Budapest University of Technology and Economics
Department of Telecommunications

Applied Methods for some Planning and Analysis Problems in Telecommunications Networking

Hung Tuan Tran

Budapest, 2002

hung@plan.hit.bme.hu

Budapesti Műszaki és Gazdaságtudományi Egyetem
Híradástechnikai Tanszék

Néhány alkalmazott módszer a távközlő hálózatok tervezéséhez és elemzéséhez

Tran Tuan Hung

Ph.D. disszertáció

Konzulens

Dr. Do Van Tien

Budapest, 2002

hung@plan.hit.bme.hu

Kivonat

A disszertáció a távközlési hálózattervezések és a teljesítőképességi vizsgálatok alkalmazott módszereivel foglalkozik.

A disszertáció első részében az ATM hálózat topológiai optimalizálásával kapcsolatos eredmények

- a komplex tervezési feladat pontos matematikai formalizáslása és megadása, és
- egy praktikus tervezési és optimalizálási folyamat ajánlása és implementálása heurisztikus algoritmusokkal.

A disszertáció további részei analitikus teljesítőképességi elemzéssel, pontosabban annak a második fázisával (az ún. analízis fázissal) foglalkoznak. A disszertáció ebben a témakörben elért tudományos eredményei két csoportba sorolhatók. Az első csoport a QBD-re vonatkozó eredményeket tartalmazza, amelyek a következők:

- a QBD-ra fejlesztett, numerikus módszerek részletes áttekintése és teljesítményi elemzése egy konkrét modellezési példán keresztül,
- ajánlások kidolgozása arra vonatkozóan, hogy mikor melyik módszert jobb használni.

Az eredmények második csoportja a következőket foglalja magában:

- a QBD-M modell és alkalmazhatóságának, valamint a numerikus módszereinek részletes áttekintése,
- két alternatív, hatékony módszer ajánlása a QBD-M folyamatok staciónárius megoldására,
- modell alkotás az MPLS hálózatokban található, terhelés-kiegyenlített útirányítást megvalósító routerekről, az ATM koncentrátorokról, valamint a DiffSserv architektúrában alkalmazható ütemezési mechanizmusról,
- az újonnan fejlesztett és meglevő numerikus módszerek alapos teljesítményi értékelése és összehasonlítása különböző szempontok alapján.

Budapest University of Technology and Economics
Department of Telecommunications

Applied Methods for some Planning and Analysis Problems in Telecommunications Networking

Hung Tuan Tran

Ph.D dissertation

Scientific supervisor

Dr. Tien Van Do

Budapest, 2002

hung@plan.hit.bme.hu

Abstract

The dissertation deals with two important issues in telecommunications networking, namely with network planning and with analytical performance evaluation. The main topics are presented according to their elaboration order.

The first part of the present dissertation deals with the task of topology optimization for an ATM network overlaid in the SDH infrastructure. The scientific contributions of this part are:

- an exact mathematical formulation of the topology planning task for an ATM network to be built on top of the existing SDH infrastructure,
- a proposal and elaboration of the practical solution for this topology optimization task.

The other parts of the dissertation deal with numerical methods of two dimensional markovian processes. The scientific contributions can be summarized in two groups. The first group is concerned with QBD processes related issues, that cover the followings:

- a survey and a complete quantitative comparison of numerical methods for QBD processes,
- proposals based on extensive experiments for the issue of what numerical method is more advisable to use under given conditions.

The second contribution group pertains to QBD-M processes, their theory and applications:

- an overview of the QBD-M model and its application range along with a survey on the latest numerical methods for QBD-M processes,
- proposal of two alternative numerical methods for infinite QBD-M processes and of a numerical method for finite QBD-M processes,
- model construction for performance analysis of a node performing multipath routing in MPLS networks, of an ATM concentrator and of a scheduling mechanism in DiffServ architecture,
- an in-depth performance comparison of some existing and newly developed numerical methods.

Acknowledgments

I would like to express my hearty thanks to all people who have provided invaluable assistance during my study towards the PhD degree.

First of all, I would like to acknowledge my supervisor Dr Tien Van Do, who introduced me to the direction of my research at preliminary times. I have learned a lot about working attitude and problem approach from him. I am profoundly indebted to him who gave me a lot of chances to make public my research results and complete my work. Without his continuous support and straight criticisms, I could not grow up to accomplish this study.

My deepest thank is dedicated to all the members of my family, who have been with me along a long time, through many difficult periods I underwent. In many times their lovely incentive helped me to retain my ambition and stamina to keep steering my course. Without their unbreakable faith and spiritual support, I could not be able to finish my study today.

I gratefully thank all the members of the network planning group at our department, particularly Dr. László Jereb, for their ingenious and versatile support during my study. It was a pleasure for me to work in such an agreeable environment. I would like to address my cordial gratitude to all of them.

Contents

A bírálatok és a védésről készült jegyzőkönyv a Dékáni Hivatalban elérhetők.

List of Tables

List of Figures

Part I

Introduction

Chapter 1

Introduction

In the field of today's telecommunications networking, (r)evolution and developments are witnessed at an increasingly rapid rate. New networking technologies and architectures, new network elements are proposed and prepared for implementation and deployment with respect to provisioning quality of services. As a natural consequence of this fast evolution and advance, many unsolved, or still not satisfactorily solved problems and tasks are subject of further engineering work and research. This thesis provides solutions to some problems arising in two such typical areas, namely in network planning and analytical performance evaluation.

To save a cost at the introduction of a network based on a new technology (e.g.: ATM), it is highly reasonable from both economic and engineering aspects to reuse as much the old infrastructure (e.g.: SDH) as possible during the deployment of the new one. This design requirement thus raises a crucial part of network planning issues we refer to as overlay network planning.

Moreover, performance of networks with new entities (be it transmission technology, architecture element etc.) should be analysed and evaluated carefully both before their deployment and during their operation. This is to gain complete control over the network and thus to assure services up to expectations. This kind of activities is widely known as performance evaluation [20, 59, 66]. Specifically, performance evaluation quantitatively identifies relevant performance parameters for the network and analytically assesses them by achieving appropriate performance analysis. The goal is to produce analytical results of performance measures of the given network and through them to make it tractable from the performance points of view. Apparently, in order to retrieve analytical results in a efficient way, the need for numerical methods is really brought out and initiates another important research area.

The contributions of this thesis are tightly related with the aforementioned overlay network planning and numerical methods for performance evaluation. The motivations and purposes of my research work are given in the next section.

1.1 Research motivations and purposes

Briefly, this dissertation addresses some solutions for the following issues

- developing a practical planning method for the optimal topology design of a realistic network, and
- developing and evaluating efficient numerical methods for some queueing models frequently applied to analytical performance evaluation of telecommunication systems.

Being motivated by realistic demands and in line with the overlay network planning aspects, the research work in the first part of the dissertation presents a practical planning method for the topological optimization of an ATM network to be built in the existing SDH infrastructure. The aims addressed in this research work are:

- to give an exact formulation of a planning task, which aim is to implement and deploy an ATM network on the top of the SDH infrastructure. Note that the idea of utilizing the existing SDH resources stems from the economic point of view and sharply corresponds to the realistic strategy of networking providers;
- to propose a practical approach to solve this planning task in an easy-to-implement and efficient way, which results in an ATM network with optimal topology.

Model-based analysis of communication systems has also received a huge amount of research work. A lot of support tools like HIT, MACOM, HiQPN [1], PEPSY, MOSEL [2], SHARPE [3] and many others have been or are being developed to assist efficient tasks of analysts. Staying in this streamline, throughout the next part of the dissertation, our work focuses mainly on the analysis stage of analytical performance evaluation. The aim of this work is to give an in-depth insight into solving techniques, and to develop new ones that serve as common efficient computational tools for performance analysis of numerous problems arising in the field of telecommunications networks and computer systems.

The survey of recent relevant literature clearly indicates that markovian two-dimensional queueing models have a wide range of applications. The most frequent markovian two-dimensional models are categorized into class of QBD (Quasi Birth Death) processes and its extended version, to which we refer under the name QBD-M[1] processes throughout this work. The use of these queueing systems is observed widespreadly in performance analysis covering issues of ATM technology, DiffServ architecture, mobile networks, computer systems and so on.

Being motivated by the fact that QBD and QBD-M processes often provide a good modelling approach, our research focuses intensively on the investigation and development of computational methods for steady state analysis of the aforementioned queueing systems.

The aim of the research work addresses the following issues:

[1]The name comes from Ram Chakka in earlier private communication between him and Tien V. Do

- Although some efficient numerical methods have been successfully developed for solving QBD processes, the lack of an overall comparison concerning the performance of different methods, including execution time, space requirement, numerical stability and accuracy, still remains. Our first probe is devoted to filling this gap and making a proposal of what method is more advisable to use in view of system's parameters.

- The second goal achieved by our work is to develop simple and efficient methods for steady state analysis of QBD-M processes. Numerical aspects considered during the development include time and space complexity, numerical accuracy and numerical stability.

- In the remainder of our work, we construct analytical models for some telecommunications systems enabling their performance analysis. The aim is two-fold. On the one hand, the wide range of application of QBD-M processes is pointed out and illustrated. On the other hand, a comprehensive comparison of computational methods is carried out to investigate benefits and drawbacks of each method.

1.2 Research methodology

1.2.1 Network planning

Network planning generally comprises the following steps in sequence [49]:

- Design issues determination: this is to clarify the incipient aspects and design tasks. For example, whether the would-be network is built on top of existing capabilities; which aspect should be the most important: network cost, its performance or manageability; what is the focused problem: node placement and sizing or link topology optimization etc.

- Input data collection: traffic demands, QoS requirements, device characteristics etc.

- Design task formalisation: this is a technical step which mathematically combines all the information retrieved in the previous steps in a consistent and systematical describing way.

- Design tool choice: according to the features of the formulated task, a quantitative method is developed for the design process.

- Presentation of design results and assessment of the proposed design method

The network planning part of the thesis strictly applies to the aforementioned design concept. The idea of building an ATM network with optimal topology on top of the existing SDH structure for given traffic demands and QoS requirements represents the principle of the first two steps. In the third step, the layered network model [47] is considered. The formulated task proves an integer programming problem, which seems to be NP-hard. Therefore, in the fourth step heuristic algorithms are utilized.

1.2.2 Analytical performance evaluation

Generally, analytical performance evaluation of any given system consists of two stages:

- **Modelling stage**: The background of analytical performance evaluation relies on queueing theory including both steady state and transient analysis. The chosen queueing model is expected to capture the essential characteristics of the system as much as possible. At the same time, it should remain analytically tractable.
- **Analysis stage**: Once the queueing model has been adopted, its numerical analysis regarding the steady state or/and transient behaviour is carried out. Performance measures of interest are computed and evaluated.

In the analysis stage, the need for efficient numerical methods is really brought out [69]. This motivated the research work (analytical assessment and development of numerical methods for QBD and QBD-M processes) presented in the remaining part of the thesis.

The research work relies firstly on the deep knowledge of the queueing models along with being aware of their existing methods. The background of queueing theory, matrix analytic principles and the use of basic numerical criteria are essentially utilized. The analytical results have been confirmed by both theoretical considerations and concrete numerical data.

1.3 Organization of the thesis

The structure of the dissertation is divided into three parts. The first part deals with the optimization model and practical approach for the topological design of an ATM network built on top of the SDH infrastructure. The second part is concerned with QBD process related issues. Within this part, Chapter 3 details mathematical description of the QBD concept and gives a survey on computational methods for steady state solution of both kinds of QBD processes (finite and infinite). Chapter 4 presents a performance comparison between mostly used computational methods via a case study of a processor system with breakdowns and repairs. Useful remarks related to the issue of which method is more advisable to use under given conditions are also discussed and pointed out in this chapter.

The third part of the dissertation deals with QBD-M processes, their theory and applications. Emphasis is put on how to obtain steady state solution of QBD-M processes in an efficient way and how to formulate some practical problems with QBD-M processes. The part begins with Chapter 5 that reviews the necessity of the QBD-M model in performance evaluation and system analysis along with the mathematical description of the QBD-M model and existing basic methods available for its steady state solution. In order to accomplish steady state analysis in a more efficient way, Chapter 6 presents the proposal of two new computational methods. In Chapter 7 the two-fold aim is achieved. On the one hand, by proposing and showing that

analytical models of two performance analysis problems trace back to QBD-M processes, the powerful applicability of the QBD-M model is demonstrated. On the other hand, the performance assessment of the newly developed methods is performed in comparison with existing methods. The comparative study highlights the advantages of each new method over the use of the existing ones.

Chapter 8 of the third part demonstrates the application of finite QBD-M model to the analysis of a priority queueing system fed by two kinds of traffic classes, which may play an important role in evaluation of scheduling mechanisms implemented in DiffServ environment. A novel steady state solution method is proposed to derive performance parameters of the queueing system. Finally, Chapter 9 summarises the main results of the dissertation.

Part II

Planning of an overlay ATM network in an SDH infrastructure

Chapter 2

Planning of an overlay ATM network in an SDH infrastructure

2.1 Introduction

State-of-the-art telecommunications and data services require more and more bandwidth, therefore, it is a modern tendency to develop high-speed transmission capabilities that can be simultaneously and economically used as an integrated infrastructure for different applications.

ATM network development to support advanced data and reliable multimedia communications is gaining the popularity of network operators who aggressively use this technology to drive new business initiatives, since an ATM backbone provides simple and effective network management for public networks and a scalable network solution for the later introduction of new integrated services as well. One can read more information on the pilot projects for the Wide Area ATM Deployment of the European operators in [4, 11] and recently the plan of the international joint venture of Deutsche Telekom, France Telekom and Sprint for the launch of a seamless Global ATM service via one of the largest and most advanced ATM-based networks in the world, which will be offered in 13 countries: Belgium, Canada, Denmark, France, Germany, Ireland, Israel, Japan, the Netherlands, Sweden, Switzerland, United Kingdom and the United States [15]

Moreover, an existing SDH infrastructure is often utilized as the transport network for ATM in order to decrease the initial deployment cost of the ATM network. For example, the Hungarian Telecommunications Company Ltd. (HTC) has implemented a country-wide high-speed (622 Mbps-2.4 Gbps) SDH based backbone network ([62]) with several network protection opportunities, that is being used as an efficient transport layer for the ATM network to fulfill the continuously growing demands initiated towards HTC.

This chapter presents an approach for planning an ATM network which is implemented on top of the SDH infrastructure and is intended for serving high-speed advanced data communications.

To design such a network we decompose the problem into the following subproblems:

1. optimize the topology of an ATM network which includes
   - selecting the location of ATM access switches and the assignment of users to the switches to minimize the unused SDH capacity,
   - selecting the location for transit switches to efficiently carry traffic among the users,
2. realize the ATM links in the SDH infrastructure (optimal routing for ATM links in the SDH)

The solutions for the above subproblems are obtained by using heuristic algorithms. Moreover, the planning system is designed in such a way that allows network planners to set up a complete planning process. The solution of the above subproblems is being implemented in the XPLANET tool developed for the support of the network planning activities of HTC [47]. The tool is being applied for the determination and comparison of some possible network solutions based on the scenarios where the public ATM network is introduced on top of the SDH infrastructure.

The main contribution of the research work presented in this chapter is the proposal of the mentioned practical planning approach. Being constructed from sophisticated theoretical considerations and appropriately modified heuristic algorithms, this planning approach is expected to have a great applicability in some related design tasks. Moreover, its credit is supported by some possible extensions as explained later.

The chapter is organized as follows. Section 2.2 describes main considerations that serve as starting points of the planning process. In Section 2.3 the network planning approach is presented. In Section 2.4 an exact mathematical formulation and the proposed optimisation procedures are discussed. In Section 2.5 some results on a realistic scenario are presented. Section 2.6 deals with the assessment of the proposed planning approach. Finally, Section 2.7 concludes this chapter.

2.2 Planning considerations

2.2.1 Network architecture

The target network is partitioned into two levels: the access and backbone levels. In more detail, both in the access and in the backbone subnetworks further network sublevels can be identified. The access network provides the access of users to the access switches. It is composed of line terminators, multiplexers and concentrators, and normally has a tree topology. The backbone network with a two-level hierarchical architecture is currently under consideration (Figure 2.1). The higher level consists of transit switches and interconnects the access switches of the regional

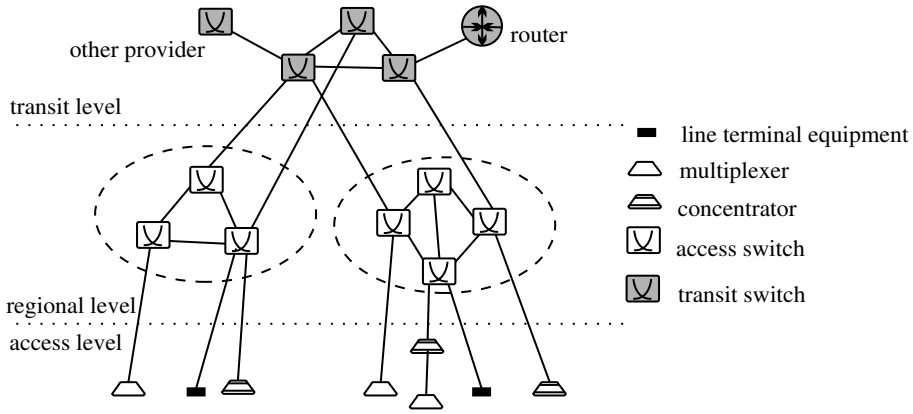


Figure 2.1: The considered ATM network hierarchy

subnetworks. For data communication services (such as LAN/MAN interworking) multiprotocol routers are used. The multiprotocol routers are usually connected to the transit nodes.

The network provides point-to-point and point-to-multipoint connections which can be flexibly allocated, maintained and reallocated by the management system. The transit and access ATM switches are sophisticated multiservice devices and can be easily extended as the user demand grows.

2.2.2 Realization of the ATM network in the SDH infrastructure

We consider the case when the ATM network is built on top of an SDH infrastructure.

