right] \quad (4.2)$$

In this case the Markov process is said to be *time-homogeneous*. Throughout this thesis, all considered Markov processes are assumed to be time homogeneous.

For the analysis of continuous-time Markov processes we confine our attention on discrete-state processes, i.e., the state space S is finite or countable infinite. Then the stochastic process is called *continuous-time Markov chain (CTMC)*. We consider a CTMC with finite state space S = {s_1, s_2, ..., s_N} of size N. The probability of being in state s_i at time t is denoted by $\pi_i(t) = P[X(t) = s_i]$. The state probabilities are combined in the *state probability vector* $\pi(t) = (\pi_1(t) ,..., \pi_N(t))$. The behavior of a CTMC is characterized by (i) the initial state probability vector given by the probability mass function of X(0), i.e., $\pi(0)$, and (ii) the transition probabilities $p_{ij}(t) = P[X(t) = s_j| X(0) = s_i]$, which are the elements of the *state transition matrix* $\mathbf{P}(t) = [p_{ij}(t)]$.

In order to determine the *transient state probability vector* $\pi(t)$ and/or the *steady-state probability vector* $\pi = \lim_{t\to\infty} \pi(t)$ for a given CTMC the generator matrix of the Markov chain is considered. Note that the steady-state probability vector is only defined if a steady-state solution exists. Sufficient conditions for the existence of a steady-state solution are stated in Theorem 4.3. The *generator matrix* of a CTMC, denoted by **Q**, has entries that are the rates at which the process jumps from state to state. These entries are defined by

$$q_{ij} = \lim_{t\to 0} \frac{p_{ij}(t)}{t}, \quad i \neq j. \tag{4.3}$$

The quantities q_{ij} describe the evolution of the CTMC in an infinitesimal unit of time. The diagonal entries of **Q** are set equal to minus the total rate out of state s_i,

$$q_{ii} = -\sum_{j\neq i} q_{ij} . \tag{4.4}$$

This implies that the row sums of matrix **Q** equal 0. Due to the Markov property (4.1) the time between state changes is memoryless. Since the exponential distribution is the only memoryless continuous distribution, the holding time of a particular state s_i is exponentially distributed with rate $-q_{ii}$. When the state space S is finite, the matrix **Q** generates the state transition matrix **P**(t) by means of the expansion of its matrix exponential, that is

$$\mathbf{P}(t) = e^{\mathbf{Q}t} = \sum_{n=0}^{\infty} \frac{t^n}{n!} \cdot \mathbf{Q}^n \tag{4.5}$$

To determine whether a steady-state solution exists, we first have to classify the states of a CTMC. A state s_i is said to be an *absorbing state* provided that $q_{ij} = 0$ for all $j \neq i$, so that, once entered, the process is destined to remain in that state. A state s_j is said to be *reachable* from state s_i if for some $t > 0$, the probability of traversing from state s_i to state s_j in time t is positive. A CTMC is said to be *irreducible* if every state is reachable from every other state. Assuming an irreducible CTMC with finite state space, the time dependent state probability vector at instant of time t is unique and can be determined by the system of ordinary differential equations:

$$\frac{d}{dt}\pi(t) = \pi(t) \cdot \mathbf{Q} \tag{4.6}$$

with given initial distribution $\pi(0)$. The system of differential equations (4.6) has the solution:

$$\pi(t) = \pi(0) \cdot e^{\mathbf{Q}t} \tag{4.7}$$

Eq. (4.7) relates the matrix exponential of the generator matrix **Q** to the time dependent distribution of a continuous-time Markov chain. Applying the *randomization technique* [GM84] (also called uniformization or Jensen's method) the transient solution can be derived numerically from Eq. (4.7). This method is based on a recursive vector-matrix multiplication of the transient probability vector of an appropriate discrete-time Markov chain with the corresponding probability matrix.

Taking the limit for t to infinity in (4.6), we get a system of linear equations, which can be easily solved to obtain the steady-state solution π. The conditions under which such a limiting probability vector exists and how it can be determined are summarized in the following theorem. For a proof of this result and further discussions on computing steady-state probabilities we refer to the standard literature, e.g., [Tri02].

Theorem 4.3 (Steady-state solution of CTMC): Let **Q** be the generator matrix of an irreducible continuous-time Markov chain. If the homogeneous system of equations

$$\pi \cdot \mathbf{Q} = 0 \tag{4.8}$$

with the additional normalization requirement that $|\pi| = 1$ has a solution π, then π is unique and constitutes the steady-state probability vector.

Note that the system of equations $\pi \cdot \mathbf{Q} = 0$ with normalization requirement $|\pi| = 1$ as considered in Theorem 4.3 indeed has a unique solution if the state space of the irreducible CTMC is finite. Since in subsequent chapters of this thesis we are concerned with irreducible CTMCs that have finite state space this provides us an appropriate vehicle for computing steady-state probabilities. Note that a CTMC exactly represents the stochastic behavior of a discrete-event stochastic system with only exponential events or immediate events. Examples are GSPNs or UML state diagrams with timed transitions that can occur only after an exponentially distributed delay. A CTMC is not the appropriate stochastic process for the analysis of DES with events that have deterministically distributed (or even generally distributed) delays since in this case the Markov property does not hold.

4.2 Generalized Semi-Markov Processes

A generalized semi-Markov process (GSMP) is a continuous-time stochastic process that makes a state transition when one or more “*events*” associated with the occupied state occur. Events associated with a state compete to trigger the next state transition, and each set of trigger events has its own distribution for determining the next state. At each state transition of the GSMP, *new* events may be scheduled. For each of these new events, a clock indicating the time until the event is scheduled to occur is set according to an independent (stochastic) mechanism, i.e., for each new event a clock reading is generated according to its *clock setting distribution*. For each scheduled event, which does not trigger a state transition but is still scheduled in the next state, its clock *continues* to run. If an event is no longer scheduled in the next state, it is *canceled*, and the corresponding clock reading is discarded. In general, in a GSMP events may occur simultaneously resulting in a set of trigger events E* rather than in a unique trigger event e* [She93].

Let $E = \{e_1, e_2, \ldots, e_K\}$ be a finite set of events and $S = \{s_1, s_2, \ldots, s_N\}$ be a finite set of states. For a state $s \in S$, let $s \mapsto E(s)$ be a mapping from the set S to a nonempty subset of E; E(s) denotes the set of all events that are scheduled to occur when the process is in state s. When the process is in state s, the (simultaneous) occurrence of one or more events of E(s) triggers a state transition to a state s'. Denote the probability that the new state is s' given that the event e* or the set of events E* occur in state s by p(s',s,e*) and p(s',s,E*), respectively. For each $s \in S$ and $e^* \in E(s)$ or $E^* \subseteq E(s)$, we assume that $p(\cdot,s,e^*)$ or $p(\cdot,s,E^*)$ is a probability mass function. Associated with each event is a clock with a reading that records the remaining time until the event is scheduled to trigger a state transition. Throughout this thesis, all clock readings are assumed to run with the same speed, although in general clock readings can run with different speeds. For $s \in S$, define the set C(s) of possible clock-reading vectors in state s as:

$$C(s) = \left\{ (c_1, \dots, c_K) \middle| c_j \ge 0 \text{ and } c_j > 0 \text{ if and only if } e_j \in E(s) \right\} \tag{4.9}$$

The j-th component of a clock-reading vector $\mathbf{c} = (c_1, c_2, \dots, c_K)$ is the clock reading associated with event e_j.

A generalized semi-Markov process $\{X(t): t \ge 0\}$ records the states of a discrete-event stochastic system as it evolves over time. Formally, it is defined in terms of a general state space Markov chain (GSSMC) $\{(S_n,\mathbf{C}_n): n \ge 0\}$ that describes the process at successive state-transition times. In order to maintain the Markov property, the state space must be extended with the clock readings of the enabled events. In fact, only clock readings of non-exponential events must be included due to the memoryless property of the exponential distribution. The random variables S_n describe the state of the process and the K-dimensional random variables $\mathbf{C}_n = (C_{n,1},\dots,C_{n,K})$ describe the corresponding clock readings. Thus, a GSSMC is a discrete-time process with continuous state space

$$\Sigma = \bigcup_{s \in S} \left(\{s\} \times C(s) \right) \subset S \times \mathbb{R}^K. \tag{4.10}$$

The term “chain” is a bit confusing since in general a chain is a discrete-state process. The term comes from the fact that the set of states S is discrete and is only extended to a continuous-state space by adding the clock reading vector. In a GSSMC, the continuous nature of the state space is expressed by the term “general state space”. Next we define a GSSMC over the state space Σ and subsequently a GSMP is defined in terms of a GSSMC.

Definition 4.4: A *general state space Markov chain (GSSMC)* is a sequence of random variables $\{(S_n,\mathbf{C}_n): n \ge 0\}$ that takes values in Σ and satisfies the Markov property:

$$P\left[(S_{n+1},\mathbf{C}_{n+1}) \in \mathcal{A} \middle| (S_n,\mathbf{C}_n),\dots,(S_0,\mathbf{C}_0)\right] = P\left[(S_{n+1},\mathbf{C}_{n+1}) \in \mathcal{A} \middle| (S_n,\mathbf{C}_n)\right] \tag{4.11}$$

for all $n \ge 0$ and measurable subsets $\mathcal{A}$ of Σ.

Definition 4.5: Let $\{(S_n,\mathbf{C}_n): n \ge 0\}$ be a GSSMC and let ζ_n be the real-valued point in time of the n-th state transition. Then, the continuous-time process $\{X(t): t \ge 0\}$ with $X(t) = S_{N(t)}$ and $N(t) = \max\left\{ n \middle| \zeta_n \le t \right\}$ is a *generalized semi-Markov process (GSMP).*

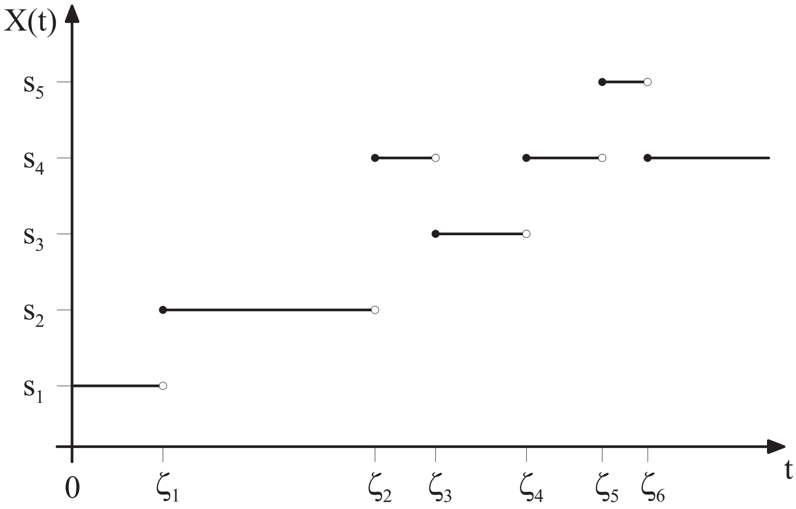


Figure 4.1. Possible evolution of a generalized semi-Markov process

Figure 4.1 shows a possible evolution of a GSMP with five states. The GSMP starts at time $\zeta_0 = 0$ in state s_1 and subsequently changes its state at time points ζ_n, n = 1,2,.... Note that the trajectory consists of a piecewise constant path. Furthermore, one can show that only a finite number of state transitions can occur in a finite time interval [She93]. The corresponding GSSMC describes the process exactly at time points ζ_n.

In the following, we assume that the GSSMC is time-homogeneous, i.e., Eq. (4.11) is independent of n:

$$P\left[(S_{n+1},\mathbf{C}_{n+1})\in\mathcal{A}\middle|(S_n,\mathbf{C}_n)\right]=P\left[(S_1,\mathbf{C}_1)\in\mathcal{A}\middle|(S_0,\mathbf{C}_0)\right] \tag{4.12}$$

With this assumption, the stochastic behavior of a GSSMC is completely specified by an initial distribution $\mu(\mathcal{B})=P\left[X_0\in\mathcal{B}\right]$, with $\mathcal{B}\subseteq\Sigma$, and a transition kernel. Heuristically, the transition kernel is a family of probability matrices specifying one-step jump probabilities. Due to the continuous state space of the GSSMC, the transition kernel must consider transitions from a state $x = (s_i,c_1,...,c_K)$ into a set of states $\mathcal{A}\subset\Sigma$ since the probability of a transition to a single state in a continuous state space equals zero.

Definition 4.6: The *transition kernel* $\mathbf{P}(\mathbf{c}, A)$ of a GSSMC is a square matrix with functional entries. Its elements $p_{ij}(\mathbf{c}, A)$ are functions of vector $\mathbf{c} = (c_1,...,c_K)$ and the set $A \subseteq C(s_j)$ such that the following conditions hold:

(i) For any fixed $(s_i,\mathbf{c}) \in \Sigma$, $p_{ij}(\mathbf{c}, A)$ is a probability distribution in $\{s_j\}\times A$.

(ii) For any valid set A, $p_{ij}(\mathbf{c}, A)$ is a piecewise continuous function of $\mathbf{c}$.

Note that in the case when all events occur after exponentially distributed delays, no clock readings need to be included in the states. Then, the set A and the vector **c** would be empty and the transition kernel reduces to the state transition matrix of a discrete-time Markov chain (DTMC) as a special case. The corresponding GSMP would be a continuous-time Markov chain.

4.3 The Transition Kernel of the General State Space Markov Chain

The numerical analysis of GSMPs can be separated into two main steps. First of all, the transition kernel of the embedded GSSMC must be computed by summation of transient state probabilities of appropriate CTMCs and second, a system of Fredholm integral equations must be solved numerically to derive the time-dependent and/or stationary solution of the GSMP. Both of these steps massively benefit from the detection of special structures in the state transition graph of the GSMP, especially those special structures that result in constant elements of the transition kernel.

4.3.1 Derivation of the Embedded General State Space Markov Chain

In this thesis, finite-state, time-homogeneous GSMPs with exponential and deterministic clock setting distributions are considered. Furthermore, events are assumed not to occur simultaneously, i.e., a unique event triggers a state transition. We divide the set of events $E = E_{exp} \cup E_{det}$ and enumerate the deterministic events by $e_1, e_2, \ldots, e_M$ and subsequently the exponential events by $e_{M+1}, e_{M+2}, \ldots, e_K$. We define D_m to be the firing delay of event e_m $(1 \leq m \leq M)$. Subsequently, we define $D = \min\{D_1, D_2, ..., D_M\}$. For the analysis of this class of GSMPs, we consider a GSSMC that is embedded at equidistant time points rather than at successive state-transition times. Thus, we consider the GSMP at a fixed sequence of time points, which is independent of the evolution of the process. This enables the transient analysis at equidistant time points nD and omits a difficult conversion of the state probabilities due to random holding times of states. To derive this GSSMC, we define a discrete-time process $X_n = \{X(nD): n \geq 0\}$ by observing the GSMP at a sequence $\{nD: n \geq 0\}$ of fixed times

$$X_n = \left(S_n, C_{n,1}, C_{n,2}, \ldots, C_{n,M}\right) \tag{4.13}$$

Here, S_n represents the state of the GSMP and $C_{n,m}$ represents the clock reading of deterministic event e_m $(1 \leq m \leq M)$ at instant of time nD. When deterministic event e_m is not enabled in state S_n, we set $C_{n,m} = 0$. The memoryless property of the exponential distribution implies that X_n is a GSSMC, i.e., it satisfies the Markov property (4.11).

In this thesis, GSMPs are considered under the restrictions that (i) at most two deterministic events are concurrently enabled, (ii) all deterministic events have the same delay D, and (iii) deterministic events cannot get canceled. Note that the presented methods can be generalized in a straightforward way such that each of these restrictions can be removed; in practice the analysis method gets more time- and space-consuming and numerical accuracy suffers. Therefore, we assume these restrictions hold if not mentioned otherwise. Nevertheless, it is described how to deal with deterministic events that can get canceled at several places in the text.

The subset of states of the GSSMC in which only exponential events are enabled is denoted by S_{exp}. Similarly, the subsets of states in which one deterministic event and two deterministic events are (concurrently) enabled are denoted by S_{det1} and S_{det2}, respectively. Subsequently, without loss of generality, we enumerate the states of the GSMP as follows:

$$S_{exp} = \{s_1, s_2, \ldots, s_{N_1}\}$$

$$S_{det1} = \{s_{N_1+1}, s_{N_1+2}, \ldots, s_{N_1+N_2}\} \tag{4.14}$$

$$S_{det2} = \{s_{N_1+N_2+1}, s_{N_1+N_2+2}, \ldots, s_N\}$$

We denote the index of the deterministic event enabled in a state $s_i \in S_{det1}$ by l(i) and the corresponding clock reading is denoted by c_1. For $s_i \in S_{det2}$ we denote the indices of the enabled deterministic events by l(i) and m(i), with l(i) < m(i), and the clock readings of $e_{l(i)}$ and $e_{m(i)}$ are denoted by c_1 and c_2, respectively. Other zero-valued clock readings in $C(s_i)$ are neglected. Given the initial distribution of the GSMP, denoted by X_0, and using (4.13), we define for the GSSMC X_n with at most two deterministic events concurrently enabled three kinds of time-dependent state probabilities [LT99]:

$$\pi_i^{(n)} = P\left[S_n = s_i \middle| X_0\right] \qquad ,\text{for } s_i \in S_{exp}$$

$$\pi_i^{(n)}(a_1) = P\left[S_n = s_i, C_{n,l(i)} \le a_1 \middle| X_0\right] \qquad ,\text{for } s_i \in S_{det1} \tag{4.15}$$

$$\pi_i^{(n)}(a_1, a_2) = P\left[S_n = s_i, C_{n,l(i)} \le a_1, C_{n,m(i)} \le a_2 \middle| X_0\right] \qquad ,\text{for } s_i \in S_{det2}$$

for n = 1,2,... and $0 < a_1, a_2 \le D$.

Subsequently, the transient state probabilities of the GSMP at instants of time t = nD are given by $\pi_i^{(n)}$ for $s_i \in S_{exp}$, $\pi_i^{(n)}(D)$ for $s_i \in S_{det1}$, and $\pi_i^{(n)}(D,D)$ for $s_i \in S_{det2}$, respectively. Corresponding stationary or time-averaged state probabilities of the GSMP are denoted as π_i, $\pi_i(D)$, and $\pi_i(D,D)$. Note that the stationary probability distribution of the embedded GSSMC exists if the corresponding GSMP has a stationary probability distribution, since a subsequence of a convergent sequence of real numbers converges and the limits are the same.

For graphical representation, we introduce the notion of the *state transition graph* of a GSMP, which is defined as a directed multigraph G with a set of vertices S, i.e., the states of the GSMP. The states of G are connected by labeled edges representing state transitions. An edge corresponding to a state transition from state s to s' which is triggered by the deterministic event $e \in E_{det}$ is denoted by a triple (s,s',e). Edges representing exponential state transitions are triples (s,s',λ) with s, s' $\in$ S and λ the rate of the exponential event that triggers the corresponding state transition. A general (weighted) edge representing both cases is denoted by (s,s',w). The state transition graph gives an intuitive understanding of the behavior of the type of GSMPs considered in this thesis.

Example 4.7 (M/D/2/K queueing system):

As an example, we consider the state transition graph of an M/D/2/K queueing system as presented in Figure 4.2. The system comprises two identical servers with constant service time D and one queue with limited capacity K. Customers arrive according to a Poisson process with rate λ. Thus, the GSMP consists of two deterministic events, e_1 and e_2, which occur when the service of a customer is finished and one exponential event, e_3, representing the arrival of a new customer. When an arriving customer finds an empty system, it enters server 1 with probability p and server 2 with probability (1-p). These next state probabilities are separated from the event rates with a semicolon in Figure 4.2.

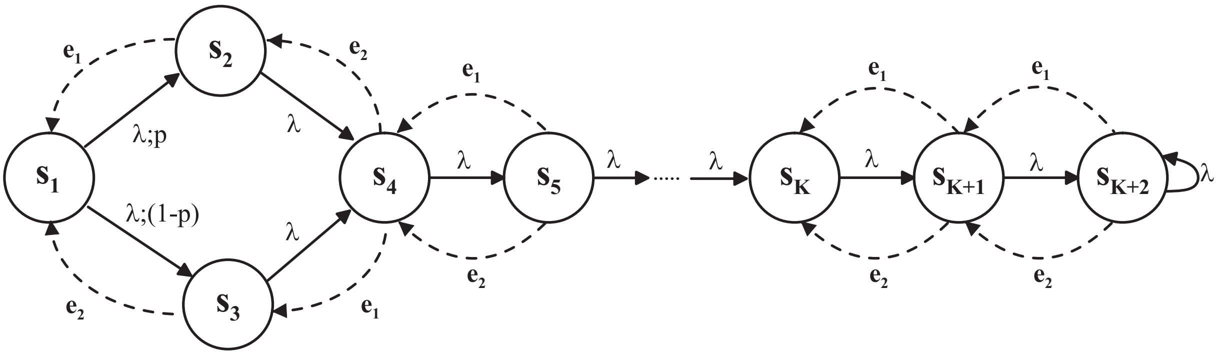


Figure 4.2. State transition graph of an M/D/2/K queueing system

The state of the corresponding GSMP is determined by the number of customers in the system. When just a single customer is in the system it must be distinguished whether this customer is served at server 1 or server 2. Thus, the state space S of the embedded GSSMC consists of K+2 states, i.e., S = {s_1, s_2, s_3, ..., s_{K+2}}. State s_1 corresponds to an empty system and state s_{K+2} represents the case with K customers in the system. The state space is divided in states S_{exp} = {s_1} with only the exponential event enabled, S_{det1} = {s_2, s_3} with exactly one deterministic event enabled, and states S_{det2} = {s_4, s_5, ..., s_{K+2}} with two deterministic events concurrently enabled. For the indices of the deterministic events enabled in states of S_{det1} holds l(2) = 1 and l(3) = 2. For states of S_{det2} holds l(i) = 1 and m(i) = 2 for i = 4,5,...,K+2. The sample path shown in Figure 4.1 represents a possible evolution of a M/D/2/K queueing system over time. At time points ζ_1, ζ_2, ζ_4, and ζ_5, customers arrive to the system and at time instants ζ_3 and ζ_6 the service of a customer is finished. Since the service time is constant it holds $\zeta_3 - \zeta_1 = \zeta_6 - \zeta_2 = D$. ■

4.3.2 General Form of the Transition Kernel

Recall that a GSSMC is completely specified by a transition kernel and an initial distribution at time t = 0. The transition kernel of the GSSMC specifies one-step jump probabilities from a given state at instant of time nD to all reachable new states at instant of time (n+1)D. As for an ordinary discrete-time Markov chain, for all states s_j not reachable from s_i corresponding jump probabilities $p_{ij}(.)$ are zero. In general, elements of the transition kernel of the GSSMC are functions of clock readings associated with the current state s_i and the new state s_j.

The transition kernel of the GSSMC $X_n = \{(S_n,\mathbf{C}_n): n \geq 0\}$ constitutes a functional matrix of the form $\mathbf{P}(\mathbf{c},A)$. In general, the elements of the transition kernel $\mathbf{P}(\mathbf{c},A)$ of the GSSMC have the form:

$$p_{ij}(\mathbf{c},A) = P\left[X_{n+1} \in \{s_j\} \times A \middle| X_n = (s_i,\mathbf{c})\right] \tag{4.16}$$

Restricting the discussion to GSMPs with at most two deterministic events concurrently enabled, the vector of old clock readings $\mathbf{c}$ and the set A for intervals of new clock readings are given by:

$$\mathbf{c} = \mathbf{c}(s_i) = \begin{cases} \varnothing & ,s_i \in S_{\exp} \\ c_1 & ,s_i \in S_{\det 1} \\ (c_1,c_2) & ,s_i \in S_{\det 2} \end{cases} \quad \text{and} \quad A = A(s_j) = \begin{cases} \varnothing & ,s_j \in S_{\exp} \\ (0,a_1] & ,s_j \in S_{\det 1} \\ (0,a_1] \times (0,a_2] & ,s_j \in S_{\det 2} \end{cases} \tag{4.17}$$

Thus, for GSMPs with at most two deterministic events concurrently enabled, the transition kernel of the GSSMC can be expressed by a functional matrix $\mathbf{P}(c_1,c_2,a_1,a_2)$. Subsequently, an element of this kernel $p_{ij}(\cdot)$ is in general a function in four variables c_1, c_2, a_1, and a_2. However, we will observe that a large number of kernel elements are constant (Theorem 4.11), i.e., $p_{ij}(c_1,c_2,a_1,a_2) = p_{ij}$. Furthermore, for most functional kernel elements new clock readings need not be considered, i.e., $p_{ij}(c_1,c_2,a_1,a_2) = p_{ij}(c_1,c_2)$.

$$\mathbf{P}(c_1,c_2,a_1,a_2)= \begin{array}{c} \left(\begin{array}{c|c|c} \mathbf{P}_{11} & \mathbf{P}_{12}(a_1) & \mathbf{P}_{13}(a_1,a_2) \\ \hline \mathbf{P}_{21}(c_1) & \mathbf{P}_{22}(c_1,a_1) & \mathbf{P}_{23}(c_1,a_1,a_2) \\ \hline \mathbf{P}_{31}(c_1,c_2) & \mathbf{P}_{32}(c_1,c_2,a_1) & \mathbf{P}_{33}(c_1,c_2,a_1,a_2) \end{array}\right) \\ 1 \qquad N_1 \mid N_1+1 \quad N_1+N_2 \mid N_1+N_2+1 \qquad N \end{array} \begin{array}{c} 1 \\ \vdots \\ N_1 \\ \hline N_1+1 \\ \vdots \\ N_1+N_2 \\ \hline N_1+N_2+1 \\ \vdots \\ N \end{array} \tag{4.18}$$

Eq. (4.18) shows the general form of the kernel $\mathbf{P}(c_1,c_2,a_1,a_2)$ as a composition of nine submatrices $\mathbf{P}_{ij}(\cdot)$ of appropriate dimension using (4.16) and (4.17). In (4.18), the submatrix $\mathbf{P}_{11}$ represents state transitions among states of $S_{\exp}$, $\mathbf{P}_{12}(a_1)$ represents state transitions from states of $S_{\exp}$ to states of $S_{\det1}$, and $\mathbf{P}_{13}(a_1,a_2)$ represents state transitions from states of $S_{\exp}$ to states of $S_{\det2}$. Furthermore, submatrix $\mathbf{P}_{22}(c_1,a_1)$ represents state transitions among states of $S_{\det1}$ and $\mathbf{P}_{21}(c_1)$ represents state transitions from states of $S_{\det1}$ to states of $S_{\exp}$. The submatrix $\mathbf{P}_{23}(c_1,a_1,a_2)$ represents state transitions from states of $S_{\det1}$ to states of $S_{\det2}$, respectively. State

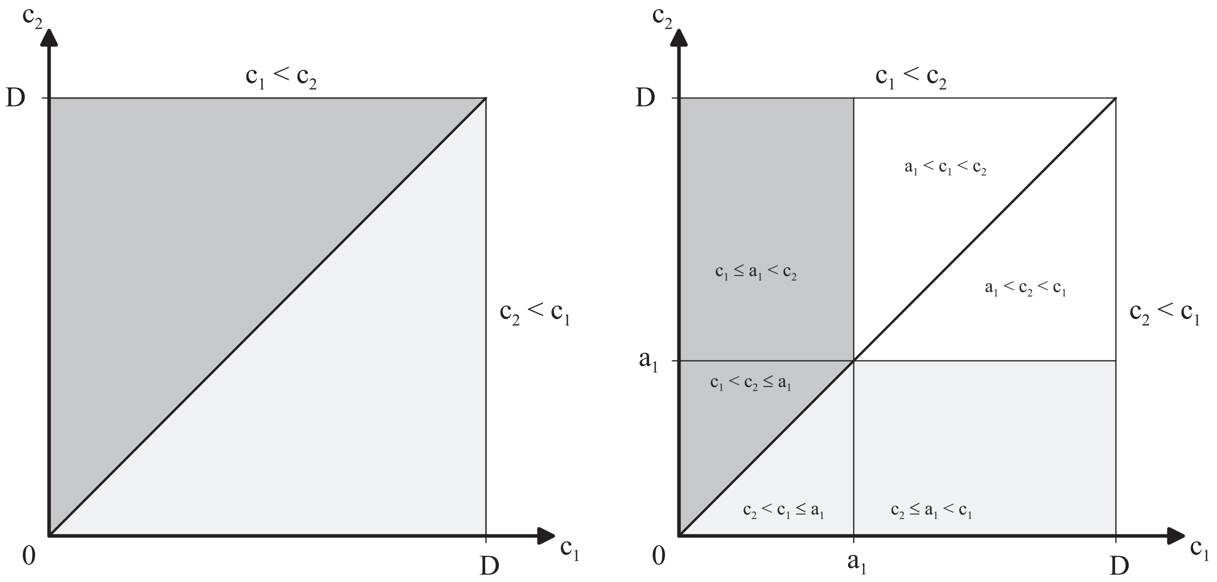


Figure 4.3. Regions with continuous kernel elements in submatrices $P_{31}(c_1,c_2)$ and $P_{32}(c_1,c_2,a_1)$

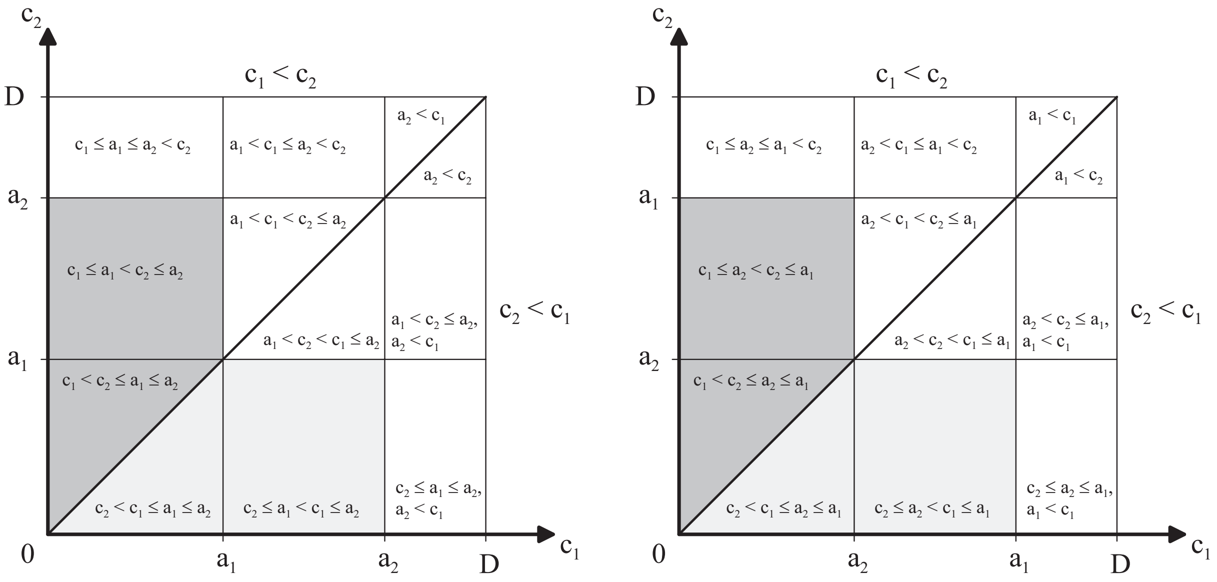


Figure 4.4. Regions with continuous kernel elements in submatrix $P_{33}(c_1,c_2,a_1,a_2)$ with $a_1 \le a_2$ and with $a_2 \le a_1$

transitions from states of S_{det2} to states of S_{det1} and S_{exp} are represented by the submatrices $\mathbf{P}_{32}(c_1,c_2,a_1)$ and $\mathbf{P}_{31}(c_1,c_2)$. The submatrix $\mathbf{P}_{33}(c_1,c_2,a_1,a_2)$ represents state transitions among states of S_{det2}.

Note that for any fixed state s_i and clock readings c_1 and c_2, $p_{ij}(c_1,c_2,a_1,a_2)$ is a probability distribution in $\{s_j\}\times(0,a_1]\times(0,a_2]$. Furthermore, functional kernel elements are continuous and differentiable for every possible ordering of the clock readings c_1, c_2, a_1, and a_2. Thus, the conditions for a transition kernel of Definition 4.6 hold. However, in general, functional kernel elements $p_{ij}(\cdot)$ are not continuous at the boundary. This is because different orderings of clock readings may lead to different functional kernel elements $p_{ij}(\cdot)$. For kernel elements of $\mathbf{P}_{33}(c_1,c_2,a_1,a_2)$, in general, there may exist 24 possible orderings for clock readings c_1, c_2, a_1, and a_2. These orderings immediately lead to the regions of integration in the system of integral equations introduced in the Section 4.4. Figure 4.3 shows the two possible regions for kernel elements in $\mathbf{P}_{31}(c_1,c_2)$ and the six possible subregions for elements in $\mathbf{P}_{32}(c_1,c_2,a_1)$. Finally, Figure 4.4 shows the 24 subregions for kernel elements in $\mathbf{P}_{33}(c_1,c_2,a_1,a_2)$. However, when deterministic events $e_{l(i)}$ and $e_{m(i)}$ associated with state s_i cannot get canceled, only the eight gray shaded regions in Figure 4.4 may occur. Furthermore, if the state probability $\pi_i(a_1,a_2)$ is symmetric with respect to a_1 and a_2 and the deterministic events $e_{l(i)}$ and $e_{m(i)}$ cannot get canceled, only the four orderings for the case $a_1 \leq a_2$ of Figure 4.4 have to be considered. Sections 4.3.3 and 4.3.4 give further insight into the structure of the transition kernel and provide the foundation for its algorithmic computation in the simplest form.

4.3.3 Computation of the Transition Kernel

In previous work, the concept of subordinated Markov chains (SMCs) has been applied for the efficient algorithmic computation of the probability matrix **P** of the discrete-time Markov chain embedded in the Markov regenerative process underlying a discrete-event stochastic system without concurrent deterministic events (see e.g. [Lin95a]). The SMC of state s_i is a continuous-time Markov chain (CTMC) whose state space is given by the transitive closure of all states reachable from s_i via a (possible empty) sequence of exponential events and corresponding next state probabilities $p(s',s_i,e^*)$ of the GSMP. For such a sequence of exponential events from s_i to s_j, we write $s_i \xrightarrow{\text{exp*}} s_j$. We define a SMC for each state of the GSMP, i.e., also for states in which only exponential events are enabled.

Definition 4.8 (Subordinated Markov chain): The continuous-time Markov chain $\{X_i(t)\text{: } t \geq 0\}$ with state transitions corresponding to the occurrence of exponential events, state space SMC$_i$ = $\{s \in S \mid s_i \xrightarrow{\text{exp*}} s\}$, and initial distribution $P\{X_i(0) = s_i\} = 1$ is called the *subordinated Markov chain (SMC)* of state s_i.

The probability for a state transition from state s_i to state s_j in time t via the occurrence of only exponential events is given by:

$$P\Big[s_i \xrightarrow[t]{\text{exp*}} s_j\Big] \overset{\text{def}}{=} P\Big[X_i(t) = s_j \Big| X_i(0) = s_i\Big] = \mathbf{1}_i^T \cdot e^{\mathbf{Q}_i t} \cdot \mathbf{1}_j \tag{4.19}$$

where the generator matrix of SMC $X_i(t)$ is denoted by $\mathbf{Q}_i$ and $\mathbf{1}_i^T$ and $\mathbf{1}_j$ denote the i-th and j-th row and column unity-vectors, respectively, of appropriate dimension. Furthermore, we define the set of states from which a state s_j is only reachable via the occurrence of exponential events as the *subordinated reachability set (SRS)* of state s_j, formally $\text{SRS}_j = \{s \in S \mid s \xrightarrow{\text{exp*}} s_j\}$. Using the randomization technique [GM84] with a fixed error tolerance ε, transient state probabilities of SMCs of (4.19) for time t = D can be computed with effort $O(\eta_i q_i D)$. Here, η_i denotes the number of nonzero entries in the generator $\mathbf{Q}_i$ and q_i

the absolute value of its maximum diagonal entry. It is important that the effort $O(\eta_i q_i D)$ also includes the computation of transient state probabilities for all time points $t < D$. For a subset $Z \subseteq S$, we define the following compound state transition probabilities, which provide the appropriate vehicle to represent the complex elements of the transition kernel:

$$P\left[s_i \xrightarrow[t_1]{\exp^*} Z \xrightarrow[t_2]{\exp^*} s_j\right] \overset{\text{def}}{=} \sum_{z \in Z} P\left[s_i \xrightarrow[t_1]{\exp^*} z\right] \cdot P\left[z \xrightarrow[t_2]{\exp^*} s_j\right] \tag{4.20}$$

Note that a summand in the right side of Eq. (4.20) is zero for $z \notin SMC_i \cap SRS_j$, since the first factor under the sum is zero if $z \notin SMC_i$ and the second factor is zero if $z \notin SRS_j$. Therefore, it is reasonable to assume that $Z \subseteq SMC_i \cap SRS_j$. Similar to Eq. (4.20), we define for the subsets $Z \subseteq SMC_i$ and $Z' \subseteq SRS_j$ the compound probabilities:

$$P\left[s_i \xrightarrow[t_1]{\exp^*} Z \xrightarrow{e} Z' \xrightarrow[t_2]{\exp^*} s_j\right] \overset{\text{def}}{=} \sum_{(z,z',e)} P\left[s_i \xrightarrow[t_1]{\exp^*} z\right] \cdot P\left[z' \xrightarrow[t_2]{\exp^*} s_j\right] \tag{4.21}$$

where (z,z',e) corresponds to an edge of the state transition graph of the GSMP with $z \in Z$, $z' \in Z'$, and $e \in E_{det}$. Since Z and Z' contain the possible intermediate states when traversing from state s_i to s_j, these sets are called *sets of intermediate states*. In order to express more complex probabilities also combinations of definitions (4.20) and (4.21) are possible, which are defined in a straightforward way. In the following, we denote by $S_m \subseteq S_{det1}$ all states in which deterministic event e_m is (exclusively) enabled. By $S_m^n \subseteq S_{det2}$, $m \neq n$, we denote all states in which both deterministic events e_m and e_n are (concurrently) enabled. Finally, we define $\tilde{S}_m^n := S_m \cup S_m^n$.

The following provides an intuitive explanation why elements of the transition kernel of a GSSMC can always be determined by appropriate sums of transient state probabilities of continuous-time Markov chains [Lin98]. Assuming the GSMP is at time nD in state s_i with two deterministic events $e_{l(i)}$ and $e_{m(i)}$ concurrently enabled. Thus, the GSSMC resides in a state, say (s_i,c_1,c_2) with $c_1 \leq c_2$, where c_1 and c_2 are clock readings associated with deterministic events $e_{l(i)}$ and $e_{m(i)}$, respectively. Noting that the state of the GSMP at time (n+1)D given the state at time nD is determined by (possibly empty) sequences of exponential events in the subintervals $((nD,nD+c_1]$, $(nD+c_1,nD+c_2]$ and $(nD+c_2,(n+1)D]$ and the occurrence of the deterministic events $e_{l(i)}$ and $e_{m(i)}$ at instants of time $nD+c_1$ and $nD+c_2$, respectively. Thus, using the property that the GSMP is time-homogeneous and by decomposing the time interval (0,D] into three subintervals $(0,c_1]$, $(c_1,c_2]$, and $(c_2,D]$, the GSMP behaves in each subinterval as a CTMC. Each of these three CTMCs is given by an SMC as defined above. Subsequently, the kernel elements of the embedded GSSMC can be computed as summations of transient state probabilities of SMCs. It is important to note that this holds irrespective of the number of deterministic events enabled in states s_i and s_j. The following theorem summarizes the discussion above and constitutes one of the main results of this chapter.

Theorem 4.9 (Numerical computation of the transition kernel): Let $\{X(t): t \geq 0\}$ be a finite-state GSMP with exponential and deterministic events. Then, all elements $p_{ij}(.)$ of the transition kernel $\mathbf{P}(c_1,c_2,a_1,a_2)$ of the embedded GSSMC $\{X_n: n \geq 0\}$ can be computed simply by summation of transient state probabilities of continuous-time Markov chains.

Proof: We prove this result by construction. A complete proof of Theorem 4.9 requires the consideration of nine different forms of kernel elements introduced in Eq. (4.18) and 24 orderings of clock readings for $\mathbf{P}_{33}(c_1,c_2,a_1,a_2)$ as shown in Figure 4.4. We spell out the derivation only for five selected forms, namely for the submatrices $\mathbf{P}_{11}$, $\mathbf{P}_{12}(a_1)$, $\mathbf{P}_{13}(a_1,a_2)$, $\mathbf{P}_{22}(c_1,a_1)$, and one ordering for $\mathbf{P}_{33}(c_1,c_2,a_1,a_2)$. Following the proof it should get clear that kernel elements of other forms could be derived in a similar way.

Recall that $X_n = \{(S_n,\mathbf{C}_n): n \geq 0\}$ is the GSSMC embedded in the GSMP at equidistant instants of time nD. For ease of exposition, we assume $p(z',z,e_m) = 1$ in the following proof. The extension of Eqs. (4.22) to (4.27) to an arbitrary pmf of next state probabilities $p(z',z,E^*)$ is straightforward; i.e., requires one additional summation. Let us first derive how to compute kernel elements p_{ij} of submatrix $\mathbf{P}_{11}$. Since no deterministic event is enabled in state s_i, we need not decompose the time interval (0,D] into subintervals. Deterministic events cannot have triggered a state transition in (0,D], though, some deterministic events may have become enabled and get canceled in (0,D]. Thus, considering the SMC of state s_i and using (4.19), we derive the kernel element as:

$$p_{ij} = P\left[S_{n+1} = s_j \middle| S_n = s_i\right] = P\left[s_i \xrightarrow[D]{\exp^*} s_j\right] \tag{4.22}$$

That is the probability for a state transition from s_i to s_j via the occurrence of only exponential events. Eq. (4.22) implies that kernel elements $p_{ij} = 0$ for states $s_j \notin SMC_i$.

Now consider the derivation of kernel elements of submatrix $\mathbf{P}_{12}(a_1)$. Corresponding kernel elements are of the form $p_{ij}(a_1)$ where a_1 denotes the boundary of the clock reading interval of the deterministic event newly enabled in state s_j; see Eq. (4.17). Subsequently, we decompose the time interval (0,D] into subintervals $(0,a_1]$ and $(a_1,D]$. Using (4.20), we have:

$$p_{ij}(a_1) = P\left[S_{n+1} = s_j, C_{n+1,l(j)} \leq a_1 \middle| S_n = s_i\right] = P\left[s_i \xrightarrow[a_1]{\exp^*} S_{l(j)} \xrightarrow[D-a_1]{\exp^*} s_j\right] \tag{4.23}$$

with the set of intermediate states $S_{l(j)}$. Subsequently, we consider the derivation of kernel elements of submatrix $\mathbf{P}_{13}(a_1,a_2)$. Recall that a_1 and a_2 denote upper bounds of clock readings of new deterministic event. Since we have to consider two clock readings, two possible orderings may occur. For $a_1 \leq a_2$, using (4.20), we get:

$$\begin{aligned} p_{ij}(a_1,a_2) &= P\left[S_{n+1} = s_j, C_{n+1,l(j)} \leq a_1, C_{n+1,m(j)} \leq a_2 \middle| S_n = s_i\right] \\ &= P\left[s_i \xrightarrow[a_1]{\exp^*} S_{l(j)} \xrightarrow[a_2-a_1]{\exp^*} S_{l(j)}^{m(j)} \xrightarrow[D-a_2]{\exp^*} s_j\right] \\ &\quad + P\left[s_i \xrightarrow[a_1]{\exp^*} S_{l(j)}^{m(j)} \xrightarrow[D-a_1]{\exp^*} s_j\right] \end{aligned} \tag{4.24}$$

For $a_2 \leq a_1$, kernel elements can be derived by exchanging a_1 with a_2 and replacing $S_{l(j)}$ with $S_{m(j)}$ in the right side of Eq. (4.24). Next, we consider the derivation of kernel elements of the submatrix $\mathbf{P}_{22}(c_1,a_1)$. Formulas (4.25) and (4.26) hold under the assumption that the deterministic event $e_{l(i)}$ cannot get canceled. The case that event $e_{l(i)}$ can get canceled is discussed below. For $c_1 \leq a_1$, using (4.20) and (4.21), we get:

$$\begin{aligned} p_{ij}(c_1,a_1) &= P\left[S_{n+1} = s_j, C_{n+1,l(j)} \leq a_1 \middle| S_n = s_i, C_{n,l(i)} = c_1\right] \\ &= P\left[s_i \xrightarrow[c_1]{\exp^*} S_{l(i)} \xrightarrow{e_{l(i)}} S_{\exp} \xrightarrow[a_1-c_1]{\exp^*} S_{l(j)} \xrightarrow[D-a_1]{\exp^*} s_j\right] \\ &\quad + P\left[s_i \xrightarrow[c_1]{\exp^*} \tilde{S}_{l(i)}^{l(j)} \xrightarrow{e_{l(i)}} S_{l(j)} \xrightarrow[D-c_1]{\exp^*} s_j\right] \end{aligned} \tag{4.25}$$

The first term at the right side of (4.25) represents the probability of state transitions from s_i to s_j via intermediate states of $S_{\exp}$ and the second term considers transitions with intermediate states of S_{det1} and S_{det2}. For $a_1 < c_1$, we get:

$$\begin{aligned} p_{ij}(c_1,a_1) &= P\left[S_{n+1} = s_j, C_{n+1,l(j)} \leq a_1 \middle| S_n = s_i, C_{n,l(i)} = c_1\right] \\ &= P\left[s_i \xrightarrow[a_1]{\exp^*} S_{l(i)}^{l(j)} \xrightarrow[c_1-a_1]{\exp^*} S_{l(i)}^{l(j)} \xrightarrow{e_{l(i)}} S_{l(j)} \xrightarrow[D-c_1]{\exp^*} s_j\right] \end{aligned} \tag{4.26}$$

If the deterministic event $e_{l(i)}$ can get canceled, kernel elements of the submatrix $\mathbf{P}_{22}(c_1,a_1)$ are derived by formulas similar to (4.25) and (4.26). The difference lies in additionally distinguishing whether the deterministic event became canceled or remained enabled until its scheduled occurrence time $nD+c_1$. As a consequence, the original SMC associated with state s_i is divided in two different SMCs: a first SMC for the case that the deterministic event scheduled in state s_i at time nD, denoted by $e_{l(i)}$, remained enabled until time $nD+c_1$ and a second SMC represented all feasible cases that event $e_{l(i)}$ gets canceled in the time interval $(nD, nD+c_1]$ due to the occurrence of an exponential event. For the former case, again, two orderings of clock readings (i.e., $c_1 \leq a_1$ and $a_1 < c_1$) are possible.

Finally, let us consider the derivation of kernel elements of the submatrix $\mathbf{P}_{33}(c_1,c_2,a_1,a_2)$ as the most general case. Recall from Figure 4.4 that in general 24 orderings of clock readings c_1, c_2, a_1, and a_2 are possible. In Eq. (4.27), we derive how to compute kernel elements of the form $p_{ij}(c_1,c_2,a_1,a_2)$ for the ordering $c_1 \leq c_2 \leq a_1 \leq a_2$ under the assumption that both deterministic events $e_{l(i)}$ and $e_{m(i)}$ cannot get canceled.

$$\begin{aligned} & p_{ij}(c_1,c_2,a_1,a_2) \\ &= P\left[S_{n+1} = s_j, C_{n+1,l(j)} \leq a_1, C_{n+1,m(j)} \leq a_2 \middle| S_n = s_i, C_{n,l(i)} = c_1, C_{n,m(i)} = c_2\right] \\ &= P\left[\begin{array}{l} s_i \xrightarrow[c_1]{\exp^*} S_{m(i)}^{l(i)} \xrightarrow{e_{l(i)}} S_{m(i)} \xrightarrow[c_2-c_1]{\exp^*} S_{m(i)} \xrightarrow{e_{m(i)}} S_{\exp} \xrightarrow[a_1-c_2]{\exp^*} S_{l(j)} \cdots \\ \cdots \xrightarrow[a_2-a_1]{\exp^*} S_{m(j)}^{l(j)} \xrightarrow[D-a_2]{\exp^*} s_j \end{array}\right] \\ &\quad + P\left[s_i \xrightarrow[c_1]{\exp^*} S_{m(i)}^{l(i)} \xrightarrow{e_{l(i)}} \tilde{S}_{m(i)}^{m(j)} \xrightarrow[c_2-c_1]{\exp^*} \tilde{S}_{m(i)}^{m(j)} \xrightarrow{e_{m(i)}} \tilde{S}_{\exp}^{m(j)} \xrightarrow[a_1-c_2]{\exp^*} S_{m(j)}^{l(j)} \xrightarrow[D-a_1]{\exp^*} s_j\right] \\ &\quad + P\left[s_i \xrightarrow[c_1]{\exp^*} S_{m(i)}^{l(i)} \xrightarrow{e_{l(i)}} \tilde{S}_{m(i)}^{l(j)} \xrightarrow[c_2-c_1]{\exp^*} \tilde{S}_{m(i)}^{l(j)} \xrightarrow{e_{m(i)}} S_{l(j)} \xrightarrow[a_2-c_2]{\exp^*} S_{m(j)}^{l(j)} \xrightarrow[D-a_2]{\exp^*} s_j\right] \\ &\quad + P\left[s_i \xrightarrow[c_1]{\exp^*} S_{m(i)}^{l(i)} \xrightarrow{e_{l(i)}} \tilde{S}_{m(i)}^{l(j),m(j)} \xrightarrow[c_2-c_1]{\exp^*} \tilde{S}_{m(i)}^{l(j),m(j)} \xrightarrow{e_{m(i)}} S_{m(j)}^{l(j)} \xrightarrow[D-c_2]{\exp^*} s_j\right] \end{aligned} \tag{4.27}$$

with additional sets of intermediate states $\tilde{S}^{m}_{exp} = S_{exp} \cup S_m$ and $\tilde{S}^{n,k}_{m} = S_m \cup S^n_m \cup S^k_m$. ■

Note that for the case if $S = S_{det2}$ the analysis simplifies considerably, since in each state of the GSSMC both deterministic events are active. Therefore, if additionally deterministic events cannot get canceled, clock readings of the GSMP once set in the initial state are the same at every embedding time point nD of the GSSMC. As a consequence, the transition kernel $\mathbf{P}(c_1,c_2,a_1,a_2)$ of the GSSMC reduces to a state transition matrix $\mathbf{P}_{\tilde{c}_1,\tilde{c}_2}$, $0 < \tilde{c}_1, \tilde{c}_2 \le D$, of an ordinary DTMC for every fixed initial clock readings $c_1 = \tilde{c}_1$ and $c_2 = \tilde{c}_2$. This provides a probabilistic explanation why an embedded DTMC exists for a structurally restricted class of deterministic and stochastic Petri nets, recently studied by German [Ger99].

Example 4.10 (Transition kernel of M/D/2/K queueing system):

As an example, the transition kernel of the M/D/2/K queuing system introduced in Example 4.7 is considered for K = 2. Thus, the transition kernel comprises a 4 × 4 matrix. Since the M/D/2/K queue has a Poisson arrival stream transient probabilities of the SMCs can be represented by closed-form formulas. This enables an exact representation of the elements of the transition kernel in all cases. From the structure of the queueing system we find two different types of SMCs, which are the building blocks of the transition kernel:

$$b_i(t) \overset{def}{=} P[\text{exactly i arrivals in time t}] = \frac{(\lambda t)^i}{i!} \cdot e^{-\lambda t}$$

$$B_i(t) \overset{def}{=} P[\text{at least i arrivals in time t}] = 1 - \sum_{k=0}^{i-1} b_k(t)$$

The following presents some elements of the transition kernel and provides an intuitive understanding of their derivation. For ease of notation we simply write the index i for a representation of state s_i. Furthermore, the probability sign in the notations defined in Eqs. (4.19), (4.20), and (4.21) is omitted. Finally, we abbreviate the positive differences $|a_1 - a_2|$ and $|c_1 - c_2|$ with Δa and Δc, respectively. Elements representing state transitions from states of S_{exp} to states of S_{exp}, S_{det1} and S_{det2} are given by:

$$p_{11} = 1 \xrightarrow[D]{exp^*} 1 = b_0(D)$$

$$p_{12}(a_1) = 1 \xrightarrow[a_1]{exp^*} 2 \xrightarrow[D-a_1]{exp^*} 2 = p \cdot b_1(a_1) b_0(D-a_1)$$

$$p_{13}(a_1) = 1 \xrightarrow[a_1]{exp^*} 3 \xrightarrow[D-a_1]{exp^*} 3 = (1-p) \cdot b_1(a_1) b_0(D-a_1)$$

$$p_{14}(a_1,a_2) = \begin{cases} 1 \xrightarrow[a_1]{exp^*} 2 \xrightarrow[a_2-a_1]{exp^*} 4 \xrightarrow[D-a_2]{exp^*} 4 + 1 \xrightarrow[a_1]{exp^*} 4 \xrightarrow[D-a_1]{exp^*} 4 & , a_1 \le a_2 \\ 1 \xrightarrow[a_2]{exp^*} 3 \xrightarrow[a_1-a_2]{exp^*} 4 \xrightarrow[D-a_1]{exp^*} 4 + 1 \xrightarrow[a_2]{exp^*} 4 \xrightarrow[D-a_2]{exp^*} 4 & , a_2 \le a_1 \end{cases}$$

$$= \begin{cases} B_2(a_1) + p \cdot b_1(a_1) B_1(\Delta a) & , a_1 \le a_2 \\ B_2(a_2) + (1-p) \cdot b_1(a_2) B_1(\Delta a) & , a_2 \le a_1 \end{cases}$$

Elements representing state transitions from states of S_{det1} to states of S_{exp} are given by:

$$p_{21}(c_1) = 2 \xrightarrow[c_1]{\text{exp*}} 2 \xrightarrow{e_1} 1 \xrightarrow[D-c_1]{\text{exp*}} 1 = b_0(D)$$

$$p_{31}(c_1) = 3 \xrightarrow[c_1]{\text{exp*}} 3 \xrightarrow{e_2} 1 \xrightarrow[D-c_1]{\text{exp*}} 1 = b_0(D)$$

Elements representing state transitions from states of S_{det1} to states of S_{det1} are given by:

$$p_{22}(c_1,a_1) = \begin{cases} 2 \xrightarrow[c_1]{\text{exp*}} 2 \xrightarrow{e_1} 1 \xrightarrow[a_1-c_1]{\text{exp*}} 2 \xrightarrow[D-a_1]{\text{exp*}} 2 & ,c_1 \le a_1 \\ 0 & ,a_1 \le c_1 \end{cases}$$

$$= \begin{cases} p \cdot b_0(c_1)b_1(a_1 - c_1)b_0(D - a_1) & ,c_1 \le a_1 \\ 0 & ,a_1 \le c_1 \end{cases}$$

$$p_{23}(c_1,a_1) = \begin{cases} 2 \xrightarrow[c_1]{\text{exp*}} 2 \xrightarrow{e_1} 1 \xrightarrow[a_1-c_1]{\text{exp*}} 3 \xrightarrow[D-a_1]{\text{exp*}} 3 + 2 \xrightarrow[c_1]{\text{exp*}} 4 \xrightarrow{e_1} 3 \xrightarrow[D-c_1]{\text{exp*}} 3 & ,c_1 \le a_1 \\ 2 \xrightarrow[a_1]{\text{exp*}} 4 \xrightarrow[c_1-a_1]{\text{exp*}} 4 \xrightarrow{e_1} 3 \xrightarrow[D-c_1]{\text{exp*}} 3 & ,a_1 \le c_1 \end{cases}$$

$$= \begin{cases} (1-p) \cdot b_0(c_1)b_1(a_1 - c_1)b_0(D - a_1) + B_1(c_1)b_0(D - c_1) & ,c_1 \le a_1 \\ B_1(a_1)b_0(D - c_1) & ,a_1 \le c_1 \end{cases}$$

The kernel elements $p_{33}(c_1,a_1)$ and $p_{32}(c_1,a_1)$ have similar form as elements $p_{22}(c_1,a_1)$ and $p_{23}(c_1,a_1)$ except that p must be replaced with (1-p). Finally, we consider elements representing state transitions from states of S_{det2} to states of S_{det2}. For $c_1 \le c_2 \le a_1 \le a_2$ we get:

$$\begin{aligned} p_{44}(c_1,c_2,a_1,a_2) = {} & 4 \xrightarrow[c_1]{\text{exp*}} 4 \xrightarrow{e_1} 3 \xrightarrow[c_2-c_1]{\text{exp*}} 3 \xrightarrow{e_2} 1 \xrightarrow[a_1-c_2]{\text{exp*}} 2 \xrightarrow[a_2-a_1]{\text{exp*}} 4 \xrightarrow[D-a_2]{\text{exp*}} 4 \\ & + 4 \xrightarrow[c_1]{\text{exp*}} 4 \xrightarrow{e_1} 3 \xrightarrow[c_2-c_1]{\text{exp*}} 3 \xrightarrow{e_2} 1 \xrightarrow[a_1-c_2]{\text{exp*}} 4 \xrightarrow[D-a_1]{\text{exp*}} 4 \\ & + 4 \xrightarrow[c_1]{\text{exp*}} 4 \xrightarrow{e_1} 3 \xrightarrow[c_2-c_1]{\text{exp*}} 4 \xrightarrow{e_2} 2 \xrightarrow[a_2-c_2]{\text{exp*}} 4 \xrightarrow[D-a_2]{\text{exp*}} 4 \end{aligned}$$

and for $c_1 \le a_1 \le c_2 \le a_2$ holds

$$p_{44}(c_1,c_2,a_1,a_2) = 4 \xrightarrow[c_1]{\text{exp*}} 4 \xrightarrow{e_1} 3 \xrightarrow[a_1-c_1]{\text{exp*}} 4 \xrightarrow[c_2-a_1]{\text{exp*}} 4 \xrightarrow{e_2} 2 \xrightarrow[a_2-c_2]{\text{exp*}} 4 \xrightarrow[D-a_2]{\text{exp*}} 4$$

Considering all cases we get $p_{44}(c_1,c_2,a_1,a_2) =$

$$\begin{cases} p \cdot b_0(\Delta c)b_1(a_1 - c_2)B_1(\Delta a) + b_0(\Delta c)B_2(a_1 - c_2) + B_1(\Delta c)B_1(a_2 - c_2) & ,c_1 \le c_2 \le a_1 \le a_2 \\ p \cdot b_0(\Delta c)b_1(a_1 - c_1)B_1(\Delta a) + b_0(\Delta c)B_2(a_1 - c_1) + B_1(\Delta c)B_1(a_1 - c_1) & ,c_2 \le c_1 \le a_1 \le a_2 \\ (1-p) \cdot b_0(\Delta c)b_1(a_2 - c_2)B_1(\Delta a) + b_0(\Delta c)B_2(a_2 - c_2) + B_1(\Delta c)B_1(a_2 - c_2) & ,c_1 \le c_2 \le a_2 \le a_1 \\ (1-p) \cdot b_0(\Delta c)b_1(a_2 - c_1)B_1(\Delta a) + b_0(\Delta c)B_2(a_2 - c_1) + B_1(\Delta c)B_1(a_1 - c_1) & ,c_2 \le c_1 \le a_2 \le a_1 \\ B_1(a_1 - c_1)B_1(a_2 - c_2) & ,c_1 \le a_1 \le c_2 \le a_2 \\ B_1(a_2 - c_2)B_1(a_1 - c_1) & ,c_2 \le a_2 \le c_1 \le a_1 \\ 0 & ,\text{else} \end{cases}$$

Note that only eight of the 24 possible orderings of clock readings must be considered for the M/D/2/K queue since deterministic events cannot get canceled (see Figure 4.4). Furthermore, two of the eight possible cases cannot occur since the M/D/2/K queue consists only of two deterministic events at all. Thus, the orderings $c_1 \le a_2 \le c_2 \le a_1$ and $c_2 \le a_1 \le c_1 \le a_2$ lead to zero state transition probabilities. ■

4.3.4 Detection of Constant Kernel Elements

In this section, we state sufficient conditions on the building blocks of the GSMP under which kernel elements are constant because jump probabilities of the GSSMC are independent of clock readings. Examples of this case for which the entire kernel comprises constant kernel elements constitute GSMPs underlying queueing systems with quasi birth-death arrival process, one or several deterministic servers, and infinite waiting room (i.e., MAP/D/c queues). This implies that for such GSMPs the corresponding GSSMC behaves in fact as an ordinary DTMC. As illustrated in Section 4.6, the GSSMC underlying queueing systems with finite waiting room behaves almost as a DTMC, i.e., almost all kernel elements are constant.

For arbitrary GSMPs, a necessary condition for kernel elements of the GSSMC to be constant is that clock readings of new deterministic events at time (n+1)D do not depend on the occurrence of exponential events in [nD,(n+1)D). For GSMPs with at most two deterministic events concurrently enabled, this implies $p_{ij}(c_1,a_1) = p_{ij}(c_1)$ and $p_{ij}(c_1,c_2,a_1,a_2) = p_{ij}(c_1,c_2)$. Recall that $z \xrightarrow{e_{l(i)}} z'$ denotes the state transition from state z to z' due to the occurrence of deterministic event $e_{l(i)}$. Recall also that we write $s_i \xrightarrow[t]{\exp^*} s_j$ for a sequence of exponential events from s_i to s_j in time t. According to (4.21), we write for $s_i, s_j \in S_{det1}$ a path $s_i \xrightarrow[c_1]{\exp^*} z \xrightarrow{e_{l(i)}} z' \xrightarrow[D-c_1]{\exp^*} s_j$ for a sequence of exponential events in $(0,c_1]$ followed by the occurrence of deterministic event $e_{l(i)}$ and another sequence of exponential events in $(c_1,D]$. If for each such path holds $z' \in S_{det1}$ and the deterministic events enabled in state s_i and state z' cannot get canceled, then the clock reading of the new deterministic event does not depend on the occurrence of exponential events in [nD,(n+1)D). In Section 4.5 we show how to check the independence of bounds a_1 and a_2 for new clock readings with the help of appropriately defined sets of states.

In order to decide whether kernel elements $p_{ij}(c_1,c_2)$ are also independent of old clock readings c_1 and c_2 we consider a path in the state transition graph of the GSMP. A *path* from state s_i to s_j is defined as a sequence of states (and corresponding enabled and occurring events) that are temporarily hold by the GSMP when traversing from state s_i to s_j in time interval [0,D]. The *length* of a path is defined as the number of states traversed. The set of *all* feasible paths from state s_i to s_j is denoted by $PATH(s_i,s_j)$. Assuming that deterministic events cannot get canceled, Figure 4.5 depicts the general form of a path $\tau \in PATH(s_i,s_j)$ of length n from state $s_i \in S_{det1}$ to an arbitrary state s_j. The deterministic event $e_{l(i)}$ occurs in the k-th state z_k of the path. Furthermore, the rates of exponential events that are enabled and occur on this path are included in Figure 4.5. Note that in general the probability of traversing a path from

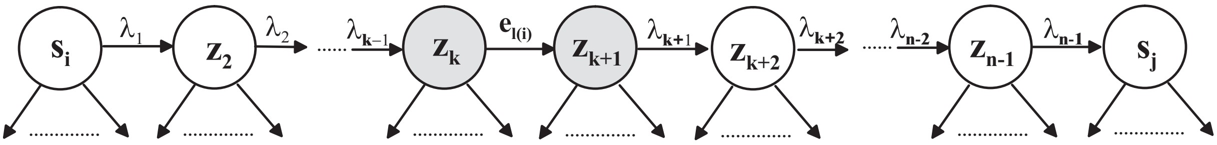


Figure 4.5. General form of a path τ from state $s_i \in S_{det1}$ to state s_j

s_i to s_j depends on the clock reading c_1 of the deterministic event $e_{l(i)}$. The key idea for detecting kernel elements that are independent of clock reading c_1 is to show that under some conditions the paths of PATH(s_i,s_j) can be grouped into a (possibly infinite) number of classes Γ_m, m = 1,2,..., such that the *proportionate jump probability* $p_{ijm}(c_1)$ of traversing the paths of Γ_m is constant. Furthermore, we show that the detection as well as the computation of constant kernel elements can be easily performed without explicitly computing the classes Γ_m.

In the following we define some notations in order to deal with a special class of states that defines a subgraph of the state transition graph of the GSMP. Let s_i and s_j be states of $S_{det1} \cup S_{det2}$. The set of all states located on a path of PATH(s_i,s_j) in which the deterministic event e_m occurs is denoted by $V_{pre}(s_i,s_j,e_m)$. The set of all states located on a path of PATH(s_i,s_j) which are reached due to the occurrence of deterministic event e_m is denoted by $V_{post}(s_i,s_j,e_m)$. Note that in general the sets $V_{pre}(\cdot)$ and $V_{post}(\cdot)$ depend on the ordering of old clock readings c_1 and c_2. The graph component which is defined by the set of states $V_{pre}(s_i,s_j,e_m)$ and $V_{post}(s_i,s_j,e_m)$ is denoted by $G_{pre}(s_i,s_j,e_m) = (V_{pre}(s_i,s_j,e_m),\ E_{pre}(s_i,s_j,e_m))$ and $G_{post}(s_i,s_j,e_m) = (V_{post}(s_i,s_j,e_m),\ E_{post}(s_i,s_j,e_m))$, respectively. The sets of edges $E_{pre}(\cdot)$ and $E_{post}(\cdot)$ should contain not only edges between states of the corresponding graph component but also all other edges starting from states of the graph component. Furthermore, let f_m: S $\rightarrow$ S be a mapping from a state $s \in S$ to the state that is reached due to the occurrence of the deterministic event e_m. For states in which deterministic event e_m is not enabled f_m is not defined.

Theorem 4.11 presents a sufficient condition under which kernel elements $p_{ij}(c_1)$ for s_i, $s_j \in S_{det1}$ are constant. Intuitively, a kernel element $p_{ij}(c_1)$ is constant if the graph component $G_{pre}(s_i,s_j,e_m)$ is mapped onto $G_{post}(s_i,s_j,e_m)$ by the function f_m, i.e., G_{pre} and G_{post} are isomorphic with isomorphism f_m. We would like to point out that Theorem 4.11 does not require the computation of an isomorphism f_m. In fact, the algorithmic implementation of Theorem 4.11 requires only the verification of f_m being an isomorphism. That is, the mapping f_m must be applied to every state of G_{pre} and the resulting subgraph must be simply compared with the graph G_{post} for identity. Such a comparison is only of asymptotic complexity $O(N \cdot K)$ with N the overall number of states and K the overall number of events, respectively. After stating the proof of Theorem 4.11, conditions under which kernel elements $p_{ij}(c_1,c_2)$ for s_i, $s_j \in S_{det2}$ are constant are derived in a straightforward way.

Theorem 4.11 (Constant kernel elements): Consider a GSMP with exponential and deterministic events. Let $\mathbf{P}(c_1,c_2,a_1,a_2)$ be the transition kernel of its GSSMC $\{X_n\colon n \geq 0\}$ and s_i and $s_j \in S_{det1}$ with $p_{ij}(c_1,a_1)$ independent of a_1. Then, the corresponding kernel element $p_{ij}(c_1,a_1)$ is constant if the following condition holds:

$(s,s',w) \in E_{pre}(s_i,s_j,e_{l(i)}) \Leftrightarrow (f_{l(i)}(s),f_{l(i)}(s'),w) \in E_{post}(s_i,s_j,e_{l(i)})$

Proof: This result is proven by constructing a constant kernel element assuming the condition of Theorem 4.11 holds. First, we show that under this condition the paths $\tau \in PATH(s_i,s_j)$ can be grouped into sets Γ_m, m = 1,2,..., fulfilling the properties (i) to (iv) as stated below. Subsequently, we show that the proportionate jump probability $p_{ijm}(c_1)$ of traversing the paths of Γ_m is constant. In order to simplify the notation we omit the index m of Γ_m and denote such an arbitrary class of paths by Γ. The paths of Γ should fulfill the following properties:

(i) Γ contains n-1 paths of length n,

(ii) for k = 1,2,...,n-1, the set Γ contains a path τ_k such that the deterministic event $e_{l(i)}$ occurs at position k, i.e., τ_k contains the state transition $z_k \xrightarrow{e_{l(i)}} z_{k+1}$,

(iii) the sets of exponential events scheduled in state z_k and z_{k+1} of path τ_k (k = 1,2,...,n-1) and corresponding next state probabilities are equal, i.e., $E(z_k) \cap E_{exp} = E(z_{k+1}) \cap E_{exp}$ and $p(\cdot,z_k,e^*) = p(\cdot,z_{k+1},e^*)$,

(iv) the multi-set of exponential events scheduled and occurring in path τ_k in states $z_1,z_2,...,z_{k-1}$ (in states $z_{k+1},z_{k+2},...,z_n$) must be a subset (a superset, respectively) of the corresponding multi-set in τ_{k+1}. Furthermore, corresponding next state probabilities must be equal.

Now we start the construction of Γ by considering an arbitrary path $\tau \in PATH(s_i,s_j)$. Let n be the length of path τ and let k be the position on the path τ where the deterministic event $e_{l(i)}$ occurs (see Figure 4.5). To emphasize the occurrence of $e_{l(i)}$ at position k, we denote the path τ by τ_k. Note that exactly one such position k must exist because we assume that the deterministic event $e_{l(i)}$ cannot get canceled. In the following, we construct a path τ_{k+1} (see Figure 4.6) from the given path τ_k such that conditions (ii) to (iv) hold for τ_k and τ_{k+1}. Then, starting with k = 1, i.e., s_i is the state where the deterministic event occurs, the paths τ_1, τ_2, ..., τ_{n-1} can be constructed fulfilling properties (i) to (iv).

The states s_i=$z_1,z_2,...,z_k$ and $z_{k+2},...,z_n$=s_j on path τ_{k+1} are the same as on path τ_k and therefore condition (iv) holds for τ_k and τ_{k+1}. Condition (ii) simply holds by the definition of τ_k and τ_{k+1}. What we have to show is that a state $z'_{k+1} \in V_{pre}(s_i,s_j,e_{l(i)})$ exists with deterministic transition to z_{k+2} and exponential transition from z_k to z'_{k+1}. Recall that the condition of Theorem 4.11 implies that $f_{l(i)}$ is an isomorphism. Therefore, the inverse mapping $f^{-1}_{l(i)}$ exists. Let $z'_{k+1} := f^{-1}_{l(i)}(z_{k+2})$, then, obviously there is a deterministic transition from z'_{k+1} to z_{k+2}.