The following assumptions are made:

- The public ATM access and transit switches are placed into the nodes of the SDH network, thus all the ATM links are realized and routed in the SDH infrastructure.
- The ATM network service will be required by business subscribers with ATM interface. These users are not located in the nodes of the SDH network, therefore, the access of users to the SDH infrastructure must be provided. The ATM link between a specific user and its first ATM access switch is realized with two consecutive sections. For the first section new optical cable will be deployed between the site of the user and the nearest SDH ADM (Add and Drop Multiplexer) point in the SDH infrastructure[1].

The cost of deploying access cables is fixed and depends only on the location of users. Therefore, this part of the network can not be optimised. The second section will be realized only in the SDH infrastructure. Since the physical capacity of the second section in the SDH infrastructure is not shared with other users in the ATM layer[2], the physical capacity of the

[1] if it has been already implemented, the particular cost for the user is decreased.

[2] in contrast with the SDH physical capacity interconnecting the ATM switches

second section is not efficiently used if the bandwidth requirement is below the port rate.

2.3 Planning method

2.3.1 Modelling the cost of the SDH usage

The following function is introduced to model the cost for the SDH usage of the ATM network on the SDH infrastructure:

$$f(i, j, C) = g_1(C)(n_{ij}\gamma + d_{ij}\lambda) + g_2(C), \tag{2.1}$$

where

- C denotes the required speed of an ATM link whose endpoints are i and j, to be realized in the SDH infrastructure. $g_1(C)$ and $g_2(C)$ are stepwise functions to take into account the fact that in ATM networks link capacities are of discrete value.
- n_{ij} is the number of SDH hops (number of ADMs) and d_{ij} is the length of the ATM link between node i and j in the SDH infrastructure.
- γ and λ are the cost coefficients.

2.3.2 Formal description of the planning task

Formally, the design problem for an ATM network topology built on top of an SDH infrastructure can be stated as follows:

Given

- the SDH infrastructure,
- the list of users and their access points to the SDH infrastructure,
- users' requirements (port speed, bandwidth and QoS requirements),
- cost and characteristics (e.g. capacity) of switches, routers and the usage of the SDH infrastructure,
- the potential location of switches (it is assumed that the ATM switches are realized in the nodes of the SDH infrastructure).

Objective: is to find an ATM network topology in order to minimize the network cost and the SDH usage.

2.3.3 Practical approach

Since the problem described in the previous subsection is complex enough and very impractical to solve in a single step, we elaborate a practical approach as follows.

First, we decompose and identify the planning task into two main subtasks:

1. optimize the topology of an ATM network which includes
   - select the location of ATM access switches and the assignment of users to the switches to minimize the unused SDH capacity,
   - select the location for transit switches to efficiently carry traffic among the users,
2. realize the ATM links in the SDH infrastructure (optimal routing for ATM links in the SDH)

The decomposition is carried out in such a way that allows the reuse of the code of the optimization algorithms (see Section 2.4) applied for problems. Moreover, the input interface of the program modules for each subproblems are designed in order to allow the network planners to build a complete planning process for this problem.

Secondly, due to the fact that the two subtasks are interrelated the solution for the planning problems can only be obtained by carrying out the appropriate subtasks iteratively.

The solution to the above problems is implemented in the framework of the XPLANET software package by using the generic network model, and therefore, both the optimization library of XPLANET and its graphical user interface are directly applied to this planning problem.

2.4 Models and heuristic algorithms for the components of the planning approach

2.4.1 Optimal topology design

Notation for input data is introduced as follows:

- $U = \{1, ..., |U|\}$ denotes the set of users.
- $T = \{T_{hi}\} \quad (h, i \in U)$ is the traffic matrix.
- G_i is the speed of a line connecting user i to the network (specified by the user and may depend on the interface required by the user).
- S is the set of potential sites.
- μ_{ij} $(i \in U,\ j \in S)$ is the cost of connecting user i to site j.

- ν_{kj} $(k, j \in S)$ is the cost of connecting sites k and j.

  Both μ_{ij} and ν_{kj} has the form of (2.1).

- without the loss of generality we assume that W is the switching capacity of a switch used in the network (it limits the maximal number of the switch ports),

- α is the installation cost of a switch.

Binary decision variables are defined as follows:

- $u_j = 1$ $(j \in S)$ if a switch is installed in site j, otherwise $u_j = 0$

$$\mathbf{u} = \{u_j : j \in S\}.$$

- $x_{ij} = 1 \quad (i \in U,\ j \in S)$ if a switch is implemented in site j and user i is connected to a switch in site j; otherwise $x_{ij} = 0$

$$\mathbf{X} = \{x_{ij} : \ (i \in U,\ j \in S)\}.$$

- $y_{kj} = 1 \ \ (k, j \in S)$ if sites k and j are interconnected with a link of capacity C_{kj}; otherwise $y_{kj} = 0$

$$\mathbf{Y} = \{y_{kj} : \quad (k \in S,\ j \in S)\}.$$

- $z_{hikl} = 1 \quad (h, i \in U), \quad (k, l \in S)$ if traffic T_{hi} between user h and i is routed in a link between sites k and l; otherwise $z_{hikl} = 0$

$$\mathbf{Z} = \{z_{hikl} : \quad h, i \in U,\ k, l \in S\}.$$

A model for an integer programming problem is formulated as follows:

Model-A

$$\min_{(\mathbf{u},\mathbf{X},\mathbf{Y},\mathbf{Z})} \sum_{j \in S} u_j \alpha + \sum_{i \in U} \sum_{j \in S} \mu_{ij} x_{ij} + \sum_{k \in S} \sum_{j \in S} \nu_{kj} y_{kj}, \tag{A1}$$

subject to

$$\sum_{j \in S} x_{ij} = 1 \quad (i \in U), \tag{A2}$$

$$x_{ij} \leq u_j, \tag{A3}$$

$$y_{kj} \leq u_j \text{ and } y_{kj} \leq u_k \quad (k \in S, j \in S), \tag{A4}$$

$$\sum_{i \in U} x_{ij} G_i + \sum_{k \in S} y_{kj} u_j C_{kj} \leq u_j W \quad (j \in S), \tag{A5}$$

$$z_{hikl} \leq y_{kl} \quad (h, i \in U), \quad (k, l \in S), \tag{A6}$$

$$\sum_{h,i \in U} z_{hikl} T_{hi} \leq y_{kl} C_{kl} \quad (k, l \in S). \tag{A7}$$

In Model-A, expression (A1) refers to the total cost of the network. Constraints (A2) and (A3) enforce that a user should be connected to a switch. Constraint (A4) expresses that a link can only be established between sites of switches. Constraint (A5) represents that the processing capacity of the switch is limited. Constraints (A6) and (A7) implicitly include the PVC routing problem.

To solve this optimization problem we decompose Model-A into two submodels. The solution of the planning task can be obtained by applying the appropriate combination of two well-known heuristic algorithms.

For example the following procedure can be set up for planning a network with a two-level hierarchical architecture. First, we select the location of ATM access switches and the assignment of users to the switches to minimize the unused SDH capacity. Second, we optimize the topology of the regional subnetwork. Third, we identify the location for transit switches to efficiently carry traffic among the users. Fourth, we design the topology of the transit network. We can observe that the first and the third planning task can be carried out with the same algorithm. This is also true for the second and the fourth task.

The submodels and heuristic algorithms are presented in the next section.

2.4.1.1 Switch location and assignment

A model for an integer programming problem is formulated as follows:

Model-B

$$\min_{(\mathbf{u},\mathbf{X})} \sum_{j\in S} u_j\alpha + \sum_{i\in U}\sum_{j\in S} \mu_{ij}x_{ij}, \tag{B1}$$

subject to

$$\sum_{j\in S} x_{ij} = 1 \quad (i \in U), \tag{B2}$$

$$x_{ij} \le u_j, \tag{B3}$$

$$\sum_{i\in U} x_{ij}G_i + KCu_j \le u_jW \quad (j \in S), \tag{B4}$$

where we assumed that each switch is connected to at least K other switches with capacity C.

At first glance, one can observe that this is similar to the concentrator location problem which can be solved with heuristic algorithms (for example the Center of Mass, the Add, Drop algorithm[3]). However, there are some differences between the concentrator location problem and this problem. Namely, in the concentrator location problem there is a center and the capacity of the center is assumed to be infinite. In our work, we modified the Add algorithm for our problem. The applied algorithm is of greedy nature. It evaluates the savings obtainable by adding a switch at each site. Then it greedily selects the switch which saves the most money.

[3]The interested reader may find more details on these algorithms in [49]

After each switch is selected, the savings by adding an additional switch are changed so all the potential savings are reevaluated. Moreover, it is also checked whether the capacity of switch is exceeded because of directly attached users.

In order to decrease the search space we can determine the minimum number ($N = \sum_{i \in S} u_i$) of necessary switches. From (B4) one obtains

$$\sum_{j \in S} \sum_{i \in U} x_{ij} G_i + KCn \leq NW, \tag{2.2}$$

and it follows:

$$\frac{\sum_{i \in U} G_i}{W - KC} \leq N. \tag{2.3}$$

2.4.1.2 Topology optimization

The formal formulation of this problem is as follows.

Given:

- the SDH infrastructure and the cost model,
- a set of the switches' locations denoted by $S^{'}$ ($S^{'} \subseteq S$).
- the "traffic matrix" between the switches.

Design objective:

- is to find a topology of the backbone network in order to efficiently serve the user demands.

A model for an integer programming problem is formulated as follows:

Model-C:

$$\min_{(\mathbf{Y},\mathbf{Z})} \sum_{k \in S^{'}} \sum_{j \in S^{'}} \nu_{kj} y_{kj}, \tag{C1}$$

subject to

$$\sum_{i \in U} x_{ij} G_i + \sum_{k \in S^{'}} y_{kj} u_j C_{kj} \leq u_j W \quad (j \in S^{'}), \tag{C2}$$

$$z_{hikl} \leq y_{kl} \quad (h, i \in U), \quad (k, l \in S^{'}), \tag{C3}$$

$$\sum_{h,i \in U} z_{hikl} T_{hi} \leq y_{kl} C_{kl} \quad (k, l \in S^{'}). \tag{C4}$$

The routing is implicitly included in this problem, therefore it seems that this problem can not be solved as a linear programming problem. Moreover, for practical size heuristic techniques must be used. For our purpose we use the MENTOR algorithm, which is capable of creating a low cost, efficient network [50].

2.4.2 ATM link routing in the SDH infrastructure

The formal formulation of this problem is stated as follows.
Given:

- ATM physical link demands between ATM switches and between users and switches
- the underlying SDH infrastructure

Design objective:

- determine the optimal routing of ATM links in the SDH infrastructure.

This problem can be formalized as a multicommodity flow problem and can be solved by standard linear programming techniques [30].

2.5 Numerical demonstrations

In this section we demonstrate the proposed planning approach with some numerical results obtained for a specific ATM network to be implemented on top of the Budapest and Hungarian core SDH networks. We present some preliminary results concerning the regional ATM network in the first scenario and some results concerning the hypothetical transit ATM network for Hungary in the second scenario, respectively.

2.5.1 Budapest metropolitan network

In this scenario the infrastructure is supposed to be the SDH network implemented in Budapest. The network structure consists of two levels: STM-4 rings and STM-16 mesh are deployed on the lower level and higher level, respectively, and it is depicted in Figure 2.2.

For the numerical study we suppose that a network will be implemented with ATM switches of 2.5 Gbps capacity. We also assume that the number of users is 42 and they all require the STM-1 access interface to the ATM network. We investigate two cases: in the first case the peak cell rate is allocated to connections in the ATM network, while in the second case the required bandwidth of connections is based on the effective bandwidth concept. For both cases connections are described by a two-state ON-OFF model where they are either in a busy (ON) state sending packets back-to-back at peak rate or in an idle (OFF) state sending no packets at all. The effective bandwidth of the connection, R is determined from the following approximation derived in [34]:

$$R = M\frac{\beta - X + \sqrt{(\beta - X)^2 + 4X\rho\beta}}{2\beta} \tag{2.4}$$

where

- M is the peak rate of the connection,
- m is the mean rate of the connection,
- b is the average duration of the busy period,
- $\beta = \ln\left(\frac{1}{\epsilon}\right) b(1-\rho)M$,
- $\rho = \frac{m}{M}$ is the probability that the connection is active,
- $X = D \times R$ is the buffer space required by the connection,
- D and ϵ are the QoS parameters required by the connection.

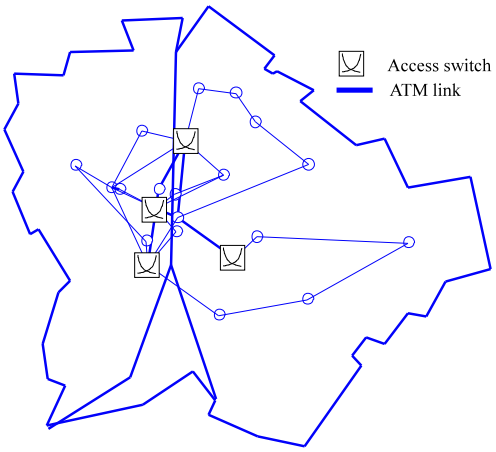
Table 2.1 shows the number of switches versus the cost coefficients λ and γ. It can be observed, as it is expected, that the larger the penalty for the unused capacity, the more switches are needed. In Table 2.2 we present the comparison of the two planning cases based on the peak cell rate and effective bandwidth scheme. It can be also observed that in average the cost of a network based on the peak cell rate allocation is 4-5 times more than the cost of a network based on the effective bandwidth allocation scheme.

In Figure 2.2 the logical topology of an ATM network structure on top of the SDH infrastructure is plotted in the simplest case when 4 ATM switches are allocated. The switch locations obtained with the approach show strong consistency with the SDH infrastructure since the transit nodes of the ATM network are chosen in the important points of the SDH network. Due to this fact this network can be considered as one of the candidates for the very first step toward the deployment of the multilevel ATM network architecture.

λ /γ	1e+03	1.35e+05	1.4e+05	2e+05	2.2e+05	2.4e+05	3.467e+05	3.4675e+05	1e+06
1e+03	4	5	5	5	5	6	9	9	9
2e+04	4	5	5	5	6	6	9	9	9
2.35e+04	5	5	5	6	6	6	9	9	9
2.36e+04	5	5	5	6	6	6	9	9	9
1.4e+05	5	7	8	8	8	8	9	9	9
1.6e+05	6	8	8	8	8	8	9	9	9
1.7e+05	6	8	8	8	8	8	9	9	9
1.8e+05	7	8	8	8	8	8	9	9	9
2.2e+05	7	8	8	8	8	9	9	9	9
2.4e+05	8	8	8	8	8	9	9	9	9
5e+05	8	9	9	9	9	9	9	9	9
1e+06	9	9	9	9	9	9	9	9	9

Table 2.1: Number of backbone switches versus λ and γ in Budapest

The ATM network on top of the SDH infrastructure



The SDH infrastructure in Budapest

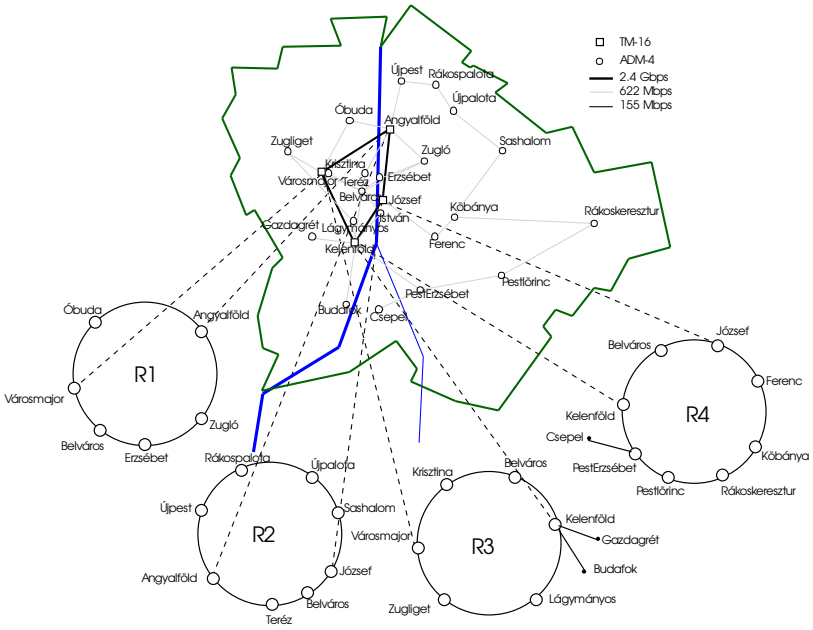


Figure 2.2: An ATM network on top of the SDH infrastructure in Budapest

λ /γ	0	1e+03	1.35e+05	1.4e+05	2e+05	2.2e+05	2.4e+05	3.467e+05	3.4675e+05	1e+06
0.00e+00	1.000	0.974	0.302	0.297	0.263	0.253	0.298	0.255	0.255	0.196
2.00e+04	0.431	0.427	0.278	0.275	0.244	0.262	0.255	0.255	0.255	0.185
2.35e+04	0.399	0.449	0.271	0.268	0.266	0.258	0.251	0.252	0.252	0.185
2.36e+04	0.451	0.448	0.271	0.268	0.266	0.258	0.251	0.252	0.252	0.185
1.40e+05	0.202	0.202	0.214	0.223	0.216	0.214	0.212	0.209	0.209	0.173
1.60e+05	0.208	0.208	0.216	0.215	0.210	0.208	0.206	0.204	0.204	0.171
1.70e+05	0.204	0.204	0.212	0.212	0.207	0.205	0.204	0.202	0.202	0.171
1.80e+05	0.212	0.212	0.209	0.208	0.204	0.203	0.201	0.200	0.200	0.170
2.20e+05	0.198	0.198	0.198	0.198	0.195	0.194	0.198	0.193	0.193	0.167
2.40e+05	0.198	0.198	0.194	0.194	0.192	0.191	0.195	0.190	0.190	0.166
5.00e+05	0.166	0.166	0.170	0.170	0.169	0.169	0.169	0.168	0.168	0.163
1.00e+06	0.152	0.152	0.152	0.152	0.153	0.153	0.154	0.154	0.154	0.154

Table 2.2: Ratio of totalcosts between the peak rate and effective bandwidth allocation schemes

2.5.2 Core network in Hungary

In the second example the ATM network with similar planning conditions for the core network is investigated. The corresponding SDH network is of two levels and consists of STM-4 rings and STM-16 mesh as in the Budapest example. The number of ATM access switches is defined to be 84, and the capacity of ATM switches to be 2.5 Gbps as well.

The number of transit switches obtained with $\lambda = 10^3, \gamma = 10^3$ is 7, and the location of transit switches found by the optimization procedure can be seen in Figure 2.3. However, in this case although the sites of the nodes are also in good relation with the important nodes of the corresponding SDH network, the definition of the first phase, taking into account the long-term network architecture, needs some further investigations.

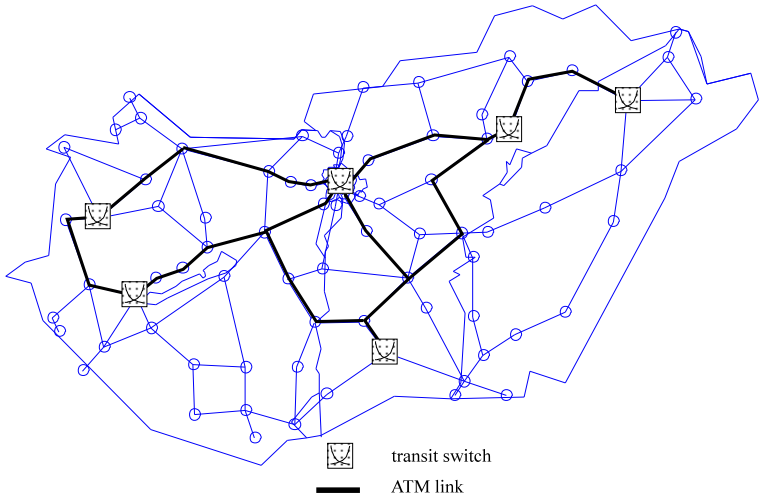


Figure 2.3: A core ATM network

2.6 Assessment of the planning approach

Recall that our planning approach defines the subtasks and then combines the application of ADD and MENTOR heuristic algorithms for assignments and topology optimisation in both regional and transit level of the considered network architecture. The rationale behind the choice of the above algorithms is their suitability and superiority over other ones which has been experienced and verified by the planning community [49].

As an illustration example, we present here one more comparison between the MENTOR algorithm and the new one developed recently for the optimization of mesh networks [22]. Both the algorithms have been applied to the design of a network covering main cities of Hungary with the following properties:

- The number of nodes is 42. These nodes are the main cities of Hungary.
- Each node communicates with all others nodes. The requirements of all connections are 8 Mbps uniformly.
- The cost of links is set proportionally to their length.
- The link capacity is 149 Mbps

Table 2.3 shows the quantitative comparisons between the two algorithms running on PCs with Pentium 233Mhz processor.

Metrics \ Algorithm	MENTOR algorithm	New algorithm
Total cost	11195.45	9372.40
Number of links	205	148
Total allocated capacities by demands	3318 sections$\times 8 = 26544$Mbps	2379 sections $\times 8 = 19032$ Mbps
$\frac{\text{Total allocated capacity}}{\text{Total capacity}}$	$\frac{26544}{205*149} = 0.869$	$\frac{26544}{148*149} = 0.863$
Average length of paths of demands	3.864 sections	2.763 sections
The max. length of paths	11	8
Running time	0.371- 0.381 (s)	191.876 (s)

Table 2.3: Comparison between the MENTOR and the new algorithm for the pattern Hungarian network

As the comparative results indicate, regarding some certain aspects like the total cost and the number of needed links, the MENTOR algorithm is outperformed by the new one. In overall, the new algorithm yields a more cost-efficient (about 15% better) solution. However, MENTOR achieves significantly faster than the new algorithm, making its integration into the complete planning process much more appealing.

This suggests that our planning approach is more useful and preferable than other possible ones particularly when being applied within interactive network design tools. This is because in

such cases, to accomplish the planning process iteratively in a reasonably small amount of time may be needed.

On the other hand, we emphasize that the presented model and the planning approach are general enough to be extended and applied to other scenarios when the ATM is directly built on other infrastructure different from SDH (e.g: on the fiber infrastructure). In that case we need only to change the cost function.

2.7 Conclusions and contributions

We have proposed an optimization model and a practical planning approach for planning an ATM network which is built on top of the SDH infrastructure. In our approach the design problem is decomposed into subproblems that can be solved in sequence with appropriately modified heuristic algorithms. Some results concerning the application of the planning approach to a realistic scenario for the introduction of a public ATM network on top of the Hungarian SDH infrastructure have also been presented. The topology of the implemented network is based on the results obtained by the planning approach. The usefulness and appealing practicality of the proposed planning approach have been pointed out through assessment.

Further research direction is to evaluate the performance of the implemented network in order to improve the planning process. Moreover, the extension of the planning approach with respect to network extensions and protection techniques is also in progress.

Parts of this chapter have been published in [21, 23, 24, 25]

Part III

QBD processes

Chapter 3

Theory of QBD processes and numerical methods for their steady state analysis

3.1 Introduction

Many problems in the performance evaluation of telecommunications networks can be solved with the model of two dimensional Markov chains. The common feature of such chains is that each of their state is defined by a couple of variables: a phase and a level. In a number of cases, the level transitions are only possible between adjacent levels, which leads to a well-known queueing model called QBD (Quasi Birth-Death) process. The concept of QBD processes, as a simple generalization of the classical birth and death M/M/1 queues was first introduced by Wallace in [72] and Evans in [27] in the late sixties. Since then, analysis of many problems arising in telecommunications and computer networks has been proved to require queueing models that are typically QBD-s, such as $M/PH/1/\infty$, $PH/M/n/\infty$, $PH/PH/1/\infty$, $MAP/PH/n/\infty$, $\sum_{i=1}^{n} MAP_i/PH/1/\infty$ and $\sum_{i=1}^{n} MAP_i/PH/1/m$ (see [32, 52, 57, 60] for more details).

Owing to its widespread applicability, the QBD process has gained a lot of research attention over recent years. For example, many research projects focus on finding numerical methods for steady state distribution of QBD processes. Moreover, evaluating the capability of those numerical methods also proves an important research issue and it is the main scope of this part of the dissertation.

Before moving ahead to detailed discussions, we now begin with a brief overview of numerical methods available in the literature that have been developed for steady state analysis of QBD processes. Exact mathematical formulation of some methods will be presented in the next section. However, offering the detailed and exhaustive description of all the methods is beyond

the scope of the dissertation, therefore, some of them are just discussed on the mentioning level. Nevertheless, for each method, the relevant related literature will be given. Interested readers can find deeper insights in the proposed materials.

The first numerical procedure known as a matrix geometric method was proposed by Neuts in [60]. In this work, the geometric relation between level probability vectors was revealed, which makes the computation more convenient. The key element of this method is the iterative calculation of a rate matrix R by which the geometric relation is defined. However, the original matrix geometric method has some disadvantages mainly in terms of computational time. Therefore, improving the efficiency (e.g. time and space requirement, numerical stability) of this computational method is a great research challenge. In recent years, research efforts have resulted in several new computational methods published in the literature [9, 16, 19, 53, 54, 57, 63, 79].

The methods proposed by Latouche et al. [53] and Naoumov et al. [57] are improved versions of the classical matrix geometric method. Having an in-depth analysis of QBD processes and using probabilistic interpretations, Latouche et al. proposed a really fast and numerically stable algorithm for computing the rate matrix. This algorithm was further speeded up by Naoumov et al. by the use of matrix factorization. These two methods are really popular and have been widely applied in several works.

Ram Chakka developed an exact computational method called spectral expansion for QBD processes [16]. Instead of using the geometric relation between level probability vectors, a spectral expansion of the level probability vectors is introduced. The expression is defined by eigenvalues and eigenvectors of the characteristic matrix polynomial constructed from the process parameters. According to the author, this method is efficient, accurate and easy to use.

Nail Akar et al. approached the solution of QBD processes from the novel side [9]. Their starting point is the observation of the close connection between solving the QBD process and solving the Algebric Ricatti Equations arising in optimal control problem in control theory. Their proposed method then basically relies on the theory of invariant subspace and on the computation of matrix sign function with an iterative procedure. The rate matrix R is obtained from the calculated invariant subspace of an adequately constructed matrix. The method is believed to be fast and stable.

In [79], the authors presented an efficient and versatile *folding method*, that can be applied to finite QBD processes. The odd-even permutation achieved inside the transition probability matrix and the use of the principle of finite Markov chain reduction are the key elements of this computational method. In contrast with matrix geometric methods, the folding algorithm solves directly the equation $\pi Q = 0$, where π is the steady state probability vector, Q is the generator matrix of the given QBD process. By taking a finite sequence of reduction steps in the *forward reduction phase*, the original transition matrix is brought to a single-level form, from which a boundary vector can be determined. Since the steady state solution π is expressed as a product of the boundary vector and a finite sequence of expansion factors, it can be calculated in the

backward expansion phase. Readers are encouraged to study [78, 79] for more details concerning the mathematical description as well as the applicability of this method.

In [12, 13, 54], Bini and Meini stated a fast, quadratically convergent and numerically stable algorithm called *cyclic reduction algorithm* for QBD problems. The algorithm is derived from a block cyclic-reduction applied for block tridiagonal Toeplitz-like probability transition matrices, supplemented with the use of FFT (*Fast Fourier Transform*) technique. From the viewpoint of performance capability, this algorithm has been considered to be equivalent with the algorithm of Naoumov et al.

Another solution method for finite QBD processes was sketched in [63]. The authors provide an exact computational method, which is only based on simple matrix operations. The explicit analytic solution can be expressed in terms of process parameters. The authors show that their computation procedure has the same asymptotic complexity as that of other solving techniques. However, the applicability of their method is limited by the non-singularity condition of certain matrices.

Recently, a novel method has been published in [19]. The method is named ETAQA (*Efficient Technique for the Analysis of QBD-processes by Aggregation*). In this method, the state space of a QBD chain is divided into several equivalence classes by a certain specific partitioning rule. Instead of computing the probability distribution of all states in the chain, only the aggregate probability distribution of the states in each class is evaluated. The authors show that those aggregate probabilities contain sufficient information to compute performance measures of interest such as the mean queue length or any higher moments. The method is proved to have better computational and storage complexity compared with other ones. ETAQA can be originally applied to a class of QBD processes, in which the downward transitions are directly towards a single state. If this condition is not fulfilled, further manipulations such as rearranging the state space partitioning are necessary.

In this chapter we first review the basic theory of the QBD concept. Then we explore mathematical details of some most familiar and recently developed numerical methods. The contents of this chapter will serve as a theoretical background for our research work presented in the next chapter.

3.2 Mathematical description of QBD processes

Consider a queueing system modeled by a discrete time, two dimensional Markov process, which has either semi-infinite or finite state space. The state of the system at observation time n is described by two integer valued random variables I_n and J_n. The former one is bounded and referred to as the phase, while the latter one may be either unbounded (infinite case) or bounded (finite case) and is referred to as the level of the system. The Markov process is denoted by $X = \{I_n, J_n; n \geq 0\}$ and its state space is $(\{0, 1, \ldots, N\} \times \{0, 1, \ldots\})$ in the infinite case and

$(\{0, 1, \ldots, N\} \times \{0, 1, \ldots L\})$ in the finite case, respectively. If the possible jumps of system's level in transition are only 0, -1 or 1, the process is known as Quasi Birth-Death (QBD) process.

The transition probabilities of the underlying Markov process are given by the following transition probability matrices:

- A_j: purely phase transitions – From state (i, j) to state (k, j) $(0 \leq i, k \leq N; j = 0, 1, \ldots)$
- B_j: one−step upward transitions–From state (i, j) to state $(k, j + 1)$ $(0 \leq i, k \leq N; j = 0, 1, \ldots)$
- C_j: one−step downward transitions–From state (i, j) to state $(k, j - 1)$ $(0 \leq i, k \leq N; j = 0, 1, \ldots)$.

A_j, B_j and C_j matrices have size of $(N+1) \times (N+1)$. We assume that for $j \geq M$ the transition probability matrices become level-independent. That is

$$A_j = A, j \geq M; \;\; B_j = B, j \geq M - 1; \;\; C_j = C, j \geq M. \tag{3.1}$$

With these matrices, the transition probability matrix of the QBD process can be easily constructed. An example is given in Figure 3.1.

$$P = \left[\begin{array}{ccccc|ccccc} A_0 & B_0 & & & & & & & & \\ C_1 & A_1 & B_1 & & & & & & & \\ & C_2 & A_2 & B_2 & & & & & & \\ & & C_3 & A_3 & B & & & & & \\ & & & C_4 & A & B & & & & \\ & & & & C & A & B & & & \\ & & & & & C & A & B & & \\ & & & & & & \ddots & \ddots & \ddots \end{array}\right]$$

Figure 3.1: The block structure of the transition probability matrix of a QBD process $(M = 4)$

For further computations we introduce the following notations:

- $p_{i,j}$: the steady state probability of the state (i, j)

$$p_{i,j} = \lim_{n \to \infty} P(I_n = i, J_n = j); \quad i = 0, 1, \ldots, N; \quad j = 0, 1, \ldots \tag{3.2}$$

  Our task is to determine these probabilities in terms of known parameters of the system.

- $\underline{v}_j$: the row vector referred to as the level probability vector and is defined as:

$$\underline{v}_j = (p_{0,j}, \; p_{1,j}, \ldots, p_{N,j}); \quad j = 0, 1, \ldots \tag{3.3}$$

- $\underline{e}$: the column vector of $(N+1)$ elements each of which is equal to 1.
- I: a unit matrix of appropriate size.

For $j = 0, 1, \ldots, M-1$, the balance equations of the system are

$$\underline{v}_j = \underline{v}_{j-1} B_{j-1} + \underline{v}_j A_j + \underline{v}_{j+1} C_{j+1}. \tag{3.4}$$

(It is assumed that $\underline{v}_{j-1} = \underline{0}$ if $j < 1$). For $j \geq M$, the corresponding $j-$independent set is

$$\underline{v}_j = \underline{v}_{j-1} B + \underline{v}_j A + \underline{v}_{j+1} C. \tag{3.5}$$

Note that if the system is finite (i.e. $j \leq L$) then one more boundary equation appears

$$\underline{v}_L = \underline{v}_{L-1} B + \underline{v}_L A. \tag{3.6}$$

In addition, since the sum of all probabilities must be one, we have

$$\sum_{j=0}^{\infty} \underline{v}_j \underline{e} = 1 \tag{3.7}$$

for the infinite case and

$$\sum_{j=0}^{L} \underline{v}_j \underline{e} = 1 \tag{3.8}$$

for the finite case.

In order to get steady state performance measures for the aim of dimensioning the system, one has to know the steady state probabilities. This steady state analysis is done by means of the boundary equations (3.4) and the equations (3.5). In the next sections some of the most well-known methods developed for steady state analysis of QBD processes will be presented. Although the algorithms will be discussed here for discrete time QBD processes, we emphasize that simple stochastic arguments approve their applicability in continuous time domain as well.

3.3 Computational methods for infinite QBD processes

3.3.1 The spectral expansion method

The main contribution of the spectral expansion method published by Ram Chakka [16] is that the solution for equations (3.5) can be expressed in the form

$$\underline{v}_j = \sum_{k=0}^{N} a_k \underline{\psi}_k \lambda_k^{j-(M-1)}, \quad j \geq M-1, \tag{3.9}$$

where λ_k is the k-th eigenvalue strictly inside the unit disk and $\underline{\psi}_k$ is the corresponding left eigenvector of the characteristic matrix polynomial

$$Q(\lambda) = B + A\lambda + C\lambda^2,$$

i.e. they satisfy the equation

$$\underline{\psi} Q(\lambda) = \lambda \underline{\psi}. \tag{3.10}$$

Combining the form (3.9) with the first M level-dependent equations (3.4) and the normalizing equation (3.7), one gets a set of linearly independent equations, which has a unique solution of $\underline{v}_0, \ldots, \underline{v}_{M-2}, \underline{a}$ where $\underline{a} = (a_0, \ldots, a_N)$ is the coefficient vector. The detailed procedure of computing all relevant eigenvalues and eigenvectors is excellently discussed in [16], therefore the readers are referred to it for deeper insights.

3.3.2 The matrix geometric method

In the classical matrix geometric method, Neuts proved that the solution for equations (3.5) is given as follows [60]:

$$\underline{v}_j = \underline{v}_{M-1} R^{j-(M-1)}, \quad j \geq M-1, \tag{3.11}$$

where matrix R (referred to as the rate matrix) is the minimal non-negative solution of the quadratic matrix equation given by

$$B + RA + R^2 C = R. \tag{3.12}$$

Once R is determined, $\underline{v}_j$ $(j \geq M)$ can be expressed in terms of $\underline{v}_{M-1}$. Combining the form (3.11) with the first M level-dependent equations (3.4) and the normalizing equation (3.7), one gets a set of linearly independent equations, which has a unique solution of $\underline{v}_0, \ldots, \underline{v}_{M-1}$.

To compute matrix R with the desired accuracy, several iterative procedures were offered in the last decade (see [53, 57, 60]). The common feature of these methods is that all of them were formulated based in terms of basic non-negative matrices G, R, U, each of which has probabilistic interpretations as follows [53]. For any $n \geq M-1$

- $G(i,l)$ entry of matrix G is the probability that starting from state $(i, n+1)$ the chain visits level n and does so by visiting the state (l, n),
- $R(i,l)$ entry of matrix R is the expected number of visits into state $(l, n+1)$ starting from state (i, n), until the first return to level n,
- $U(i,l)$ entry of U matrix is the taboo probability that starting from the state $(i, n+1)$, the Markov chain eventually returns to the level $n+1$ and does so by visiting the state $(l, n+1)$, under taboo of level n (i.e. without visiting any state in the level n).