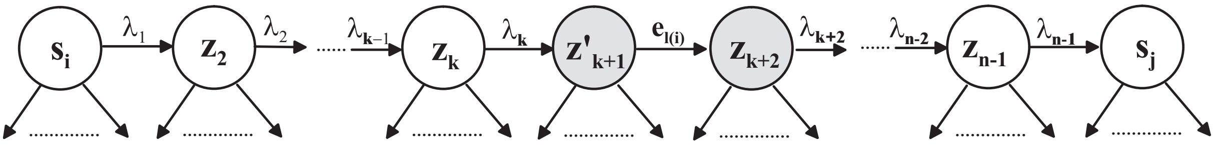


Figure 4.6. Path τ_{k+1} with deterministic event occurring at position k+1

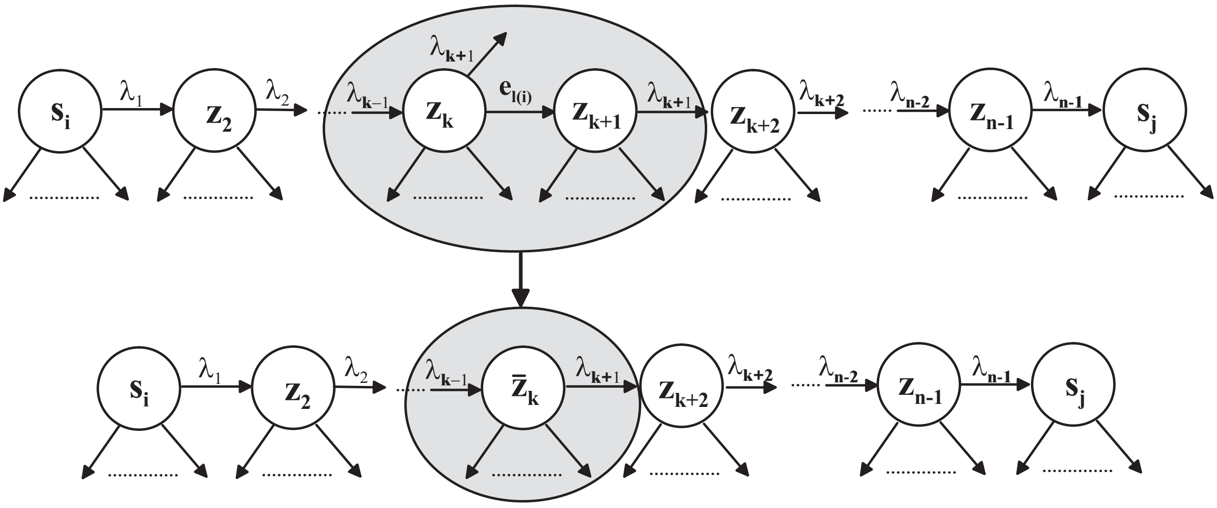


Figure 4.7. Combining states in path τ_k leading to constant jump probabilities

Furthermore, the same exponential events are enabled in state z'_{k+1} as in state z_{k+2} because $f_{l(i)}$ is an isomorphism, i.e., condition (iii) holds. To complete the path τ_{k+1} we have to show finally, that an exponential transition from state z_k to z'_{k+1} exists. Path τ_k contains an exponential transition from $f_{l(i)}(z_k)=z_{k+1}$ to state z_{k+2}. Applying the inverse function $f^{-1}_{l(i)}$ shows that there exists also an exponential transition from z_k to z'_{k+1}.

To complete the proof it remains to show that the proportionate jump probability, from state s_i to state s_j via paths of class Γ is constant, i.e., independent of clock reading c_1. Following Theorem 4.9 the proportionate jump probability p_{ijm} is given by:

$$p_{ijm}(c_1) = \sum_{k=1}^{n-1} P\left[s_i \xrightarrow[c_1]{\exp^*} z_k\right] \cdot P\left[z_{k+1} \xrightarrow[D-c_1]{\exp^*} s_j\right] \qquad (4.28)$$

Because of condition (iii), the same set of exponential events is scheduled in states z_k and z_{k+1} and corresponding next state probabilities are equal for k = 1,2,...,n-1. Thus, we can combine the two states to one state $\bar{z}_k$ as illustrated in Figure 4.7. Subsequently, we can rewrite (4.28) as:

$$p_{ijm}(c_1) = \sum_{k=1}^{n-1} P\left[s_i \xrightarrow[c_1]{\exp^*} \bar{z}_k\right] \cdot P\left[\bar{z}_k \xrightarrow[D-c_1]{\exp^*} s_j\right] \qquad (4.29)$$

Recall that according to condition (iv), each path contains the same (multi)-set of n-2 scheduled and occurring exponential events and corresponding next state probabilities are equal. Note that Eq. (4.29) can be interpreted as the convolution of two exponential phase-type random variables given by the exponential events occurring in (0, c_1] and in (c_1, D]. Since according to (ii) the class Γ contains exactly one path τ_k such that the deterministic event $e_{l(i)}$ occurs at position k, the summation in Eq. (4.29) is taken over all possible cases. Thus, the jump probability p_{ijm} is given by the probability that the considered n-2 exponential events occur in (0,D]. In order to compute the constant proportionate jump probability we simply set $c_1 = 0$.

$$\begin{aligned} p_{ijm} = p_{ijm}(0) &= \sum_{k=1}^{n-1} P\left[s_i \xrightarrow[0]{\exp^*} \overline{z}_k\right] \cdot P\left[\overline{z}_k \xrightarrow[D]{\exp^*} s_j\right] \\ &= P\left[\overline{z}_1 \xrightarrow[D]{\exp^*} s_j\right] = P\left[z_2 \xrightarrow[D]{\exp^*} s_j\right] \end{aligned} \tag{4.30}$$

At the beginning of the proof the path τ_k was chosen arbitrarily and therefore we can construct for every path of PATH(s_i,s_j) the corresponding class Γ with constant proportionate jump probability. Subsequently, these probabilities can be added to the overall constant kernel element p_{ij}. ■

Note that the condition of Theorem 4.11 can be generalized in a straightforward way to kernel elements of the form $p_{ij}(c_1,c_2)$ for s_i, $s_j \in S_{det2}$. For this case we have separate conditions for each clock reading c_1 and c_2:

$$p_{ij}(c_1,c_2) = p_{ij}(c_2) \quad \text{if} \;\; (s,s',w) \in E_{pre}(s_i,s_j,e_{l(i)}) \Leftrightarrow (f_{l(i)}(s),f_{l(i)}(s'),w) \in E_{post}(s_i,s_j,e_{l(i)}) \tag{4.31}$$

$$p_{ij}(c_1,c_2) = p_{ij}(c_1) \quad \text{if} \;\; (s,s',w) \in E_{pre}(s_i,s_j,e_{m(i)}) \Leftrightarrow (f_{m(i)}(s),f_{m(i)}(s'),w) \in E_{post}(s_i,s_j,e_{m(i)}) \tag{4.32}$$

The kernel element $p_{ij}(c_1,c_2,a_1,a_2) = p_{ij}(c_1,c_2)$ is constant if both, conditions (4.31) and (4.32) hold. Note that both conditions must be checked for each ordering of clock readings c_1 and c_2 since in general the graph components are different for the two possible orderings. The proof follows the same ideas as for the proof of Theorem 4.11. Therefore, we omit the details. Note that from a methodological point of view Theorem 4.11 as well as Theorem 4.9 can be generalized to GSMPs with more than two deterministic events concurrently enabled. In fact, a larger number of different cases must be considered. To provide an example for detecting constant kernel elements, we show that almost all kernel elements of queueing systems with Markovian arrival process and deterministic servers are constant.

Example 4.12 (Constant kernel elements of M/D/2/K queueing system):

Considering the M/D/2/K queueing system introduced in Example 4.7, we show that most of the kernel elements are constant. A first observation is that kernel elements $p_{ij}(\cdot) = 0$ for $5 \le j+2 < i \le K+2$ since not more than two customers can be served by the deterministic servers from time nD to (n+1)D. Furthermore, for $i < 6$, state transitions from s_i to s_j traverse states $s_1 \in S_{exp}$ and/or states s_2, $s_3 \in S_{det1}$. Thus, these transitions depend on clock readings a_1 and/or a_2 and are no candidates for constant kernel elements. On the other for $i \ge 6$ kernel elements depend only on old clock readings c_1 and/or c_2. To determine whether these kernel elements are also independent of c_1 and c_2 we have to consider the sets $V_{pre}(\cdot)$ and $V_{post}(\cdot)$. For $4 \le i\text{-}2 \le j \le K\text{-}1$ we have:

$$V_{pre}(s_i,s_j,e_1) = \{s_i,\ldots,s_{j+2}\} \text{ and } V_{post}(s_i,s_j,e_1) = \{s_{i-1},\ldots,s_{j+1}\}, \quad \text{for } c_1 < c_2 \tag{4.33}$$

$$V_{pre}(s_i,s_j,e_1) = \{s_{i-1},\ldots,s_{j+1}\} \text{ and } V_{post}(s_i,s_j,e_1) = \{s_{i-2},\ldots,s_j\}, \quad \text{for } c_2 < c_1 \tag{4.34}$$

Considering the corresponding graph components in Figure 4.2 we find an isomorphism with the given mapping $f_1(s_i) = s_{i-1}$ for $c_1 < c_2$ and $c_2 < c_1$. According to condition (4.31), corresponding kernel elements are independent of clock reading c_1. For deterministic event e_2 the sets $V_{pre}(\cdot)$ and $V_{post}(\cdot)$ and the mapping can be derived in a similar way. Putting it together, corresponding kernel elements are independent of clock readings c_1 and c_2, i.e., are constant. For $i > 6$ and $j = K$ we have a somewhat different situation. Considering Eq. (4.33) we find no independence of clock reading c_1 for $c_1 < c_2$ since in state s_{K+2} no exponential event is enabled but in the state s_{K+1}, which is the result of mapping $f_1(s_{K+2})$, the exponential event corresponding to customer arrivals is enabled. On the other hand, for $c_2 < c_1$ the kernel elements $p_{i,K}(c_1,c_2)$ are independent of c_1 with respect to Eq. (4.34). Considering deterministic event e_2 we find the opposite behavior, i.e., $p_{i,K}(c_1,c_2)$ is independent of c_2 for $c_1 < c_2$ but depends on c_2 for $c_2 < c_1$. For $j = K+1$ and $j = K+2$ kernel elements depend in general on both clock readings c_1 and c_2. Note that the discussion about constant kernel elements for the M/D/2/K queueing system can be extended to queueing systems with more general arrival processes, e.g., MAP/D/2/K queues or even BMAP/D/2/K queues [Luc93]. The results in Section 4.6 show that for such queueing systems almost all kernel elements are constant. ■

4.4 Transient and Stationary Analysis of the GSSMC

Recall that an element $p_{ij}(c_1,c_2,a_1,a_2)$ of the transition kernel constitutes the conditional one-step jump probability of been in state s_i with deterministic clock readings c_1 and c_2 and jumping to state s_j with clock readings less or equal than a_1 and a_2, respectively. To solve the time-dependent and stationary equations (4.15) of the GSSMC, kernel elements have to be unconditioned. This can be done by applying the law of total probability for the joint random variable $X_n = (S_n,\mathbf{C}_n)$, which consists of one discrete and at most two continuous random variables representing the state and the clock readings, respectively. The result constitutes a system of Fredholm integral equations over a piecewise continuous transition kernel for the stationary and transient solution, respectively. To write these system of integral equations in vector notation, we define three vectors of state probabilities $\pi^{(n)}_{\exp}$, $\pi^{(n)}_{\det 1}(a_1)$, and $\pi^{(n)}_{\det 2}(a_1,a_2)$ for the states of S_{exp}, S_{det1}, and S_{det2}, respectively. To further simplify the notation in the system of integral equations (4.36) to (4.38), we introduce two vectors $\mathbf{y}^{(n)}(c_1)$ and $\mathbf{z}^{(n)}(c_1,c_2)$ for the derivatives of state probabilities, i.e., the density functions:

$$\mathbf{y}^{(n)}(c_1) \overset{\mathrm{def}}{=} \frac{d\pi^{(n)}_{\det 1}(c_1)}{dc_1} \qquad \text{and} \qquad \mathbf{z}^{(n)}(c_1,c_2) \overset{\mathrm{def}}{=} \frac{\partial^2 \pi^{(n)}_{\det 2}(c_1,c_2)}{\partial c_1 \partial c_2} \tag{4.35}$$

Then, using the piecewise continuous submatrices $\mathbf{P}_{ij}(.)$ of the transition kernel defined in (4.18), time-dependent state probabilities for the GSMP at instants of time nD are given by:

$$\begin{aligned}\pi^{(n+1)}_{\exp} &= \pi^{(n)}_{\exp} \cdot \mathbf{P}_{11} + \int_0^D \mathbf{y}^{(n)}(c_1) \cdot \mathbf{P}_{21}(c_1) dc_1 \\ &+ \int_0^D \int_0^{c_2} \mathbf{z}^{(n)}(c_1,c_2) \cdot \mathbf{P}_{31}(c_1,c_2) + \mathbf{z}^{(n)}(c_2,c_1) \cdot \mathbf{P}_{31}(c_2,c_1) dc_1 dc_2\end{aligned} \tag{4.36}$$

$$\begin{aligned}\pi_{\text{det1}}^{(n+1)}(a_1) = \pi_{\exp}^{(n)} \cdot \mathbf{P}_{12}(a_1) + \int_0^{a_1} \mathbf{y}^{(n)}(c_1) \cdot \mathbf{P}_{22}(c_1, a_1) dc_1 + \int_{a_1}^{D} \mathbf{y}^{(n)}(c_1) \cdot \mathbf{P}_{22}(c_1, a_1) dc_1 \\ + \int_0^{a_1}\int_0^{c_2} \mathbf{z}^{(n)}(c_1, c_2) \cdot \mathbf{P}_{32}(c_1, c_2, a_1) + \mathbf{z}^{(n)}(c_2, c_1) \cdot \mathbf{P}_{32}(c_2, c_1, a_1) dc_1 dc_2 \\ + \int_{a_1}^{D}\int_0^{a_1} \mathbf{z}^{(n)}(c_1, c_2) \cdot \mathbf{P}_{32}(c_1, c_2, a_1) + \mathbf{z}^{(n)}(c_2, c_1) \cdot \mathbf{P}_{32}(c_2, c_1, a_1) dc_1 dc_2\end{aligned} \tag{4.37}$$

$$\begin{aligned}\pi_{\text{det2}}^{(n+1)}(a_1, a_2) = \pi_{\exp}^{(n)} \cdot \mathbf{P}_{13}(a_1, a_2) + \int_0^{a_1} \mathbf{y}^{(n)}(c_1) \cdot \mathbf{P}_{23}(c_1, a_1, a_2) dc_1 + \int_{a_1}^{a_2} \mathbf{y}^{(n)}(c_1) \cdot \mathbf{P}_{23}(c_1, a_1, a_2) dc_1 \\ + \int_0^{a_1}\int_0^{c_2} \mathbf{z}^{(n)}(c_1, c_2) \cdot \mathbf{P}_{33}(c_1, c_2, a_1, a_2) + \mathbf{z}^{(n)}(c_2, c_1) \cdot \mathbf{P}_{33}(c_2, c_1, a_1, a_2) dc_1 dc_2 \\ + \int_{a_1}^{a_2}\int_0^{a_1} \mathbf{z}^{(n)}(c_1, c_2) \cdot \mathbf{P}_{33}(c_1, c_2, a_1, a_2) + \mathbf{z}^{(n)}(c_2, c_1) \cdot \mathbf{P}_{33}(c_2, c_1, a_1, a_2) dc_1 dc_2\end{aligned} \tag{4.38}$$

for $a_1 \leq a_2$

where $0 \leq a_1$, $a_2 \leq D$ and $\pi_{\text{det1}}^{(n)}(0) = \pi_{\text{det2}}^{(n)}(c_1, 0) = \pi_{\text{det2}}^{(n)}(0, c_2) = 0$. Exchanging a_1 and a_2 at the boundary of the integrals in Eq. (4.38) leads to the corresponding equation for $\pi_{\text{det2}}^{(n+1)}(a_1, a_2)$ with $a_2 \leq a_1$. The system of integral equations (4.36) to (4.38) consists of integrals over the triangular and rectangular continuous regions of the transition kernel as represented by the gray shaded regions in Figure 4.3 and Figure 4.4. Note that for the case that deterministic events can get canceled all continuous regions of Figures 4.3 and 4.4 must be considered for integration in the system (4.36) to (4.38).

Transient solutions at instants of time $t = nD$ can be derived by performing n iterations of (4.36) to (4.38). According to a successive approximation algorithm for Volterra integral equations [Lin85] we call this iterative scheme *Picard iteration*. Taking the limits $n \to \infty$ in (4.36) to (4.38) and replacing the left side of (4.37) and (4.38) with integrals over the corresponding density functions, we derive a system of Fredholm integral equations with unknown functions $\pi_{\exp}$, $\mathbf{y}(c_1)$, and $\mathbf{z}(c_1,c_2)$. This system of integral equations can be solved numerically based on a direct quadrature of the integrals and subsequent solution of one large but very sparse linear system of equations. According to the direct solution we call this algorithm the *direct quadrature method (DQ-method).*

Noting that if state probabilities are symmetric with respect to clocks of concurrent deterministic events leads to $\pi_{\text{det2}}(a_1,a_2) = \pi_{\text{det2}}(a_2,a_1)$ for $0 < a_1,a_2 \leq D$. As a consequence, Eq. (4.38) for $a_2 \leq a_1$ of the systems of integral equations can be omitted for the numerical analysis of the corresponding GSMP. In the most general setting, the state probability $\pi_i(a_1,a_2)$ is symmetric when the state transition graph G of the GSMP is isomorphic to the state transition graph G' of the GSMP in which all edges labeled with $e_{l(i)}$ are replaced by edges labeled with $e_{m(i)}$ and vice-versa. An example of such a case constitutes the M/D/2/K queue, if both servers are attended with probability p = $\frac{1}{2}$. The isomorphism g:G→G' is simply given by $g(s_i) = s_i$ for i = 1,4,5,...,K+2 and $g(s_2)=s_3$ and $g(s_3)=s_2$.

4.5 Algorithmic Description of the Numerical Approach

The methodological result that all elements of the transition kernel of the GSSMC can be expressed by summation of transient state probabilities of CTMCs (Theorem 4.9) reduces the computation of jump probabilities of a stochastic process with continuous state space, i.e., a GSSMC to transient analysis of a number of simple stochastic processes, i.e., the SMCs $X_i(t)$.

Figure 4.8 presents a high level description of the algorithm for generating the transition kernel $\mathbf{P}(c_1,c_2,a_1,a_2)$ of the GSSMC. The algorithm assumes that the building blocks of the GSMP are given by the state transition graph as introduced in Section 4.3.1. Using an equidistant discretization 0, Δ, 2Δ,.., MΔ of the clock readings the transient state probabilities $P[s_i \xrightarrow[m\Delta]{exp^*} s_j]$, m = 0,1,2,...,M, of SMC $X_i(t)$ can be computed (see step (1) of Figure 4.8). Recall that transient analysis of an SMC using the randomization technique requires an asymptotic effort of $O(\eta_i q_i M\Delta)$. Furthermore, step (1) significantly benefits from the exploitation of repetitive structures among the SMCs as proposed in [Lin95], i.e., the randomization technique must be only applied for a small subclass of all states.

```
(1)  Compute the transient state probabilities $P[s_i \xrightarrow[m\Delta]{exp^*} s_j]$ for every discretization step m
     by exploiting special structures of the SMCs according to [Lin95a]
(2)  Compute the sets $V_{pre}(s_i,s_j,e_{l(i)})$, $V_{post}(s_i,s_j,e_{l(i)})$, $V_{pre}(s_i,s_j,e_{m(i)})$ and $V_{post}(s_i,s_j,e_{m(i)})$ according to
     Eqs. (4.39) to (4.44)
(3)  FOR k = 1 TO 3 DO
(4)    FOR l = 1 TO 3 DO
(5)      Determine type of submatrix $\mathbf{P}_{kl}(\cdot)$ of $\mathbf{P}(c_1,c_2,a_1,a_2)$ according to Eq. (4.18)
(6)      FOR EACH kernel element $p_{ij}(\cdot)$ of matrix $\mathbf{P}_{kl}(\cdot)$ DO
(7)        Check kernel element $p_{ij}(\cdot)$ to independence of new clock readings $a_1$ and $a_2$ according to
           Eqs. (4.45) and (4.46)
(8)        Check kernel element $p_{ij}(\cdot)$ to independence of old clock readings $c_1$ and $c_2$ according to
           Theorem 4.11 and Eqs. (4.31) and (4.32)
(9)        FOR EACH ordering of the remaining clock readings in $p_{ij}(\cdot)$ DO
(10)         Compute kernel element $p_{ij}(\cdot)$ for every discretization step of the remaining clock
             readings by summation of transient state probabilities $P[s_i \xrightarrow[m\Delta]{exp^*} s_j]$ according to
             Theorem 4.9 with independent clock readings equal to zero
(11)       OD
(12)     OD
(13)   OD
(14) OD
```

Figure 4.8. Algorithmic description for computation of transition kernel $\mathbf{P}(c_1,c_2,a_1,a_2)$

A prerequisite for the detection of constant kernel elements according to Theorem 4.11 is the computation of the sets $V_{pre}(s_i,s_j,e_{l(i)})$, $V_{post}(s_i,s_j,e_{l(i)})$, $V_{pre}(s_i,s_j,e_{m(i)})$ and $V_{post}(s_i,s_j,e_{m(i)})$ in step (2) of Figure 4.8. These sets of states can be easily derived from the sets SMC_i and SRS_j and the mapping f_m introduced in Section 4.3.3 and Section 4.3.4, respectively. For $s_i \in S_{det1}$ we get:

$$V_{pre}(s_i,s_j,e_{l(i)}) = SMC_i \cap f_{l(i)}^{-1}\left(SRS_j\right) \tag{4.39}$$

$$V_{post}(s_i,s_j,e_{l(i)}) = f_{l(i)}\left(SMC_i\right) \cap SRS_j \tag{4.40}$$

For $s_i \in S_{det2}$ we must distinguish the two possible orderings of old clock readings c_1 and c_2, i.e., for $c_1 < c_2$ we get:

$$V_{pre}(s_i,s_j,e_{l(i)}) = SMC_i \cap f_{l(i)}^{-1}\left(SRS_{f_{m(i)}^{-1}(SRS_j)}\right) \tag{4.41}$$

$$V_{post}(s_i,s_j,e_{l(i)}) = f_{l(i)}\left(SMC_i\right) \cap SRS_{f_{m(i)}^{-1}(SRS_j)} \tag{4.42}$$

$$V_{pre}(s_i,s_j,e_{m(i)}) = SMC_{f_{l(i)}(SMC_i)} \cap f_{m(i)}^{-1}\left(SRS_j\right) \tag{4.43}$$

$$V_{post}(s_i,s_j,e_{m(i)}) = f_{m(i)}\left(SMC_{f_{l(i)}(SMC_i)}\right) \cap SRS_j \tag{4.44}$$

For $c_2 < c_1$ the right sides of Eqs. (4.41) and (4.42) must be exchanged with the corresponding right sides of Eqs.(4.43) and (4.44).

Steps (3) to (14) of Figure 4.8 show the main part of the algorithm for generating the transition kernel. In general, each submatrix $\mathbf{P}_{kl}(\cdot)$ of the transition kernel $\mathbf{P}(c_1,c_2,a_1,a_2)$ depends on different clock readings c_1, c_2, a_1, a_2 (see Eq. (4.18)). In order to reduce the computational effort of a kernel element $p_{ij}(\cdot)$ the independence of clock readings must be checked in advance (see steps (7) and (8)). As discussed in Section 4.3.4, a kernel element $p_{ij}(\cdot)$ is independent of new clock readings a_1 and a_2 if clock readings of new deterministic events at time (n+1)D do not depend on the occurrence of exponential events in [nD,(n+1)D). This implies that deterministic events enabled in states s_i and s_j cannot get canceled. Furthermore, the following conditions must hold:

$$V_{post}(s_i,s_j,e_{l(i)}) \subset S_{det1} \qquad ,\text{for } s_i \in S_{det1} \tag{4.45}$$

$$V_{post}(s_i,s_j,e_{l(i)}) \cup V_{post}(s_i,s_j,e_{m(i)}) \subset S_{det2} \qquad ,\text{for } s_i \in S_{det2} \tag{4.46}$$

The independence of clock readings c_1 and c_2 can be checked according to Theorem 4.11 and Eqs. (4.31) and (4.32), i.e., whether $G_{pre}(s_i,s_j,e_{l(i)})$ and $G_{post}(s_i,s_j,e_{l(i)})$ are isomorphic with isomorphism $f_{l(i)}$ and/or $G_{pre}(s_i,s_j,e_{m(i)})$ and $G_{post}(s_i,s_j,e_{m(i)})$ are isomorphic with isomorphism $f_{m(i)}$, respectively. Recall that we do not have to find an isomorphism; we only have to check whether the given mapping $f_{l(i)}$ and $f_{m(i)}$ is an isomorphism. Step (10) of Figure 4.8 shows the computation of a kernel element for one ordering of the remaining clock readings by summing up transient state probabilities according to Theorem 4.9. As an example we show the computation of a kernel element $p_{ij}(c_1,c_2,a_1,a_2)$ that was detected to be constant in steps (7) and (8). Let S' be the set of states that can be reached due to the (immediate) occurrence of both deterministic events in state s_i:

$$S' = \left\{ z' \in S_{det2} \middle| s_i \xrightarrow{e_{l(i)}} z \xrightarrow{e_{m(i)}} z' \right\} \tag{4.47}$$

Then, the constant kernel element p_{ij} can be computed by

$$p_{ij} = \sum_{z' \in S'} P\left[z' \xrightarrow[D]{exp^*} s_j \right] \tag{4.48}$$

which simply constitutes a sum of transient state probabilities already computed in step (1) of Figure 4.8. Note if next state probabilities $p(s_i,z,e_{l(i)})$ and $p(z,z',e_{m(i)})$ are equal to one then the set S' contains only one state, i.e., S' = {z'}.

After the numerical computation of the transition kernel the system of integral equations (4.36) to (4.38) must be solved numerically. The discretization $a_1 = 0, \Delta, 2\Delta,.., M\Delta$ and $a_2 = 0, \Delta, 2\Delta,.., M\Delta$ leads to a two-dimensional grid. Subsequently, one-dimensional and two-dimensional integrals are transformed using appropriate quadrature and cubature rules, respectively [Eng80]. Note that for numerical integration only rules, which allow an equidistant discretization, can be applied since, e.g., for the computation of $\pi_{det1}^{(n+1)}\left((m+1)\Delta\right)$ the values $\mathbf{y}^{(n)}\left(\Delta\right),...,\mathbf{y}^{(n)}\left(m\Delta\right)$ must have been computed previously. For one-dimensional integrals we applied Newton-Cotes quadrature formulas. Newton-Cotes formulas are an extremely useful and straightforward family of numerical integration techniques with equidistant discretization steps. The weights for the approximation formulas are derived by integrating the (uniquely determined) Lagrange interpolating polynomial in the discretization points. One-dimensional integrals are approximated by

$$\int_0^{M\Delta} f(c)dc \approx \Delta \cdot \sum_{m=0}^{M} w_{M+1,m} \cdot f(m\Delta) \tag{4.49}$$

with pre-calculated weights

$$w_{M,m} = \frac{(-1)^{M-m}}{m!(M-m)!} \int_0^M t(t-1)\cdots(t-m+1)(t-m-1)\cdots(t-M)dt \tag{4.50}$$

Two-dimensional rectangular integrals are derived by applying the product rule to the Newton-Cotes weights for one-dimensional integrals. For two-dimensional triangular integrals the Newton-Cotes weights are derived directly by integrating the interpolating polynomial over the two-dimensional triangular grid. In order to perform one step of the iteration (4.36) to (4.38) the density functions (4.35) must be computed numerically. One-dimensional numerical derivatives are computed by a five-point rule and two-dimensional derivatives by a nine-point rule. Similar to the Newton-Cotes weights, these rules are also derived from the interpolating polynomial [Gau97].

The detection of constant kernel elements as well as the detection of kernel elements that depend only on a subset of the clock readings c_1, c_2, a_1, a_2 not only simplifies the generation of the transition kernel but also increases efficiency and accuracy in the numerical solution of the system of integral equations. The reason is that two-dimensional integrals over kernel elements, which do not depend on c_1 or c_2, can be simplified to one-dimensional integrals or even solved completely. Therefore, we consider a submatrix $\mathbf{P}_{ij}(c_1,c_2,a_1,a_2)$ to be *separated* as follows:

$$\mathbf{P}_{ij}(c_1,c_2,a_1,a_2) = \mathbf{U}_{ij}(a_1,a_2) + \mathbf{V}_{ij}(c_2,a_1,a_2) + \tilde{\mathbf{V}}_{ij}(c_1,a_1,a_2) + \mathbf{W}_{ij}(c_1,c_2,a_1,a_2) \quad (4.51)$$

Note that with this notation constant kernel elements are included in submatrix $\mathbf{U}_{ij}(a_1,a_2)$. Applying the separation (4.51) to the first part of the rectangular integral of Eq. (4.38) we get the following representation:

$$\begin{aligned}&\int_{a_1}^{a_2}\int_{0}^{a_1} \frac{\partial^2 \pi_{\det 2}(c_1,c_2)}{\partial c_1 \partial c_2}\cdot \mathbf{P}_{ij}(c_1,c_2,a_1,a_2)dc_1dc_2 = \\ &\left(\pi_{\det 2}(a_1,a_2)-\pi_{\det 2}(a_1,a_1)\right)\cdot \mathbf{U}_{ij}(a_1,a_2)+\int_{a_1}^{a_2}\frac{d\pi_{\det 2}(a_1,c_2)}{dc_2}\cdot \mathbf{V}_{ij}(c_2,a_1,a_2)dc_2 \qquad (4.52) \\ &+\int_{a_1}^{a_2}\int_{0}^{a_1} \frac{\partial^2 \pi_{\det 2}(c_1,c_2)}{\partial c_1 \partial c_2}\cdot \left(\tilde{\mathbf{V}}_{ij}(c_1,a_1,a_2)+\mathbf{W}_{ij}(c_1,c_2,a_1,a_2)\right)dc_1dc_2\end{aligned}$$

Other rectangular and triangular integrals as well as the one-dimensional integrals can be transformed in a similar way to increase accuracy of the numerical computation.

Given the submatrices of the transition kernel $\mathbf{P}(c_1,c_2,a_1,a_2)$ of (4.18), the initial distribution of the GSSMC at each mesh point (kΔ,lΔ) with $1 \le k,l \le M$, and the mission time $t = n_0D$, the main steps of the Picard iteration for transient analysis are summarized in Figure 4.9. For steady-state analysis the Picard iteration must be performed until a predefined accuracy of the

```
FOR n = 0 TO $n_0$ DO
    FOR k = 0 TO M DO
        Compute the vector of derivatives $\mathbf{y}^{(n)}(k\Delta)$ according to Eq. (4.35)
        FOR l = 0 TO M DO
            Compute the vector of derivatives $\mathbf{z}^{(n)}(k\Delta,l\Delta)$ according to Eq. (4.35)
        OD
    OD
    Compute the vector of state probabilities $\pi_{\exp}^{(n+1)}$ according to Eq. (4.36)
    FOR k = 1 TO M DO
        Compute the vector of state probabilities $\pi_{\det 1}^{(n+1)}(k\Delta)$ according to Eq. (4.37)
        FOR l = 1 TO M DO
            Compute the vector of state probabilities $\pi_{\det 2}^{(n+1)}(k\Delta,l\Delta)$ according to Eq. (4.38)
        OD
    OD
OD
```

Figure 4.9. Main steps of the Picard iteration for numerical analysis of GSMPs

solution vector is achieved. Alternatively, the DQ-method can be applied for steady-state solution. In this case, using numerical integration and differentiation, the system of integral equations (4.36) to (4.38) is transformed into a linear system of equations. For the solution of this large but very sparse system a GMRES equation solver is applied [Ste94].

The first prototype of the algorithm for generating the transition kernel and the solution of the system of integral equations was implemented in two diploma theses written at the chair of Computer Systems and Performance Evaluation at the University of Dortmund in the year 2000. In a first diploma thesis, we realized an efficient implementation of the randomization technique that identifies special structures among the SMCs for fast computation of transient state probabilities [Wal00]. Furthermore, we implemented a prototype of the results of Theorem 4.9 using a matrix notation for summing up the transient state probabilities. Using such matrices avoids a costly re-computation of proportionate jump probabilities for suffixes of paths that have already been computed. In a second diploma thesis, we implemented a prototype of the algorithms for numerical transient and steady-state solution of the system of integral equations [Loh00]. We derived the Newton-Cotes weights for one-dimensional and two-dimensional integrals as well as the weights of the five-point and nine-point rule for numerical differentiation. The prototype implementation of the GSMP analysis software was finalized in 2002 and 2003 with many corrections and a rigorous test of the implemented software. Furthermore, the GSMP software was included into the new release of DSPNexpress, DSPNexpress-NG. An online demonstration of the GSMP analysis algorithm is provided on the Web [DSPN].

4.6 Application Examples

To illustrate the impact of the methodological results of the previous sections, two application examples of high interest for communication network performance analysis are considered. The first example constitutes an MMPP/D/2/K queueing system with a two-state Markov-modulated Poisson process representing customer arrivals. The second example is a model representing a single cell in a cellular mobile communication network taking into consideration new call arrivals as well as handover calls. We present curves for relevant QoS measures for the application examples that are validated by extensive two-day simulation runs. Furthermore, results concerning the performance of the GSMP analysis method applied to these models are presented. In particular, the amount of time and space needed for generating the transition kernel and solving the system of integral equations is depicted. The experiments have been performed on a PC workstation with a 2.3 GHz processor and 2 GByte main memory running the operating system Linux with kernel version 2.4.18. For the performance tests the user CPU time has been measured with the system call `times`.