The relation between the three matrices is expressed by the following equations (see [35, 53])

$$G = (I-U)^{-1} C, \tag{3.13}$$

$$R = B(I-U)^{-1}, \tag{3.14}$$

$$U = A + BG = A + RC. \tag{3.15}$$

In addition, G, R, U matrices are the minimal non-negative solutions of the non-linear equations

$$G = C + AG + BG^2, \tag{3.16}$$

$$R = B + RA + R^2C, \tag{3.17}$$

$$U = A + B(I - U)^{-1}C. \tag{3.18}$$

Note that once one of the three basic matrices has been calculated, the other two are automatically obtained by means of equations (3.13), (3.14), (3.15). Based on equations (3.16), (3.17), (3.18), iterative procedures are proposed to compute G, R, U. The simplest one named *Simple Substitution (SS)* algorithm is shown in Figure 3.2.

$R_0 = 0$
$k = 0$
DO
  $k = k + 1$
  $R_k = (-B - R_{k-1}^2 C)(A - I)^{-1}$
$WHILE\ (||R_k - R_{k-1}|| \geq \epsilon)$

Figure 3.2: The iterative procedure SS for computing the rate matrix R

In case of positive recurrent QBD processes, another successive substitution procedure shown in Figure 3.3 can be used [53]. This numerical algorithm has a complexity of $O\left(\frac{7}{3}(N+1)^3 I_U\right)$, where I_U is the necessary iterations needed to achieve a given accuracy ϵ.

$k = 1$
$U = A$
$G = (I - U)^{-1}C$
DO
  $k = k + 1$
  $U = A + BG$
  $G = (I - U)^{-1}C$
$WHILE\ (||e - Ge||_\infty \geq \epsilon)$
$R = B(I - U)^{-1}$

Figure 3.3: The iterative procedure of the matrix geometric method

3.3.3 The logarithmic reduction algorithm by Latouche et. al.

Latouche and Ramaswami revealed in [53] the probabilistic interpretation hidden in the iterative procedure shown in Figure 3.3. At each k-th iterative step, matrix G_k is evaluated. The element

$G_k(i,l)$ is the probability that, starting from the state $(i,1)$, the chain eventually visits the level 0, and does so by visiting the state $(l,0)$, *under taboo* of the level $k+1$ and above. In other words, in the k-th step only those paths are considered, whose length does not exceed k. Thus, at the beginning of the algorithm, the chain is allowed to move no higher than the level 1. With each new iteration, the chain is allowed to visit one level above the previous maximum.

The main idea of the logarithmic reduction algorithm is derived from the stochastic observation mentioned above. In the new approach, during each iterative step, the chain is always allowed to proceed up to a multiple of twice the level attained at the previous iteration step. As a consequence, a logarithmic reduction in the number of iterations required to achieve convergence is performed. The iterative procedure is shown in Figure 3.4.

$T_0 = I - A^{-1}B$
$T_2 = I - A^{-1}C$
$k = 0$
$S = T_2$
$\Pi = T_0$
DO
$\quad k = k+1$
$\quad T_i = (I - T_0T_2 - T_2T_0)^{-1}(T_i)^2 \quad i = 0,2$
$\quad S = S + \Pi T_2$
$\quad \Pi = \Pi T_0$
$WHILE\ (||e - Se|| \geq \epsilon)$
$G = S$
$U = A + BS$
$R = B(I-U)^{-1}$

Figure 3.4: The logarithmic reduction algorithm by Latouche et al.

The complexity of this algorithm is $O\left(\frac{25}{3}(N+1)^3 I_{LA}\right)$, where I_{LA} is the necessary iterations needed to achieve a given accuracy ϵ.

3.3.4 The algorithm by Naoumov et. al.

Based on the theory of matrix factorization, Naoumov et al. [57] developed further the logarithmic algorithm. Their computation algorithm reduces the complexity of the basic loop of each iteration step, herewith produces better performance. This improved iterative procedure is detailed in Figure 3.5.

The complexity of this algorithm is $O\left(\frac{19}{3}(N+1)^3 I_{NA}\right)$, where I_{NA} is the necessary iterations needed to achieve a given accuracy ϵ.

$S = A - I$
$V = B$
$T = C$
$W = A - I$
DO
    $X = -S^{-1}V$
    $Y = -S^{-1}T$
    $Z = VY$
    $W = W + Z$
    $S = S + Z + TX$
    $V = VX$
    $T = TY$
$WHILE\ (||Z|| \geq \epsilon)$
$R = -BW^{-1}$

Figure 3.5: The logarithmic reduction algorithm improved by Naoumov et al.

3.3.5 Invariant subspace based method

Observing the close interconnection between solving a QBD process and solving the Algebric Ricatti Equations arising in optimal control problem in control theory, Nail Akar et al. approached the solution of the rate matrix R from a novel side [9]. Their proposed method basically relies on the theory of invariant subspace and on the computation of matrix sign function with iterative procedure.

By defining the matrix polynomials

$$F(z) = zI - (B + Az + Cz^2) \text{ and } H(s) = \sum_{i=0}^{2} H_i s^i = (1-s)^2 F(z)|_{z=\frac{1+s}{1-s}}, \tag{3.19}$$

the authors construct the matrices $\hat{H}_0 = \left[H_0 H_2^{-1}\right]^T$, $\hat{H}_1 = \left[H_1 H_2^{-1}\right]^T$ and consider the block companion matrix

$$E = \begin{bmatrix} 0 & I \\ -\hat{H}_0 & -\hat{H}_1 \end{bmatrix}.$$

Let T be a matrix of $2(N+1) \times (N+1)$, whose columns are a basis of the closed left invariant subspace of the matrix E, and partition matrix T as $T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$, where T_1 and T_2 are matrices of $(N+1) \times (N+1)$. Then the rate matrix R is defined as

$$R = (T_1 - T_2)^{-T}(T_1 + T_2)^T. \tag{3.20}$$

In order to find the closed left invariant subspace of the matrix E, the authors make use of the

matrix sign function iteration by introducing the matrix

$$E_m = E - \frac{\underline{y}\underline{x}^T}{\underline{x}^T\underline{y}}, \tag{3.21}$$

where $\underline{x}^T = [\ \underline{x}_0^T\hat{H}_1 \quad \underline{x}_0^T\ ]$, $\underline{y} = \begin{bmatrix} \underline{y}_0 \\ 0 \end{bmatrix}$ and $\underline{x}_0$, $\underline{y}_0$ are two vectors such that $\underline{x}_0^T\hat{H}_0 = 0$ and $\hat{H}_0\underline{y}_0 = 0$. The closed left invariant subspace of E and E_m are the same. Moreover, the matrix sign function of matrix E_m, $Z = sgn(E_m)$ is well-defined and can be computed by an iterative procedure shown in Figure 3.6. The basis of the closed left invariant subspace of E_m in turn is easily computed from its matrix sign function Z by performing the rank revealing QR decomposition of the matrix $I - Z$ (see Section 6.2.1 in Chapter 6 for more details).

$k = 0$
$Z_0 = E_m$
$d = dim(E_m)$
DO
  $t = |detZ_k|^{-1/d}$
  $Z_{k+1} = \frac{1}{2}(tZ_k + t^{-1}Z_k^{-1})$
  $k = k + 1$
$WHILE\ (||Z_k - Z_{k-1}||_1 \geq \epsilon||Z_{k-1}||_1)$

Figure 3.6: The iterative procedure to obtain matrix sign function

The complexity of this algorithm is $O\left(\frac{4}{3}[2(N+1)]^3 I_{IS} + 4(N+1)^3\right)$, where the former component refers to the complexity of the computation of the matrix sign function and the latter one is the complexity of the rank revealing performed by the Householder QR with column pivoting algorithm [33].

3.4 Computational methods for finite QBD processes

Imposing a limit on the maximum value of J_n leads to a QBD process having finite state-space $\{I_n, J_n; n \geq 0\}$. Let the maximum value of variable J_n be L, then the equations (3.5) still hold, except that the range of j is limited to L.

3.4.1 The spectral expansion method

Using spectral expansion implies the form [16]

$$\underline{v}_j = \sum_{k=0}^{N} a_k\underline{\psi}_k\lambda_k^{j-(M-1)} + \sum_{k=0}^{N} b_k\underline{\phi}_k\beta_k^{L-j}, \qquad M-1 \leq j \leq L. \tag{3.22}$$

Here, $\lambda^{'}$s are the $N+1$ eigenvalues of least absolute value defined in the way as in subsection 3.3.1, $\beta^{'}$s are the $N+1$ eigenvalues of least absolute value satisfying the equation

$$\underline{\phi}(C + A\beta + B\beta^2) = \beta\underline{\phi}, \tag{3.23}$$

where $\underline{\psi}^{'}$s and $\underline{\phi}^{'}$s are eigenvectors corresponding to $\lambda^{'}$s and $\beta^{'}$s, respectively. Once the necessary eigenvalues and eigenvectors are determined, the set of linear simultaneous equations, which is composed of equations (3.4), (3.6) and (3.8) must be solved to determine $\underline{v}_0, \ldots, \underline{v}_{M-2}, \underline{a}, \underline{b}$, where $\underline{a} = (a_0, \ldots, a_N)$ and $\underline{b} = (b_0, \ldots, b_N)$.

3.4.2 Matrix geometric, Latouche's and Naoumov's methods

The matrix geometric solution is given by [9, 16] as follows:

$$\underline{v}_j = \underline{w}_1 R_1^{j-(M-1)} + \underline{w}_2 R_2^{L-j}, \quad M-1 \leq j \leq L. \tag{3.24}$$

Here, $\underline{w}_1$ and $\underline{w}_2$ are the unknown vectors of size $N+1$. R_1 and R_2 are the minimal non-negative solution of the quadratic matrix equations

$$B + R_1 A + R_1^2 C = R, \tag{3.25}$$

$$C + R_2 A + R_2^2 B = R. \tag{3.26}$$

To compute R_1 and R_2 the matrix-geometric, Latouche's algorithm or Naoumov's algorithm can be used. Once those matrices are calculated, the set of linear simultaneous equations, which is composed of equations (3.4), (3.6) and (3.8) must be solved to determine $\underline{v}_0, \ldots, \underline{v}_{M-2}, \underline{w}_1, \underline{w}_2$. For more detailed description, see [9, 16].

3.4.3 Invariant subspace based method

In fact, invariant subspace based method is a unifying algorithmic approach for the solution of both finite and infinite QBD processes. In [9], the authors prove that in case of finite QBD chains, the solution form is

$$\underline{v}_j = \underline{w}_1 R_1^{j-(M-1)} + \underline{w}_2 R_2^{L-j}, \quad M-1 \leq j \leq L, \tag{3.27}$$

where $\underline{w}_1$ and $\underline{w}_2$ are the unknown vectors. Recall that this form of two matrix geometric terms is also introduced in case of matrix geometric method and its enhanced versions. In this method, the computation of R_1 and R_2 is done by finding the left and right invariant subspace through the matrix sign function as follows.

Let us define the matrix polynomials

$$F(z) = zI - (B + Az + Cz^2) \text{ and } H(s) = \sum_{i=0}^{2} H_i s^i = (1-s)^2 F(z)|_{z=\frac{1+s}{1-s}} \tag{3.28}$$

and matrices $\hat{H}_0 = \left[H_0 H_2^{-1}\right]^T$, $\hat{H}_1 = \left[H_1 H_2^{-1}\right]^T$. Also let us construct the matrix $E = \begin{bmatrix} 0 & I \\ -\hat{H}_0 & -\hat{H}_1 \end{bmatrix}$. R_1 and R_2 matrices now are calculated through the left and right invariant subspace of matrix E, respectively. Assuming that the basis for the left invariant subspace is $T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$ then we have

$$R_1 = (T_1 - T_2)^{-T}(T_1 + T_2)^T. \tag{3.29}$$

Similarly, let the basis of the right invariant subspace be $U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$ then

$$R_2 = (U_1 + U_2)^{-T}(U_1 - U_2)^T. \tag{3.30}$$

The boundary vectors $\underline{v}_0, \ldots, \underline{v}_{M-2}, \underline{w}_1, \underline{w}_2$ can be obtained by solving the set of linearly independent equation, which is built up from equations (3.4), (3.6) and (3.8).

3.5 Conclusions

In this chapter, a class of Markov chains called Quasi Birth Death processes has been negotiated. The application of this QBD class is widely observed in performance analysis of computer systems and telecommunications networks. A comprehensive survey on the latest numerical methods for steady state analysis of QBD processes has been given. From this point of view, this chapter may be considered as a tutorial on numerical methods for QBD processes, which hopefully provides a brief but easy-to-understand material for interested readers.

In the next chapter, a comprehensive performance-related comparison between some methods dealt in this chapter will be presented. The results of this comparative study will help analysts in choosing the most adequate method to solve each individual QBD model arising in practice.

Chapter 4

Numerical comparison of computational methods for QBD processes

4.1 Motivation

The extensive application of computational methods developed for QBD processes is witnessed in the modelling and analysis of telecommunications and computer networks. Therefore, one frequently faces with the task of solving a given QBD process in steady state. In order to do it in an efficient way an overall picture related to the performance and capability of the proposed methods is needed. Along this line, in the literature one can find some work by Haverkort ([36]) on the comparison between spectral expansion and Latouche's algorithms; by Mitrani et al. ([56]) on the comparison between spectral expansion and matrix geometric methods; by Nail Akar et al. ([7, 8]) on the comparison between Latouche's algorithm and invariant subspace method; by B. Meini [54] on the comparison between the cyclic reduction algorithm and the invariant subspace based method and recently the work by A. Ost [61] and N. Akar [6] on the comparison of some of the aforementioned methods. The common feature of these works is that each of them gives comparative results based on an individual concrete application study, and all of them, except [6], deal with infinite QBD processes.

This chapter will report the results of our comparative study achieved towards a more complete numerical comparison between some mostly used computational methods of QBD processes. Our results have been performed for both infinite and finite QBD cases, hence the comparison is somewhat more comprehensive than the previous works. Moreover, the fact that the basic case study is different from the ones in some of previous works, will hopefully make the comparative picture related to performance of different methods more complete and clearer.

Our comparison is concerned with

- SE: spectral expansion method,
- MG: simple substitution matrix geometric method,
- LA: logarithmic reduction method proposed by Latouche et. al.,
- NA: improved version of LA method, developed by Naoumov et. al., and
- IS: invariant subspace based method.

The criteria for comparison in this study includes computational complexity (computational time) and numerical stability. All the aforementioned methods except the spectral expansion are iterative, therefore the time and steps needed for convergence are investigated too. Theoretically, the spectral expansion provides exact numerical results. The question when it could be considered to give reference results is also pointed out in this section.

A case study for numerical comparison is the multiprocessor system with repairs and breakdowns, that can be modelled with a QBD process.

4.2 Implementation

The implementation begins with the observation that the computational procedure for obtaining the steady state probabilities of a QBD process consists of two phases:

- In the first phase either all relevant eigenvalues and corresponding eigenvectors of the characteristic matrix polynomial are determined if the spectral expansion method is applied; or the R matrix (R_1 and R_2 in case of finite processes) is calculated if the MG, LA, NA or IS are applied.
- In the second phase a finite set of linearly independent equations must be solved for the fundamental unknown probability vectors $\underline{v}_j$ ($0 \leq j \leq M-2$) as well as for either additional vectors ($\underline{v}_{M-1}$ in the infinite case; $\underline{w}_1, \underline{w}_2$ in finite case) if MG, LA, NA or IS are used; or the coefficient vectors ($\underline{a}$ in the infinite case; $\underline{a}, \underline{b}$ in the finite case) if SE is used.

Although all the MG, LA, NA, and IS methods are based on iterative computation, a major difference between them should be clearly pointed out. When either MG, LA or NA is used, their iterative procedure has a stopping criteria regarding to the rate matrix itself. In contrast, the IS method has an iterative procedure concerning the matrix sign function. In case of IS, finishing the iterative procedure does not result straightforwardly in the rate matrix (or matrices), but further computational steps are needed.

Exploiting the fact that the spectral expansion method theoretically provides exact numerical results, we adopt and implement a comparative framework consisting of two kinds of stopping scenarios named object and performance parameter based scenarios for the iterative methods.

1. $k = 0,\ T_1 = 0$
2. *Run SE*
3. *Determine* $T_{SE}, E(j)_{SE}$
4. *DO*
   - 4.1. $k = k + 1$
     $T_2 = 0$
   - 4.2. *Run step k of METHOD*
     $t =$ *time of step k*
     $T_1 = T_1 + t$
   - 4.3. *Solve the set of equations*
     $T_2 =$ *Time of solving the set of equations*
   - 4.4. *Compute* $E(j)_{METHOD}$

   $WHILE\ (\frac{1}{E(j)_{SE}}(|E(j)_{SE} - E(j)_{METHOD}|) \geq \xi)$
5. *# of iterations* $= k$
6. $T_{METHOD} = T_1 + T_2$

Figure 4.1: The iterative procedure applied for comparison

- *Scenario I: Performance parameter based criteria*

  Theoretically, performance parameters (such as the mean number of jobs in the system) can be exactly calculated with the spectral expansion method due to its non-iterative feature. If one of them is adopted as *reference value* then the stopping criteria for the rest of iterative methods may be introduced as follows.

  Let one of the performance measures (e.g. the mean queue length) calculated by the matrix geometric method or by its improved versions be *comp. value*. The term "*relative bias*" or "*relative difference*" is defined as

$$relative\ bias = \frac{abs(\text{reference value} - \text{comp. value})}{\text{reference value}} \tag{4.1}$$

  and is compared with the stopping criterion ξ in order to stop the iterative procedure shown in Figure 4.1 where the term METHOD may be MG, LA or NA. In case of infinite processes, during one iterative step, the matrix R is evaluated. In case of finite processes the evaluation is done for both R_1 and R_2.

  Our experiences show that increasing the number of iterations assures the convergence of the rate matrix (or matrices) and the mean number of jobs approaches the value calculated by the SE method in one direction, i.e. it monotonously increases (decreases) to the result of SE. In case of the IS method, however, the mean number of jobs oscillates around the value calculated by SE and does not show such definite monotonous convergence. For example, after iteration step n the relative bias is less than, let say 10^{-5}. One then may believe that it is fair to stop the iterative procedure at this point and measure the time.

However, if one achieves iteration step $n+1$, it turns out that the relative bias changes to the value which may be greater than 10^{-4}. Thus, obviously it is not fair to stop after n iterations. The IS method, therefore, is not drawn into scenario I.

- *Scenario II: Object based criteria*

  The stopping criterion of the first scenario sometimes does not work, because the bias between the results belonging to the SE and other methods can not be reduced further, however many iterations are performed (see later). In these cases, we must use the original stopping criterion ϵ described in Section 3.3.

The implementation of the methods was carried out in standard C using the Meschach library that was developed for matrix operations at the School of Mathematical Sciences, Australian National University by David E. Stewart and Zbigniew Leyk and it is free via netlib (ftp.netlib.org/c/meschach). All the reported results are obtained by programs running on a Sun SPARCstation Ultra60 with SPARC processor. The reported time is measured in seconds and is composed of the time of both computation phases.

4.3 The case study

The example is taken from [16]. The system is a set of homogeneous processors whose number is N. The processors break down from time to time. Single and independent failures of processors, as well as multiple and simultaneous failures are possible. Failed processors return to operative state after successful repair. Single and independent repairs of processors, as well as multiple and simultaneous repairs are possible. Being in its operative state, each processor serves jobs, one at time. Each job can occupy at most one operative processor at a time. Failure, repair and service time are assumed to be exponentially distributed. Those arriving jobs who do not obtain service immediately are stored in a buffer. If the buffer has infinite size, it is the case of an infinite system. If the buffer size is limited to L, it is the case of a finite system. Both the infinite and finite cases will be taken into the comparison.

The system is modelled by a two dimensional Markov process in the following way:

- $I(t)$ is the operative state of the system, representing the number of operative processors at time t, $I(t) = 1, \ldots, N$,
- $J(t)$ is the number of jobs in the system at time t, $J(t) = 0, 1, \ldots$

Now, we have to construct the transition matrices A, B, and C, which have analogous interpretation with the discrete definitions given in Section 3.2. Let us notice that *when a new job arrives or when a completed job departs from the system, the operative state does not change, unless there is an independent coincidence towards such a change.* Hence, change in the operative state of the system is reflected only in the matrices A and A_j. Taking into account that

- the individual processors break down independently at rate ξ and are repaired independently at rate η,
- the global simultaneous breakdowns of all currently operative processors occur at rate ξ_0 and the global simultaneous repairs of all currently inoperative processors occur at rate η_N,

the matrix A is given by

$$A = A_j(j = 0, 1, \ldots) = \begin{bmatrix} 0 & N\eta & & & & \eta_N \\ \xi_0 + \xi & 0 & (N-1)\eta & & & \eta_N \\ \xi_0 & 2\xi & 0 & & & \eta_N \\ & \ddots & \ddots & \ddots & & \\ & & & & & \eta + \eta_N \\ \xi_0 & & & & N\xi & 0 \end{bmatrix}. \tag{4.2}$$

When the operative state $I(t) = i$, jobs are assumed to arrive according to an independent Poisson process with rate σ_i. Hence for all j ($j = 0, 1, \ldots$) the rate matrix of the one-step upward transitions (initiated by the arrivals of single jobs) has the form

$$B = B_j = diag[\sigma_0, \sigma_1, \ldots, \sigma_N]. \tag{4.3}$$

Let us assume that each operative processor has an exponentially distributed service time with parameter μ. The one-step downward transitions take place by the departures of single jobs. The departure rate depends on the current state $(I(t), J(t)) = (i, j)$ and it is given by the entry $C_j(i, i)$ of matrix C_j. If $i > j$, then every job has a processor for getting service, and not all operative processors are occupied. Hence the departure rate of jobs is $j.\mu$. If $i \leq j$, then all the operative processors are occupied by jobs, hence the departure rate of jobs is $i.\mu$. We arrive at

$$C_j = diag[0, min(j, 1).\mu, \ldots, min(j, N).\mu]. \tag{4.4}$$

Note that for $j \geq N$, C_j does not depend on j. Consequently, the threshold M is given by $M = N$.

In all the cases taken into account in Section 4.4, in order to keep the service capacity constant, the service rate of each processor was set to $\mu = 1$. For the sake of simplicity $\sigma_i = \sigma$ for all $0 \leq i \leq N$. Other parameters are $\eta = \eta_N = 0.1$ and $\xi = \xi_0 = 0.05$.

4.4 Numerical comparison

In what follows we compare the algorithms through a case study of the processor system with repairs and breakdowns presented in the previous subsection. The criteria of comparison are computation complexity and numerical stability.

The system is negotiated in both infinite and finite cases. The relevant characteristics of the system that affect the operation of the computational methods are the total average incoming load, the total average service rate (service capacity) and the dimension of the system. These parameters are defined by means of parameters σ, μ, N and the transition matrices.

In the infinite case, the total average incoming load should be less than the total average service rate, otherwise the system becomes unstable. In the finite case, the system is always stable, but there will be lost events if the number of jobs exceeds the size of buffer (overload condition). For a system with finite buffer, our experiences show that the buffer's size has negligible impact on computation time, therefore in all the cases reported below the buffer size was chosen to be $L = 100$. Although we have produced a numerous set of numerical results according to several scenarios, in what follows we only present and heighten the most representative and informative results to emphasize our important comparative observations.

4.4.1 Computational complexity

First, we examine the dependency of computational time on the system load, the stopping accuracy and the system's size. In the following tables, the terms I and T will refer to the number of needed iterations and the total computation time (measured in seconds), respectively.

Effect of system load When the system has an infinite buffer, the condition of stability is $\sigma < 0.666667 * N$. Table 4.1 shows the number of necessary iteration steps and computation time for a given relative bias, while system load is increasing.

$\xi = 10^{-3}$, $N = 10$								
σ	Stability	T_{SE}	I_{MG}	T_{MG}	I_{NA}	T_{NA}	I_{LA}	T_{LA}
6	Y	0.03	629	0.27	9	0.03	8	0.03
6.6	Y	0.03	6633	3.00	12	0.03	11	0.03
6.66	Y	0.04	66461	29.73	16	0.03	15	0.03
6.666	Y	0.04	664736	274.89	19	0.03	18	0.04
6.6666	Y	0.04	6721786	2808.67	22	0.03	22	0.04
6.66666	Y	0.04	-	-	-	-	-	-
6.666666	Y	0.04	-	-	-	-	-	-
6.6666666	N	-	-	-	-	-	-	-

Table 4.1: Stability and complexity of computational methods in infinite QBD case

In Table 4.1 the term "-" is jotted down several times, even the system is still stable. This is the case when the software package was unable to compute the rate matrix meeting a given relative bias either because the number of iterations is extremely high (MG case) or because the fossilized result calculated by LA and NA differs from the result of SE considerably. The first cause is illustrated in Figure 4.2, where the needed iterations of MG increase with the system load in an exponential way.

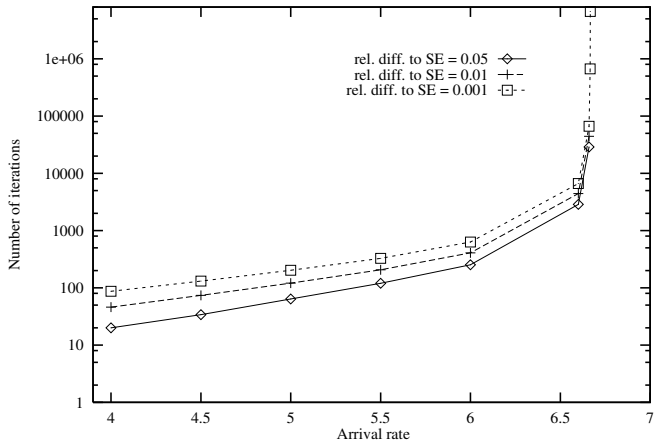


Figure 4.2: Complexity of the MG method versus the load of an infinite system ($N = 10$)

To expose the second cause, we have carried out further experiments as follows. We check the system on the verge of stability and choose the stopping criteria according to Scenario II with $\epsilon = 10^{-12}$. Numerical results are presented in Table 4.2.

σ (arrival rate)	I_{LA}	Rel. diff. of LA	I_{NA}	Rel. diff. of NA
6.66666	29	0.0069	28	0.0077
6.666664	30	0.0015	29	0.0012
6.666665	31	0.0383	30	0.0449
6.6666652	31	0.0209	30	0.0217
6.6666654	31	0.0154	30	0.0246
6.6666656	31	0.0343	30	0.0497
6.6666658	32	0.2473	31	0.2736
6.666666	33	0.3933	32	0.4013

Table 4.2: On the verge of stability of an infinite system ($N = 10$, $\epsilon = 10^{-12}$)

As one can see, if the stability edge is approached very closely (recall that the service capacity of the studied system is $0.666667 * N$), the methods' results show quite large bias and practically the $\xi = 0.001$ relative bias was never reachable. That explains why the computation procedure was unable to finish when Scenario I was run with 0.001 relative bias.

For finite systems, impact of the system load on the computation time and the number of necessary iterations can be seen in Tables 4.3 and Table 4.4.

For both finite and infinite cases, numerical results of Tables 4.1, Table 4.3 and Table 4.4 clearly show that if the system load is not high, the bias related with total computations time between SE, LA and NA is negligible. The computational time of MG tends to

σ	T_{SE}	I_{MG}	T_{MG}	I_{NA}	T_{NA}	I_{LA}	T_{LA}
3.00	0.07	42	0.06	5	0.05	4	0.05
5.00	0.07	188	0.22	7	0.06	6	0.06
6.00	0.06	455	0.35	9	0.06	8	0.06
6.60	0.07	1230	0.56	10	0.06	9	0.06
6.66	0.07	1353	0.73	10	0.06	9	0.06
6.666	0.06	1357	0.78	10	0.06	9	0.06
6.6666	0.07	1358	0.87	10	0.06	9	0.06

Table 4.3: Computation time and number of iterations versus load σ (finite system, $N = 10$, $\xi = 10^{-3}$)

overstep the time of other methods if system load approaches the saturation condition (meanwhile the system is kept stable or is still able to operate without loss). Otherwise it operates nearly with the same efficiency.

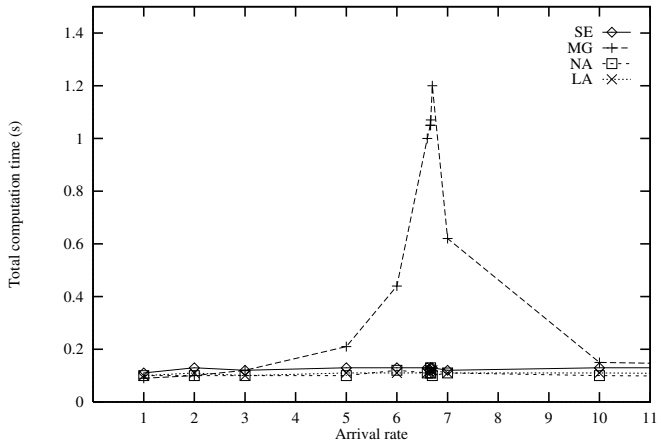


Figure 4.3: Computation time versus load of a finite system ($N = 10$, $\xi = 10^{-3}$)

Increasing the system load (by increasing the arrival rate σ) has no significant effect on the computation time of the spectral expansion, Latouche's and Naoumov's methods, i.e. their execution times are almost constant (see Figure 4.3). The significant impact on the computation time of MG can be explained by slow convergence leading to a high number of iterations needed (as shown in Figure 4.2).

It is interesting that in the finite case, for a given relative bias, the number of necessary iterations of the MG method is not strictly increasing with the traffic load. After the saturation point, the number of iterations needed for a given relative bias tends to decrease at the expense of the loss probability (see Figure 4.3).

As mentioned earlier, the IS method can only be involved in Scenario II of the comparison. The time requirement of this method for increasing system load is presented in Table 4.5

$N = 10$, $\sigma = 3.0$, $E(j)_{spect.exp}$=5.19970358							
ξ	T_{SE}	I_{MG}	T_{MG}	I_{NA}	T_{NA}	I_{LA}	T_{LA}
10^{-3}	0.07	42	0.06	5	0.05	4	0.05
10^{-6}	0.07	107	0.13	7	0.06	6	0.06
10^{-9}	0.07	172	0.13	7	0.06	6	0.05
10^{-12}	0.07	236	0.28	8	0.06	7	0.06
$N = 10$, $\sigma = 6.0$, $E(j)_{spect.exp}$=33.55243677							
ξ	T_{SE}	I_{MG}	T_{MG}	I_{NA}	T_{NA}	I_{LA}	T_{LA}
10^{-3}	0.06	455	0.35	9	0.06	8	0.06
10^{-6}	0.06	1119	0.68	10	0.06	9	0.06
10^{-9}	0.07	1784	0.95	11	0.06	10	0.07
10^{-12}	0.07	2428	1.32	11	0.06	10	0.06
$N = 10$, $\sigma = 6.666$, $E(j)_{spect.exp}$= 51.0021848							
ξ	T_{SE}	I_{MG}	T_{MG}	I_{NA}	T_{NA}	I_{LA}	T_{LA}
10^{-3}	0.06	1357	0.78	10	0.06	9	0.06
10^{-6}	0.07	43377	24.78	15	0.07	14	0.07
10^{-9}	0.07	510683	278.35	19	0.07	18	0.06
10^{-12}	0.07	1176710	635.94	20	0.07	19	0.07

Table 4.4: Effect of the system load and the desired relative bias on computation time (finite system)

and Table 4.6 together with the time of the SE, LA and NA methods for infinite and finite cases, respectively. The stopping criteria was fixed at $\epsilon = 10^{-12}$. For representative purposes, we also report the relative difference between the mean number of jobs in the system computed by the IS and by the SE method. The notable remark is that in case of an infinite system, the IS method proves a little bit more time consuming than the SE, LA and NA methods, which is true over all the system load. At the same time, the IS method performs slightly faster than the other ones in case of a finite system. Moreover, it produces numerical results with high precision. The relative difference between the mean number of jobs computed by the IS and by the SE method in all cases is practically zero (with the coincidence of up to 8 digits after the floating point).

Effect of stopping accuracy If we claim more precise relative bias, it takes longer to get numerical results because more iterations are needed. This tendency, however, is only observed considerably when MG is used. Generally speaking, the stricter the stopping criterion (related to the first scenario), the larger the complexity of the iterative procedure is (see Table 4.4). However, we just cannot assert that an arbitrarily small relative bias is always reachable. A possible explanation for this is that after a certain number of iterations, the performance measure produced by the iterative methods converges to a certain value. Letting the iterative procedure go on by making the stopping criteria of *Scenario I* smaller does not bring a significant change in this value, and thus in the relative bias. This

$N = 30$, $\epsilon = 10^{-12}$					
σ	T_{SE}	T_{NA}	T_{LA}	T_{IS}	Rel. diff. of IS
$0.3 * N$	15.48	15.40	15.47	16.77	0 %
$0.6 * N$	15.48	15.44	15.48	16.82	0 %
$0.66 * N$	15.50	15.45	15.52	16.85	0 %
$0.666 * N$	15.60	15.49	15.54	16.93	$1.4 * 10^{-6}$
$0.6666 * N$	15.53	15.48	15.56	19.41	$5.6 * 10^{-5}$

Table 4.5: Effect of the system load on the computation time of IS method (infinite system)

$N = 30$, $\epsilon = 10^{-12}$					
σ	T_{SE}	T_{NA}	T_{LA}	T_{IS}	Rel. diff. of IS
$0.3 * N$	18.95	18.73	18.85	18.76	0 %
$0.6 * N$	18.91	18.76	18.87	18.78	0 %
$0.66 * N$	18.92	18.88	18.94	18.82	0 %
$0.666 * N$	18.94	18.87	19.05	18.89	0 %
$0.6666 * N$	18.92	18.94	19.04	23.43	0 %

Table 4.6: Effect of the system load on the computation time of IS method (finite system)

situation occurs mainly on the verge of stability border as illustrated in Figure 4.4. As one can see, allowing the LA and NA iterative algorithms to run longer by decreasing the stopping criteria ϵ does not lead to a better relative bias, because it becomes unchanged after while.

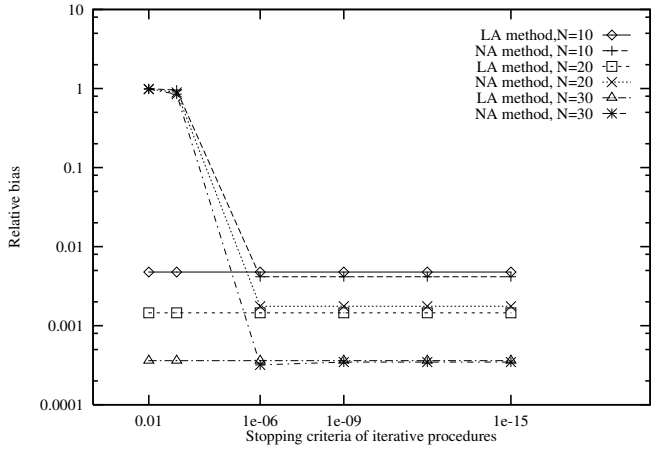


Figure 4.4: Relative bias ξ versus stopping criteria ϵ of *Scenario II* ($\sigma = 0.666666 * N$)

Effect of the system's size The capability of the applied methods in question can also be examined by changing the dimension of the system. Table 4.7 shows the results for the infinite case that are obtained when N is changing. The parameter set is $\sigma = 0.6 * N$

(moderately loaded system), $\mu = 1.0$ and the relative bias $\xi = 10^{-3}$. Table 4.8 reports the same investigation for the finite case with $\sigma = 0.6 * N$ (moderately loaded system), $\mu = 10/N$ and the relative bias $\xi = 10^{-3}$.