4.6.1 Performance Results for the MMPP/D/2/K Queue

The first example constitutes an MMPP/D/2/K queueing system [FM93] comprising two identical servers with constant service time D = 1.0 seconds, as already introduced in Section 3.1. Arrivals occur according to a Poisson process which is controlled by an irreducible CTMC with two states, i.e., N = 1, representing bursty and non-bursty mode of arrivals of customers. The duration of bursty mode and normal mode, i.e., non-bursty mode, is $1/\alpha_{\text{bursty}} = 0.5$ seconds and $1/\alpha_{\text{normal}} = 100$ seconds, respectively. The arrival rates λ_{bursty} and λ_{normal} are determined from the average arrival rate $\lambda_{\text{avg}} = 0.5$ and the burstfactor B, which represents the burstiness of the arrival process:

$$\lambda_{\text{normal}} = \lambda_{\text{avg}} \cdot \frac{\alpha_{\text{bursty}} + \alpha_{\text{normal}}}{\alpha_{\text{bursty}} + B \cdot \alpha_{\text{normal}}} \tag{4.53}$$

$$\lambda_{\text{bursty}} = B \cdot \lambda_{\text{normal}} \tag{4.54}$$

Figure 4.10 presents two experiments concerning the loss probability of customers. Numerical results computed with M = 12 discretization steps of kernel elements employed in each dimension are shown in solid lines. In order to verify the numerical results, lower and upper bounds of confidence intervals from simulation runs with one billion customer arrivals and confidence level 99% are shown in dashed lines. Note that these simulations take about two-days for one experiment comprising 30 single simulation runs (i.e. one curve of Figure 4.10 with a fixed value of K and varying burstfactor) to obtain the desired confidence level. The curves clearly indicate that numerical results are within the confidence intervals. Furthermore, numerical results with very low probability, i.e., 10^{-12}, can be computed whereas the accuracy of simulation results is bounded by 10^{-7}. The left side of Figure 4.10 shows the loss probability for varying burstfactor and different queue capacities K. Indeed, for non-bursty traffic, i.e., B = 1, no customers are lost anyway, whereas the loss probability increases rapidly for larger burstfactors. Recall that in every experiment the average arrival rate is the same, i.e., $\lambda_{\text{avg}} = 0.5$. Thus, the figure inherently shows the influence of burstiness on queueing behavior, which is an important topic in modeling IP networks [PF95]. The right

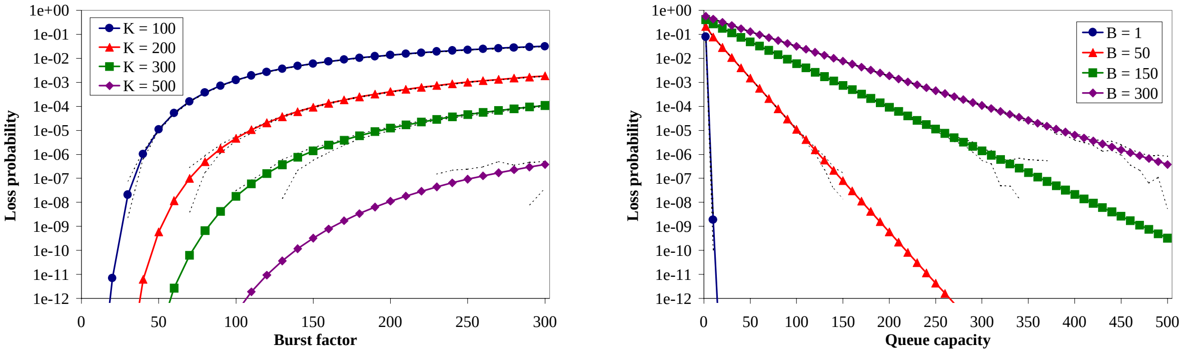


Figure 4.10. Loss probability for different queue capacities and burstfactors

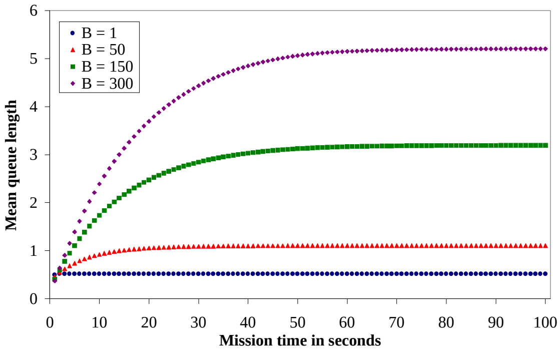


Figure 4.11. Transient development of mean queue length

side of Figure 4.10 shows the loss probability for varying queue capacity and given burstfactor B. This experiment exactly shows what system designers are interested in, i.e., what queue capacity is needed for a given average traffic with burstiness B in order to keep the loss probability below a certain threshold. As an example, for λ_{avg} = 0.5 and B = 150 the queue must consist of at least 310 buffer places in order to keep the loss probability below 10^{-6}.

To provide an example of the transient development of a QoS measure Figure 4.11 shows the mean queue length for K = 100 at different time steps. The initial distribution is set such that with probability 1.0 no customers reside in the system and that the arrival process has rate λ_{bursty} at time t = 0. Note that the mean queue length at time n·D is computed from the probability distribution after the n-th iteration of the Picard algorithm. Thus the Picard-algorithm produces the transient development as an intermediate result. The mean queue length presented in the figure underlies the effect of bursty traffic on queueing behavior: For B = 300 on average only about 5.2 buffer places are filled up, although the loss probability is considerably high, i.e., about 10^{-2} as shown in Figure 4.10.

The next experiment considers the performance of kernel generation. The left side of Figure 4.12 shows the CPU time needed to generate the transition kernel, i.e., transient analysis of SMCs (randomization) plus summation of transient state probabilities. The right side of Figure 4.12 shows the corresponding memory requirements of the transition kernel.

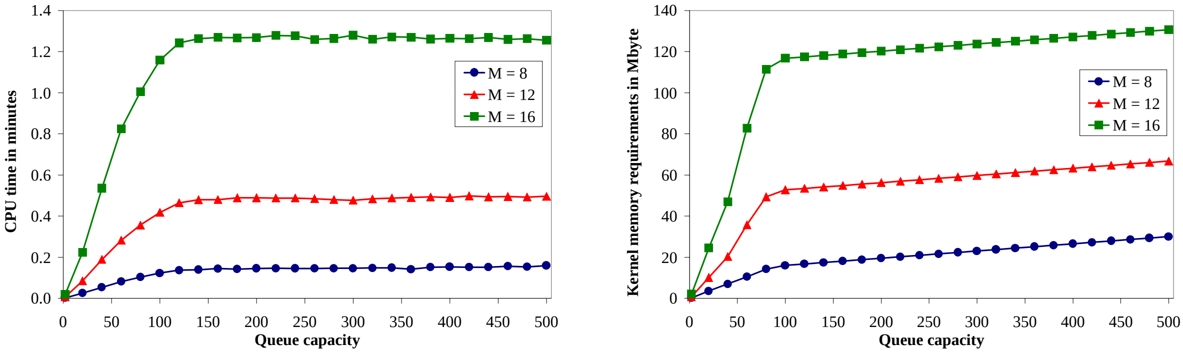


Figure 4.12. Generation of transition kernel: CPU time and memory requirements

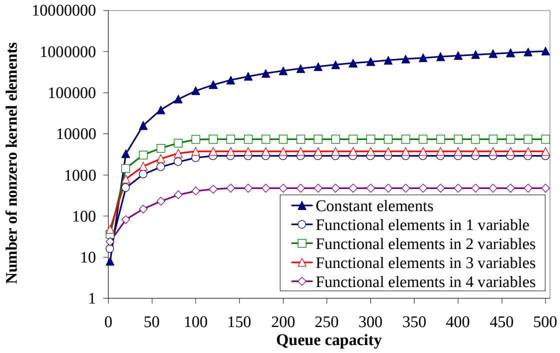


Figure 4.13. Classification of elements of the transition kernel

The experiments are presented for varying model size and different numbers of discretization steps M. The remaining parameters of the model are kept fixed, i.e., $\lambda_{avg} = 0.5$ and B = 150. Note that model size depends on the queue capacity. That is, for queue capacity K the model consists of 2·(K+2) states. As expected, the computation as well as the memory requirement is much more time and space consuming for increasing number of discretization steps M. Increasing the model size results in a different observation. For queue capacities K = 2 to K = 120 a significant increase in time and space is observed. For larger queue capacities only a very small further increase in time and space can be seen. This is due to the fact that for increasing queue capacity more than 98% of nonzero kernel elements are constant as shown in Figure 4.13. Furthermore, the kernel generation employs a dynamic sparsing method by setting both constant and functional kernel elements smaller than a given threshold $\varepsilon = 10^{-16}$ to zero. This results in an almost linear growth of the nonzero kernel elements and stagnation in number of functional kernel elements for this class of GSMP models.

The following experiment shows the CPU time required for solving the system of integral equations. The left side of Figure 4.14 presents results for 100 iterations of the Picard algorithm and the right side of Figure 4.14 shows the time requirements for the DQ-method with 100 iterations of GMRES. Note that time requirements of the DQ-method includes the generation of the system of linear equations as well as the GMRES solution time. For the GMRES algorithm, we consider a maximal Krylov space of dimension 20, i.e., the algorithm

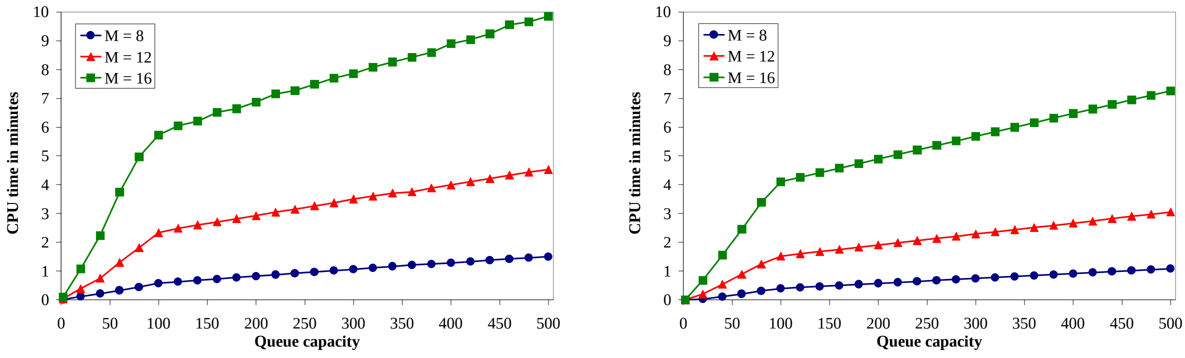


Figure 4.14. Picard- and DQ-method for solving system of integral equations

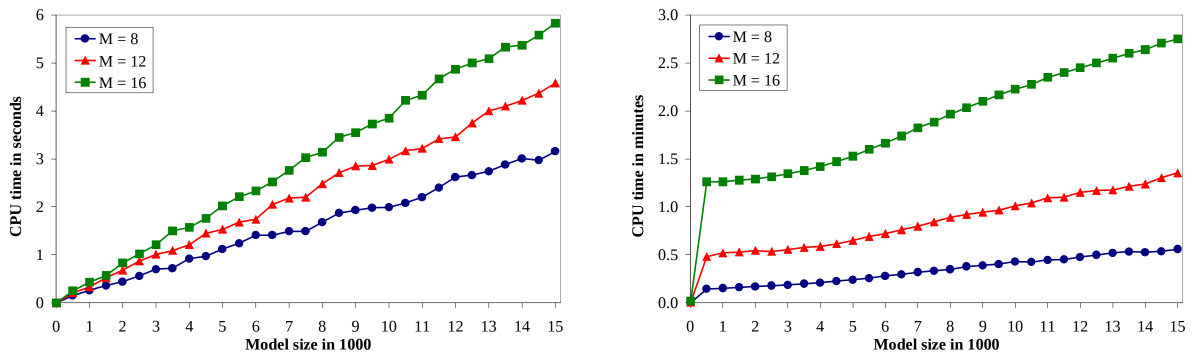


Figure 4.15. Generation of transition kernel: CPU time required for transient analysis of SMCs (left) and summation of transient state probabilities (right)

restarts after 20 iterations. As expected, the curves have a similar shape as the curves for kernel generation of Figure 4.12 since the time consumed by both algorithms depends on the number of nonzero kernel elements. Comparing the Picard iteration with the DQ-method we conclude that the DQ-method is slightly faster. This is due to the fact that no additional computation is required after each iteration as in the Picard-algorithm where the derivatives $\mathbf{y}(c_1)$ and $\mathbf{z}(c_1,c_2)$ and the numerical integrals must be computed after each iteration. The DQ-method requires this computation only one time before starting the GMRES iteration, since the derivatives and integrals are stored in the large system of equations. Obviously, the disadvantage of this pre-computation is the higher memory requirements, e.g., for K = 500 and M = 16 about 600 MBytes of memory are required to store the system of equations.

To show the scalability of the solution algorithm we consider the CPU time required for models with up to 15,000 states, i.e., K = 7,498. The left side of Figure 4.15 shows the time needed for transient analysis of the SMCs and the right side shows the time spent summing up the transient probabilities to compute constant and functional kernel elements. As in Figure 4.12, we observe that the CPU time grows nearly linear for increasing model size. Furthermore, for models with up to 15,000 states the generation of the transition kernel with M = 8 discretization steps requires less than 30 seconds of CPU time. This is due to the sparseness of the transition kernel and the exploitation of constant kernel elements according to Section 4.3.4. Thus, for such GSMPs underlying finite-capacity queueing systems with two deterministic servers, the exploitation of constant kernel elements is key for their efficient transient and steady-state analysis.

4.6.2 Performance Results for a Single Cell Model

The second application example constitutes a model of a single cell in a cellular network with terminal mobility. The overall number of radio channels available in the cell is denoted by N. New call requests and handover requests arrive according to a Poisson processes with arrival rates λ. We assume a portion of 2/3 of the incoming call requests to be new calls and 1/3 to be handover calls. The amount of time a mobile terminal with an ongoing call remains within the cell is called *dwell time*. If the call is still active after the dwell time, a handover toward an adjacent cell takes place. The *call duration* is defined as the amount of time that the call will be active, assuming it completes without being forced to terminate due to handover failure. We assume the dwell time and call duration to be exponentially distributed random variables with means $1/\mu_h$ = 60 seconds and $1/\mu$ = 120 seconds, respectively.

New call requests and handover requests can be accepted in the cell only if at least one free channel is available. In order to prioritize handover calls over new calls a certain number of guard channels, denoted by r ($0 \leq r \leq N$), is exclusively reserved for handover calls. That is, a new call request is accepted only if at least r+1 free channels are available in the cell. If a new call request is accepted in the cell an attach procedure that identifies the new terminal in the cell introduces a deterministic overhead. Similar, a handover call from an adjacent cell introduces a significant overhead due to the transfer of the call from one cell to another. This overhead is modeled by a deterministic delay with duration D = 200 ms. Since the attachment procedure for new calls is executed sequentially a queue with accepted calls must be considered. The same holds for accepted handover calls. We assume accepted new calls to be queued in the *new-calls queue* and accepted handovers to be queued in the *handover-queue*. Each of these queues has finite capacity K.

Figures 4.16 to 4.18 show a representation of the single cell model with UML state diagram notation as introduced in Chapter 3. In particular, the model is represented by four different concurrent states, *CallRequest*, *CallQueues-NewCalls*, *CallQueues-HandoverCalls*, and *CallService*. These superstates communicate with each other through the variables *FreeChannels*, *OngoingCalls*, *HewCallsQueue*, and *HandoverQueue*. The exponential event

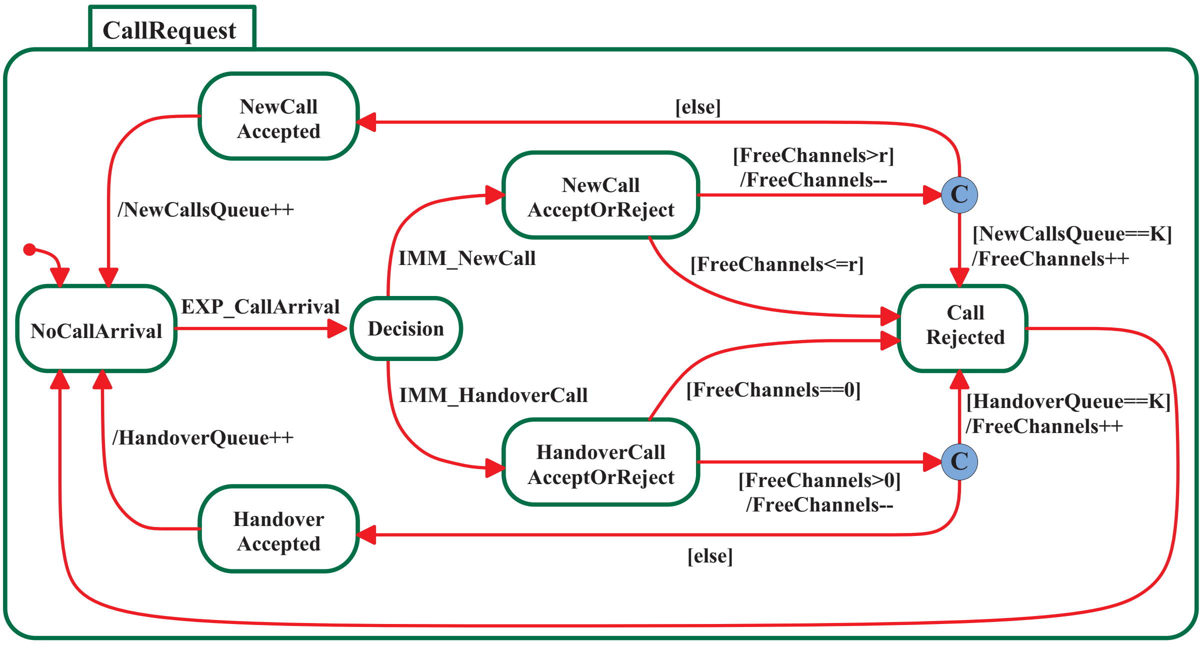


Figure 4.16. Single Cell Model: UML state diagram of call request process

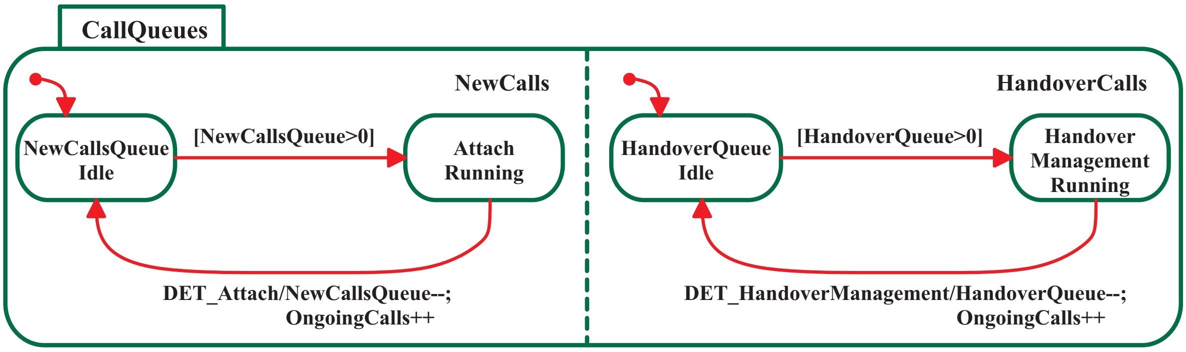

CallQueues
NewCalls
NewCallsQueue Idle
[NewCallsQueue>0]
Attach Running
DET_Attach/NewCallsQueue--; OngoingCalls++
HandoverCalls
HandoverQueue Idle
[HandoverQueue>0]
Handover Management Running
DET_HandoverManagement/HandoverQueue--; OngoingCalls++

Figure 4.17. Single Cell Model: UML state diagram of management processes

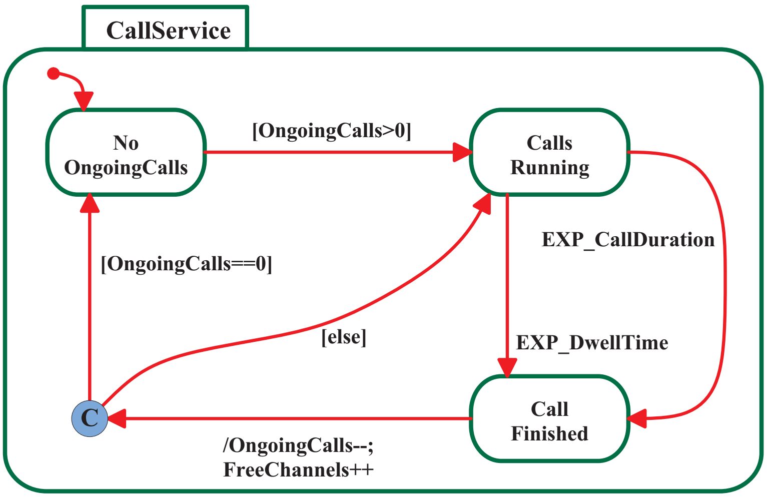


Figure 4.18. Single Cell Model: UML state diagram of call service process

CallArrival has rate λ and the exponential events *DwellTime* and *CallDuration* have rates depending on the number of ongoing calls, i.e., *OngoingCalls*$\cdot\mu_h$ and *OngoingCalls*$\cdot\mu$, respectively. The delay of the (concurrent) deterministic events *Attach* and *HandoverManagement* is D. Immediate events *NewCall* and *HandoverCall* have same priority and associated weights 2/3 and 1/3, respectively.

Figure 4.19 presents two experiments concerning the probability of rejecting a call request due to queue overflow even if sufficient free channels are available. The curves are presented for varying call arrival rate and different queue capacities K. The overall number of channels is set to N = 20. Furthermore, r = 2 channels are exclusively reserved for handover calls. As for the MMPP/D/2/K application example, numerical results with M = 8 discretization steps are shown in solid lines and results from a simulation run with one billion call arrivals and confidence level 99% are shown in dashed lines. The curves clearly indicate that numerical results are within the confidence intervals. The intention of the experiments is to determine a sufficient queue capacity K such that the probability of rejecting a call request even if sufficient free channels are available is insignificant. Since for K = 3 the queue overflow probability is below 10^{-5} for the entire call arrival spectrum, we conclude that K = 3 is an appropriate value for the queue capacity for the considered setting.

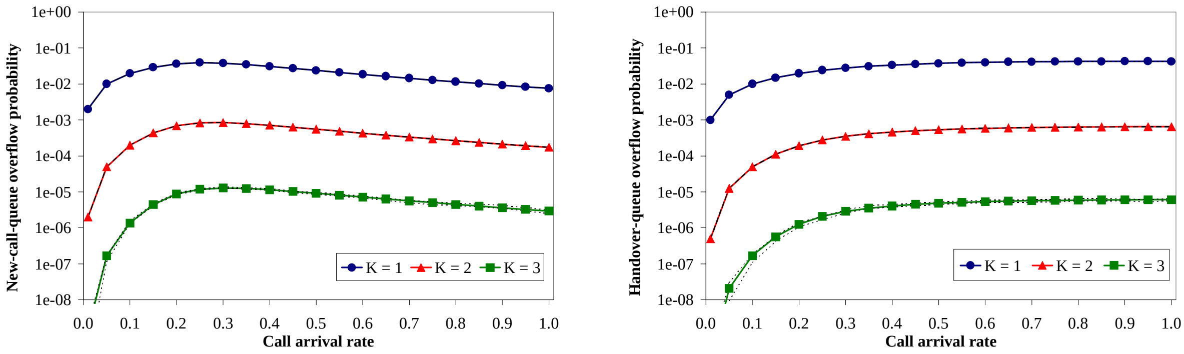


Figure 4.19. Queue overflow probability for new-calls-queue and handover-queue

The next experiment considers the impact of exclusively reserving a certain number of channels for handover calls. According to the result of the previous experiments, the queue capacity of the new-calls queue and the handover queue is assumed to be K = 3. Figure 4.20 shows the new call blocking probability and the handover failure probability for varying call arrival rate and different numbers of exclusively reserved handover channels r. The presented curves clearly indicate that the increase in new call blocking probability due to the reservation of handover channels is negligible compared to improvement of the handover failure probability. As an example, for a call arrival rate of 0.2 call requests per second the blocking probability of new calls is increased from 1% to 7% whereas the handover failure probability decreases from 1% to 0.01% when reserving r = 0 and r = 4 channels, respectively.

Figure 4.21 presents an example of the transient evolution of the number of ongoing calls obtained with the Picard iteration. A fixed call arrival rate of 0.15 call requests per second is considered. The experiments are conducted for different initial distributions of the state probability vector, that is, no ongoing calls at t = 0, a uniformly distributed number of ongoing calls at t = 0, and N = 20 ongoing calls at t = 0 are considered. For increasing mission time the curves show the time-dependent average number of ongoing calls as it converges to an average number of 6.2 ongoing calls in steady-state.

The next experiments consider the performance of the GSMP analysis methodology for varying model sizes. The number of states of the state transition graph of the single cell model

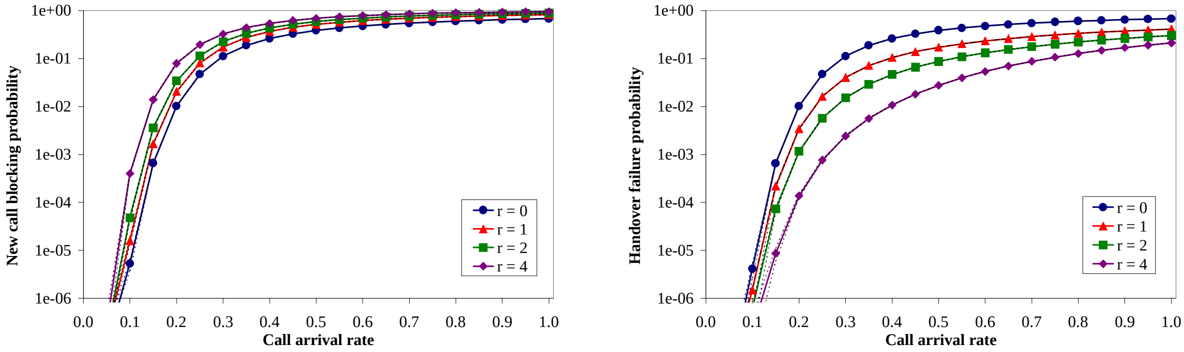


Figure 4.20. Prioritization of handover calls

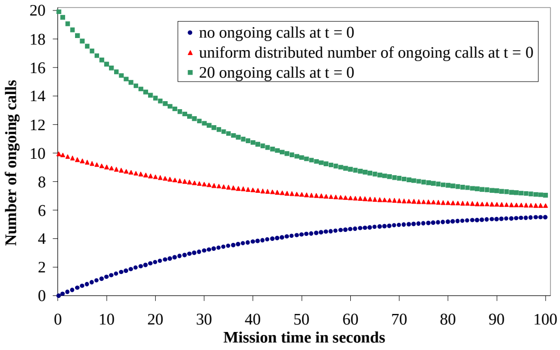


Figure 4.21. Transient evolution of number ongoing calls for different initial state probability distributions

depends on the choice of N, K, and r. The number of states can be computed by the following formula:

$$\sum_{i=0}^{N}\sum_{j=0}^{\min(N-i,K)}\left(1+\min(N-i-j,N-r,K)\right) \tag{4.55}$$

For N sufficiently large compared to K and r, i.e., N ≥ 2·K and N ≥ K+r, Eq. (4.55) can be simplified:

$$\left(N-K+1\right)\cdot\left(K+1\right)^2 \tag{4.56}$$

Figure 4.22 shows the CPU time needed to generate the transition kernel and the corresponding memory requirements versus increasing model size in terms of the overall number of channels N. It can be seen that more time is needed for kernel generation as in the example for the MMPP/D/2/K queueing system. The reason is the following: Figure 4.23 indicates that for this class of GSMPs the dominating number of kernel elements depends on two and three clock readings and almost no kernel elements are constant. Noting the high percentage of functional kernel elements, the single cell model constitutes a worst-case example. Nevertheless, the generation of the transition kernel for the considered setting, i.e., N = 20, K = 3, and r = 2, requires less than 1.3 minutes of CPU time for M = 8 discretization steps.

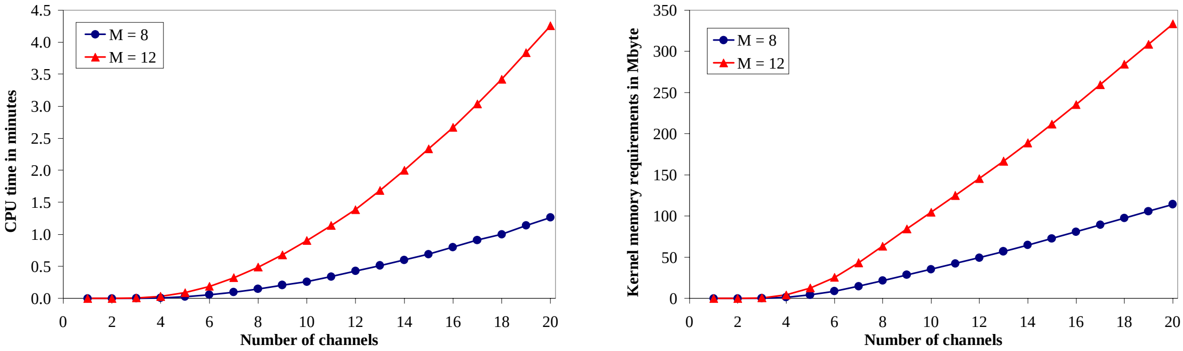


Figure 4.22. Generation of transition kernel: CPU time and memory requirements

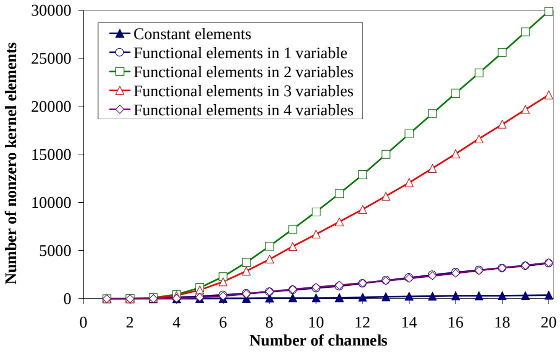


Figure 4.23. Classification of elements of the transition kernel

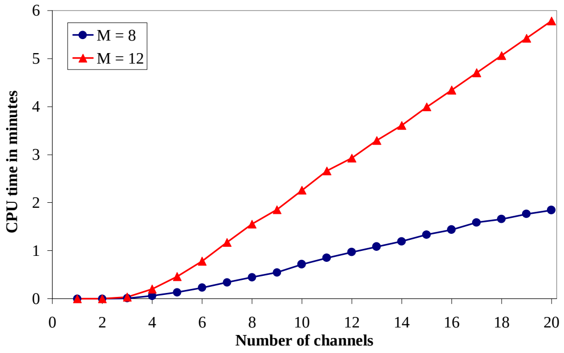


Figure 4.24. Picard iteration for solving system of integral equations

Figure 4.24 presents results for 100 iterations of the Picard algorithm. According to the high number of functional kernel elements the time consumed by the Picard iteration is also considerably high. For computation of the steady-state probability vector for the single cell model an average of 500 Picard iterations is required. Thus, the complete GSMP solution process requires approximately 10 minutes of CPU time for calculating steady-state probabilities. Nevertheless, comparing this solution time with a simulation run which requires more than two hours to achieve appropriate accuracy the research contribution and the importance of the GSMP methodology is pointed out.

4.7 Discussion and further Comments on the GSMP Analysis Algorithm

This section gives further insight into the proposed GSMP analysis algorithm and comments on the experience made by the author during the time the solution algorithm was developed and the application examples were analyzed. In particular, the limits in terms of state space and numerical accuracy are discussed.

The limits in terms of state space for the GSMP analysis algorithm are mainly determined by the memory requirements to store the transition kernel. In fact, the memory requirements are inherently determined by the structure of the model, i.e., how many nonzero kernel elements exist and how many kernel elements are constant or depend on one, two, three, or four clock readings. Let η_i, i = 0,1,2,3,4, be the number of nonzero kernel elements that depend on i clock readings. Applying M discretization steps in each dimension, the space complexity is given by

$$O\left(\eta_0 + \eta_1 \cdot M + \eta_2 \cdot M^2 + \eta_3 \cdot M^3 + \eta_4 \cdot M^4\right) \quad (4.57)$$

If deterministic events are not concurrently enabled, the space complexity reduces to

$$O\left(\eta_0 + \eta_1 \cdot M + \eta_2 \cdot M^2\right) \quad (4.58)$$

Note that two types of zero kernel elements can be distinguished. First, a kernel element $p_{ij}(\cdot)$ can be zero because there is simply no feasible path of state transitions from state s_i to state s_j. An example for such a kernel element is a state transition in the M/D/2/K (or MMPP/D/2/K) queueing system from a state with $k > 2$ customers in the queue to a state with less than k-2 customers in the queue (see also Example 4.12). The second type of zero kernel elements results from applying a dynamic sparsing method during kernel generation. That is, all kernel elements smaller than a given threshold $\varepsilon = 10^{-16}$ are set to zero. As an example, consider again the kernel statistic for the MMPP/D/2/K queue presented in Figure 4.13. Indeed, for increasing model size (i.e., increasing queue capacity) from K = 150 to K = 500, the number of functional kernel elements stagnates. Without dynamic sparsing the number of functional kernel elements would grow linearly (even if constant kernel elements were detected according to Theroem 4.11) as can be concluded from the detailed discussion of the kernel structure for an M/D/2/K queueing system (see Example 4.12).

Can we expect to reduce the number of nonzero kernel elements by dynamic sparsing for almost all models? When trying to answer this question we should keep in mind that we consider models, which can be represented in a compact notation, e.g., as DSPN or UML diagram. Increasing the state space of the model is usually done by simply varying a model parameter (e.g., the queueing capacity K) and not by adding additional structural information to the model. Thus, increasing this model parameter results in a somewhat regular development of the corresponding state transition graph with repeating sub-structures and state transitions between these sub-structures. Obviously, there is a limit n_0, which is depending on the model structure and model parameter setting, such that the probability of a transition of the GSMP traversing at least n_0 of these sub-structures in a given time t is less than ε. Thus, the corresponding state transition probability and also the corresponding kernel element are set to zero by the dynamic sparsing method. For the M/D/2/K queueing system the sub-structure mentioned above comprises only one state (representing a certain number of customers queued) and the regular development of the state transition graph for varying queue capacity results in a sequence structure. The author believes that dynamic sparsing will play an important role for almost all DES derived from DSPNs or UML diagrams.

Considering Eqs. (4.57) and (4.58), two factors limiting the size of the state space are observed. First, the number of functional nonzero kernel elements and second, the number of discretization steps M. Considering the number of discretization steps, it can be concluded from the experiments presented in Section 4.6 that M = 12 or even M = 8 discretization steps are sufficient for the computation of performance measures with reasonable accuracy. The number of functional kernel elements depends on the structure of the model, the cardinality of S_{det1} and S_{det2}, and, more important, whether constant kernel elements are found according to Theorem 4.11. For the MMPP/D/2/K queueing system the transition kernel can be generated for models with several thousand states. The corresponding memory requirements to store nonzero kernel elements strongly depends on the parameter setting of the queueing system (i.e., the number of kernel elements that are set to zero due to dynamic sparsing). Applying the same parameter setting as in Section 4.6.1 with a model size of 50,000 states (i.e., K = 24,998), M = 8 discretization steps, and burstfactor B = 1, 10, and 20 the nonzero elements of the transition kernel require about 129 MByte, 260 MByte, and 355 MByte memory, respectively. For model size of 100,000 states and B = 1 the memory requirements are 254 MByte. Note that the computation of the transition kernel requires just a few seconds of CPU time for these examples.

For a comparison of the GSMP analysis algorithm with an approach based on supplementary variables, an MMPP/D/1/K queueing system similar to that studied in [GH99] is considered. Note that this model does not contain concurrently enabled deterministic events. Applying the GSMP analysis algorithm, a model with 10,000 states, M = 8 discretization steps, and burstfactor B = 50, requires about 18 MByte memory to store the transition kernel. The CPU time required for kernel generation and transient solution is 2 seconds and 13 seconds, respectively. This is a clear advantage compared to the supplementary variables approach, which requires for an MMPP/D/1/K queue with failure and repair and similar parameter setting about 100 hours of CPU time for transient solution (see Figure 8 in [GH99]). It is concluded, that key for an efficient analysis algorithm is the observation that kernel elements are piecewise continuous functions and, more important, the detection of constant kernel elements, which has not been considered in previous related approaches.