N	T_{SE}	I_{MG}	T_{MG}	I_{NA}	T_{NA}	I_{LA}	T_{LA}
5	0.01	410	0.06	8	0.01	7	0.01
10	0.04	629	0.38	9	0.02	8	0.03
15	0.22	848	0.88	9	0.20	8	0.20
20	1.05	1067	2.92	10	1.03	9	1.05
25	4.80	1285	8.42	10	4.75	9	4.79
30	15.77	1504	22.30	10	15.68	9	15.68
35	42.78	1722	53.98	10	42.55	9	42.58
40	97.03	1941	114.61	11	96.77	10	96.81
45	202.38	2159	229.61	11	201.86	10	201.90
50	377.31	2377	420.58	11	376.99	10	377.08

Table 4.7: Computation time versus the system's dimension (infinite system, $\xi = 10^{-3}, \sigma = 0.6 * N$)

N	T_{SE}	I_{MG}	T_{MG}	I_{NA}	T_{NA}	I_{LA}	T_{LA}
5	0.02	13	0.01	3	0.01	2	0.01
10	0.06	455	0.35	9	0.06	8	0.06
15	0.37	112	0.44	7	0.34	6	0.34
20	1.37	42	1.35	5	1.30	4	1.30
25	5.61	26	5.44	4	5.41	3	5.41
30	17.88	19	17.55	4	17.58	3	17.69
35	47.46	15	47.07	4	47.07	3	47.08
40	107.08	12	106.56	4	106.43	3	106.34
45	217.29	11	216.64	3	216.77	2	216.62
50	405.62	10	404.67	3	404.70	2	404.73

Table 4.8: Computation time versus the system's dimension (finite system, $\xi = 10^{-3}, \sigma = 0.6 * N$)

One can observe that the total computation time increases with the system's size. In case of infinite systems, the MG method seems a bit more time-consuming than the other ones. However, in case of finite systems, all the four methods show almost the same efficiency as regards to computation time. The reasons for this lie in two factors. First, the required relative bias is quite loose and secondly, the system is not considered under heavily loaded conditions.

The next two tables (Table 4.9 and Table 4.10) include results related to the IS method for infinite and finite systems. The regular tendency is observable again. The IS method is slightly slower than the SE, LA and NA methods for the case of infinite systems. At the same time, it performs slightly faster than the other methods for the case of finite systems. Again, the relative bias related to performance parameters between the IS and

$\sigma = 0.66 * N$, $\epsilon = 10^{-12}$					
N	T_{SE}	T_{NA}	T_{LA}	T_{IS}	Rel. diff. of IS
10	0.03	0.03	0.03	0.04	0 %
20	1.04	1.01	1.02	1.15	0 %
30	15.50	15.45	15.52	16.85	0 %
40	96.86	96.78	96.86	103.41	0 %
50	377.08	376.93	377.02	402.59	0 %
60	1150.30	1150.60	1150.48	1237.96	0 %

Table 4.9: Effect of the system size on the computation time of the IS method (infinite system)

$\sigma = 0.66 * N$, $\epsilon = 10^{-12}$					
N	T_{SE}	T_{NA}	T_{LA}	T_{IS}	Rel. diff. of IS
10	0.07	0.07	0.07	0.07	0 %
20	1.48	1.46	1.47	1.45	0 %
30	18.91	18.81	18.91	18.82	0 %
40	113.16	112.99	113.03	112.86	0 %
50	426.87	426.94	427.12	426.57	0 %
60	1286.34	1286.21	1286.92	1300.54	0 %

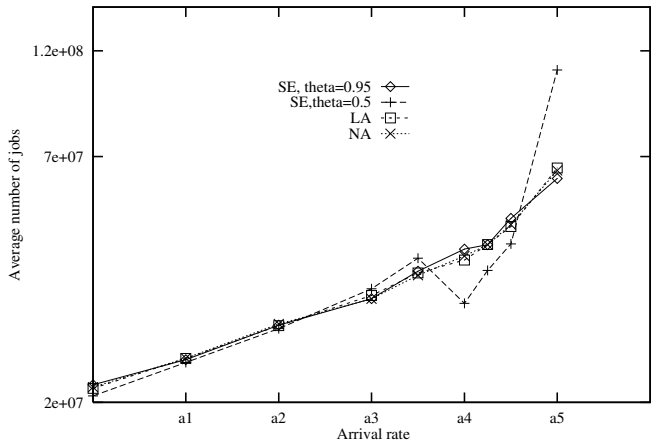
Table 4.10: Effect of the system size on the computation time of the IS method (finite system)

SE method is practically zero.

4.4.2 Numerical stability

Table 4.1 also confirms the stability of SE, LA and NA over MG. Differing from the case of MG method, the deterioration is not observed on the verge of the system's stability. However, if the considered system has a finite buffer, the computational methods do not show any failure during their operation.

Experiences show that the SE method is sensitive to the accuracy of the computed eigenvalues. When a turning parameter (referred to as θ) should be used during the procedure calculating eigenvalues and eigenvectors (see [16]) and the system load is very close to the saturation point, some violations in numerical results occur. The situation is illustrated in Figure 4.5. Taking a look at the neighborhood of $a4$, one can observe a failure exhibited in two facts. The first one is that the results calculated with different θ-s are diverse. The second one is that the fundamental rule, according to which the average number of jobs in the system must increase with the offered load, is violated.



	arrival rate (σ)
a1	6.6666652
a2	6.6666654
a3	6.6666656
a4	6.6666658
a5	6.666666

Figure 4.5: On the verge of the system's stability

4.5 Conclusions and contributions

In this chapter, through a case study of a non-trivial repair-breakdown processors system, capability of some numerical methods mostly used for steady state analysis of QBD processes has been evaluated and compared to each other. Introducing two different stopping scenarios for iterative methods, we have carried out the performance comparison for both infinite and finite cases. Numerical aspects of interest were time complexity and numerical stability.

The contributions of this chapter are the comparative conclusions gained from numerical tests. Moreover, the choosing policy has been addressed to the issue of what numerical method should be chosen for solving a QBD process with given parameters.

In case of slightly loaded infinite systems, all the methods exhibit nearly the same efficiency. Increasing system load makes the use of the matrix geometric method impractical due to its extremely long running time. The invariant subspace method becomes a little bit more time consuming than the spectral expansion, Latouche's and Naoumov's method. Approaching closely the saturation point of the system, Latouche's and Naoumov's methods may produce results with large difference compared to that of the spectral expansion due to some numerical problems of SE. Therefore, in a tight neighborhood of the saturation point it may not be advisable to use the spectral expansion method.

In case of finite systems, the invariant subspace method seems to be slightly superior over the other methods. Moreover, numerical results of the invariant subspace method shows great

coincidence with that obtained by the spectral expansion method, provided that the stability border is not approached too closely to deteriorate the goodness of the latter one.

The comparative work can be extended in several directions. First of all, the set of practical examples may be included in a wider range. The more queueing phenomena are studied, the more overall and authentic states may be concluded. Other direction of moving ahead is to take other methods mentioned in Section 3.1 into comparison. Obviously, this trend will require more efforts concerning implementation, as well as evaluation of numerical results.

Parts of this chapter have been published in our papers [38, 37].

Part IV

QBD-M processes

Chapter 5

Extension of the QBD model - QBD-M processes

5.1 Motivations

Recall that jumps in the level dimension of the process of interest so far have been restricted to three possible values: -1, 0 or $+1$, hence the interpretation is accepted as a *quasi* birth-death (QBD) process. One of the plausible generalizations of QBD processes is the relaxation of the limitation hiding on level jumps, i.e. taking queueing systems having batch arrivals and batch departures into consideration. Furthermore, the original feature according to which jumps on level have to be finite, should be retained. This motivated us to begin examining such queueing systems in which *upper-bounded* arrival, as well as *upper-bounded* departure batches are presence. Queueing models applied to such systems will be referred to as QBD-M processes, where the letter M alludes to multiple jumps in level dimension.

It should be emphasized that there are a number of practical phenomenon fitting the QBD-M model. Systems with bursty arrivals and with batch services are familiar not only in many telecommunications related fields but also in other ones such as manufacturing, transportation etc. Without the attempt of an exhaustive review, we will recall some of the typical applications below.

The first application scenario is concerned with ATM technology. In ATM technology, one of the most important design issues is the performance evaluation of applied ATM devices such as ATM multiplexers, concentrators and inverse multiplexers. The common feature of such devices is that their input line(s) may operate at different speeds from those of the output line(s). Due to the slotted time feature of ATM systems, discrete queueing analysis is typically and widely used. Discrete analysis in turn requires a common time scale, that is often chosen as the largest time slot corresponding to the lowest speed line. The occurrence of batch arrivals and/or batch departures then comes from the fact that the chosen time scale consists of a finite number of

time slots corresponding to the other lines (with higher speed).

On the other hand, in today's packet switched telecommunications infrastructure, a number of networking technologies such as TCP/IP, ATM, SDH etc. are co-operating. Owing to the individual protocols and technologies, data packets of different size are handled within each subnetwork operating with its own technology. At the edge devices bridging such subnetworks, the fact that incoming and outgoing packets have different size leads to the idea of applying a model with batch arrivals and batch departures. The simplest example is when an ATM link serves internet traffic. All IP packets have the size of several ATM cells. So if we consider an ATM cell as a "job unit", then, obviously, the arrival of each IP packet is interpreted as a batch arrival of ATM cells.

In a wireless environment, the analysis of high capacity cellular communication systems also highlights the occurrence of batch arrivals and batch departures. Let us assume that mobile platforms such as buses, trains, taxicabs etc. may support multiple calls simultaneously. When these vehicles are crossing the boundary of their current cellular territory, multiple handoff event occurs. To treat with this multiple handoff problem during analysis of cellular systems, one should make use of a queueing model in which batch arrivals and batch departures (related to both handoff calls and new calls, as well) are considered. Since a given mobile platform can only support a finite number of simultaneous calls, it is realistic to assume that batches have finite upper bound.

In other areas, such as manufacturing or transportation, the use of batch servers is really popular. Each server has the service facility of maximum simultaneous jobs. If an arrival process is batch arrival process reflecting the bursty nature of the traffic, then the analysis of such systems clearly requires analytical techniques related to systems with batch arrivals and batch departures.

Some introductory examples described above altogether with many others confirm the fact that the QBD-M process could be a useful queueing model applicable to the analysis of many problems arising in different scientific areas. Efficient numerical solution for QBD-M processes, therefore, has received extensive research efforts and has been a grate challenge with growing practical importance.

In what follows we will focus on one of the basic computational tasks. Namely, we deal with the issue of how to determine efficiently the steady state probabilities of a given QBD-M process. The efficiency here covers many aspects, such as computation time, numerical complexity, numerical accuracy and numerical stability. First of all, let us start with some fundamentals of QBD-M processes.

5.2 Mathematical description of QBD-M processes

Consider a discrete markovian process described by two-dimensional states. Each state of the process at observation time n is denoted by two integer valued random variables I_n and J_n. I_n is taking a finite set of values $\{0, 1, \ldots N\}$ and J_n is taking an infinite set of values $\{0, 1, \ldots\}$. The states with the same value of J_n ($J_n = j$) define a level (j), whereas the states with the same value of I_n ($I_n = i$) define a phase (i). Upward and downward multiple jumps in the level dimension are possible but are required to be upper-bounded. The process is referred to as a QBD-M process. Note that QBD processes are a special case of QBD-M processes when both the upward bound and downward bound equal to 1. The possible transitions underlying this Markov process are given by the following matrices

- A_j: purely phase transitions – From state (i, j) to state (k, j) $(0 \leq i, k \leq N; j = 0, 1, \ldots)$
- $B_{j,s}$: bounded $s-$step upward transitions – From state (i, j) to state $(k, j + s)$ $(0 \leq i, k \leq N; 1 \leq s \leq y_1; y_1 \geq 1; j = 0, 1, \ldots)$
- $C_{j,s}$: bounded $s-$step downward transitions – From state (i, j) to state $(k, j - s)$ $(0 \leq i, k \leq N; s \leq j; 1 \leq s \leq y_2; y_2 \geq 1; j = 0, 1, \ldots)$. If $j < s$ then $C_{j,s} = 0$.

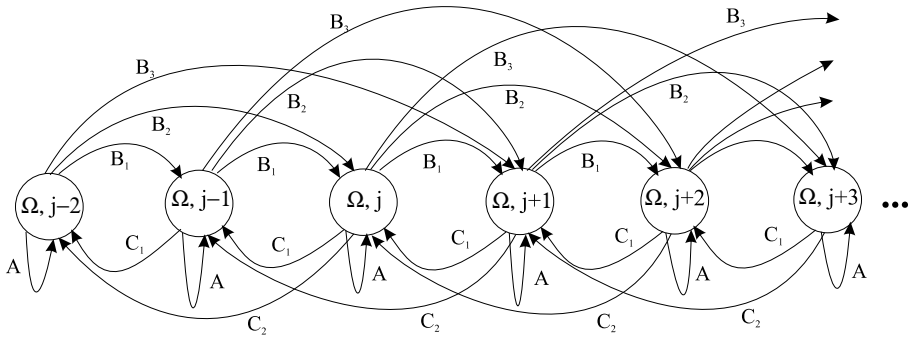
The QBD-M process is assumed to have a multiple boundary of size M, i.e. there is a threshold M, $M \geq y_1$, such that

$$A_j = A, j \geq M; \;\; B_{j,s} = B_s, j \geq M - y_1; \;\; C_{j,s} = C_s, j \geq M. \tag{5.1}$$

The transition probability matrix of the process has the form shown in Figure 5.1. An example of the state diagram is illustrated in Figure 5.2.

$$P = \left[\begin{array}{cccccc|ccccc}
A_0 & B_{0,1} & B_{0,2} & B_{0,3} & & & & & & & \\
C_{1,1} & A_1 & B_{1,1} & B_{1,2} & B_{1,3} & & & & & & \\
C_{2,2} & C_{2,1} & A_2 & B_{2,1} & B_{2,2} & B_{2,3} & & & & & \\
 & C_{3,2} & C_{3,1} & A_3 & B_{3,1} & B_{3,2} & B_3 & & & & \\
 & & C_{4,2} & C_{4,1} & A_4 & B_{4,1} & B_2 & B_3 & & & \\
 & & & C_{5,2} & C_{5,1} & A_5 & B_1 & B_2 & B_3 & & \\
\hline
 & & & & C_{6,2} & C_{6,1} & A & B_1 & B_2 & B_3 & \\
 & & & & & C_{7,2} & C_1 & A & B_1 & B_2 & B_3 \\
 & & & & & & C_2 & C_1 & A & B_1 & B_2 \\
 & & & & & & & C_2 & C_1 & A & B_1 \\
 & & & & & & & & \ddots & \ddots & \ddots
\end{array}\right]$$

Figure 5.1: The block structure of the transition probability matrix of a QBD-M process ($y_1 = 3, y_2 = 2$, $M = 6$)



$y_1 = 3$, $y_2 = 2$
Ω: the set of phases
A,B_1, B_2, B_3, C_1,C_2: trans. prob. matrixes

Figure 5.2: State diagram of a QBD-M process

Using the same notations introduced earlier in Chapter 3, we are now ready to give basic equations of the process as follows. For $j = 0, 1, \ldots, M-1$, the balance equations of the system are

$$\underline{v}_j = \sum_{s=1}^{y_1} \underline{v}_{j-s} B_{j-s,s} + \underline{v}_j A_j + \sum_{s=1}^{y_2} \underline{v}_{j+s} C_{j+s,s}. \tag{5.2}$$

(It is assumed $\underline{v}_{j-s} = \underline{0}$ if $j < s$). For $j \geq M$, the corresponding $j-$independent set is

$$\underline{v}_j = \sum_{s=1}^{y_1} \underline{v}_{j-s} B_s + \underline{v}_j A + \sum_{s=1}^{y_2} \underline{v}_{j+s} C_s. \tag{5.3}$$

In addition, since the sum of all probabilities must be one, we have:

$$\sum_{j=0}^{\infty} \underline{v}_j \underline{e} = 1. \tag{5.4}$$

Investigation now is focused on determining steady state probabilities of the process, i.e. calculating all $p_{i,j} = \lim_{n\to\infty} Pr(I_n = i, J_n = j)$ from the given parameters. In the remainder of this chapter and in the next chapter we assume the knowledge of the probability transition matrix of a given QBD-M process. The concrete construction of the probability transition matrix is case-dependent, i.e. it must be done based on the characteristic of the system under analysis. In general, this task is not too complicated and can be performed by means of simple probabilistic considerations.

5.3 Existing computational methods for the steady state solution of QBD-M processes

Solution approach for finding steady state distribution of QBD-M processes may rely on the fact that QBD-M processes are viewed as an extension of QBD processes in the sense of batch arrivals and departures. Naturally, the fact that jumps in the level dimension may occur in the form of batches makes the numerical solution more complicated and more time-consuming compared to the case of QBD processes. Therefore, either well-known methods need to be extended or new algorithms need to be developed to solve QBD-M processes. In the line of this approach, numerical methods proposed so far for steady state analysis of QBD-M processes are found e.g. in [16, 36, 55, 75, 76]. We can classify these methods with respect to their features into the following categories.

5.3.1 Reblocking related solutions

Since the QBD-M process is the extended version of the QBD process, it is straightforward that one can somewhat exploit the capability of computational methods developed for QBD processes. This observation leads to the solution way presented in [36], which requires the use of reblocking technique to get a standard QBD process having level probability vectors of augmented size. The transformed QBD process then can be solved by the methods developed for QBD processes available e.g. in [9, 12, 16, 19, 53, 57]

As regards the details, first the adequate enlargement of the block size of the transition probability matrix P is carried out to obtain a transition probability matrix in a form similar to that of a standard QBD process. In this way, new blocks with size $l(N+1) \times l(N+1)$ are introduced, where $l = max(y_1, y_2)$. This step is often referred to as block size enlargement or reblocking. It is important to point out that the reblocking is only done for the regular part of the transition probability matrix P (the part with repeating row structure) regardless of the size of the boundary part. In other words, it is not required that the boundary part in the transition probability matrix P contains an integer number of blocks of augmented size. In Figure 5.1 this kind of reblocking is illustrated. Next, it is straightforward to apply the existing methods detailed in Chapter 3 to the transformed QBD process in order to calculate the steady state probabilities.

This kind of solution is the simplest and most convenient from the viewpoint of implementation, since one can re-use the existing numerical methods without any modifications. However, it is important to emphasize that reblocking increases both the execution time and the storage requirement. Particularly, when the maximum number of arrivals and departures in batches is considerably large, this solution method becomes impractical. The significant increase in the execution time will be demonstrated later in Chapter 7.

Another set of methods published in [74, 75, 76] also benefit from the advantage of reblocking.

Namely, they do utilize the geometric relation between level probability vectors of the transformed QBD process. However, by observing some additional specific relations, they provide reduction in the time and space requirements of the computing task.

5.3.2 Direct solutions

The spectral expansion method is capable to solve systems with upper-bounded batch arrivals and batch departures without any additional manipulations such as re-blocking [16].

Let $y = y_1 + y_2$ and rewrite equation (5.3) as

$$\sum_{k=0}^{y} \underline{v}_{j+k} D_k = 0, \qquad j \geq M - y_1, \tag{5.5}$$

where

$$D_k = \begin{cases} B_{y_1-k} & \text{for } k = 0, 1, \ldots, y_1 - 1 \\ A - I & \text{for } k = y_1 \\ C_{k-y_1} & \text{for } k = y_1 + 1, y_1 + 2, \ldots, y_1 + y_2 \end{cases} \tag{5.6}$$

Applying the spectral expansion method implies that all the level probability vectors can be expressed in the form

$$\underline{v}_j = \sum_{k=0}^{y_1(N+1)-1} a_k \underline{\psi}_k \lambda_k^{j-(M-y_1)}, \qquad j \geq M - y_1, \tag{5.7}$$

where λ_k is the k-th eigenvalue strictly inside the unit disk and $\underline{\psi}_k$ is the corresponding left eigenvector of the characteristic matrix polynomial

$$D(\lambda) = \sum_{k=0}^{y} D_k \lambda^k,$$

i.e. they satisfy the equations

$$\underline{\psi} D(\lambda) = 0. \tag{5.8}$$

Using the form (5.7) and taking the first M level-dependent equations (5.2), as well as the normalized equation (5.4) one gets a set of linearly independent equations, which has a unique solution of $\underline{v}_0, \ldots, \underline{v}_{M-y_1-1}, \underline{a}$, where $\underline{a} = (a_0, \ldots, a_{y_1(N+1)-1})$ is the co-efficient vector.

The key issue related to this direct solution of QBD-M processes is the computation of all relevant eigenvalues and eigenvectors. For this purpose, in [16] a detailed computational procedure was proposed and discussed in an excellent way. Note that due to its non-iterative nature, the spectral expansion method theoretically provides exact numerical results of steady state probabilities, and through them exact performance measures can be calculated. However, according to our numerical experiments sometimes it has problems with numerical stability due to the "goodness" of the procedure computing eigenvalues and eigenvectors. This observation

is also mentioned by other authors in [36, 75]. So far, no satisfactory explanation for such operation failures are brought out and this should still be a research challenge.

We note that the cyclic reduction technique based on Fast Fourier Transforms available in [12] is another candidate method for the steady state solution of QBD-M processes. In addition, for a certain subclass of QBD-M processes, a recent method of [55] also provides the exact solution in a known number of steps. This method exploits the role of a so called companion matrix and operates on it.

5.4 Conclusions

In this chapter, the concept of QBD-M processes has been introduced. We have pointed out the necessity of QBD-M processes and its interpretation as the generalization of QBD processes. The extension of QBD processes is required because modeling and analysis of numerous problems arising in several areas such as traditional and wireless telecommunications, computer systems, manufacturing, transportation etc. fit well this class of queueing model. For illustrations, some practical phenomenon have been described in this chapter .

We have summarized existing relevant and related computational methods for the steady state solution of QBD-M processes. Within this brief literature survey, advantages and drawbacks of these methods have been delineated. For example, the block size enlargement technique allows the application of all methods developed for QBD processes at the expense of increased storage requirements and computational complexity. Spectral expansion is one of the direct solving techniques (i.e. without re-blocking) with a quite complicated procedure of computing eigenvalues and eigenvectors, which has some unanswered numerical problems.

In the next chapter, some new computational methods developed for steady state solution of QBD-M processes will be presented.

Chapter 6

Developed numerical methods for the steady state solution of QBD-M processes

In this chapter two new computational methods developed for the steady state solution of QBD-M processes are presented. One of them is a simple iterative method, which is an extension of the method proposed in [74, 75]. This method makes use of the block size enlargement (reblocking) technique. Its gains in computation complexity and storage requirements are offered by revealing special relations between steady state level probability vectors. Another method is derived from the theory of generalized invariant subspace with the application of the matrix sign function.

6.1 A simple iterative method

6.1.1 Mathematical description

With the use of block size enlargement, the introduction of blocks of size $l(N+1) \times l(N+1)$, $l = max(y_1, y_2)$ brings the probability transition matrix to the form of a standard QBD process. For the transformed QBD, if the classical matrix geometric method or its improved versions are used, a rate matrix $\mathbf{R}$ of size $l(N+1) \times l(N+1)$ should be computed. For the case $y_1 \geq y_2$ (i.e. $l = y_1$), denote the $(N+1) \times (N+1)$ submatrices of this $\mathbf{R}$ as

$$\mathbf{R} = \begin{bmatrix} R_{1,1} & \cdots & R_{1,y_2} & \cdots & R_{1,y_1} \\ \vdots & & & & \vdots \\ R_{y_1,1} & \cdots & R_{y_1,y_2} & \cdots & R_{y_1,y_1} \end{bmatrix}. \tag{6.1}$$

We will show that the steady state probabilities and performance measures of the original QBD-M process in the case $y_1 \geq y_2$, can be calculated based only on $y_1 y_2$ submatrices of the rate

matrix **R**.

Theorem 6.1.1 *If the steady state distribution of the QBD-M process exists then the following equations hold*

$$\underline{v}_j = \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i} T_{i,0}, \quad \forall j \geq M \tag{6.2}$$

$$\underline{v}_{j+k} = \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i+1} T_{i,k-1}, \quad \forall j \geq M, \quad \textit{and } 1 \leq k \leq y_2 \leq y_1 \tag{6.3}$$

where $T_{i,k-1} = R_{i+1,k}$.

Proof: The transition probability matrix P of a given QBD-M process has a repeating row structure from level M. A very essential observation now is that in this matrix P, re-blocking can be done from any level $j \geq M$ without destroying the regular structure of the process.

Once reblocking has been done from level j ($j \geq M$), there is a geometric relation for "new" level vectors that have a size of $y_1(N+1)$. Therefore, we have

$$[\underline{v}_j, \ldots, \underline{v}_{j+y_1-1}] = [\underline{v}_{j-y_1}, \ldots, \underline{v}_{j-1}]\mathbf{R}, \quad \forall j \geq M. \tag{6.4}$$

After some algebra we obtain (6.2) from (6.4).

A similar thought leads to

$$[\underline{v}_{j+1}, \ldots, \underline{v}_{j+k}, \ldots, \underline{v}_{j+y_2}, \ldots, \underline{v}_{j+y_1}] = [\underline{v}_{j+1-y_1}, \ldots, \underline{v}_j]\mathbf{R}, \quad \forall j \geq M. \tag{6.5}$$

After some algebra we obtain (6.3) from (6.5) □

The special relations revealed in this theorem denote the fact that if we are able to compute either the first $(N+1)$ or the first $(N+1)y_2$ columns of **R** ($T_{i,k-1},\ 0 \leq i \leq y_1-1,\ 1 \leq k \leq y_2$) then the steady state distribution of the process becomes available. In what follows we will provide a simple iterative procedure to compute the first $(N+1)y_2$ columns of **R**.

Theorem 6.1.2 *The* $T_{i,j}$, $0 \leq i \leq y_1-1$, $0 \leq j \leq y_2-1$ *matrices introduced in Theorem 6.1.1 are the minimal non-negative solutions of the following system of matrix equations:*

$$T_{0,0} = B_{y_1} + T_{0,0}(A + \sum_{k=1}^{y_2} T_{y_1-1,k-1} C_k) \tag{6.6}$$

$$T_{i,0} = B_{y_1-i} + T_{i,0}(A + \sum_{k=1}^{y_2} T_{y_1-1,k-1} C_k) + \sum_{k=1}^{y_2} T_{i-1,k-1} C_k \tag{6.7}$$

$$\text{for } 1 \leq i \leq y_1 - 1$$

$$T_{i,k} = \sum_{ii=1}^{k} T_{i,ii-1} T_{ii+y_1-(k+1),0} \tag{6.8}$$

$$\text{for } 1 \leq k \leq y_2 - 1 \text{ and } 0 \leq i \leq k-1$$

$$T_{i,k} = \sum_{ii=1}^{k} T_{i,ii-1} T_{ii+y_1-(k+1),0} + T_{i-k,0} \tag{6.9}$$

$$\text{for } 1 \leq k \leq y_2 - 1 \text{ and } k \leq i \leq y_1 - 1$$

In order to prove Theorem 6.1.2, we must use the result of Theorem 6.1.1 and the assumption $y_1 \geq y_2$. To see a proof please refer to Appendix A.1.

Theorem 6.1.3 *Setting* $X_{i,j}^{(0)} = 0$, $0 \leq i \leq y_1 - 1$, $0 \leq j \leq y_2 - 1$ *the iteration*

$$X_{0,0}^{(n+1)} = B_{y_1} + X_{0,0}^{(n)} \left(A + \sum_{k=1}^{y_2} X_{y_1-1,k-1}^{(n)} C_k \right) \tag{6.10}$$

$$X_{i,0}^{(n+1)} = B_{y_1-i} + X_{i,0}^{(n)} \left(A + \sum_{k=1}^{y_2} X_{y_1-1,k-1}^{(n)} C_k \right) + \sum_{k=1}^{y_2} X_{i-1,k-1}^{(n)} C_k \tag{6.11}$$

$$\text{for } i = 1, \ldots, y_1 - 1$$

$$X_{i,k}^{(n+1)} = \sum_{ii=1}^{k} X_{i,ii-1}^{(n)} X_{ii+y_1-(k+1),0}^{(n)} \tag{6.12}$$

$$\text{for } 1 \leq k \leq y_2 - 1 \text{ and } 0 \leq i \leq k-1$$

$$X_{i,k}^{(n+1)} = \sum_{ii=1}^{k} X_{i,ii-1}^{(n)} X_{ii+y_1-(k+1),0}^{(n)} + X_{i-k,0}^{(n)} \tag{6.13}$$

$$\text{for } 1 \leq k \leq y_2 - 1 \text{ and } k \leq i \leq y_1 - 1$$

converges to the minimal non-negative solutions of equations written in Theorem 6.1.2

The proof of this theorem can be found in Appendix A.2.

6.1.2 Computing algorithm

Using Theorem 6.1.3 the iterative algorithm shown in Figure 6.1 can be used to obtaine $T_{i,j}$, $0 \leq i \leq y_1 - 1$, $0 \leq j \leq y_2 - 1$ and through them the vectors $\underline{v}_j$ $(j = 0, 1, \ldots)$.

Once the relevant matrices $T_{i,j}$ $(0 \leq i \leq y_1 - 1, 0 \leq j \leq y_2 - 1)$ have been obtained by the new iterative algorithm, the boundary level probability vectors $\underline{v}_0, \ldots, \underline{v}_{M-1}$ need to be computed to make the steady state distribution available. This is because once $\underline{v}_0, \ldots, \underline{v}_{M-1}$ are known, the rest of $\underline{v}_j$, $j \geq M$ can be determined by means of equation (6.2). In order to get $\underline{v}_0, \ldots, \underline{v}_{M-1}$, equations (5.2) and (5.4) must be used. First of all, an additional expression is stated as follows

For all $0 \leq i \leq y_1 - 1, 0 \leq j \leq y_2 - 1$
  set $T_{i,j}^{(0)} = 0$
$n = 0$
DO
  -Compute $T_{i,j}^{(n+1)}$ by Theorem 6.1.3
  for all $0 \leq i \leq y_1 - 1, 0 \leq j \leq y_2 - 1$
  $n = n + 1$
WHILE $(\max_{i,j}(||T_{i,j}^{(n)} - T_{i,j}^{(n-1)}|| \geq \epsilon)$

Figure 6.1: The proposed numerical method to obtain the T matrices

Corollary 6.1.1 *For any* $1 \leq k \leq y_2$

$$\underline{v}_{M+k} = \underline{v}_{M-y_1} T_{0,0} T_{y_1-1,k-1} + \sum_{i=1}^{y_1-1} \underline{v}_{M-y_1+i} \left(T_{i-1,k-1} + T_{i,0} T_{y_1-1,k-1} \right) \tag{6.14}$$

Proof: By using Theorem 6.1.1, for $j \geq M$ and $1 \leq k \leq y_2$ we have

$$\begin{aligned}
\underline{v}_{j+k} &= \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i+1} T_{i,k-1} \\
&= \sum_{i=0}^{y_1-2} \underline{v}_{j-y_1+i+1} T_{i,k-1} + \underline{v}_j T_{y_1-1,k-1} \\
&= \sum_{i=1}^{y_1-1} \underline{v}_{j-y_1+i} T_{i-1,k-1} + \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i} T_{i,0} T_{y_1-1,k-1} \\
\underline{v}_{j+k} &= \underline{v}_{j-y_1} T_{0,0} T_{y_1-1,k-1} + \sum_{i=1}^{y_1-1} \underline{v}_{j-y_1+i} \left(T_{i-1,k-1} + T_{i,0} T_{y_1-1,k-1} \right)
\end{aligned}$$

By substituting $j = M$, the Corollary is proven. □

Writing equations (5.2) and taking Corollary 6.1.1 into account we see that the unknown vectors from the boundary balance equations of level j $(0 \leq j \leq M-1)$ are $\underline{v}_0, \ldots, \underline{v}_{M-1}$ and the number of linearly independent equations is the same as the number of unknown vectors (M). However, to get a set of linearly independent equations, one of the vector equations above must be replaced by the normalizing equation (5.4). The infinite sum in equation (5.4) is resolved as follows

Theorem 6.1.4

$$\sum_{j=M}^{\infty} \underline{v}_j = \sum_{i=0}^{y_1-1} \underline{v}_{M-y_1+i} \left(\sum_{n=0}^{i} T_{n,0} \left(I - \sum_{l=0}^{y-1} T_{l,0} \right)^{-1} \right) \tag{6.15}$$

Proof: This proof is from [75]. Let $s = \sum_{j=M}^{\infty} \underline{v}_j$ and make the following transformations:

$$\begin{aligned} s &= \sum_{j=M}^{\infty} \underline{v}_j = \sum_{i=0}^{y_1-1} \sum_{j=0}^{\infty} \underline{v}_{M+jy_1+i} \\ &= \sum_{i=0}^{y_1-1} \sum_{j=0}^{\infty} \sum_{n=0}^{y-1} \underline{v}_{M+(j-1)y_1+i+n} T_{n,0} \\ &= \sum_{n=0}^{y_1-1} \left(\sum_{i=0}^{y_1-1} \sum_{j=0}^{\infty} \underline{v}_{M+(j-1)y_1+i+n} \right) T_{n,0} \\ &= \sum_{n=0}^{y_1-1} \left(\sum_{i=n}^{y_1-1} \underline{v}_{M-y_1+i} + s \right) T_{n,0} \end{aligned}$$

A simple rearrangement of the equation above justifies the Theorem □.

6.1.3 Time and space complexity of the proposed method

The time complexity is measured in FLOP unit: one FLOP unit consists of one ADD and one MULTIPLY in floating point. With matrices of size $(N+1)\times(N+1)$, each matrix multiplication requires $(N+1)^3$ FLOPs. Matrix additions will be considered to have negligible complexity.

According to the iterative procedure detailed in Theorem 6.1.3, in each iteration step the computational effort is composed of the following items:

- the calculation of $T_{0,0}$ requires $(y_2+1)(N+1)^3$ FLOPs, since there are y_2+1 matrix multiplications;
- the calculation of each $T_{i,0}$ $(i = 1, 2, \ldots, y_1 - 1)$ requires further $(y_2+1)(N+1)^3$ FLOPs, since there are $2y_2+1$ matrix multiplications, among which y_2 matrix multiplications have been performed during the calculation of $T_{0,0}$.

  Thus, $(y_1-1)(y_2+1)(N+1)^3$ FLOPs are needed to calculate all $T_{i,0}$ $(i = 1, 2, \ldots, y_1-1)$.
- the calculation of each $T_{i,k}$ $(i = 0, 1, \ldots, y_1 - 1;\ k = 1, 2, \ldots, y_2 - 1)$ requires $k(N+1)^3$ FLOPs, since k matrix multiplications must be done. Thus, for all $T_{i,k}$ $(i = 0, 1, \ldots, y_1-1;\ k = 1, 2, \ldots, y_2 - 1)$ the number of needed FLOPs is

$$\sum_{k=1}^{y_2-1} k(N+1)^3 y_1 = y_1 \frac{y_2(y_2-1)}{2}(N+1)^3.$$

Based on the results above, the total number of FLOPs needed in each iteration step of the proposed algorithm is $\left[\frac{1}{2}y_1y_2(y_2-1) + y_1(y_2+1)\right](N+1)^3$. Comparing with the existing algorithms, it turns out that this amount of complexity is surely less than $\frac{7}{3}(y_1)^3(N+1)^3$ (applying

the classical matrix geometric after reblocking), and obviously is less than $\frac{19}{3}(y_1)^3(N+1)^3$ (using Naoumov's algorithm after reblocking) and $\frac{25}{3}(y_1)^3(N+1)^3$ (using Latouche's algorithm after reblocking)[1].

The space requirement of the new method is $y_1y_2(N+1)^2$, whereas in all the other iterative methods the whole $\mathbf{R}$ must be stored, which means $(y_1)^2(N+1)^2$ space requirement.