Next, the numerical accuracy of the GSMP analysis algorithm is considered for both the kernel generation and the solution of the system of integral equations. First of all, we take into account the generation of the transition kernel of the GSSMC. Recall that elements of the transition kernel are sums of appropriate transient state transient probabilities of subordinated Markov chains, which are computed with the randomization technique proposed by Gross and Miller [GM84]. The randomization technique is widely applied and generally accepted to produce accurate transient probabilities of a CTMC for a given error tolerance ε. For a stable calculation of Poisson probabilities during randomization the algorithm proposed by Fox and Glynn [FG88] is applied. Furthermore, it is a known fact that the computation of a sum of positive values (i.e., the transient probabilities) is numerically stable. Thus, it can be concluded that the computation of the transition kernel can be performed with the same accuracy as the randomization when choosing an appropriate value for ε.

Recall that for the numerical computation of one- and two-dimensional integrals, quadrature and cubature methods based on Newton-Cotes weights are applied. Note that Newton-Cotes weights obtained from Lagrange interpolating polynomials with more than eight discretization steps contain negative values, which result in numerically unstable subtractions during integration. As a consequence, Newton-Cotes formulas are only applied up to degree seven (i.e., eight discretization steps). If integration with more than eight discretization steps should be applied, the region of integration is divided into several parts each considered separately. With this solution the integrals are computed with highest possible precision. A possible extension to increase accuracy could be an adaptive choice of discretization steps depending on the stiffness of the function to be integrated [Gau97]. That is, if the function to be integrated is not sufficiently smooth (i.e., stiff) the number of discretization steps is increased. Such a numerical integration with adaptive step size control could be subject for future research.

4.8 Summary

In this chapter, two theorems that provide the foundation for the effective algorithmic generation of the transition kernel of the general state space Markov chain (GSSMC) underlying a GSMP with exponential and possibly concurrent deterministic events are presented. Key contribution constitutes the derivation of an algorithmic approach how kernel elements can always be computed by appropriate summation of transient state probabilities of continuous-time Markov chains (Theorem 4.9). Furthermore, conditions on the building blocks of the GSMP under which kernel elements of its GSSMC are constant, i.e., are independent of clock readings, are derived (Theorem 4.11). Thus, for such state transitions the GSSMC behaves like a discrete-time Markov chain. Applying Theorem 4.11, it is shown that for queueing systems with Markov-modulated arrival process almost all kernel elements are constant. Comparing the solution time of the GSMP methodology with the time consumed by a simulation run of appropriate accuracy the research contribution and the importance of the GSMP methodology is pointed out.

Chapter 5

Performance Analysis of the General Packet Radio Service

This chapter presents an efficient and accurate analytical model for the radio interface of the General Packet Radio Service (GPRS) in a GSM network. The model is utilized for investigating how many packet data channels should be allocated for GPRS under a given amount of traffic in order to guarantee appropriate quality of service. The GPRS model is presented in time-enhanced UML state diagram notation and the steady-state distribution of the underlying stochastic process is computed with DSPNexpress-NG. The model represents the sharing of radio channels by circuit-switched GSM connections and packet-switched GPRS sessions under a dynamic channel allocation scheme. Section 5.1 describes the basic GPRS radio interface, which provides the technical background of the GPRS model. In Section 5.2, the GPRS model is described and its parameters are introduced. Section 5.3 presents the model realization with UML state diagrams. Comprehensive performance studies for GPRS are presented in Section 5.4. A detailed comparison of the performance of different network configurations and different percentages of GPRS users is provided.

5.1 GPRS Channel Allocation

The introduction of GPRS as an additional service in second generation GSM networks requires several modifications to the network architecture as introduced in Chapter 2. Furthermore, modifications in the channel allocation scheme are required. On the physical layer, GSM uses a combination of FDMA and TDMA for multiple access. Two frequency bands are reserved for GSM operation, one for transmission from the mobile station to the BTS (uplink) and one for transmission from the BTS to the mobile station (downlink). Each of these bands is divided into 124 single carrier channels of 200 kHz width. A certain number of these frequency channels is allocated to a BTS, i.e., to a cell. Each of the 200 kHz frequency channels is divided into eight time slots that form a TDMA frame. A time slot lasts 0.577 ms and carries 114 bits of information. The recurrence of one particular time slot defines a physical channel. GSM channels are called *Traffic Channels (TCH)* and channels allocated for GPRS are called *Packet Data Channels (PDCH).*

The channel allocation in GPRS is different from the original allocation scheme of GSM. GPRS allows a single mobile station to transmit on multiple time slots of the same TDMA frame. This results in a very flexible channel allocation: one to eight time slots per TDMA frame can be allocated to one mobile station. On the other hand a time slot can be assigned temporarily to a mobile station, so that one to eight mobile stations can use one time slot. Moreover, uplink and downlink channels are allocated separately, which efficiently supports asymmetric data traffic flows, i.e., non real-time traffic like WWW browsing or FTP.

In conventional GSM, a channel is permanently allocated to a particular user during the entire call period (whether data is transmitted or not). In contrast to this, in GPRS the channels are only allocated when data packets are sent or received, and they are released after the transmission. For bursty traffic this results in a much more efficient usage of the scarce radio resource. With this principle, multiple users can share one physical channel. GPRS includes the functionality to increase or decrease the amount of radio resources allocated to GPRS on a dynamic basis. The PDCHs are taken from the common pool of all channels available in the cell. The mapping of physical channels to either packet-switched (GPRS) or circuit-switched (conventional GSM) services can be performed statically or dynamically ("capacity on-demand"), depending on the current traffic load. A load supervision procedure monitors the load of the PDCHs in the cell. According to the current demand, the number of channels allocated for GPRS can be changed. Physical channels not currently in use by conventional GSM can be allocated as PDCHs to increase the quality of service for GPRS. When there is a resource demand for services with higher priority, e.g., GSM voice calls, PDCHs can be de-allocated.

5.2 Description of the GPRS Model

Following [ACL+99], the performance model considers a single cell in an integrated GSM/GPRS network, serving circuit-switched voice and packet-switched data calls. We assume that GSM calls and GPRS calls arrive according to two mutually independent Poisson processes, with arrival rates λ_{GSM} and λ_{GPRS}, respectively. GSM calls are circuit-switched, so that one physical channel is exclusively dedicated to the corresponding mobile station. After the arrival of a GPRS call, a *GPRS session* begins. During this time, the base station schedules the radio interface (i.e., the physical channels) among different GPRS users. GPRS users receive packets according to a specified traffic model explained below. Similar as for the single cell model considered in Section 4.6.2, the amount of time that a mobile station with an ongoing call remains within the cell is called *dwell time*. If the call is still active after the dwell time, a handover toward an adjacent cell takes place. The *call duration* is defined as the amount of time that the call will be active, assuming it completes without being forced to terminate due to handover failure. As shown in [BJ99], the dwell time can be best modeled by a lognormal distribution. To keep the model analytically tractable, we assume the dwell time to be an exponentially distributed random variable with mean $1/\mu_{h,GSM}$ for GSM calls and $1/\mu_{h,GPRS}$ for GPRS sessions. The call durations are also exponentially distributed with mean values $1/\mu_{GSM}$ and $1/\mu_{GPRS}$ for GSM calls and GPRS sessions, respectively. This is a common and quite realistic assumption derived by measurements in telecommunication networks. In order to limit the amount of packet traffic in the cell we restricted the maximal number of active GPRS sessions by a value M. This provides a form of first-come first-served admission control in order to guarantee certain QoS for the GPRS users.

At this place, it should be noted that the assumption of exponentially distributed delays may be relaxed by including phase-type distributions in order to incorporate a slightly more realistic mobility process, while still allowing Markov chain analysis. However, the impact on the anticipated qualitative results and trends is expected to be insignificant, while obtaining the steady-state distribution of the ensuing higher-dimensional Markov chain would be computationally much more expensive. As a consequence of simplifications such as these, the *absolute* values of the numerical results have limited validity, which in fact is true for even the most detailed simulation studies encountered in the literature. Incorporating more realistic details into the model precludes analytical treatment yet requires more time-consuming simulations. This would cause the already intensive search for the optimal system parameters to become unacceptably slow. As is the raison d'être of virtually all analytical studies found in this field, the numerical inaccuracies are out-weighted by the enhanced qualitative insight that is generated by truly optimizing the considered mobile communication system.

To exactly model the user behavior in the cell, we consider the handover flow of active GSM calls and GPRS sessions from adjacent cells. It is impossible to specify in advance the intensity of the incoming handover flow. This is due to the fact that the handover rate out of the cell depends on the number of active customers within the cell. On the other hand, the handover rate into the cell depends on the number of customers in the neighboring cells. Thus, the iterative procedure introduced in [ACL+99] is employed for balancing the incoming and outgoing handover rates. The iteration is based on the assumption that the incoming handover rate $\lambda_{h,GSM}^{(i+1)}$ of GSM calls and $\lambda_{h,GPRS}^{(i+1)}$ of GPRS sessions at step i+1 is equal to the corresponding outgoing handover rate computed at step i.

Since in the end-to-end data path, the wireless link is typically the performance bottleneck, the model represents the radio interface of an integrated GSM/GPRS network. The functionality of the GPRS core network is not included. Because of the anticipated traffic asymmetry (most of the GPRS traffic will be WWW browsing), the model focuses on

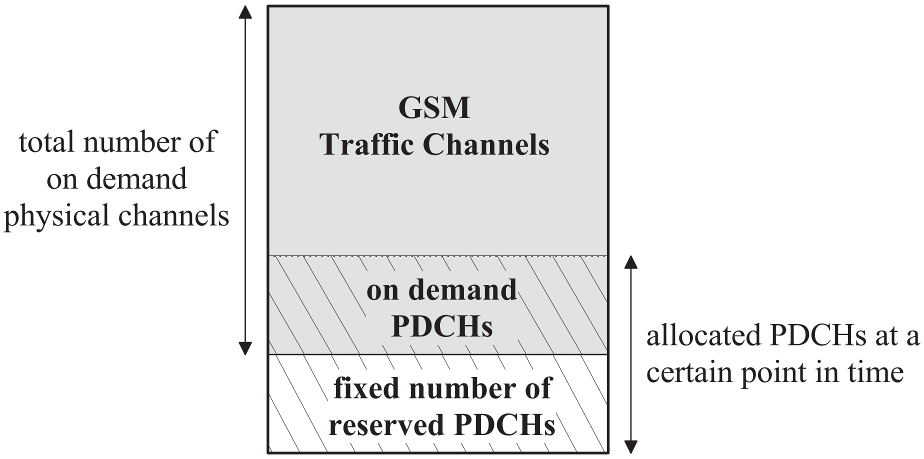

total number of on demand physical channels
GSM Traffic Channels
on demand PDCHs
fixed number of reserved PDCHs
allocated PDCHs at a certain point in time

Figure 5.1. Partitioning of physical channels among GSM and GPRS

resource contention in the downlink (i.e., the path BSC → BTS → MS) of the radio interface. The amount of uplink traffic, e.g., induced by acknowledgments, is assumed to be negligible. The arrival stream of data packets is modeled at the network layer, assuming a data packet size of 480 byte [ETSI98]. Data packets arriving at the base station are stored in a FIFO buffer with limited size of K data packets until they are transmitted on a free physical channel. Let N be the overall number of physical channels available in the cell. We assume that N_{GPRS} channels are permanently reserved as PDCHs for GPRS and the remaining $N_{GSM} = N - N_{GPRS}$ channels can be used either as GSM traffic channels or “on-demand” as PDCHs. Among the on-demand channels, GSM calls have priority. That is, on-demand channels allocated as PDCHs are immediately released, when requested by a GSM call. Figure 5.1 illustrates the partitioning scheme of physical channels in GSM traffic channels and GPRS packet data channels.

In order to describe the GPRS traffic, we adopt the model defined by the 3rd Generation Partnership Project in [ETSI98]. This model again relies on the traffic model defined by Anderlind and Zander [AZ97]. Active users within a cell execute a *packet service session,* which is an alternating sequence of *packet calls* and *reading times* (see Figure 5.2). During a

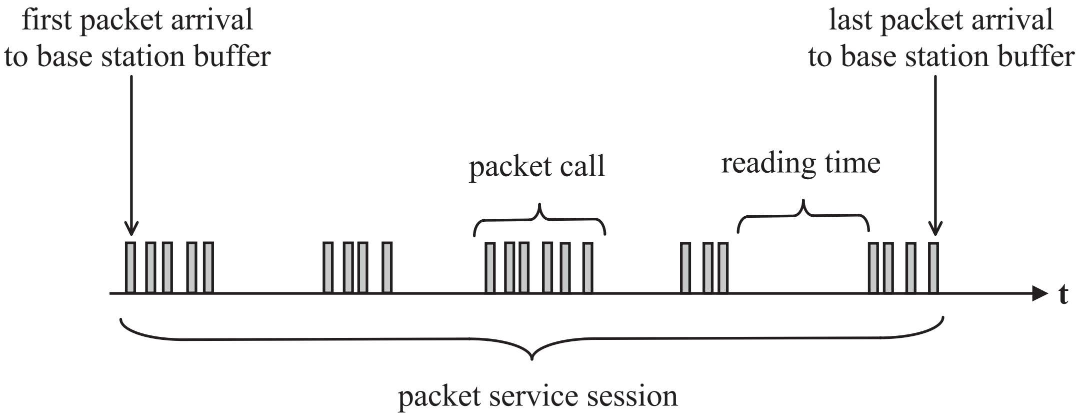

first packet arrival to base station buffer
last packet arrival to base station buffer
packet call
reading time
t
packet service session

Figure 5.2. Typical characteristic of a packet service session [ETSI98]

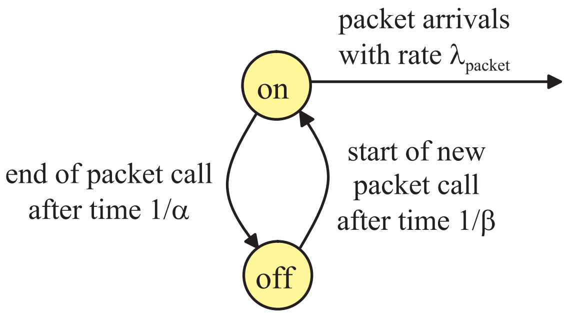


Figure 5.3. Interrupted Poisson process traffic model for one GPRS session

packet call several packets may be generated. Therefore, a packet call constitutes a bursty sequence of packets. Note that the burstiness during a packet call is a characteristic feature of packet transmissions that must be taken into account in an accurate traffic model [AZ97]. For example, if WWW browsing is considered, a packet call corresponds to the downloading of a WWW document. After the document is received at the terminal, the user is consuming a certain amount of time, i.e., the reading time, studying the information. It is also possible that the packet service session contains only one packet call. In fact this is the case for a file transfer via FTP.

According to the measurement-based results from [ETSI98], the number of packet calls within a packet session should be a geometrically distributed random variable with mean N_{pc}. The reading time between packet calls is an exponentially distributed random variable with parameter $1/D_{pc}$. Each packet call comprises a geometrically distributed number of data packets with mean N_d and the interarrival time between packets in a packet call is an exponentially distributed random variable with parameter $1/D_d$. Such a traffic model can be represented by a Markov-modulated Poisson process (MMPP, [FM93]). In particular, we consider an MMPP with two alternating states named “on” and “off”. The on-state corresponds to an active packet call of one GPRS user and the off-state represents the reading time of the GPRS user (see Figure 5.3). During on-state packets are generated by an exponentially distributed random variable with parameter $\lambda_{packet} = 1/D_d$. In off-state no packets are generated. The average on and off times are exponentially distributed random variables with parameter $\alpha = 1/(N_d \cdot D_d)$ and $\beta = 1/D_{pc}$. Thus, we consider a special case of an MMPP, i.e. an *interrupted Poisson process* (IPP) that is specified by the infinitesimal generator matrix **Q** and rate matrix Λ:

$$\mathbf{Q} = \begin{pmatrix} -\alpha & \alpha \\ \beta & -\beta \end{pmatrix} \qquad \Lambda = \begin{pmatrix} \lambda_{packet} & 0 \\ 0 & 0 \end{pmatrix} \tag{5.1}$$

The average packet service session time corresponds to the GPRS session duration defined by $1/\mu_{GPRS} = N_{pc} \cdot (D_{pc} + N_d \cdot D_d)$.

Because most of today's internet traffic (around 90%) is transported via the TCP/IP protocol, we take into account a TCP flow control mechanism in the GPRS model. In case of network congestion, the buffer of the router at the beginning of the bottleneck link, i.e., at the base station, overflows and packets get lost. TCP detects lost packets due to timer expiration or the reception of three duplicate acknowledgements from the receiver and reacts on these losses by reducing the packet sending rate of the source. As illustrated by the validation with simulation results in Section 5.4.2, the following approximate representation of a TCP flow control in the GPRS model can effectively represent the reaction of TCP sources to network congestion, i.e., buffer overflow at the base station. Therefore, in the GPRS model the sending rate of the TCP sources is reduced when the buffer occupancy exceeds a certain percentage η of the buffer size K.

To provide a reliable wireless link for data transfer a forward error correction (FEC) mechanism on the physical layer as well as an automatic repeat request (ARQ) mechanism in the radio link control (RLC) protocol on the link layer are specified for GPRS [CG97]. For FEC four different coding schemes CS-1 to CS-4 based on convolutional coding are currently defined. Coding scheme CS-1 corresponds to the coding required for a channel with high block error rate, i.e., code rate 1/2, and coding scheme CS-4 corresponds to no coding, i.e., code rate 1. In order to take into account the influence of block errors on performance measures we consider the fixed coding scheme CS-2 in the GPRS model. It allows a data transfer rate of 13.4 Kbit/sec on each PDCH [CG97].

Note that we do not consider packet losses due to interference on the wireless link. Because TCP is unaware of these losses, it could be possible that TCP will react falsely on wireless losses with congestion control, i.e., slowing down its sending rate. As shown in [Mey99], GPRS provides a sufficiently fast working ARQ mechanism, which allows typically several retransmissions before TCP recognizes a loss due to timer expiration. Therefore, TCP observes just packet delays rather than losses. Nevertheless, in the GPRS model we assume that almost all packet losses can be recovered by the forward error correction mechanism of the coding scheme and therefore no retransmissions of lost packets are necessary. Taking into account packet retransmissions, which would lead to a decrease in overall throughput, could be considered in future work.

5.3 Time-enhanced UML State Diagrams for the GPRS Model

The analysis of the GPRS model is performed by means of UML state diagrams and the mapping onto the underlying Markov chain as introduced in Chapter 3. From the steady-state distribution of the Markov chain performance measures of interest can be computed. First of all, we take into consideration the state space of the GPRS model. A *state* of the model representing the considered cell is determined by the number of GSM connections currently active, denoted by n ($0 \leq n \leq N_{GSM}$), the number of active GPRS sessions, denoted by m ($0 \leq m \leq M$), the number of packets in the base station buffer denoted by k ($0 \leq k \leq K$), and the states r_i of the two-state MMPPs for active GPRS sessions with $0 \leq i \leq m$. As a consequence, the state can be specified as the vector s = (n, k, m, r_1, ..., r_M) with $r_i = 1$ or $r_i = 2$ for $1 \le i \le m$ and $r_i = 0$ for $m+1 \le i \le M$. This leads to $\left(2^{M+1}-1\right)\left(N_{GSM}+1\right)(K+1)$ feasible states. Due to its large state space, such a model can be analyzed by discrete-event simulation only. Note that discrete-event simulation is a very time consuming analysis method and the results only have limited validity for large confidence intervals. This holds especially when dealing with large discrepancies of event rates as in an MMPP with bursty and non-bursty arrival rates. A comprehensive discussion of pros and cons of simulation and numerical analysis can be found in the introductory part of [Lin98].

Making the common assumption that all GPRS users behave statistically identical allows us to derive an aggregated model whose state space is tractable for numerical solution. The rationale behind the aggregation lies in the fact that m identical two-state MMPPs corresponding to m active GPRS sessions can be represented by one MMPP with m+1 states. This result is recapitulated in the following theorem.

Theorem 5.1 (Superposition of n identical two-state MMPPs, [FM93]): Let $\mathbf{Q}_n$ and Λ_n be the generator and the rate matrix of the composite MMPP resulting from the superposition of n identical processes with generator $\mathbf{Q}$ and rate matrix Λ.

$$\mathbf{Q} = \begin{pmatrix} -\alpha & \alpha \\ \beta & -\beta \end{pmatrix} \qquad \Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \tag{5.2}$$

The matrices $\mathbf{Q}_n$ and Λ_n are given by

$$\left(q_n\right)_{i,j} = \begin{cases} i\cdot\alpha & ,\text{if } j = i-1 \\ -i\cdot\alpha-(n-i)\cdot\beta & ,\text{if } j = i \\ (n-i)\cdot\beta & ,\text{if } j = i+1 \\ 0 & ,\text{otherwise} \end{cases} \tag{5.3}$$

$$\Lambda_n = \operatorname{diag}\left(i\cdot\lambda_1+(n-i)\cdot\lambda_2\right) \tag{5.4}$$

for $0 \le i, j \le n$. For the dimensions dim($\mathbf{Q}_n$) and dim(Λ_n) holds: dim($\mathbf{Q}_n$) = dim(Λ_n) = n+1.

Employing this aggregation, the state of the GPRS model for the considered cell can be expressed by a vector s = (n, k, m, r). In this tuple, r represents the state of the MMPP for m concurrently active GPRS sessions. The state r of the aggregated MMPP models that r MMPPs of individual GPRS sessions are in on-state and the remaining m-r MMPPs are in off-state. This reduces the state space significantly to an overall number of $\frac{1}{2}(M+1)(M+2)\left(N_{GSM}+1\right)(K+1)$ states.

The behavior of GSM users in the considered cell can be represented by an M/M/c/c queue with c = N_{GSM} servers. This is because GSM users are not effected by data traffic of GPRS sessions due to their higher priority. The arrival process of GSM voice calls is the superposition of two Poisson processes corresponding to newly arriving voice calls and incoming handover requests. Therefore, the arrival rate of the M/M/c/c queue is given by

$\lambda_{GSM}+\lambda_{h,GSM}$. In the same way, the service rate of the M/M/c/c queue is derived as $\mu_{GSM}+\mu_{h,GSM}$. Moreover, the behavior of GPRS users can be represented in the same way by an M/M/c/c queue with c = M servers and arrival and service rates $\lambda_{GPRS}+\lambda_{h,GPRS}$ and $\mu_{GPRS}+\mu_{h,GPRS}$, respectively. The GPRS model constitutes a compound queueing system whose arrival process is governed by the number of active GPRS users (i.e., the customers of the latter M/M/c/c queue) and whose service process is governed by the number of active GSM connections (i.e., the customers of the former M/M/c/c queue).

Figures 5.4 to 5.7 present time-enhanced UML state diagrams for the GPRS model. In particular, the model is represented by four different concurrent states, *CallRequest*, *GSM CallService*, *GPRS Session Service*, and *PacketTransmission*. These superstates communicate with each other through the variables *n*, *k*, *m*, and *r*. The overall incoming call requests are

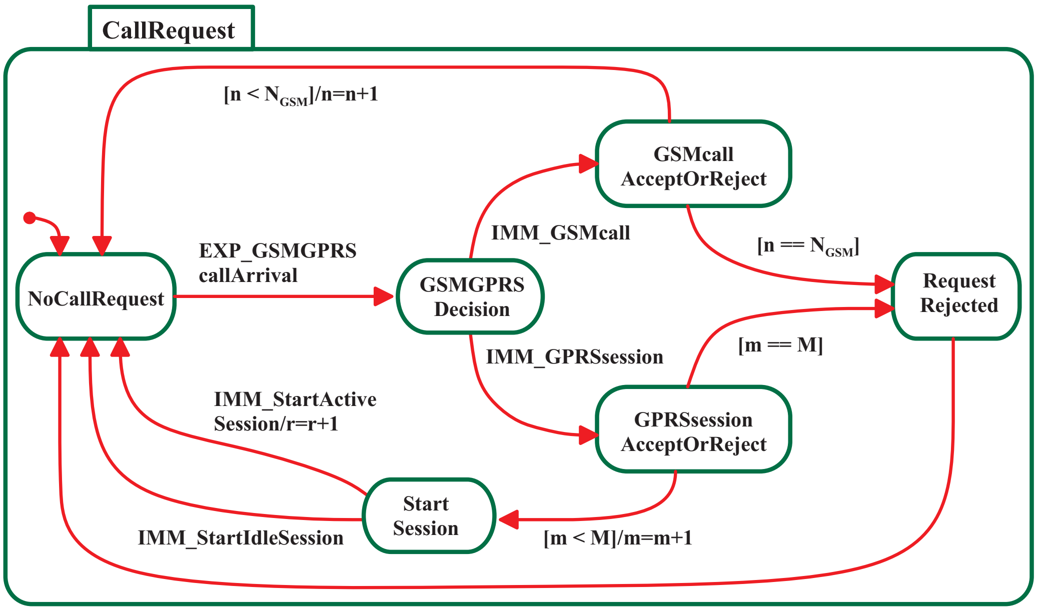


Figure 5.4. UML state diagram for call request process

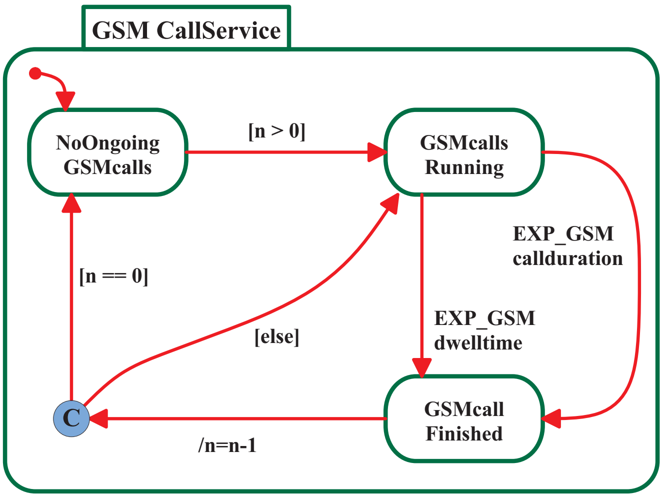


Figure 5.5. UML state diagram for GSM call service process

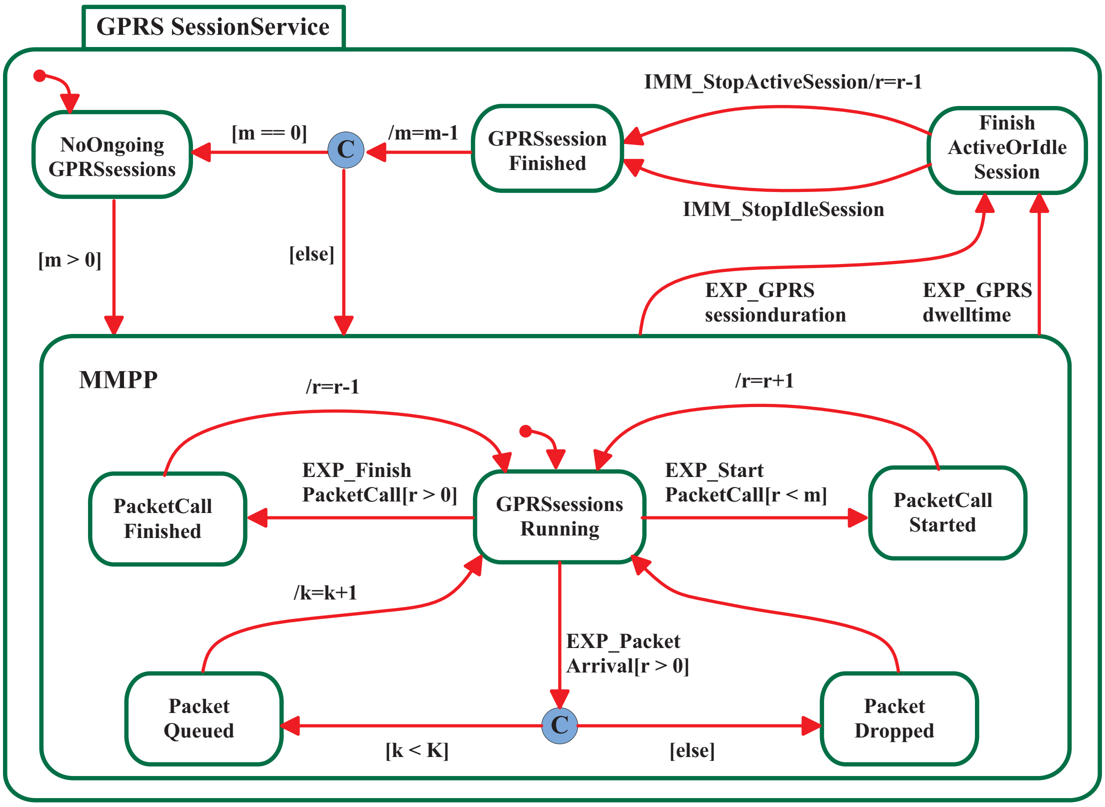


Figure 5.6. UML state diagram for GPRS session service process

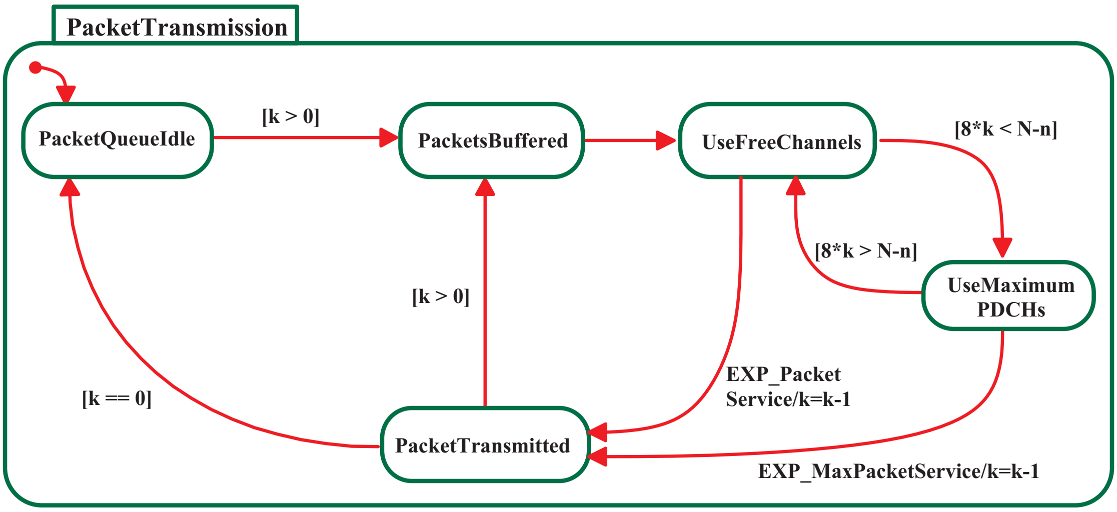


Figure 5.7. UML state diagram for GPRS packet transmission process

represented by the exponential event *GSMGPRScallArrival* which has firing rate $\lambda_{GSM}+\lambda_{h,GSM}$ + $\lambda_{GPRS}+\lambda_{h,GPRS}$. A portion $\lambda_{GSM}+\lambda_{h,GSM}$ and $\lambda_{GPRS}+\lambda_{h,GPRS}$ corresponds to GSM calls and GPRS sessions, respectively. This is represented by the immediate events *GSM call* and *GPRS session*. Incoming GSM calls and handovers are accepted in the cell if the number of free channels, excluding those reserved as PDCHs, is such that the call can be accommodated. Incoming GPRS sessions and handovers are accepted in the cell if the current number of ongoing GPRS sessions is less than the maximal number of GPRS sessions M. A new GPRS session in the cell starts sending packets according to an MMPP described above. We assume the MMPP to start in steady-state, that is in on-state with probability $\beta/(\alpha+\beta)$ and in off-state with probability $\alpha/(\alpha+\beta)$. This assumption guaranties that the MMPP is still in steady-state when the GPRS session is terminated. Thus, the immediate events *StartActiveSession* and *StartIdleSession* have associated firing weights β and α, respectively.

Both the completion of calls and outgoing handover requests have the effect of freeing a channel in the cell. Thus, with n active GSM calls the rate of freeing a channel due to call completion (exponential event *GSMcallduration*) and due to handover (exponential event *GSMdwelltime*) is $n\cdot\mu_{GSM}$ and $n\cdot\mu_{h,GSM}$, respectively. Similar, with m active GPRS sessions the events *GPRSsessionduration* and *GPRSdwelltime* have rates $m\cdot\mu_{GPRS}$ and $m\cdot\mu_{h,GPRS}$, respectively. For GPRS sessions leaving the cell we have to distinguish if the packet arrival process, i.e., the IPP, of the terminated session is in on-state or in off-state. Since r GPRS sessions (out of m) are in on-state and the remaining m-r sessions are in off-state, the leaving session is with probability r/m in on-state and with probability (m–r)/m in off-state. Thus, the immediate events *StopActiveSession* and *StopIdleSession* have associated firing weights r and m-r, respectively.

If the number of data packets queued in the base station buffer is less or equal $\eta\cdot K$ the arrival rate of data packets is determined by the number of ongoing GPRS sessions in the cell and by the state of the aggregated MMPP. In this case the average arrival rate of data packets corresponds to r sessions in on-state, i.e., exponential event *PacketArrival* has associated firing rate $r\cdot\lambda_{packet}$. The same argument holds for the time spent in a particular state of the (m+1)-state Markov chain controlling the arrival process of data packets. With rate $r\cdot\alpha$ the aggregated MMPP changes to a less bursty state, i.e. one GPRS session changes from on-state to off-state (exponential event *FinishPacketCall*), and with rate $(m-r)\cdot\beta$ it changes to a more bursty state (exponential event *StartPacketCall*), respectively. For a queue length of more than $\eta\cdot K$ data packets the arrival rate is simply bounded by the transmission rate as described next.

In the transmission process for data packets, the PDCHs are fairly shared by all packets in transfer up to a maximum of 8 PDCHs per data packet (“multislot mode”) and a maximum of 8 packets per PDCH [ETSI99]. With k packets residing in the base station buffer a maximum of $8\cdot k$ PDCHs could be used for data transfer. We assume that at each time all free on-demand channels are allocated as PDCHs. Furthermore, N_{GPRS} fixed PDCHs are utilized for data transfer. This results in an overall number of N–n physical channels that are available for the transfer of GPRS packet data. Thus, if $8\cdot k < N-n$ the maximum number of $8\cdot k$ PDCHs is utilized, i.e., exponential event *MaxPacketService* has associated firing rate $8\cdot k\cdot\mu_{service}$. Otherwise, the exponential event *PacketService* with rate $(N-n)\cdot\mu_{service}$ considers the case that all free channels can be used for transfer.

Recall that the arrival and service behavior for GSM calls and GPRS sessions constitute a M/M/c/c queueing systems. Since the steady-state solution for such a queue is known in closed-form, we can immediately derive some performance measures. With

$$\rho_{GSM} = \frac{\lambda_{GSM} + \lambda_{h,GSM}}{\mu_{GSM} + \mu_{h,GSM}} \quad \text{and} \quad \rho_{GPRS} = \frac{\lambda_{GPRS} + \lambda_{h,GPRS}}{\mu_{GPRS} + \mu_{h,GPRS}} \tag{5.5}$$

the steady-state solutions $\pi_{GSM,n}$ for n active GSM calls in the cell and $\pi_{GPRS,m}$ for m active GPRS sessions in the cell is given by:

$$\pi_{GSM,0} = \left(\sum_{n=0}^{N_{GSM}} \frac{\rho_{GSM}^{n}}{n!} \right)^{-1}, \qquad \pi_{GSM,n} = \pi_{GSM,0} \cdot \frac{\rho_{GSM}^{n}}{n!}, \qquad \text{for n = 1,2,...,} N_{GSM} \tag{5.6}$$

$$\pi_{GPRS,0} = \left(\sum_{m=0}^{M} \frac{\rho_{GPRS}^{m}}{m!} \right)^{-1}, \qquad \pi_{GPRS,m} = \pi_{GPRS,0} \cdot \frac{\rho_{GPRS}^{m}}{m!}, \qquad \text{for m = 1,2,...,M} \tag{5.7}$$

We apply the steady-state solutions (5.6) and (5.7) to iteratively balance the handover flows of GSM calls and GPRS sessions in advance. Assuming that in steady-state the average handover flow entering the cell equals the average handover flow leaving the cell and the initialization $\lambda_{h,GSM}^{(0)} = \lambda_{GSM}$ and $\lambda_{h,GPRS}^{(0)} = \lambda_{GPRS}$, the handover flows can be balanced as follows (i ≥ 0):

$$\lambda_{h,GSM}^{(i+1)} = \mu_{h,GSM} \cdot \sum_{n=1}^{N_{GSM}} n \cdot \pi_{GSM,n}^{(i)} \qquad \text{,for GSM calls} \tag{5.8}$$

$$\lambda_{h,GPRS}^{(i+1)} = \mu_{h,GPRS} \cdot \sum_{m=1}^{M} m \cdot \pi_{GPRS,m}^{(i)} \qquad \text{,for GPRS sessions} \tag{5.9}$$

Furthermore, from the solution (5.6) and (5.7) performance measures such as *Carried Voice Traffic (CVT)* and *Average Number of GPRS Sessions (AGS)* can be calculated:

$$\text{CVT} = \sum_{n=1}^{N_{GSM}} n \cdot \pi_{GSM,n} \tag{5.10}$$

$$\text{AGS} = \sum_{m=1}^{M} m \cdot \pi_{GPRS,m} \tag{5.11}$$

The *GSM call* blocking *probability* and *GPRS session blocking probability* is simply given by the steady-state probabilities $\pi_{GSM,N_{GSM}}$ and $\pi_{GPRS,M}$, respectively. In fact, the steady-state probability of N_{GSM} ongoing GSM calls or M ongoing GPRS sessions constitutes Erlang's loss formula (also known as Erlang-B formula). Note that a straight-forward evaluation of (5.6) and (5.7) for large values of N_{GSM} and M leads to overflow problems due to the computation of large factorials and powers of the loads ρ_{GSM} and ρ_{GPRS}. Therefore, a well known numerical stable method, which is based on recursive relations must be applied for the computation (see e.g. [HMP+01]).

In the following, we show how to compute additional performance indices that are plotted in the curves presented in Section 5.4. These performance measures are obtained from the steady-state solution π of the GPRS model, which is determined by numerical analysis of the underlying continuous-time Markov chain as explained in Section 4.1. The *Carried Data Traffic (CDT)* is the average number of channels in use for data transfer, i.e., PDCHs, and is given by:

$$\mathrm{CDT} = \sum_{\text{all states i}} c(i) \cdot \pi_i \tag{5.12}$$

where c(i) is the number of PDCH utilized in state i and π_i is the steady-state probability of state i. Recall that the number of PDCH used in a particular state i = (n, k, m, r) is determined by the minimum of 8·k and N−n. Furthermore, we can derive the average packet arrival rate λ_{avg} from the steady-state distribution by summing up the arrival rates in states i weighted by the probabilities π_i. The *Packet Loss Probability (PLP)* is the probability that an arriving data packet finds a base station buffer with already K packets queued and, thus, cannot be stored. It can be computed from the average packet arrival rate and the overall throughput of data packets $\mathrm{CDT} \cdot \mu_{service}$:

$$\mathrm{PLP} = \quad - \frac{\mathrm{CDT} \cdot \mu_{service}}{\lambda_{avg}} \tag{5.13}$$

The *Queueing Delay (QD)* is the time packets are waiting in the base station buffer until a free PDCH is available for transfer. It can be computed by the quotient of the *Mean Queue Length (MQL)*, which can be directly derived from the steady-state distribution, and the overall throughput of data packets:

$$\mathrm{QD} = \frac{\mathrm{MQL}}{\mathrm{CDT} \cdot \mu_{service}} \tag{5.14}$$

A last performance measure of interest is the *Average Throughput per User (ATU)* that can be derived by the overall throughput of data packets and the average number of GPRS sessions in the cell:

$$\mathrm{ATU} = \frac{\mathrm{CDT} \cdot \mu_{service}}{\mathrm{AGS}} \tag{5.15}$$

Note that all these performance measures can be defined in the supplementary timing specification file for the UML state diagrams of the GPRS model.

5.4 Performance Results for the GPRS Model

5.4.1 The Base Parameter Setting

The base parameter setting underlying the performance experiments is summarized in Table 5.1. These values are used for the derivation of all numerical results unless specified otherwise. The overall number of physical channels in a cell is set to N = 20 among which at least one channel is reserved for GPRS. Our study is mainly focussed on the introduction of GPRS into the GSM network. Therefore, we assume as base value that only 5% of the arriving calls corresponding to GPRS session requests and the remaining 95% are GSM calls. GSM call duration is set to 120 seconds and call dwell time to 60 seconds, so that users make 1-2 handovers on average. These values are quite often used in design and planning of mobile telephony systems. For GPRS sessions the average session duration is obtained from the different traffic model parameters described below. The session dwell time is assumed to be 120 seconds. We assume slower movement of GPRS users than for GSM users because higher visual attention is required for GPRS services like WWW browsing that do not allow fast movement in many cases. In all experiments, we fix the modulation and coding scheme to CS-2 [CG97]. It allows a data transfer rate of 13,4 Kbit/sec on each PDCH.

The traffic models are derived from the traffic parameter characterization defined by the 3rd Generation Partnership Project in [ETSI98]. In particular, we consider two traffic models, (1) for 8 Kbit/sec and (2) for 32 Kbit/sec WWW browsing. In both models, the average number of packet calls per session is N_{pc} = 5, the average reading time between packet calls is D_{pc} = 412 seconds. The average number of data packets within a packet call is N_d = 25. The models only differ in the burstiness of the packet arrival process. That is the interarrival time between data packets during a packet call. For the 8 Kbit/sec model D_d = 0.5 and for 32 Kbit/sec D_d = 0.125, respectively. The corresponding parameters of the GPRS model are obtained as described in Section 5.2.

Table 5.2 specifies the parameters of the traffic models. Traffic model 1 corresponds to 8 Kbit/sec bandwidth and traffic model 2 to 32 Kbit/sec bandwidth for WWW browsing,

Parameter	Base Value
Number of physical channels, N	20
Number of fixed PDCHs, N_{GPRS}	1
BSC buffer size, K	100 data packets
Transfer rate for one PDCH (CS-2), $\mu_{service}$	13.4 Kbit/sec
Average GSM voice call duration, $1/\mu_{GSM}$	120 sec
Average GSM voice call dwell time, $1/\mu_{h,GSM}$	60 sec
Average GPRS session dwell time, $1/\mu_{h,GPRS}$	120 sec
Percentage of GSM users	95%
Percentage of GPRS users	5%

Table 5.1. Base parameter setting of the GPRS model

Parameter	Traffic Model 1	Traffic Model 2	Traffic Model 3
Maximum number of active GPRS sessions, M	50	50	20
Average GPRS session duration, $1/\mu_{GPRS}$	2122.5 sec	2075.6 sec	312.5 sec
Average arrival rate of data packets, λ	8 Kbit/sec	32 Kbit/sec	32 Kbit/sec
Average duration of a packet call, $1/\alpha$	12.5 sec	3.1 sec	3.1 sec
Average reading time between packet calls, $1/\beta$	412 sec	412 sec	3.1 sec

Table 5.2. Parameter setting of different traffic models

respectively. As we will observe from the curves presented in Section 5.4.3, these two traffic models produce a low traffic load that can be managed by one or two PDCHs. In order to study the cell under heavier traffic load (i.e., the usage of on-demand PDCHs), a third traffic model is introduced. This model is obtained from traffic model 2 by setting the off-duration of the traffic process equal to the on-duration and assuming a GPRS session duration for 50 packet calls. The refined traffic model corresponds to traffic model 3 in Table 5.2.

5.4.2 Validation of the GPRS Model against Simulation

Recall that the major simplification in the GPRS model stems from the choice of studying just one cell in isolation, instead of considering the entire cell cluster and the interactions among adjacent cells. This simplification relies on the assumption that under operating conditions of the cellular network (i.e., in steady-state) the average incoming handover flow is equal to the average outgoing handover flow. Furthermore, the model consists of a simplified TCP flow control mechanism.

To validate these simplifications of the GPRS model, we additionally implemented a detailed simulator [Prz00] using the simulation library CSIM [CSIM]. This simulator represents a cellular network comprising seven hexagonal cells and takes explicitly into account the handover procedures for GSM and GPRS users. Moreover, the transmission of data packets over the wireless link is modeled in more detail than in the UML model of GPRS. That is, we explicitly consider the segmentation of data packets into TDMA frames. Furthermore, all relevant TCP mechanisms, such as slow start, congestion avoidance, and retransmission based on both timeouts and duplicate acknowledgements, have been implemented. The simulation results of the mid cell of the cell cluster are compared with corresponding results obtained from the GPRS model. Confidence intervals with confidence level of 95% for simulation results are computed using batch means. For the validation we considered traffic model 3 because in this configuration most valuable statements can be derived from the presented experiments.

In the first experiment, we determine the optimal value for the threshold η in order to closely approximate the flow control of TCP. Recall that in the GPRS model the arrival rate of data packets slows down when the queue at the base station reaches a length of more than

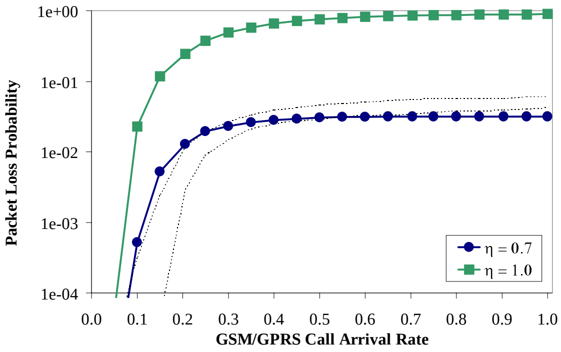


Figure 5.8. Calibrating the threshold η to closely represent the flow control of TCP

η·K packets. Figure 5.8 shows the packet loss probability for different values of η in comparison to the simulation result. The borders of the confidence intervals are drawn as dashed lines. Numerical results are drawn in solid lines. From the curves, we conclude that a setting η equal to 0.7 is best suitable for modeling a TCP flow control in the GPRS model. A value of η below 0.7 slows down the traffic, even if the network is not really congested. Subsequently, we consider η = 0.7 in the following experiments. A threshold of η = 1.0 corresponds to the case without flow control. In this case the packet loss probability approaches the value 1.0 for increasing call arrival rate.

In order to validate relevant performance measures, Figure 5.9 plots curves for carried data traffic and throughput per GPRS user for different percentages of GPRS users in comparison to the numerical results. These curves clearly indicate that the simplifications introduced in the GPRS model do not alter significantly the performance measures of interest. Thus, the GPRS model is highly accurate and can effectively be utilized for studying the performance of GPRS. Furthermore, the analytical GPRS model can be solved within a few minutes of CPU solution time on a modern PC whereas the simulator requires simulation runs in the order of hours. In fact, for a model with 489,951 states (i.e., N_{GSM} = 20, M = 20, K = 100), the memory requirements are 122.8 MByte (5,040,651 nonzero matrix elements) and the solution process consumes about 13 minutes of CPU time (applying an SOR equation system solver)

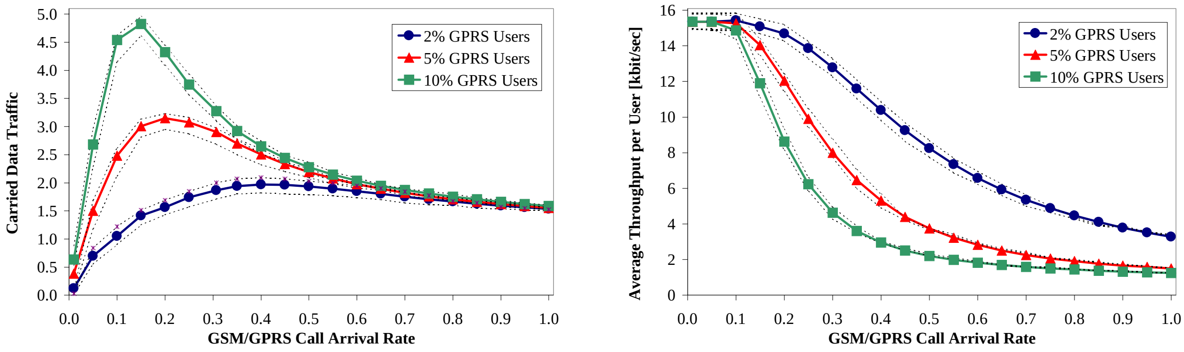


Figure 5.9. Validation of numerical results with detailed simulator, 1 reserved PDCH

on a PC with 2.3 GHz processor and 2 GByte main memory. For a GPRS model with 2,812,446 states (i.e., $N_{GSM} = 20$, $M = 50$, $K = 100$), the memory requirements are 725.7 MByte (29,829,636 nonzeros) and the CPU solution time is about 78 minutes.

The shape of the carried data traffic curve of Figure 5.9 can be explained as follows: For low traffic the fraction of the channel utilization corresponding to GPRS users increases up to 4.8 in case of 10% GPRS users. However, with increasing traffic the fraction of the channel utilization of GPRS users decreases because more and more GSM users occupy the radio resources. This is due to the assumption that GSM users have priority over GPRS users. Therefore, for very high traffic the fraction of the channel utilization corresponding to GPRS users decreases to its minimum, which corresponds to the one reserved PDCH. The reduction of carried data traffic for increasing traffic load clearly decreases the throughput for every GPRS user as depicted in the right curve of Figure 5.9.

5.4.3 A Comparative Performance Study of GPRS

This section presents numerous performance curves of the cellular mobile communication network derived from steady-state solutions of the GPRS model. In particular, we investigate the impact of the number of PDCHs reserved for GPRS users on the performance of the cellular network. This results give valuable hints for network designers on how many PDCHs should be allocated for GPRS for a given amount of traffic in order to guarantee appropriate quality of service. In the curves presented in this section, we assume the base parameter setting of Table 5.1 if not mentioned otherwise. In all curves the arrival rate of GSM and GPRS users is varied to study the cell under increasing traffic intensity due to more user requests.

Figures 5.10 to 5.12 present a comparative study of the mobile network considering traffic models 1 and 2. As performance measures, we consider carried data traffic, packet loss probability, and queueing delay as defined in Section 5.3. In each figure we vary the number of reserved PDCHs (1, 2, and 4). The maximum number of GPRS sessions that can be concurrently active in the cell is restricted to $M = 50$. From the curves presented in Figure 5.9 we see that for both traffic models the carried data traffic remains nearly the same even if we reserve 1, 2 or 4 PDCHs for GPRS. For a GSM/GPRS call arrival rate of 1 call per second only 0.6 PDCHs are used on average. Note that a GSM/GPRS call arrival rate of one call per second corresponds to 0.05 new GPRS session requests per second in case of 5% GPRS users. To derive the overall GPRS session request rate, we have to add the handover request rate that is obtained by the balancing procedure (see Section 5.2 and Section 5.3). In case of traffic model 1 and 2 this handover rate is very high for GPRS users because their dwell time in the cell is very low compared to their average session duration. Therefore, we conclude that for both traffic models one PDCH is sufficient to carry the data traffic of the GPRS users.

From Figures 5.11 and 5.12, we observe that reserving more PDCHs decreases queueing delay and the probability of packet loss due to buffer overflow. This is surely important to provide certain QoS guaranties to GPRS users. Consequently, we conclude that on the one hand reserving more PDCHs will decrease queueing delay and packet loss (in bursty packet arrival phases) but on the other hand these extra physical channels will be idle most of the time. It is noteworthy that due to the scarce radio resources the reservation of PDCHs has to be decided carefully. Reserving two or even more PDCHs would only be desirable for providing certain QoS guaranties to GPRS users. Comparing the curves in Figures 5.11 and 5.12, we find that traffic model 2 which produces more bursty traffic (arrival rate of 32 Kbit/sec during a packet call) results in longer delay and higher packet loss probability.

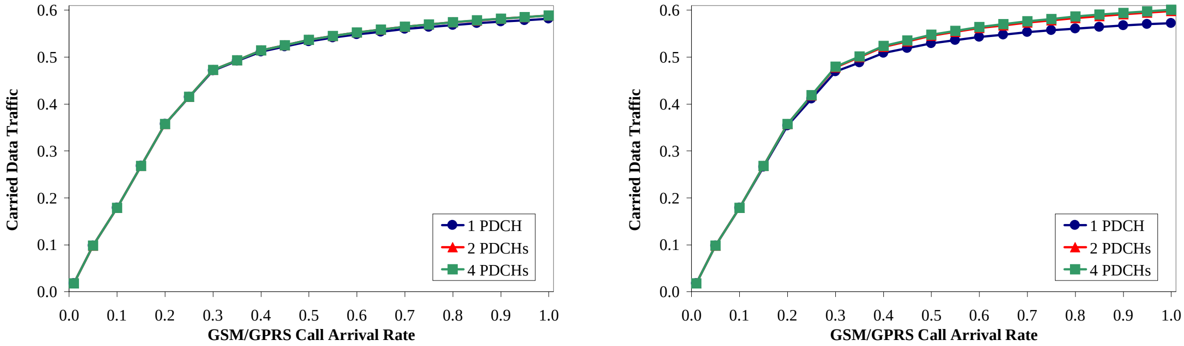


Figure 5.10. Carried data traffic for traffic model 1 (left) and 2 (right)

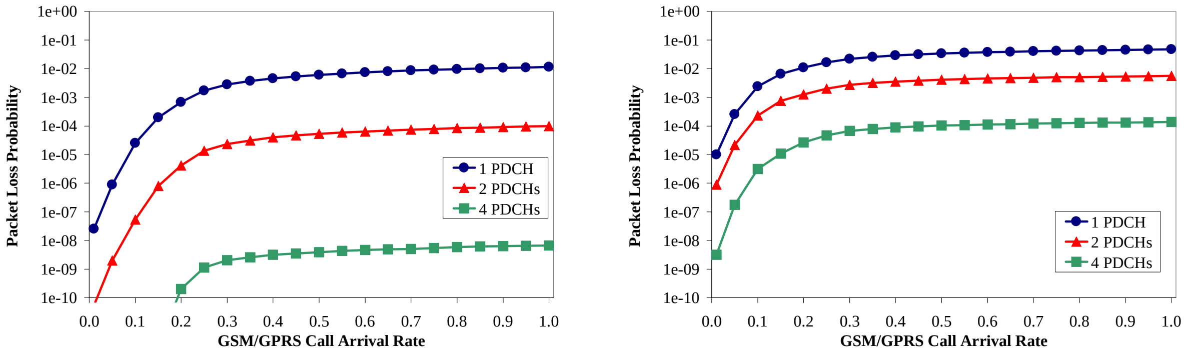


Figure 5.11. Packet loss probability for traffic model 1 (left) and 2 (right)

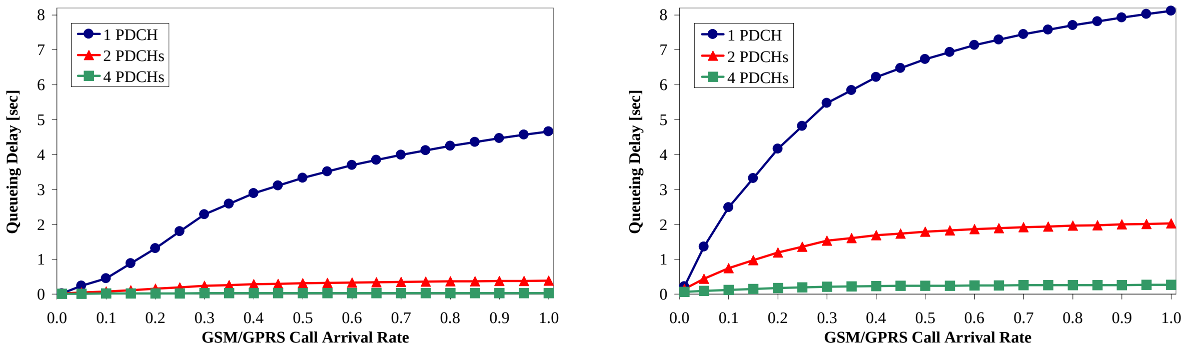


Figure 5.12. Queueing delay for traffic model 1 (left) and 2 (right)

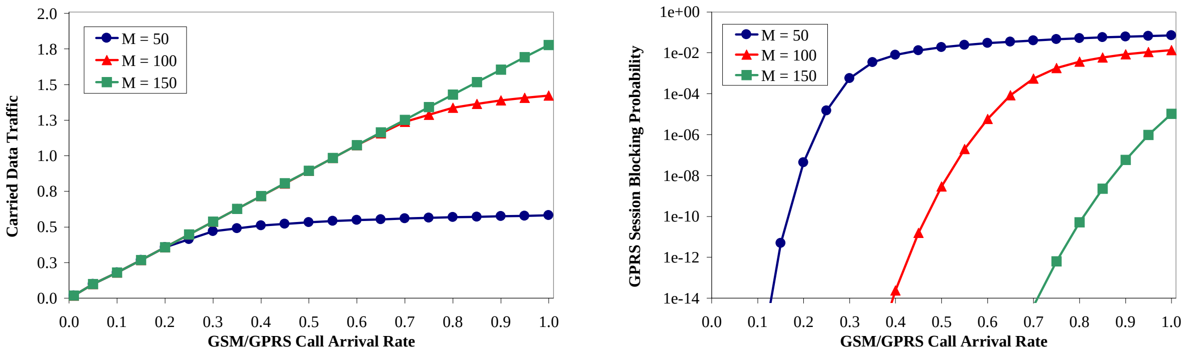


Figure 5.13. Carried data traffic and GPRS session blocking probability

Due to the long GPRS session duration of approximately 2100 seconds (equals to 35 minutes) in traffic models 1 and 2, the handover arrival rates of active GPRS sessions is very high (about 0.3 GPRS handover requests per second at an GSM/GPRS arrival rate of 1 call per second). Therefore, the blocking probability of arriving GPRS sessions is also high because the maximal number of active GPRS sessions in the cell is restricted to M = 50 and this limit is reached very quickly (about 10% of GPRS users are not admitted in the cell at an arrival rate of 1 GSM/GPRS call per second). This effect justifies the following experiment where we studied how many PDCHs are needed to satisfy *almost all* GPRS session requests up to a GSM/GPRS call arrival rate of 1 call per second (see Figure 5.13). Therefore, we increase the maximum number of active GPRS sessions allowed in the cell to values M = 50, 100, and 150, respectively. The curves of Figure 5.13 plot carried data traffic and GPRS session blocking probability versus GSM/GPRS call arrival rate. They are computed using traffic model 1. For M = 150 we find a maximal GPRS session blocking probability that is below 10^{-5} with an utilization of 1.8 PDCHs on average: In fact no more PDCHs are needed! We conclude from Figure 5.13 that the reservation of 2 PDCHs for GPRS is sufficient to satisfy *almost all* GPRS session requests up to a new call arrival rate of 1 call per second.

In the next experiments, we investigate the system under higher GPRS traffic load, i.e. traffic model 3. Figures 5.14 to 5.16 present a comparison of the mobile network for different system configurations. The comparison is made in two dimensions: the amount of GPRS users and the number of reserved PDCHs. In each curve, we vary the number of reserved PDCHs (0, 1, 2, and 4) and the fraction of GPRS users among newly arriving calls (2%, 5%, and 10%). As performance measure, we consider the carried data traffic and average throughput per user.

For low traffic the utilization of physical channels for packet transfer is independent from the numbers of reserved PDCHs. This is because the low amount of traffic can be completely managed by the cell even with no reserved PDCH. However, for increasing traffic intensity the channel utilization for data transfer decreases. This can be explained by the same argument as for Figure 5.9. Furthermore, we observe that the decrease of allocated PDCHs due to high traffic intensity becomes less significant when more PDCHs are reserved. This observation can also be concluded from the curves plotting the average throughput per GPRS user in Figures 5.14 to 5.16. Surely, under very low traffic intensity for GSM and GPRS users, each GPRS user reaches the maximum throughput. With increasing load, the throughput per user decreases much more slightly in case of 4 reserved PDCHs. This is opposed to the case of no reserved PDCHs where the throughput approaches nearly zero.

Comparing the different GPRS user populations, we discuss an example of high interest for network designers: Consider GPRS users with a QoS profile that allows a throughput degradation of at most 50%. Then, we can conclude that for 2% GPRS users the reservation of 4 PDCHs is sufficient up to an GSM/GPRS call arrival rate of 1 call per second. However,

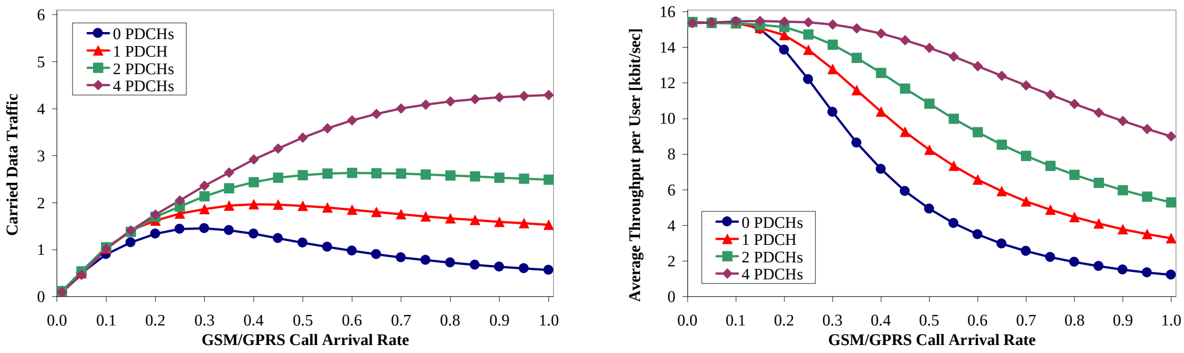


Figure 5.14. Carried data traffic and throughput per user for 2% GPRS users

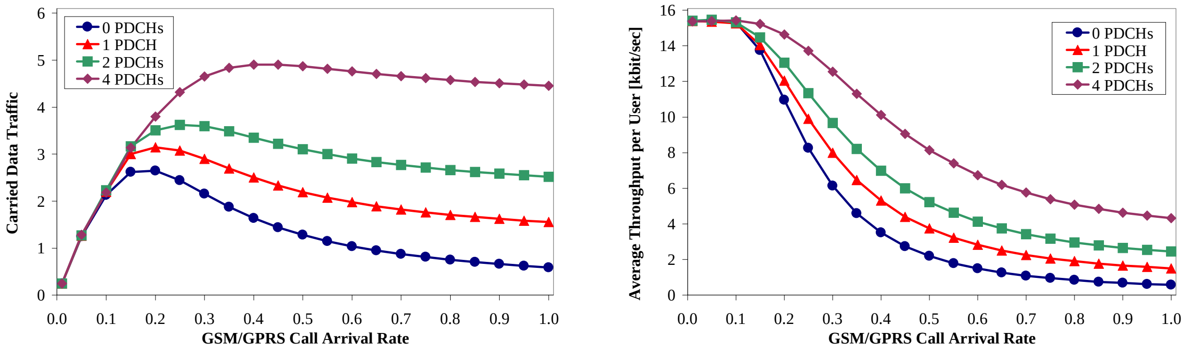


Figure 5.15. Carried data traffic and throughput per user for 5% GPRS users

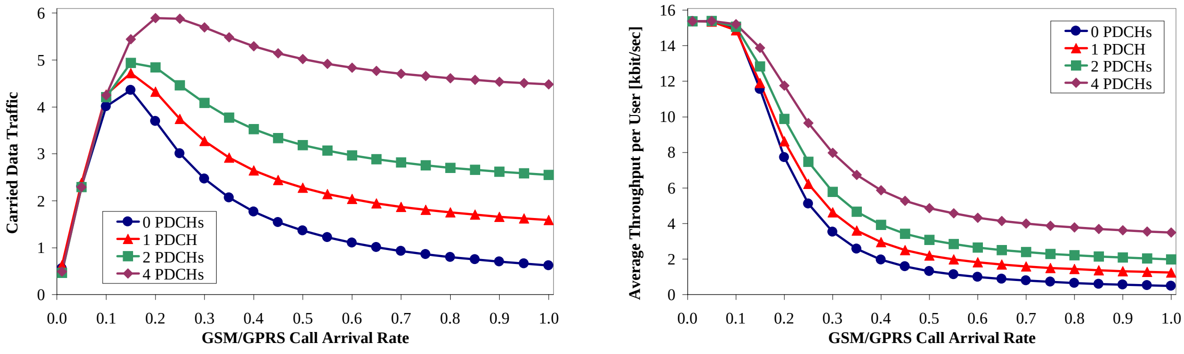


Figure 5.16. Carried data traffic and throughput per user for 10% GPRS users

for the case of 5% and 10% GPRS users, the QoS profile can only be guaranteed up to a call arrival rate of 0.5 and 0.3 calls per second, respectively. In this case network designers should think about more restrictive call admission conditions to meet the requirements.

In an additional experiment, we study the performance loss in the GSM voice service due to the introduction of GPRS. Figure 5.17 plots the carried voice traffic and voice blocking probability for different numbers of reserved PDCHs. The presented curves indicate that the decrease in channel capacity and, thus, the increase of the blocking probability of the GSM voice service is negligible compared to the benefit of reserving additional PDCHs for GPRS. Figure 5.18 presents curves for average number of GPRS users in the cell and blocking probabilities of GPRS session requests due to reaching the limit of M active GPRS sessions. We observe that for 2% GPRS users the maximum number of 20 active GPRS sessions is not reached. Therefore, the blocking probability remains below 10^{-5}. For 10% GPRS users and increasing call arrival rate, the average number of sessions approaches its maximum. Thus, some GPRS users will be rejected. It is important to note that the curves of Figure 5.18 can be utilized for determining the average number of GPRS users in the cell for a given call arrival rate. In fact, together with the curves of Figures 5.14 to 5.16, we can provide estimates for the maximum number of GPRS users that can be managed by the cell without degradation of quality of service. For example, for 5% GPRS users and 4 PDCHs reserved, the QoS profile of maximal 50% throughput degradation is achieved until the call arrival rate exceeds 0.5 calls per second, i.e., until there are on average eight active GPRS users in the cell.

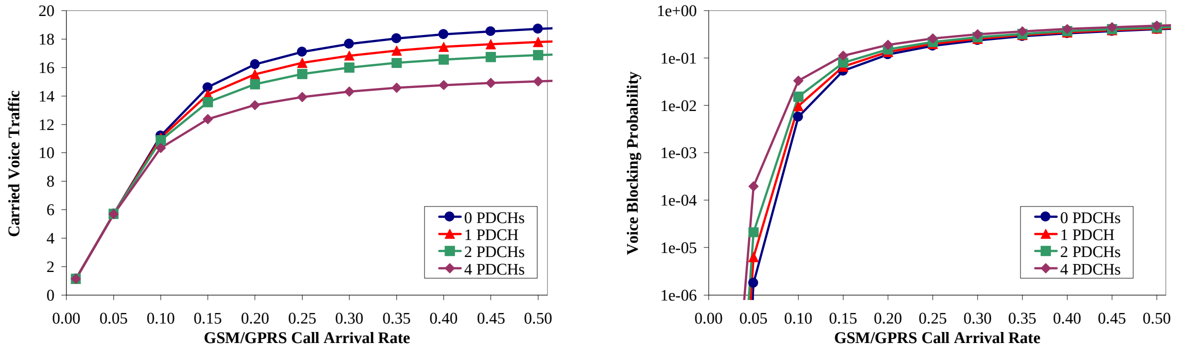


Figure 5.17. Influence of GPRS on GSM voice service (95% GSM calls)

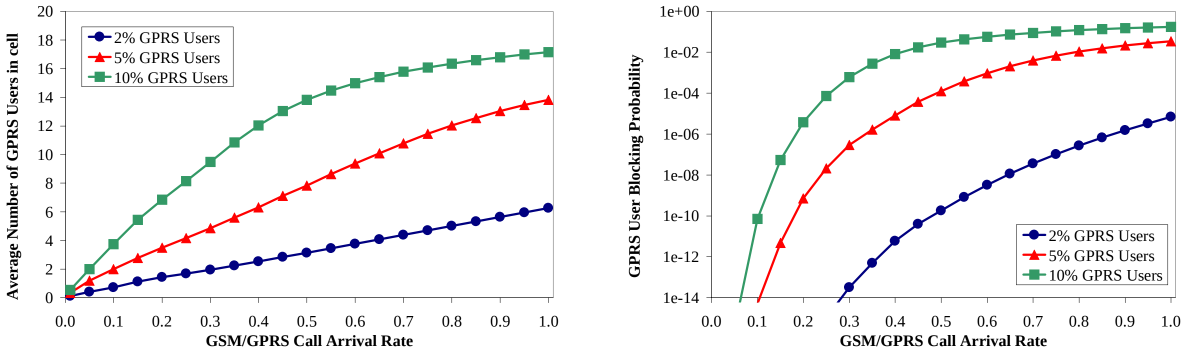


Figure 5.18. Average number of GPRS users in cell and GPRS user blocking probability

5.5 Summary

In this chapter, a comprehensive performance study of the radio resource sharing by circuit-switched GSM connections and packet-switched GPRS sessions under a dynamic channel allocation scheme is presented. A fixed number of physical channels are assumed to be permanently allocated to GPRS and the remaining channels are assumed to be on-demand channels that can be used by GSM voice service and GPRS packets. Performance results are derived from the steady-state analysis of a GPRS model. A validation of the GPRS model with a detailed simulator on the network level shows that almost all performance curves derived from the GPRS model lie in the confidence intervals of the corresponding curve of the simulator. The presented GPRS model can be analyzed within a few minutes of CPU solution time on a modern PC whereas the simulator requires simulation runs in the order of hours.

The impact of the number of packet data channels reserved for GPRS users on the performance of the cellular network is investigated. That is for example, for GPRS users with a QoS profile allowing a throughput degradation of at most 50%, we concluded that for 2% GPRS users among all incoming calls, the reservation of four PDCHs is sufficient up to an GSM/GPRS call arrival rate of one call per second. However, for the case of 5% and 10% GPRS users, the QoS profile can only be guaranteed up to a call arrival rate of 0.5 and 0.3 calls per second, respectively. Such results give valuable hints for network designers on how many PDCHs should be allocated for GPRS for a given amount of traffic in order to guarantee appropriate quality of service.

Note that determining the number of PDCHs for GPRS is a tradeoff between GSM and GPRS performance. Therefore, an optimal value of PDCHs can be only determined with respect to the desired performance requirements for GSM and GPRS that must be selected by the network operator. Applying adaptive performance management [LLT02], future work considers the dynamic adjustment of the number of PDCHs with respect to the current GSM and GPRS traffic load and the desired performance requirements.

Chapter 6

Adaptive Quality of Service Management

In this chapter, an approach for the adaptive control of 3G mobile networks in order to improve quality of service (QoS) for mobile subscribers is introduced. The approach constantly monitors QoS measures such as packet loss probability and handover failure probability during operation of the network. Based on the values of the QoS measures just observed, system parameters of an admission controller are adjusted by an adaptive QoS management entity. Thus, the adaptive control framework closes the loop between network operation and network control. Section 6.1 introduces the approach for adaptive QoS management and describes its embedding in the system architecture of 3G mobile networks. Section 6.2 introduces strategies for controlling the parameters of the admission controller in order to improve QoS. In Section 6.3, a UMTS system simulator [Hän01] that was developed in a diploma thesis at the University of Dortmund is introduced. Section 6.4 presents performance curves showing that handover failure probability is improved by more than one order of magnitude. Moreover, the packet loss probability can be effectively regulated to a predefined level.

6.1 The Framework for Adaptive QoS Management

This section introduces the approach for regularly adjusting system parameters to changing traffic load, packet arrival pattern, or population of users, etc. We consider a cellular mobile network in which a different transceiver station serves each cell. Furthermore, a base station controller (BSC) is considered that is responsible for a cluster of cells, i.e., several transceiver stations. The BSC manages the radio resources, i.e., schedules data packets, and controls handovers inside the cell cluster as well as handovers towards and from neighboring cell clusters. Note that the term BSC represents a general controlling entity. In case of UMTS, a BSC is meant to be a radio network controller (RNC) (see Chapter 2).

To improve QoS for mobile subscribers, an entity for *Adaptive QoS Management*, abbreviated with APM according to [LLT03a], is included in a BSC. Furthermore, a BSC has to be extended by an *Online QoS Monitoring* component that derives QoS measures in a

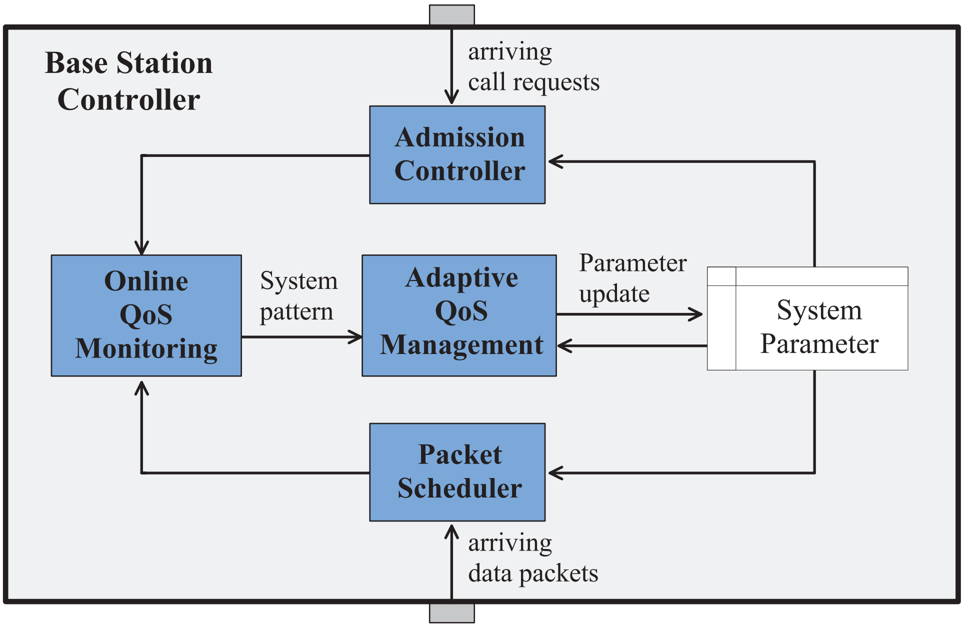


Figure 6.1. System architecture for adaptive QoS management

certain time window (e.g., handover failure probabilities of mobile users or packet loss probabilities). These QoS measures form a *system pattern* that is submitted in fixed time intervals (i.e., a control period) to the APM entity, which subsequently updates corresponding system parameters (i.e., parameters of traffic controlling components like the admission controller and packet scheduler). Thus, the proposed approach closes the loop between network operation and network control. Figure 6.1 shows the system architecture for QoS management embedded in a BSC.

6.1.1 Online QoS Monitoring

System parameters of a BSC can be effectively updated by monitoring QoS measures, which are immediately affected by these parameters. A current value for a QoS measure is determined online based on a set of *relevant events* corresponding to this QoS measure (e.g., packet arrivals are relevant events for computing packet loss probabilities). To determine an estimate for a QoS measure from the observed relevant events, estimation techniques such as a sliding window, an exponential-weighted moving average, or more recently developed techniques based on a combination of agile and stable estimators involving Kalman filters must be applied [KN01]. However, for the study presented in this thesis the online monitoring of QoS measures is performed by a sliding window technique shown in Figure 6.2. Studying QoS monitoring with other estimation techniques is out of the scope of this thesis and could be subject for future work.

The width of the sliding window over time depends on the number of relevant events that are occurred according to a given QoS measure. The upper part of Figure 6.2 shows the

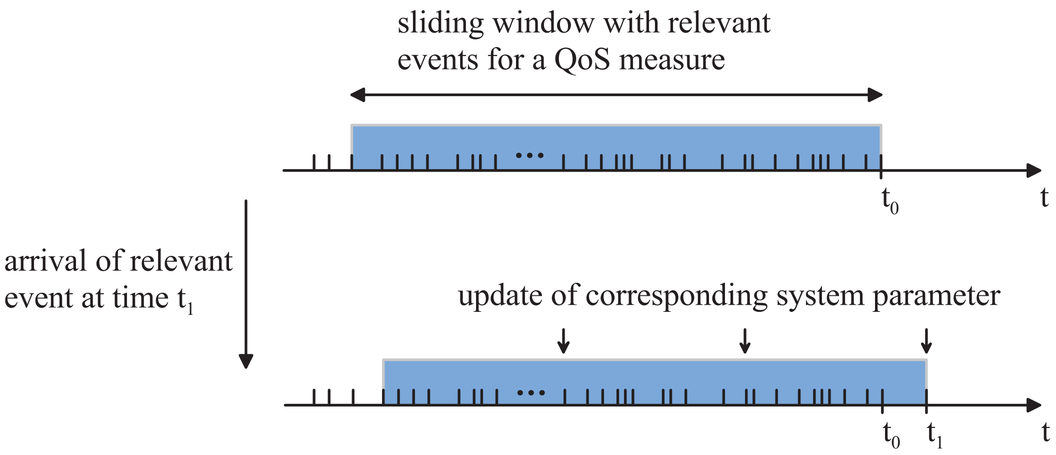

sliding window with relevant events for a QoS measure
•••
t_0
t
arrival of relevant event at time t_1
update of corresponding system parameter
•••
t_0
t_1
t

Figure 6.2. Online QoS monitoring and system parameter update

sliding window at a certain point in time t_0. Upon arrival of a new relevant event at time t_1 the sliding window moves in time as shown in the lower part of Figure 6.2. At the end of a control period the QoS measures are derived for each sliding window (e.g., packet loss probability can be derived from number of lost packets divided by number of all packet arrivals in the sliding window). These QoS measures and the number of events occurred in the last control period form the system pattern that is transferred to the adaptive QoS management entity (see Figure 6.1).

Note that an accurate online monitoring of QoS measures requires a specific width for the sliding window. A certain number of events representing the history of the QoS measure have to be considered to get an expressive measure. On the other hand considering a big sliding window prevents the APM entity from fast reaction on changing traffic conditions. A bigger sliding window contains more history and, thus, more events have to be collected to cause a significant change in the online monitored QoS measure. This tradeoff between accurate online monitoring and fast reaction of the APM to changing traffic conditions has to be studied carefully in several experiments to get the optimal width of the sliding window for each QoS measure.

6.1.2 Adaptive QoS Management

Whenever a system pattern $S = \{(P_1, n_1), \ldots, (P_m, n_m)\}$, consisting of online monitored QoS measures $P_1,\ldots,P_m$ and the numbers of relevant events $n_1,\ldots,n_m$, which occurred in the last control period, is transmitted to the APM an update of the system parameters can be performed. In general, an update of a system parameter σ is made according to a function f depending on a subset of the QoS measures $P_1,\ldots,P_m$ and the previous value $\sigma^{(old)}$ of the system parameter. Let $P_{\tau(1)},\ldots,P_{\tau(k)}$, $k \leq m$, be the QoS measures corresponding to system parameter σ, then the update is made if a certain minimum number $n(\sigma)$ of relevant events occurred in the last control period. That is:

$$\sigma^{(\text{new})} = f(P_{\tau(1)},\ldots,P_{\tau(k)},\sigma^{(\text{old})}), \quad \text{if } \min\{n_{\tau(1)},\ldots,n_{\tau(k)}\} \geq n(\sigma) \tag{6.1}$$

Update functions are classified in *relative* functions, that perform a parameter update relative to the old parameter value and *absolute* functions that set the new parameter value independent of the old value, i.e., f is independent of $\sigma^{(\text{old})}$ in (6.1). With relative update functions strong fluctuations of the corresponding system parameter in one update step can be avoided, since the old value of the system parameter contributes to the computation of the new one. Thus, the update function can bound the difference between the two values to a predefined maximum to avoid fluctuations. In Section 6.2, a special class of relative update functions in order to set the parameters of an admission controller is studied. Furthermore, in [LLT03a] we developed an absolute update function for adjusting the weights of a weighted fair queueing packet scheduler.

6.2 Adaptive Admission Control for QoS Improvement

The proposed approach distinguishes three different types of services: circuit-switched services, packet-switched real-time services (RT), and packet-switched non real-time services (NRT). Typically, circuit-switched services are voice calls from a GSM mobile station. As proposed by 3GPP, RT services belong to conversational and streaming classes and NRT services are part of interactive and background classes [3GPP01a] (see also Chapter 2). The bandwidth available in a cell must be shared by calls of these different service classes and the different service requirements have to be met.

Before a mobile session begins, the user needs to specify its traffic characteristics and desired performance requirements by a *QoS profile*. Then, an admission controller decides to accept or reject the user's request based on the QoS profile and the current network state as e.g., given by queueing length. The purpose of the admission controller is to guarantee the QoS requirements of the user who requested admission while not violating the QoS profiles of already admitted users. The call admission criteria will be different for each service class. The QoS profile for RT sessions specifies a guaranteed bandwidth to be provided for the application in order to meet its QoS requirements. If the network cannot satisfy the desired bandwidth, the corresponding admission request is rejected. NRT sessions will be admitted by concerning the current network state in terms of current queue length as described below.

6.2.1 Architecture of the Admission Controller

In the following, a partitioning of the available bandwidth in one cell is proposed in order to meet the QoS requirements of the three considered service classes: voice calls, RT sessions, and NRT sessions. The bandwidth partitioning constitutes the rationale behind the admission

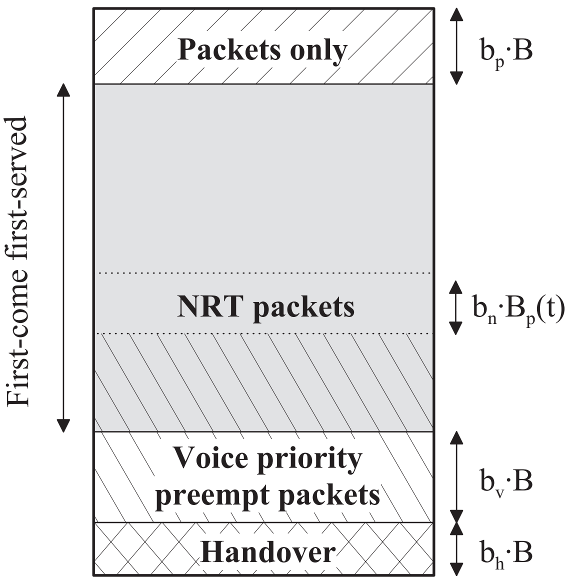


Figure 6.3. Dynamic partition of the available bandwidth

control. Figure 6.3 illustrates the partitioning of the available bandwidth into different areas. Let B be the overall bandwidth available in one cell. A portion b_h of the bandwidth B is exclusively reserved for *handover* calls from neighboring cells in order to reduce handover failures. A portion b_p is reserved for real-time and non real-time data packets, denoted by *packets only*. The remaining bandwidth $(1 - b_h - b_p) \cdot B$ can be allocated “on-demand” by voice calls and data sessions, respectively.

Because in future 3G mobile networks (around the year 2010) voice calls will still play a major role in bandwidth requirements [UMTS99], a portion b_v of the overall bandwidth with priority of voice calls, denoted by *voice priority preempt packets*, is introduced. This means, if more bandwidth than $(1 - b_h - b_v) \cdot B$ is allocated for packet data calls and the remaining bandwidth is not sufficient for accommodating a newly arriving voice call, real-time sessions have to be degraded below their guaranteed bandwidth. In fact, in this case the bitrate guarantee given to RT users will be violated, but as can be seen by the experiments presented in Section 6.4 the probability of real-time session degradation is negligible for small values b_v. The remaining on-demand bandwidth is allocated on a first-come first-served basis to voice calls or RT sessions. In order to give NRT traffic a certain amount of bandwidth, a portion b_n of the bandwidth actually available for packet data is exclusively reserved for NRT packets, denoted by *NRT packets*. Let $B_v(t)$ be the bandwidth reserved for all voice calls at a certain time point t, then for packet data bandwidth of size $B_p(t) := B - B_v(t)$ is available.

Table 6.1 shows the different call types arriving in the cell and the corresponding conditions under which these call arrivals will be admitted in a compact notation. Let t be the point in time when an admission request arrives at the admission controller. For real-time users, admission is based on the availability of the guaranteed bandwidth specified in the QoS profile. Let $B_r(t)$ be the bandwidth already allocated for real-time traffic at time t and let B_r be

	Conditions for Admission
Voice call	$B_v + B_v(t) \le \max\left\{b_v \cdot B, \min\left\{(1-b_h)\cdot B - \max\left\{b_p \cdot B, B_r(t)\right\}, B - \frac{1}{1-b_n}\cdot B_r(t)\right\}\right\}$
RT session	$B_r + B_r(t) \le \min\left\{(1-b_h)\cdot B - B_v(t), (1-b_n)\cdot(B - B_v(t))\right\}$
NRT session	NRT queue length at admission request time ≤ $\eta \cdot K_{NRT}$
Voice handover	$B_v + B_v(t) \le \max\left\{b_v \cdot B, (1-b_h)\cdot B - \max\left\{b_p \cdot B, \frac{1}{1-b_n}\cdot B_r(t)\right\}\right\}$
RT handover	$B_r + B_r(t) \le (1-b_n)\cdot(B - B_v(t))$

Table 6.1. Call admission conditions for different call types

the bandwidth required by the user who requested admission. The user will be admitted according to the bandwidth partitioning illustrated in Figure 6.3. That is, if after call admission the handover bandwidth is still available, i.e., $B_r + B_r(t) \le (1-b_h)\cdot B - B_v(t)$, and a portion b_n of the overall packet bandwidth $B_p(t)$ is also still available for non real-time sessions, i.e., $B_r + B_r(t) \le (1-b_n)\cdot(B - B_v(t))$. A minimum operator combines these two cases in the corresponding formula of Table 6.1. New voice calls with bandwidth requirements B_v will be admitted, if either less than $b_v \cdot B$ bandwidth (voice priority preempt packets area of Figure 6.3) is allocated for voice calls or if the voice call can be accommodated in the first-come first-served area without violating bandwidth requirements of ongoing calls. The corresponding formula presented in Table 6.1 can be derived in a similar way as for real-time sessions.

Data packets arriving at the BSC are queued in two distinct queues until they are scheduled to be transmitted over the radio link. We distinguish a RT queue with capacity K_{RT} and a NRT queue with capacity K_{NRT}. For NRT sessions, the admission is based on the availability of buffer space in the NRT queue [DJK+00]. In order to prevent buffer overflow once a call is admitted, the current queueing length is set against certain buffer availability threshold of the capacity K_{NRT}, denoted by η. The admission criteria for voice and RT handovers are the same as for new voice calls and RT sessions except that additional handover bandwidth can be utilized. The considered admission controller does not prioritize NRT handovers over new NRT sessions.

6.2.2 Derivation of Formulas for Adaptive Control

This section shows how to apply Eq. (6.1) for setting the parameters η and b_h of the admission controller in order to reduce packet loss probability and handover failure probability. For updating the system parameters, the general function presented in Eq. (6.1) is split into separate functions each depending only on one QoS measure. Let $P_1,...,P_k$ be the QoS measures corresponding to a system parameter σ. Then, Eq. (6.1) is simplified to:

$$\sigma^{(\text{new})} = \frac{f_1(P_1)+\cdots+f_k(P_k)}{k}\cdot\sigma^{(\text{old})}, \qquad L \le \sigma^{(\text{new})} \le R \tag{6.2}$$

The interpretation of (6.2) is the following: Each update function f_i describes the influence that the QoS measure P_i should have on the system parameter σ. Subsequently, the overall update is performed by computing the arithmetic mean of the functions f_i, i = 1,...,k, multiplied with the old value of the system parameter. Note that the value $\sigma^{(\text{new})}$ must be truncated at a certain lower bound L and an upper bound R in order to guarantee that the computation of $\sigma^{(\text{new})}$ results in a valid value of the system parameter. As basic update function we consider a logarithmic linear function of the form:

$$f_i(P_i) = m_i \cdot \log P_i + b_i \tag{6.3}$$

The reason for this choice is that we want to consider QoS measures like loss probabilities and failure/blocking probabilities, which are in the range of 10^{-5} to 1. Therefore, a logarithmic shape is more suitable. In [LLT02], we have studied update schemes of system parameters of an admission controller and a packet scheduler based on a look-up table. In order to determine the optimal entries of this look-up table extensive offline simulation experiments have been conducted. Applying regression statistics to the entries of this look-up table shows that these entries are well represented by functions with logarithmic shape. Thus, besides the motivation of the update functions given here, their choice is to a large extent originated from regression statistics conducted in earlier work. The strength of the influence of f_i on $\sigma^{(\text{new})}$ can be adjusted with the gradient m_i. The parameter b_i can be determined by the following interpretation: Suppose the *desired level* of the QoS measure P_i is β_i (e.g. the desired packet loss probability is 0.001). That is, if the online measured value of P_i is β_i the system parameter σ should not be changed in the update step from the point of view of measure P_i. Therefore, we chose $f_i(\beta_i) = 1$ and from this relation we get $b_i = 1 - m_i \cdot \log \beta_i$. Inserting in Eq. (6.3) results in the final form of the update function:

$$f_i(P_i) = m_i \cdot \log \frac{P_i}{\beta_i} + 1 \tag{6.4}$$

For ease of notation, we abbreviate the QoS measures handover failure probability and new call/session blocking probability corresponding to voice calls and RT sessions by HFP and CBP, respectively. The probability of a packet loss due to buffer overflow in the NRT queue is abbreviated by PLP. The update strategy according to Eqs. (6.2) to (6.4) is justified by its intuitive understanding and the performance results presented in Section 6.4 and in [LLT02] and [LLT03a]. The suitability of update functions other than (6.2) to (6.4), is subject for further study.

6.2.2.1 Update of Non Real-Time Queue Threshold

Recall that a system parameter update is performed each time a system pattern arrives at the APM entity and the minimum number of relevant events corresponding to this system parameter is reached. Determining the update for the system parameter η, i.e., determining $\eta^{(new)}$, is performed corresponding to the old value $\eta^{(old)}$ and the actually observed QoS measure PLP. That is:

$$\eta^{(new)} = f(PLP) \cdot \eta^{(old)}, \qquad 0.001 \le \eta^{(new)} \le 1 \tag{6.5}$$

The truncation of $\eta^{(new)}$ at the lower bound guaranties that the value does not accumulate near zero for long periods of low traffic load. The minimum number of relevant events required for an update of η is counted in data volume rather than in packet arrivals (in the experiments this number is 5MB). The setting of the gradient m of the corresponding update function is derived from a couple of experiments for different values of the gradient. The value m = -0.02 was found to be suitable. Choosing a suitable value for the gradient is a similar tradeoff as explained for the sliding window size. A large gradient results in a fast update of the system parameter in a few update steps, but also introduces higher fluctuations of the system parameter over time. The speed of the parameter adjustment is demonstrated in an experiment in Section 6.4. Furthermore, several experiments for different desired loss values β are presented.

6.2.2.2 Update of Fraction of Bandwidth Reserved for Handover

The update for the system parameter b_h, i.e., determining $b_h^{(new)}$, is performed based on the old value and the actually observed QoS measures HFP and CBP. That is:

$$b_h^{(new)} = \frac{f_1(HFP) + f_2(CBP)}{2} \cdot b_h^{(old)}, \qquad 0.001 \le b_h^{(new)} \le R \tag{6.6}$$

The value $b_h^{(new)}$ is truncated at a lower bound of 0.1% and a certain upper bound R which is a fraction of the overall bandwidth available (in the experiments we fix R = 0.7). The truncation at the lower bound is for the same reason as explained above. In fact, for computing $b_h^{(new)}$ two QoS measures corresponding to the actually observed HFP and CBP are taken into account. A high HFP should increase $b_h^{(new)}$ but this obviously also increases the CBP because less bandwidth is available for new voice calls and RT sessions. Therefore, the HFP and the CBP influences the handover bandwidth $b_h^{(new)}$. In fact, $m_1 = -m_2$ holds in the update functions f_1 and f_2. From a couple of experiments for different gradients, m_1 = 0.08 was found to be suitable. A common assumption in cellular networks is to prioritize handover calls over new calls. Therefore, the desired handover failure level β_1 should be smaller than the desired call blocking level β_2. According to these values the handover bandwidth is slightly increased, if HFP is equal to CBP.

With the presented strategy the parameters of the update functions can be chosen in an intuitive way and optimal parameter configuration can efficiently be determined. This is the major advantage over the approach based on a Performance Management Information Base (P-MIB) introduced in [LLT02], which requires extensive offline simulation experiments.

6.3 The UMTS Simulation Environment

In order to evaluate the proposed approach for adaptive control, we developed a simulation environment [Hän01] for a UMTS access network, i.e., a *UMTS Terrestrial Radio Access Network* (UTRAN) (see Chapter 2 for an introduction to the UMTS terminology). The simulator considers a cell cluster comprising seven hexagonal cells with corresponding transceiver stations (i.e., *Node B* elements), that are managed by a base station controller (i.e., a *Radio Network Controller*, *RNC*). We assume that a mobile user requests a new *session* in a cell according to a Poisson process. When a mobile user starts a new session, the session is classified as voice-, RT, or NRT session, i.e., with the session the user utilizes voice-, RT, or NRT services mutually exclusive. RT sessions consist of streaming downlink traffic corresponding to the UMTS streaming class (see Chapter 2.3) and NRT sessions consist of elastic traffic and correspond to the UMTS interactive class or background class, respectively. For the year 2010 an amount of about 50% voice calls is anticipated [UMTS99]. We assume that one half of the voice calls are served over the frequency spectrum for traditional GSM services (i.e., 890-915 and 935-960 MHz) and the second half is served over the new frequency spectrum allocated for UMTS. Nevertheless, the simulator considers only the new frequency spectrum. Therefore, we assume that 25% of the call requests are voice calls whereas RT and NRT sessions constitute 15% and 60% of the overall arriving requests (see Table 6.2).

Subsequently, we have to specify the QoS profile for RT and NRT sessions. For RT sessions the simulator considers two QoS profiles, i.e., a low bandwidth profile comprising a guaranteed bit rate of 64 kbps corresponding to streaming audio and a high bandwidth profile comprising a guaranteed bit rate of 192 kbps corresponding to streaming video. According to the RT traffic model presented in [KM98], we assume that 80% of the RT sessions utilize the low bandwidth profile whereas the remaining 20% utilize the high bandwidth profile. Following the single user traffic model [KN00], [LLT02], NRT sessions are partitioned according to different bandwidth classes as follows: 60% for 64 kbps, 30% for 144 kbps, and 10% for 384 kbps, comprising different priorities (see Table 6.2), respectively. We refer to [LLT02] for details of the NRT traffic model, especially for the parameterization of the traffic

	Circuit switched voice service	Streaming real time (RT)		Interactive non real time (NRT)		
		Audio	Video	high priority	normal priority	low priority
Portion of arriving requests	25%	12%	3%	6%	18%	36%
Session duration	120 sec	180 sec		determined by session volume distribution		
Session dwell time	60 sec	120 sec		120 sec		

Table 6.2. Characteristics for different UMTS session types

characteristics. A thorough approach to traffic modeling for 3G mobile networks is considered in [Loh04].

We assume the duration of voice calls and RT sessions to be exponentially distributed. As proposed in [BJ99], the dwell time is modeled by a lognormal distribution. All corresponding mean values are shown in Table 6.2. A NRT session remains active until a specific data volume drawn according to a bandwidth-dependent lognormal distribution is transferred. To distinguish between NRT traffic classes, the UMTS simulator implements a weighted fair queueing (WFQ, [DKS89]) scheduler with three packet priorities. These priorities correspond to the traffic handling priorities specified by 3GPP. To model the user behavior in the cell, the simulator considers the handover flow of active mobile users from adjacent cells. Similar to Chapter 5, the iterative procedure introduced in [ACL+99] is employed for balancing the incoming and outgoing handover rates. The iteration is based on the assumption that the incoming handover rate of a user class at step i+1 is equal to the corresponding outgoing handover rate computed at step i.

The simulator exactly mimics UMTS system behavior on the IP level. The focus is not on studying link level dynamics. Therefore, we assume a reliable link layer as provided by the automatic repeat request (ARQ) mechanism of the radio link control (RLC) protocol. As shown in [Mey99] for GPRS, the ARQ mechanism is fast enough to recover from packet losses before reliable protocols on higher layers (e.g. TCP) recognize these losses due to timer expiration. Thus, a reliable link level can be assumed when considering higher layer protocol actions (see e.g., [LKJ99]). To accurately model the UMTS radio access network, the simulator represents the functionality of one radio network controller and seven Node B transceiver stations, one for each of the considered cells. Since in the end-to-end path, the wireless link is typically the bottleneck, and given the anticipated traffic asymmetry, the simulator focuses on resource contention in the downlink (i.e., the path RNC → Node B → MS) of the radio interface.

The simulator considers the UTRAN access scheme based on Wideband Code Division Multiple Access (WCDMA) in Frequency Division Duplex mode (FDD) proposed by 3GPP. In FDD downlink, a division of the radio frequencies into 4 physical code channels with data rates of 1,920 kbps each up to 512 physical code channels with 15 kbps data rates each is possible. Therefore, the overall bandwidth B that is available in one cell is 7,680 kbps. We assume this bandwidth to be constant over time. Considering an overall bandwidth B(t) depending on the actual interference situation is beyond the scope of this simulation study and is subject for future research. For the channel coding, we assume a convolution-coding scheme with coding factor 2. In the experiments without adaptive control the handover bandwidth portion b_h is 5% and the NRT queue threshold η is set to 95%. The simulation environment was implemented using the simulation library CSIM [CSIM]. In a pre-simulation run the handover flow is balanced, for each cell at the boundary of the seven-cell cluster. All simulation results are derived with confidence level of 95% using the batch means method. The execution of a single simulation run requires about 40 to 60 minutes of CPU time (depending on the call arrival rate) on a dual processor Sun Sparc Enterprise with 1 GByte main memory.

6.4 Evaluation of the Framework for Adaptive QoS Management

Using simulation experiments, we illustrate the benefit of the proposed approach for adaptive QoS management of UMTS systems. In particular, we show the improvement of QoS measures. The presented curves plot the mean values of the confidence intervals for the considered QoS measures. In almost all figures, the overall call/session arrival rate of new mobile users is varied to study the cell under increasing load conditions. For ease of notation, results with and without adaptive QoS management are abbreviated by *APM on* and *APM off*, respectively.

In a first experiment, we investigate the effect of adaptive control on the threshold for the buffer size of the NRT queue (denoted by η). Figure 6.4 shows the NRT packet loss probability (left side) and the average number of NRT users in the cell (right side) for the UMTS system with and without adaptive control. Furthermore, the figures distinguish between different desired loss levels β as introduced in Section 6.2.2. We observe that the APM achieves a substantially decrease in packet loss probability. Moreover, the packet loss probability can be kept below a constant level for increasing arrival rates of mobile users. Note that this level slightly differs from the desired level of the QoS measure. This is due to the fact that the update function only decreases the NRT threshold if the online measured packet loss probability is greater than β. Therefore, the packet loss probability is in steady-state also slightly greater than β. Nevertheless, Figure 6.4 shows that the resulting packet loss probability can be adjusted quite well. For very low arrival rates, the packet loss probability is increased compared to the case without adaptive control. This is, because the packet loss probability is below the desired level and η is adjusted towards 100%.

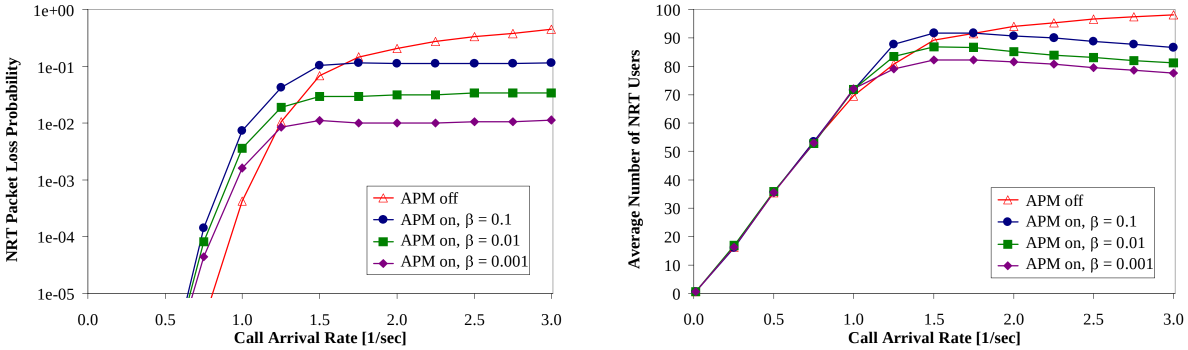


Figure 6.4. Impact of adaptive QoS management on non real-time traffic

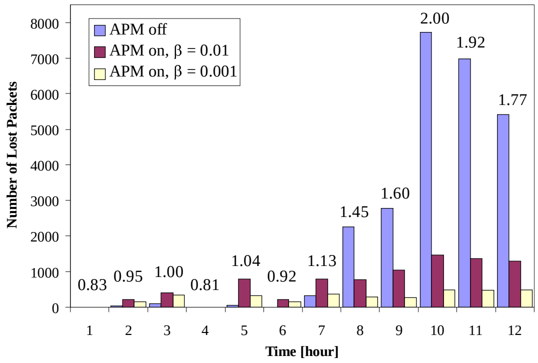


Figure 6.5. Number of packet losses for a half-day window of a weekly usage pattern

Figure 6.4 (right side) shows the average number of NRT users admitted in the cell. For all curves, the number of NRT users in the cell first increases up to about 70 users for an arrival rate of 1.0 arrivals per second. For higher arrival rates the admission controller decides to reject requests depending on the choice of the NRT threshold. In the case without APM the number of NRT users approaches 100 whereas in the cases with adaptive control less users are admitted in the cell because the threshold parameter η is decreased (e.g. about 80 users for β = 0.001). For high arrival rates a slightly decrease of the average number of NRT users can be observed. This is due to the fact that with increasing arrival rate the competition between voice, RT and NRT traffic decreases the bandwidth capacity available for NRT traffic. Therefore, less NRT users are admitted.

In the experiment presented in Figure 6.5, we study the absolute number of packet losses observed in one hour for a transient scenario, i.e., the arrival rate of new calls is changing every hour according to a half day window of a weekly usage pattern [KLL01]. The purpose of this experiment is to show that the adaptive QoS management is fast enough to react on changing traffic conditions, i.e., to effectively adjust the NRT threshold in order to reduce packet losses. The bars shown in Figure 6.5 correspond to the number of packet losses for experiments with and without adaptive control. Furthermore, the figure distinguishes between a desired loss level β of 0.01 and 0.001, respectively. The new call arrival rates considered in one hour are depicted above the bars. We conclude from Figure 6.5 that for a real-life pattern of changing arrival rates the packet losses can be effectively controlled by the APM. This justifies the choice of the gradient m = -0.02 in the update function for the NRT threshold.

Next, we study the effect of the APM on the handover traffic. Figure 6.6 shows the handover failure probability (left side) and the new call blocking probability (right side) for the UMTS system with and without APM. Similar to Figure 6.4, we distinguish between different desired levels β for the handover failure probability. The desired level for new call blocking is fixed to 0.1. Note that for controlling the handover bandwidth the desired level β can be used only to adjust the degree of prioritization of handover failure over new call blocking. Distinct from the packet loss probability, it cannot be expected to keep the handover

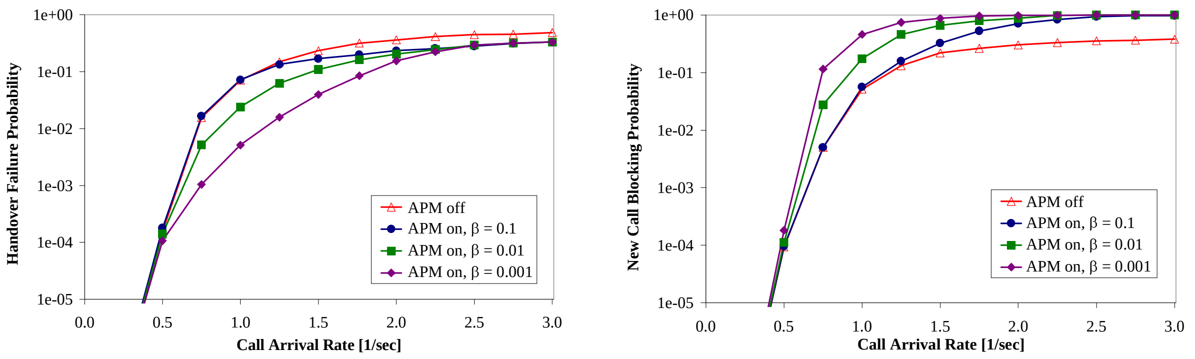


Figure 6.6. Impact of adaptive QoS management on handover traffic

failure probability at a constant level for increasing traffic load. That is for two reasons: (i) the handover bandwidth is adjusted according to two QoS measures that have a contrary influence and (ii) the increase of the handover bandwidth must be limited by a certain portion of the overall available bandwidth (see Section 6.2.2). If this limit is reached handover failures occur more frequently for further increasing call arrival rate. These two effects can be observed in the curves of Figure 6.6. Nevertheless, the handover failure probability is improved more than one order of magnitude for call arrival rates between 0.75 and 1.25 call requests per second and a desired loss level β = 0.001. When studying the blocking probability of new voice calls and RT sessions (see Figure 6.6 right side), we surly observe a higher blocking probability of new calls in the case with adaptive control and high arrival rate. In fact, almost all call requests are blocked if system load is high.

In a further experiment, we study the effect of QoS provisioning for real-time sessions, i.e., the probability of violating the QoS guarantees of corresponding QoS profiles. Figure 6.7 plots curves for the probability of real-time session degradation below the guaranteed bitrate. Recall that the degradation of real-time sessions steams from the introduction of the voice priority preempt packets bandwidth b_v. That is, a newly arriving voice call will be admitted if less than the bandwidth portion b_v is allocated to voice calls even than if bandwidth guarantees for RT sessions will not be fulfilled anymore. Therefore, we studied the probability of RT session degradation for different values of b_v. Note that for b_v = 0% the

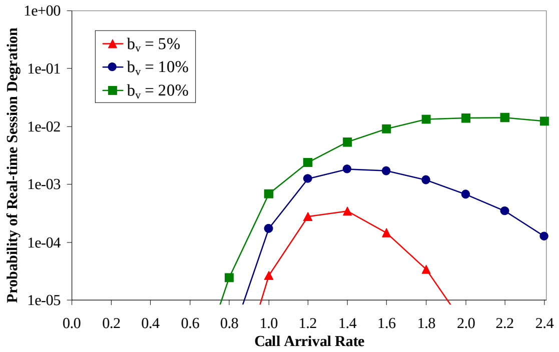


Figure 6.7. Probability of real-time sessions degradation for different values of b_v

probability of RT sessions degradation equals zero. Note that the degradation of RT sessions is performed stepwise. The probabilities shown in Figure 6.7 correspond to a degradation step of 32 kbps assuming that the real-time session continues in lower quality (e.g., only the audio component of a video real-time session is transmitted). The probability of degrading a real-time session more than one step was in all experiments less than 10^{-5}. An interesting effect that can be observed from the curves of Figure 6.7 is that for increasing traffic load the probability of RT session degradation first increases and then decreases. The decrease for high traffic load is due to the fact that RT sessions get no chance to be accommodated in the voice priory preempt packets area because at all time there are sufficient voice calls in the cell. From Figure 6.7 we conclude that average bandwidth requirements can almost always be maintained. In fact, the degradation of RT sessions below their guaranteed bandwidth requirements takes place only in 1 out of 1,000 sessions for $b_v = 10\%$.

6.5 Discussion of the Proposed Approach

This section provides a comparison of the framework for adaptive QoS management presented in this chapter with the approach for adaptive QoS/revenue management proposed in [LLT03b].

In [LLT03b], it is shown how online management of both QoS and provider revenue can be performed in 3G mobile networks by adaptively adjusting an admission controller to changing traffic conditions. As a main result, the approach is based on a novel call admission control and bandwidth degradation scheme for real-time traffic. Real-time calls with two priority levels are considered: calls of high priority have a guaranteed bitrate whereas calls of low priority can be temporarily degraded to a lower bitrate in order to reduce forced termination of calls due to handover failures. Opposed to previous work [CS02], [MHT02], a graceful degradation of bandwidth in several steps is considered. Furthermore, calls of low priority are degraded equally rather than picking out one call randomly for degradation. Clearly due to fairness reasons this approach should be preferred over a random choice of calls as applied in [CS02].

A second contribution in [LLT03b] constitutes the development of a Markov model for the admission controller that incorporates important features of 3G cellular networks, such as CDMA intra- and inter-cell interference [Lee91] and soft handover [KAL+01]. From the online quantitative analysis of the Markov model the threshold for maximal call degradation is periodically adjusted according to the currently measured traffic in the radio access network and a predefined optimization goal. Three different goals for the optimization of QoS are considered: (i) minimizing call degradation subject to a hard constraint on handover failure probability, (ii) maximizing a QoS function, and (iii) maximizing a QoS/revenue function.

Curves for measures of interest derived from the numerical steady-state analysis of the Markov model are presented in [LLT03b]. Besides the evaluation of the optimization goals, the proposed degradation scheme is compared with existing admission control policies based on adaptive guard channels [CS00], [ZL01]. It is shown that overall utilization of cell capacity is higher with the degradation scheme, which can be considered as an “on-demand” reservation of cell capacity whereas the guard channel scheme implements an “a-priori” reservation. Thus, the degradation scheme could be the method of choice in future mobile networks, which support service degradation, since it can guarantee a certain handover failure probability and also high capacity utilization.

In order to contrast the approaches considered in this chapter and in [LLT03b] let us summarize the main distinguishing aspects of these two approaches:

- Prioritization of handover calls based on *bandwidth reservation* vs. *bandwidth degradation*
- *QoS monitoring* vs. *traffic monitoring*
- Adaptive control based on *heuristical formulas* vs. *Markov model*

Comparing the effectiveness of an adaptive bandwidth degradation scheme with adaptive bandwidth reservation schemes (i.e., guard channel schemes), it can be concluded that for 3G networks with different QoS classes and call priorities, the graceful degradation of bandwidth should be the method of choice for prioritization of handover calls. This is because in the guard channel scheme is a high probability that a new call request will be rejected, although bandwidth is still available, i.e., the guard channels are unused (see Figures 14 and 15 in [LLT03b]). Contrasting monitoring of QoS measures versus monitoring of traffic characteristics, it can be concluded that traffic monitoring is only applicable for traffic with smooth characteristics, e.g., the average arrival rate of calls or the average call duration. Online monitoring of the average arrival rate of IP packet traffic is an inadequate method to represent the traffic characteristics since packet traffic usually has a bursty arrival pattern, which cannot be represented by average values. On the other hand optimization of system parameters according to a Markov model can be only applied if the Markov model is fed with the current traffic pattern. Online monitoring of QoS measures and adjusting system parameters according to heuristical formulas is the method of choice if no efficient solvable Markov model for the system component to be controlled is available. If an accurate Markov model is available the system component can be optimized after each control period whereas with heuristical formulas a desired configuration can be only approximated in several control periods. Thus, the approach based on a Markov model allows a faster and more precise control of system parameters.

6.6 Summary

In this chapter, a framework for the adaptive management of QoS in 3G mobile networks is introduced. In general, the framework aims at improving both QoS for mobile subscribers and increasing revenue earned by service providers. Nevertheless, only the improvement of QoS is presented in this thesis. Adaptive management of QoS measures for improving provider revenue by adjusting the weights of a weighted fair queueing packet scheduler is considered in [LLT03a]. System parameters controlled by adaptive QoS management constitute the portion of bandwidth reserved for handover calls and the buffer threshold of the queue for non real-time traffic. Using the UMTS traffic model of [LLT02] and a simulator on the IP level for the UMTS system [Hän01], performance curves for various QoS measures are presented to illustrate the benefit of the approach for adaptive QoS management. Two update functions that effectively control the packet loss probability and the handover failure probability are introduced. Throughout this chapter, the services and QoS profiles standardized for UMTS are considered. Thus, the proposed approach for adaptive control is tailored to UMTS networks. However, by considering other services and QoS profiles, the basic ideas underlying the framework for adaptive QoS management can also be applied for the adaptive control of other kinds of multi-service IP networks.

Chapter 7

Concluding Remarks

This chapter summarizes the main results presented in this thesis and comments on some possible directions for future research. Most of the results are published in international journals or in the proceedings of major international conferences. In addition to the publications, this thesis provides substantial more in-depth information of the proposed solutions. This especially holds for the contributions presented in Chapter 4, which exceed the currently published results.

7.1 Main Results of this Thesis

The original research contributions presented in this thesis are summarized in three categories. In particular, this thesis proposes

(i) new techniques for stochastic modeling and quantitative analysis of discrete-event systems with UML diagrams,

(ii) new methodological results for the numerical transient and steady-state analysis of the stochastic process underlying a discrete-event system, i.e., a generalized semi-Markov process, and

(iii) new approaches for the adaptive QoS management for 3G mobile networks by dynamically adjusting system parameters of an admission controller.

With respect to (i), extensions to UML state diagrams and activity diagrams to allow the association of events with exponentially distributed and deterministic delays are proposed. A particular stochastic process, the generalized semi-Markov process (GSMP), is identified as the appropriate vehicle on which quantitative analysis is performed. The main contribution is the efficient algorithm for the automated derivation of the state space underlying a UML state diagram or activity diagram that additionally deals with history pseudostates as well as configuration-depended firing rates and branching probabilities. The applicability of the presented approach for the quantitative analysis of UML system specifications is illustrated throughout this thesis with several examples. In particular, an MMPP/D/1/K queue (Chapter 3), an UML model representing a single cell in a cellular system (Chapter 4), and a performance study for the General Packet Radio Service based on UML state diagrams (Chapter 5) are considered. Automated tool support for quantitative analysis is provided by DSPNexpress-NG, which provides open interfaces for utilizing the numerical solvers for GSMPs for quantitative evaluation of system specifications modeled as DSPNs or UML diagrams. Open-source software download of DSPNexpress-NG is provided on the Web [DSPN].

With respect to (ii), two theorems that provide the foundation for the effective algorithmic generation of the transition kernel of the general state space Markov chain (GSSMC) underlying a GSMP with exponential and deterministic events are presented. Key contributions constitute the observation that kernel elements can always be computed by appropriate summation of transient state probabilities of continuous-time Markov chains (Theorem 4.9). Furthermore, conditions on the building blocks of the GSMP are derived under which kernel elements of its GSSMC are constant; i.e., are independent of clock readings (Theorem 4.11). Thus, for such state transitions the GSSMC behaves like a discrete-time Markov chain. Applying Theorem 4.11, it is shown that for queueing systems with Markov-modulated arrival process almost all kernel elements are constant. In fact, the GSMP analysis methodology extends the class of numerically solvable stochastic processes. Prior to this work, no efficient method for numerical analysis of GSMPs with exponential and possibly concurrent deterministic events was known at all. From a theoretical point of view, the presentation of the analysis methodology provides a thorough understanding of the behavior of GSMPs of the considered type.

With respect to (iii), a framework for the adaptive QoS management of 3G mobile networks is introduced. A new approach for adaptive QoS management applied to system parameters of an admission controller for real-time and non real-time traffic is proposed. The approach proposes an admission controller based on a bandwidth reservation scheme. Controlled parameters constitute the portion of bandwidth reserved for handover calls and the buffer threshold of the queue for non real-time traffic. This approach is based on online monitoring of QoS measures like packet loss probability and handover failure probability. Based on the values of the QoS measures just observed, system parameters are adjusted according to a mathematical framework with heuristical formulas. The effectiveness of the approach is demonstrated by several experiments. A contrasting discussion of this approach and a second approach, which is currently submitted for publication [LLT03b], is provided.

Throughout this thesis, the services and QoS profiles standardized for UMTS are considered. Thus, the proposed approach for adaptive control is tailored to UMTS networks. However, by considering other services and QoS profiles, the basic ideas underlying the approach for adaptive QoS management can also be applied for the adaptive control of other kinds of multi-service IP networks.

7.2 Directions for Future Research

Some future research issues of interest building on the results of this thesis are summarized in the following:

- **Analysis of discrete-event systems:** Numerical transient and steady-state analysis of GSMPs with more than two concurrently enabled deterministic events. To achieve this goal, future work consist of the following three work packages: (1) refining the numerical integral equation solver in order to effectively deal with higher dimensional integrals; (2) exploitation of symmetries both in the integral equations (i.e., clock readings) and in the transition graph (repetitive model components); and (3) efficient parallelization of the GSMP analysis method for a network of workstations.
- **Adaptive QoS/revenue management:** Adaptive/Online optimization of both QoS for mobile users and provider revenue for 3G mobile networks by adaptive control of system parameters with respect to changing traffic conditions. Identification of dependencies between provider revenue maximization and the QoS demands of mobile users. Investigating the online management of traffic controlling components other than the admission controller, e.g. CDMA power control, packet scheduling of delay-sensitive traffic or adaptive routing [OO00]. Furthermore, congestion based pricing policies for revenue maximization could be considered [Hei02].
- **Cooperative call admission control:** Cooperative call admission control schemes for intra-system handovers in heterogeneous networks. Heterogeneous network architectures, e.g., Bluetooth, IEEE 802.11, HYPERLAN, UMTS, will be supported in the open radio access network architecture (OpenRAN) [MWIF01]. The heterogeneous character requires the mapping of the users' QoS profile to the QoS attributes of the currently serving access technology.

References

[3GPP] 3GPP, `http://www.3gpp.org`.

[3GPP00] 3GPP, Network architecture, *Technical Specification TS 23.002*, 2000.

[3GPP01a] 3GPP, QoS Concept and Architecture, *Technical Specification TS 23.107*, 2001.

[3GPP01b] 3GPP, UTRAN Overall description, *Technical Specification TS 25.401*, 2001.

[ABB+89] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani, The Effect of Execution Policies on the Semantics of Stochastic Petri Nets, *IEEE Trans. on Softw. Engin.* **15**, 832-845, 1989.

[ABC84] M. Ajmone Marsan, G. Balbo, and G. Conte, A Class of Generalized Stochastic Petri Nets for the Performance Analysis of Multiprocessor Systems, *ACM Trans. on Comp. Systems* **2**, 93-122, 1984.

[ABC+95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Francheschinis, *Modelling with Generalized Stochastic Petri Nets*, John Wiley & Sons, 1995.

[AC87] M. Ajmone Marsan and G. Chiola, On Petri Nets with Deterministic and Exponentially Distributed Firing Times, in: G. Rozenberg (Ed.), *Advances in Petri Nets 1986*, *Lecture Notes in Computer Science* **266**, 132-145, Springer, 1987.

[ACL+99] M. Ajmone Marsan, G. De Carolis, E. Leonardi, R. Lo Cigno, and M. Meo, How Many Cells Should Be Considered to Accurately Predict the Performance of Cellular Networks?, *Proc. European Wireless, Munich, Germany*, 1999.

[AH01] K. Aretz, M. Haardt, W. Konhäuser, and W. Mohr, The Future of Wireless Communications beyond the Third Generation, *Computer Networks* **37**, 83-92, 2001.

[AM00] M. Ajmone Marsan and M. Meo, Approximate Analytical Models for Dual-Band GSM Networks Design and Planning, *Proc. 19th IEEE Conf. on Computer Communications (Infocom), Tel-Aviv, Israel*, 2000.

[AMM+00] M. Ajmone Marsan, S. Marano, C. Mastroianni, and M. Meo, Performance Analysis of Cellular Mobile Communication Networks Supporting Multimedia Services, *Mobile Networks and Applications* **5**, 167-177, 2000.

[AZ97] E. Anderlind and J. Zander, A Traffic Model for Non Real-Time Data Users in a Wireless Radio Network, *IEEE Communications Letters* **1**, 37-39, 1997.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An Architecture for Differentiated Services, *Request for Comments 2475, Internet Engineering Task Force*, 1998.

[BBK98] F. Bause, P. Buchholz, and P. Kemper, A Toolbox for Functional and Quantitative Analysis of DEDS, *Proc. 10th Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation, Palma de Mallorca, Spain, Lecture Notes in Computer Science* **1469**, 356-359, Springer, 1998. `http://ls4-www.cs.uni-dortmund.de/APNN-TOOLBOX/`

[BCF+87] G. Balbo, G. Chiola, G. Franceschinis, and G. Molinar Roet, On the Efficient Construction of the Tangible Reachability Graph of Generalized Stochastic Petri Nets, *Proc. 2nd Int. Workshop on Petri Nets and Performance Models (PNPM), Madison, Wisconsin*, 85-92, 1987.

[BEM+00] K. Begain, M. Ermel, T. Müller, J. Schüler, and M. Schweigel, Analytical Comparison of Different GPRS Introduction Strategies, *Proc. 3rd ACM Int. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Boston, MA*, 3-10, 2000.

[BJ99] F. Barceló and J. Jordán, Channel Holding Time Distribution in Public Cellular Telephony, *Proc. 16th Int. Teletraffic Congress, Edinburgh, Scotland*, 107-116, 1999.

[BJR99] G. Booch, I. Jacobson, and J. Rumbaugh, *The Unified Modeling Language User Guide*, Addison Wesley, 1999.

[BL00] R.J. Boucherie and R. Litjens, Radio Resource Sharing in a GSM/GPRS Network, *Proc. 12th ITC Specialist Seminar on Mobile Systems and Mobility, Lillehammer, Norway*, 261-274, 2000.

[BW97] G. Brasche and B. Walke, Concepts, Services, and Protocols of the New GSM Phase 2+ General Packet Radio Service, *IEEE Comm. Magazine* **35**, 94-104, 1997.

[Cas93] C.G. Cassandras, *Discrete Event Systems, Modeling and Performance Analysis*, Aksen Associates, 1993.

[CKT93] H. Choi, V.G. Kulkarni, and K.S. Trivedi, Transient Analysis of Deterministic and Stochastic Petri Nets, in: M. Ajmone Marsan (Ed.), *Application and Theory of Petri Nets 1993, Lecture Notes in Computer Science* **691**, 166-185, Springer, 1993.

[CKT94] H. Choi, V.G. Kulkarni, and K.S. Trivedi, Markov Regenerative Stochastic Petri Nets, *Performance Evaluation* **20**, 336-353, 1994.

[CG97] J. Cai and D.J. Goodman, General Packet Radio Service in GSM, *IEEE Comm. Magazine* **35**, 122-131, 1997.

[CL99] G. Ciardo and G. Li, Approximate transient analysis for subclasses of deterministic and stochastic Petri nets, *Performance Evaluation* **35**, 109-129, 1999.

[CM00a] V. Cortellessa and R. Mirandola, Deriving a Queueing Network Based Performance Model from UML Diagrams, *Proc. 2nd Int. Workshop on Software and Performance (WOSP), Ottawa, Canada*, 58-69, 2000.

[CM00b] V. Cortellessa and R. Mirandola, UML Based Performance Modeling of Distributed Systems, in: A. Evans, S. Kent, B. Selic (Eds.), *3rd Int. Conf. on the Unified Modeling Language, York, UK*, *Lecture Notes in Computer Science* **1939**, 178-193, Springer, 2000.

[CMT89] G. Ciardo, J. Muppala, and K.S. Trivedi, SPNP: Stochastic Petri Net Package, *Proc. 3rd Int. Workshop on Petri Nets and Performance Models (PNPM), Kyoto, Japan*, 142-151, 1989.

[CS00] S. Choi and K.G. Shin, A Comparative Study of Bandwidth Reservation and Admission Control Schemes in QoS-Sensitive Cellular Networks, *Wireless Networks* **6**, 289-305, 2000.

[CS02] C.T. Chou and K.G. Shin, Analysis of Combined Adaptive Bandwidth Allocation and Admission Control in Wireless Networks, *Proc. 21th IEEE Conf. on Computer Communications (Infocom), New York*, 2002.

[CSIM] CSIM18-The Simulation Engine, `http://www.mesquite.com`.

[DF01] M. Dinis and J. Fernandes, Provision of Sufficient Transmission Capacity for Broadband Mobile Multimedia: A Step Towards 4G, *IEEE Comm. Magazine* **39**, 46-54, 2001.

[DJK+00] S.K. Das, R. Jayaram, N.K. Kakani, and S.K. Sen, A Call Admission and Control Scheme for Quality-of-Service Provisioning in Next Generation Wireless Networks, *Wireless Networks* **6**, 17-30, 2000.

[DKS89] A. Demers, S. Keshav, and S. Shenker, Analysis and Simulation of a Fair Queueing Algorithm, *Proc. ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM), Austin, Texas*, 1-12, 1989.

[DKS02] S. Derisavi, P. Kemper, and W.H. Sanders, The Möbius state-level abstract functional interface, *Proc. 12th Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation, London, UK, Lecture Notes in Computer Science* **2324**, 31-50, Springer, 2002.

[Dou99] B.P. Douglass, *Real-time UML: Developing Efficient Objects for Embedded Systems*, 2nd Edition, Addison Wesley, 1999.

[DP00] S. Dennett and G. Patel, The 3GPP and 3GPP2 Movements Towards an All-IP Mobile Network, *IEEE Personal Comm.* **7**, 62-64, 2000.

[DSPN] DSPNexpress-NG, `http://rul-www.cs.uni-dortmund.de/gsmp/`.

[Eng80] H. Engels, *Numerical Quadrature and Cubature*, Academic Press, 1980.

[ETSI98] ETSI, Universal Mobile Telecommunication System (UMTS); Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS, *Technical Report TR 101 112 v3.2.0*, 1998.

[ETSI99] ETSI, Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Service description; Stage 2, *GSM recommendation 03.60*, 1999.

[FG88] B.L. Fox and P.W. Glynn, Computing Poisson Probabilities, *Communications of the ACM* **31**, 440-445, 1988.

[FM93] W. Fischer and K. Meier-Hellstern, The Markov-modulated Poisson Process (MMPP) Cookbook, *Performance Evaluation* **18**, 149-171, 1993.

[Gau97] W. Gautschi, *Numerical Analysis: An Introduction*, Birkhäuser, Boston, 1997.

[Ger99] R. German, Cascaded Deterministic and Stochastic Petri Nets, *Proc. 3rd Int. Workshop on the Numerical Solution of Markov Chains, Zaragoza, Spain*, 111-130, 1999.

[GH99] R. German and A. Heindl, A Fourth Order Algorithm with Automatic Stepsize Control for the Transient Analysis of DSPNs, *IEEE Trans. Softw. Engin.* **25**, 194-206, 1999.

[GL94] R. German and C. Lindemann, Analysis of Stochastic Petri Nets by the Method of Supplementary Variables, *Performance Evaluation* **20**, 317-335, 1994.

[Gly89] P.W. Glynn, A GSMP Formalism for Discrete Event Systems, *Proc. of the IEEE* **77**, 14-23, 1989.

[GM84] D. Gross and D.R. Miller, The Randomization Technique as a Modeling Tool and Solution Procedure for Transient Markov Processes, *Operations Research* **32,** 343-361, 1984.

[Gra82] W.K. Grassmann, The GI/PH/1/ Queue: A Method to find the Transition Matrix, *INFOR* **20**, 144-156, 1982.

[GY92] P. Glasserman and D.D. Yao, Monotonicity in Generalized Semi-Markov Processes, *Math. of Operations Research* **17**, 1-21, 1992.

[Hän01] J. Hänsel, Entwicklung eines Verfahrens zum adaptiven Performance Management für UMTS Netze, *Diplomarbeit (Master Thesis), Department of Computer Science, University of Dortmund*, Nov. 2001.

[Har87] D. Harel, Statecharts: A Visual Formalism for Complex Systems, *Science of Computer Programming* **8**, 231-274, 1987.

[Hei02] T.M. Heikkinen, On Congestion Pricing in a Wireless Network, *Wireless Networks* **8**, 347-354, 2002.

[HMP+01] G. Haring, R. Marie, R. Puigjaner, and K.S. Trivedi, Loss Formulas and Their Application to Optimization for Cellular Networks, *IEEE Trans. on Vehicular Technology* **50**, 664-673, 2001.

[HWB00] J.F. Huber, D. Weiler, and H. Brand, UMTS, the Mobile Multimedia Vision for IMT-2000: A Focus on Standardization, *IEEE Comm. Magazine* **38**, 129-136, 2000.

[KAL+01] H. Kaaranen, A. Ahtiainen, L. Laitinen, S. Naghian, and V. Niemi, *UMTS Networks, Architecture, Mobility and Services*, John Wiley & Sons, 2001.

[Kle99] A. Klemm, Entwicklung eines quantitativen Bewertungsverfahrens für Statechart Modelle, *Diplomarbeit (Master Thesis), Department of Computer Science, University of Dortmund*, Nov. 1999.

[KLL01] A. Klemm, C. Lindemann, and M. Lohmann, Traffic Modeling and Characterization for UMTS Networks, *Proc. IEEE Global Telecommunications Conference (Globecom), San Antonio, Texas*, 1741-1746, 2001.

[KLL03] A. Klemm, C. Lindemann, and M. Lohmann, Modeling IP Traffic Using the Batch Markovian Arrival Process, *Performance Evaluation*, 2003 (to appear).

[KM98] M. Krunz and A. Makowski, A Source Model for VBR Video Traffic Based on M/G/∞ Input Processes, *Proc. 17th IEEE Conf. on Computer Communications (Infocom), San Francisco, California*, 1441-1449, 1998.

[KMM00] R. Kalden, I. Meirick, and M. Meyer, Wireless Internet Access Based on GPRS, *IEEE Personal Comm.* **7**, 8-18, 2000.

[KN00] J. Kilpi and I. Norros, Call Level Traffic Analysis of a Large ISP, *Proc. 13th ITC Specialist Seminar on Measurement and Modeling of IP Traffic, Monterey, California*, 6.1-6.9, 2000.

[KN01] M. Kim and B. Noble, Mobile Network Estimation, *Proc. 7th ACM Conf. on Mobile Computing and Networking (MobiCom), Rome, Italy*, 298-309, 2001.

[KP00] P. King and R. Pooley, Derivation of Petri Net Performance Models from UML Specifications of Communications Software, in: B.R. Haverkort, H.C. Bohnenkamp, C.U. Smith (Eds.), *11th Int. Conf. on Tools and Techniques for Computer Performance Evaluation, Schaumburg, Illinois, Lecture Notes in Computer Science* **1786**, 262-276, Springer, 2000.

[Lee91] W.C.Y. Lee, Overview of Cellular CDMA, *IEEE Trans. on Vehicular Technology* **40**, 291-302, 1991.

[Lin85] P. Linz, *Analytical and Numerical Methods for Volterra Equations*, SIAM, Philadelphia, 1985.

[Lin93] C. Lindemann, An Improved Numerical Algorithm for Calculating Steady-State Solutions of Deterministic and Stochastic Petri Net Models, *Performance Evaluation* **18**, 75-91, 1993.

[Lin95a] C. Lindemann, Exploiting Isomorphisms and Special Structures in the Analysis of Markov Regenerative Stochastic Petri Nets, in: W.J. Stewart (Ed.), *Computations with Markov Chains*, 383-402, Kluwer, 1995.

[Lin95b] C. Lindemann, DSPNexpress: A Software Package for the Efficient Solution of Deterministic and Stochastic Petri Nets, *Performance Evaluation* **22**, 3-21, 1995.

[Lin98] C. Lindemann, *Performance Modelling with Deterministic and Stochastic Petri Nets*, John Wiley & Sons, 1998.

[LKJ99] R. Ludwig, A. Konrad, and A.D. Joseph, Optimizing the End-to-End Performance of Reliable Flows over Wireless Links, *Proc. 5th ACM Conf. on Mobile Computing and Networking (MobiCom), Seattle, Washington*, 113-119, 1999.

[LLT02] C. Lindemann, M. Lohmann, and A. Thümmler, Adaptive Performance Management for UMTS Networks, *Computer Networks* **38**, 477-496, 2002.

[LLT03a] C. Lindemann, M. Lohmann, and A. Thümmler, A Unified Approach for Improving QoS and Provider Revenue in 3G Mobile Networks, *Mobile Networks and Applications* **8**, 209-221, 2003.

[LLT03b] C. Lindemann, M. Lohmann, and A. Thümmler, Adaptive Call Admission Control for QoS/Revenue Optimization in CDMA Cellular Networks, *(submitted for publication)*, 2003.

[Loh00] M. Lohmann, Realisierung direkter und iterativer Löser für Systeme zweidimensionaler Integralgleichungen, *Diplomarbeit (Master Thesis), Department of Computer Science, University of Dortmund*, Mar. 2000.

[Loh04] M. Lohmann, Traffic Modeling and Adaptive QoS/Revenue Management for Third Generation Mobile Networks, *Ph.D. dissertation, Department of Computer Science, University of Dortmund, (to appear)*, 2004.

[LS96] C. Lindemann and G.S. Shedler, Numerical Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions, *Performance Evaluation* **27&28**, 565-582, 1996.

[LT99] C. Lindemann and A. Thümmler, Transient Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions, *Performance Evaluation* **36&37**, 35-54, 1999.

[LT01a] C. Lindemann and A. Thümmler, Evaluating the GPRS Radio Interface for Different Quality of Service Profiles, in: U. Killat, W. Lamerdorf (Eds.), *Informatik aktuell: 12th GI/ITG Fachtagung Kommunikation in Verteilten Systemen (KiVS), Hamburg, Germany*, 291-301, Springer, 2001.

[LT01b] C. Lindemann and A. Thümmler, Performance Analysis of the General Packet Radio Service, *Proc. 21st Int. Conference on Distributed Computing Systems (ICDCS), Phoenix, Arizona*, 673-680, 2001.

[LT03a] C. Lindemann and A. Thümmler, Performance Analysis of the General Packet Radio Service, *Computer Networks* **41**, 1-17, 2003.

[LT03b] C. Lindemann and A. Thümmler, Numerical Analysis of Generalized Semi-Markov Processes, *(submitted for publication)*, 2003.

[LTK+00] C. Lindemann, A. Thümmler, A. Klemm, M. Lohmann, and O. Waldhorst, Quantitative System Evaluation with DSPNexpress 2000, *Proc. 2nd Int. Workshop on Software and Performance (WOSP), Ottawa, Canada*, 12-17, 2000.

[LTK+02] C. Lindemann, A. Thümmler, A. Klemm, M. Lohmann, and O. Waldhorst, Performance Analysis of Time-enhanced UML Diagrams Based on Stochastic Processes, *Proc. 3rd Int. Workshop on Software and Performance (WOSP), Rome, Italy*, 25-34, 2002.

[Luc93] D.M. Lucantoni, The BMAP/G/1 Queue: A Tutorial, *Lecture Notes in Computer Science* **729**, 330-358, Springer, 1993.

[Mey99] M. Meyer, TCP Performance over GPRS, *Proc. 1st IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, Mississippi*, 1248-1252, 1999.

[MHT02] Y. Ma, J.J. Han, and K.S. Trivedi, Call Admission Control for Reducing Dropped Calls in Code Division Multiple Access (CDMA) Cellular Systems, *Computer Communications* **25**, 689-699, 2002.

[MNT+00] Sz. Malomsoky, Sz. Nádas, G. Tóth, and P. Zarándy, Simulation Based GPRS Network Dimensioning, *Proc. 12th ITC Specialist Seminar on Mobile Systems and Mobility, Lillehammer, Norway*, 2000.

[MWIF01] Mobile Wireless Internet Forum (MWIF), OpenRAN Architecture in 3rd Generation Mobile Systems, *Technical Report MTR-007*, September 2001, `http://www.mwif.org`.

[Nel95] R. Nelson, *Probability, Stochastic Processes, and Queueing Theory*, Springer, 1995.

[OMG99] Object Management Group, RFP: UML Profile for Scheduling, Performance, and Time, *OMG Document ad/99-03-13*, March 1999, `http://www.omg.org`.

[OMG01a] Object Management Group, OMG Unified Modeling Language Specification, *OMG Document formal/2001-09-67*, September 2001, `http://www.omg.org`.

[OMG01b] Object Management Group, Response to the OMG RFP for Schedulability, Performance, and Time, *OMG Document ad/2001-06-14*, June 2001, `http://www.omg.org`.

[OO00] E. Oubagha and S. Oueslati-Boulahia, A Comparative Study of Routing Algorithms for Elastic Flows in a Multiservice Network, *Proc. 13th ITC Specialist Seminar, Monterey, California*, 12.1-12.10, 2000.

[PF95] V. Paxson and S. Floyd, Wide-Area Traffic: The Failure of Poisson Modeling, *IEEE/ACM Transactions on Networking* **3**, 226-244,1995.

[Pro03] D. Proba, Quantitative Analyse zeitbehafteter UML Diagramme mit Anwendungen im Bereich mobiler Kommunikationssysteme, *Diplomarbeit (Master Thesis), Department of Computer Science, University of Dortmund*, Apr. 2003.

[Prz00] J. Przybilke, Leistungsbewertung von General Packet Radio Service auf IP-Kernnetzen, *Diplomarbeit (Master Thesis), Department of Computer Science, University of Dortmund*, Oct. 2000.

[PS01] J.M. Peha and A. Sutivong, Admission Control Algorithms for Cellular Systems, *Wireless Networks* **7**, 117-125, 2001.

[PSJ00] D. Petriu, C. Shousha, and A. Jalnapurkar, Architecture-Based Performance Analysis Applied to a Telecommunication System, *IEEE Trans. Softw. Engin.* **26**, 1049-1065, 2000.

[Rhap] Rhapsody, `http://www.ilogix.com`.

[Ryd96] T. Ryden, An EM Algorithm for Parameter Estimation in Markov Modulated Poisson Processes, *Computational Statistics and Data Analysis* **21**, 431-447, 1996.

[Sch95] M. Schwartz, Network Management and Control Issues in Multimedia Wireless Networks, *IEEE Personal Comm.* **2**, 8-16, 1995.

[SGM95] E. de Souza e Silva, H.R. Gail, and R.R. Muntz, Efficient Solution for a Class of Non-Markovian Models, in: W.J. Stewart (Ed.), *Computations with Markov Chains*, 483-506, Kluwer, 1995.

[She93] G.S. Shedler, *Regenerative Stochastic Simulation*, Academic Press, 1993.

[SM00] P. Stuckmann and F. Müller, GPRS Radio Network Capacity and Quality of Service using Fixed and On-Demand Channel Allocation Techniques, *Proc. 51th Vehicular Technology Conference, Tokyo, Japan*, 2000.

[Smi90] C.U. Smith, *Performance Engineering of Software Systems*, Addison Wesley, 1990.

[SW98] C.U. Smith and L.G. Williams, Performance Evaluation of Software Architectures, *Proc. 1st Int. Workshop on Software and Performance (WOSP), Santa Fe, New Mexico*, 164-177, 1998.

[Ste94] W.J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, 1994.

[TH98] M. Telek and A. Horváth, Supplementary variable approach applied to the transient analysis of Age-MRSPNs, *Proc. 4th Int. Computer Performance and Dependability Symposium (IPDS), Durham, North Carolina*, 44-51, 1998.

[TH02] M. Telek and A. Horváth, Time Domain Analysis of Non-Markovian Stochastic Petri Nets with PRI Transitions, *IEEE Trans. Softw. Engin.* **28**, 933-943, 2002.

[Toge] Together, `http://www.togethersoft.com`.

[Tri02] K.S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, Second Edition, John Wiley & Sons, 2002.

[UMTS99] UMTS-Forum, UMTS/IMT-2000 Spectrum, *Report No. 6*, 1999.

[Wal00] O. Waldhorst, Realisierung eines Verfahrens zur Generierung des Kerns einer General State Space Markov Chain, *Diplomarbeit (Master Thesis), Department of Computer Science, University of Dortmund*, Mar. 2000.

[Whi80] W. Whitt, Continuity of Generalized Semi-Markov Processes, *Mathematics of Operations Research* **5**, 494-501, 1980.

[WWRF] Wireless World Research Forum (WWRF), `http://www.wireless-world-research.org`.

[ZL01] Y. Zhang and D. Liu, An Adaptive Algorithm for Call Admission Control in Wireless Networks, *Proc. IEEE Global Telecommunications Conference (Globecom), San Antonio, Texas*, 3628-3632, 2001.