The total time to obtain the steady state probabilities, of course, also depends on the convergence of the proposed iterative procedure. In other words, the number of necessary iteration steps plays a key role in this question. Later, in Chapter 7, by the performance test performed via some case studies, we will be able to give answer to the question of how well the new method performs in comparison to other methods with regard to system parameters.

	MG	LA	NA	New method
Complexity in each step (measured in FLOPs)	$\frac{7}{3}y_1^3(N+1)^3$	$\frac{25}{3}y_1^3(N+1)^3$	$\frac{19}{3}y_1^3(N+1)^3$	$[\frac{1}{2}y_1y_2(y_2-1)+$ $y_1(y_2+1)](N+1)^3$
Space requirement	$y_1^2(N+1)^2$	$y_1^2(N+1)^2$	$y_1^2(N+1)^2$	$y_1y_2(N+1)^2$

Table 6.1: Comparison between computational methods in terms of time and space complexity

6.2 Generalised invariant subspace based method

The fundamental material of the second method we propose is a theory of generalised invariant subspace in linear algebra, which has been first introduced and applied by Nail Akar et al. (see [5, 10]) to teletraffic problems. In order to assist better understanding, we first begin with an overview of generalised invariant subspaces and the relation between invariant subspaces and the matrix sign function. Afterwards, the computational method will be presented in the light of the first part.

6.2.1 Theory of invariant subspaces and matrix sign functions

6.2.1.1 Invariant subspaces

Let us recall briefly some notations and results related with the theory of invariant subspaces, which will play a key role in the computation method described later. The main part of this subsection is from [5]. Let $\mathbb{R}^m$ be the real linear space of column vectors of m real numbers, $\mathbb{R}^{m\times n}$ be the linear space of $m\times n$ matrices with real entries. A *subspace* is a subset of $\mathbb{R}^m$ that is closed under the operations of addition and scalar multiplication. For arbitrary subspaces

[1]Keep in mind that y_1 and y_2 are integers and $y_2 \leq y_1$

$\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{S}_1 \subset \mathcal{S}_2$ denotes either inclusion or equality. If $A \in \mathbb{R}^{m\times n}$, then the image of A is defined as

$$ImA = \{x \in \mathbb{R}^m \mid x = Ay \text{ for some } y \in \mathbb{R}^n\}.$$

If $ImA = ImB$, then there exists a nonsingular matrix U such that $B = AU$. The set of all eigenvalues of a matrix $A \in \mathbb{R}^{m\times m}$ is called the *spectrum* of A and written $\sigma(A)$.

Let $A\mathcal{S}$ denote the image of a subspace $\mathcal{S}$ under A, i.e. $A\mathcal{S} = \{x \in \mathbf{R}^m \mid x = Ay \text{ for some } y \in \mathcal{S}\}$. A subspace $\mathcal{S}$ of $\mathbb{R}^m$ is said to be A *invariant* where $A \in \mathbb{R}^{m\times m}$, if $A\mathcal{S} \subset \mathcal{S}$. If a $k-$dimensional subspace $\mathcal{S}$ is A invariant, then $AS = SA_1$ holds for a $k \times k$ matrix A_1 and an $m \times k$ matrix S whose columns form a basis for $\mathcal{S}$, i.e. $\mathcal{S} = ImS$.

Let $\mathcal{S} + \mathcal{T}$ and $\mathcal{S} \oplus \mathcal{T}$ be the sum and direct sum, respectively, of the subspaces $\mathcal{S}$ and $\mathcal{T}$. Let $\mathcal{S} \oplus \mathcal{T} = \mathbb{R}^m$ and assume that $\mathcal{S}$ and $\mathcal{T}$ are invariant subspaces of a square matrix A of size m. Then, $\mathcal{S} = ImS$ and $\mathcal{T} = ImT$ and U defined by $U = [\ S \quad T\ ]$ satisfy

$$U^{-1}AU = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}.$$

If $\sigma(A_{11})$ ($\sigma(A_{22})$) lies in the closed right-half (open left-half) plane, then $\mathcal{S}$ ($\mathcal{T}$) is said to be the right (left) invariant subspace of A. When $\sigma(A_{11})$ ($\sigma(A_{22})$) lies outside (inside) the open unit disk, then $\mathcal{S}$ ($\mathcal{T}$) is called the unstable (stable) invariant subspace of A.

Let us assume a regular matrix pencil $\lambda E - A$, which is a polynomial matrix (in the indeterminate λ) of degree one. The *generalised eigenvalue problem* for the matrices A and E of size m is equivalent to finding the scalar λ for which the equation $Ax = \lambda Ex$ has solutions $x \neq 0$. Such scalars λ are called generalised eigenvalues. A solution $x \neq 0$ corresponding to an eigenvalue λ is called a generalised eigenvector. A generalised eigenvalue satisfies the relation

$$\lambda \in \sigma(E, A) := \{\mu \in C \mid det(\mu E - A) = 0\},$$

where $\sigma(E, A)$ denotes the generalised spectrum of the matrix pair (E, A) and C is the field of complex numbers.

Any subspace $\mathcal{S}$ satisfying

$$\mathcal{T} = E\mathcal{S} + A\mathcal{S}, \quad dim(\mathcal{S}) = dim(\mathcal{T})$$

is called a *generalised invariant subspace (or deflating subspace)* of the pencil $\lambda E - A$. Note that when $E = I$, we indeed have an ordinary invariant subspace.

Now, let $\mathcal{S}$ and $\mathcal{S}_c$ be two complementary deflating subspaces of the pencil $\lambda E - A$, i.e. $\mathcal{S} \oplus \mathcal{S}_c = \mathbb{R}^m$. Define $\mathcal{T} = E\mathcal{S} + A\mathcal{S}$ and $\mathcal{T}_c = E\mathcal{S}_c + A\mathcal{S}_c$. These two subspaces are also complementary [5]. Let $\mathcal{S} = ImS$, $\mathcal{T} = ImT$, $\mathcal{S}_c = ImS_c$, $\mathcal{T}_c = ImT_c$, then there exists a decomposition

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix}, \quad U^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix},$$

where

$$U = [\ T \quad T_c\ ], \quad V = [\ S \quad S_c\ ].$$

If $\sigma(E_{11}, A_{11})$ ($\sigma(E_{22}, A_{22})$) lies in the closed right-half (open left-half) plane, then $\mathcal{S}$ ($\mathcal{S}_c$) is called the right (left) deflating subspace of the matrix pencil $\lambda E - A$. When $\sigma(E_{11}, A_{11})$ ($\sigma(E_{22}, A_{22})$ lies outside (inside) the open unit disk, then $\mathcal{S}$ ($\mathcal{S}_c$) is called the unstable (stable) deflating subspace of the matrix pencil $\lambda E - A$.

6.2.1.2 Computation of an invariant subspace via a matrix sign function

An invariant subspace of a given matrix can be calculated through its matrix sign function. The definition of a matrix sign function is given as follows.

Definition 1 *Let $X \in \mathbb{R}^{m\times m}$ be a matrix with no pure imaginary eigenvalues. Let X have a Jordan decomposition $X = T(D + Y)T^{-1}$ where $D = diag\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ and Y is nilpotent and commutes with D. Then the matrix sign of X is given by*

$$\begin{aligned} Z &= sgn(X) \\ &:= T.diag\{sgn(\lambda_1), sgn(\lambda_2), \ldots, sgn(\lambda_m)\}.T^{-1} \end{aligned}$$

where for a complex scalar z with $Re(z) \neq 0$, the sign of z is defined by

$$sgn(z) = \begin{cases} 1 & \text{if } Re(z) > 0 \\ -1 & \text{if } Re(z) < 0 \end{cases}$$

From our aspect, the most important property of $Z = sgn(X)$ is that $Im(Z - I)$ and $Im(Z + I)$ yield the left and right invariant subspace of X, respectively. This property means that an orthogonal basis for the left (right) invariant subspace of X which has a dimension r is given by the first r columns of the orthogonal matrix in a rank-revealing QR decomposition of matrix $Z - I$ ($Z + I$).

The matrix sign function can be computed efficiently in several ways. For example, a scaling Newton's scheme with Byers scalar shown in Figure 6.2 is popularly adopted to this aim. It has been verified that this iterative algorithm converges quadratically for all matrix X for which the matrix sign is well defined.

6.2.2 Generalised invariant subspace based method

6.2.2.1 Formal description

We state the following theorem.

Theorem 6.2.1 *With the application of the theory of generalised invariant subspaces, the level probability vectors of a QBD-M process can be expressed in the form*

$$\underline{v}_{k+(M-y_1)} = \underline{g}F^k H \text{ with } k \geq 0, \tag{6.16}$$

$k = 0$
$Z_0 = X$
DO
    $t = |detZ_k|^{-1/m}$
    $Z_{k+1} = \frac{1}{2}(tZ_k + t^{-1}Z_k^{-1})$
    $k = k + 1$
$WHILE\ (||Z_k - Z_{k-1}||_1 \geq \epsilon||Z_{k-1}||_1)$

Figure 6.2: A iterative procedure for obtaining the matrix sign function

where the vector $\underline{g}$ and matrices F, H are well-defined.

Proof: Let us construct the following matrix polynomial and hypermatrices:

$$D(\lambda) = D_0 + D_1\lambda + \ldots + D_y\lambda^y, \tag{6.17}$$

$$G = \begin{bmatrix} 0 & 0 & \ldots & \ldots & -D_0 \\ I & 0 & \ldots & \ldots & -D_1 \\ 0 & I & \ldots & \ldots & -D_2 \\ \vdots & \vdots & \ddots & & \\ 0 & 0 & \ldots & I & -D_{y-1} \end{bmatrix}, \quad E = \begin{bmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & D_y \end{bmatrix}, \quad T = \begin{bmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{6.18}$$

where matrices D_k $(0 \leq k \leq y)$ are determined in expression (5.6). By introducing the row vectors

$$\underline{w}_k = [\ \underline{v}_{k+(M-y_1)} \quad \cdots \quad \underline{v}_{k+M} \quad \cdots \quad \underline{v}_{k+(M+y_2-1)}\ ] \quad \text{for } k \geq 0, \tag{6.19}$$

and by algebraic manipulations one can easily show that

$$\underline{w}_{k+1}E = \underline{w}_kG, \quad k \geq 0, \tag{6.20}$$

$$\underline{v}_{k+(M-y_1)} = \underline{w}_kT, \quad k \geq 0. \tag{6.21}$$

Note that matrices G and E have a size of $y(N+1) \times y(N+1)$, $\underline{w}_k$ vector has size of $y(N+1)$.

The main observation is that the singularities of the matrix pencil $\lambda E - G$, i.e. the roots of $det(\lambda E - G) = 0$ are exactly the roots of the equation $det(D(\lambda)) = 0$. In [16], an excellent discussion shows that when the system is stable, the equation $det(D(\lambda)) = 0$ has exactly $y_1(N+1)$ roots in the open unit disk and $y_2(N+1)$ roots outside the unit disk (including the one at $\lambda = 1$). Consequently, if the system is stable then the matrix pencil $\lambda E - G$ has $m_u = y_2(N+1)$ singularities outside the unit disk (including the one at $\lambda = 1$) and $m_s = y_1(N+1)$ singularities in the open unit disk. Let $m = m_u + m_s = y(N+1)$. Let $\mathcal{V}_1$

and $\mathcal{V}_2$ be the unstable and stable deflating subspaces of the pencil $\lambda E - G$, respectively. Let $\mathcal{V}_1 = ImV_1$ and $\mathcal{V}_2 = ImV_2$ for some matrices V_1 and V_2 of size $m \times m_u$ and $m \times m_s$, respectively. Also let $\mathcal{U}_1 := E\mathcal{V}_1 + G\mathcal{V}_1 = ImU_1$ and $\mathcal{U}_2 := E\mathcal{V}_2 + G\mathcal{V}_2 = ImU_2$ for some matrices U_1 and U_2 of size $m \times m_u$ and $m \times m_s$, respectively. Define

$$U = [\ U_1 \quad U_2\ ] \text{ and } V = [\ V_1 \quad V_2\ ] \tag{6.22}$$

then from the theory of generalised invariant subspace presented in Section 6.2.1 we have

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix} \text{ and } U^{-1}GV = \begin{bmatrix} G_{11} & 0 \\ 0 & G_{22} \end{bmatrix} \tag{6.23}$$

and $\sigma(E_{11}, G_{11})$ and $\sigma(E_{22}, G_{22})$ lie outside and in the open unit disk, respectively. Defining

$$[\ \underline{p}_k \quad \underline{q}_k\ ] = \underline{w}_k[\ U_1 \quad U_2\ ]$$

and post-multiplying the model (6.20) by V, we have two un-couple generalised difference equations for $\underline{p}_k$ and $\underline{q}_k$

$$\underline{p}_{k+1}E_{11} = \underline{p}_k G_{11}, \quad k \geq 0, \tag{6.24}$$

$$\underline{q}_{k+1}E_{22} = \underline{q}_k G_{22}, \quad k \geq 0. \tag{6.25}$$

Since the system is considered under the condition of stability, $\underline{w}_k$ must not be divergent as $k \to \infty$, which is fulfilled if and only if

$$\underline{p}_0 = \underline{w}_0 U_1 = 0. \tag{6.26}$$

Moreover, since $\sigma(E_{22}, G_{22})$ lie in the open unit disk, E_{22} is nonsingular and we can write

$$\underline{q}_k = \underline{q}_0 F^k, \tag{6.27}$$

where the $m_s \times m_s$ matrix F is found as

$$F = G_{22}E_{22}^{-1}. \tag{6.28}$$

Let us partition U^{-1} as

$$U^{-1} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}, \tag{6.29}$$

where the size of L_1 and L_2 is $m_u \times m$ and $m_s \times m$, respectively, then

$$\underline{v}_{k+(M-y_1)} = \underline{w}_k T = (\underline{p}_k L_1 + \underline{q}_k L_2)T = \underline{w}_0 U_2 F^k L_2 T, \quad k \geq 0. \tag{6.30}$$

By this equation we have derived the geometric form of the probability vector

$$\underline{v}_{k+(M-y_1)} = \underline{g}F^k H \text{ with } k \geq 0 \quad \underline{g} = \underline{w}_0 U_2 \text{ and } H = L_2 T. \tag{6.31}$$

□

Equation (6.31) reveals the fact that all vectors $\underline{v}_j$ $(j \geq M - y_1)$ can be expressed in terms of $\underline{v}_{M-y_1}, \ldots, \underline{v}_M, \ldots, \underline{v}_{M+y_2-1}$. Note that the boundary equations (5.2) with $0 \leq j \leq M-1$ and equations (6.26) then form the set of linear equations in which the number of unknowns is $(M + y_2)(N + 1)$. The number of equations is the same, but among them there are only $(M + y_2)(N + 1) - 1$ linearly independent ones. That means we have to replace one equation by the normalizing equation (5.4) for getting a set of linearly independent equations. Using (6.31), it can be easily shown that the equivalent form of (5.4) is

$$\sum_{j=0}^{M-y_1-1} \underline{v}_j \underline{e} + \underline{w}_0 U_2 (I - F)^{-1} L_2 T \underline{e} = 1. \tag{6.32}$$

Solving this set of equations one gets the probability vectors $\underline{v}_0, \ldots, \underline{v}_{M-y_1}, \ldots, \underline{v}_{M+y_2-1}$, based on which $\underline{w}_0$ and through it the rest $\underline{v}_j$ $(j \geq M - y_1)$ can be calculated.

6.2.2.2 Computational algorithm

Based on the previous section, one can see that the main computational task is to find bases for the unstable and stable deflating subspaces of the pencil $\lambda E - G$, leading to construction of the matrices U and V defined in (6.22). Following the same way in [5], we note that the stable (unstable) subspaces of the matrix pencil $\lambda E - G$ is equal to the left (right) deflating subspaces of the pencil $\lambda X - Y$, where the two matrices X and Y are defined as

$$X = G + E, \quad Y = G - E.$$

With this transformation, the generalised eigenvalues of the pencil $\lambda E - G$ in (outside) the unit disk are moved to the open left-half (closed right-half) plane. There is one generalised eigenvalue of $\lambda X - Y$ at the origin.

Since $\lambda E - G$ does not have any generalised eigenvalue at $\lambda = -1$, the matrix X is nonsingular and we can define $Z = X^{-1}Y$. The left (right) invariant subspace of Z is equal to the left (right) deflating subspace of the pencil $\lambda X - Y$. One now may make use of the matrix sign function to compute the left and right invariant subspace of Z. However, Z has one eigenvalue on the imaginary axis, at the origin because of which the matrix sign function can not be directly applied and therefore further transformation is needed. Let $\underline{\gamma}$, $\underline{\mu}$ be left and right eigenvectors of Z corresponding to the eigenvalues at the origin, i.e.

$$\underline{\gamma} Z = 0, \quad Z \underline{\mu} = 0, \tag{6.33}$$

then the matrix Z_e defined as

$$Z_e = Z + \frac{\underline{\mu} . \underline{\gamma}}{\underline{\gamma} . \underline{\mu}} \tag{6.34}$$

is free of imaginary-axis eigenvalues, and the left (right) invariant subspace of Z_e is equal to the left (right) invariant subspace of Z. It is not difficult to show that the vectors $\underline{\gamma}$ and $\underline{\mu}$ defined

as

$$\underline{\gamma} = [\ \underline{\pi}\ \ \underline{\pi}\ \ \ldots\ \ \underline{\pi}\ ]X, \quad \underline{\mu} = \begin{bmatrix} \underline{\mu}_0 \\ \underline{\mu}_1 \\ \vdots \\ \underline{\mu}_{y-1} \end{bmatrix}, \tag{6.35}$$

where $\underline{\pi}$ is the stationary probability vector of $D(1)$, i.e. $\underline{\pi}D(1) = 0$, $\underline{\pi e} = 1$ and

$$\underline{\mu}_0 = D_0\underline{e},\ \ \underline{\mu}_i = \underline{\mu}_{i-1} - D_i\underline{e} \ \text{ for } 1 \le i \le y-2,\ \ \underline{\mu}_{y-1} = \underline{e}$$

satisfy (6.33).

Using matrix-sign function iterations on Z_e to find bases for the unstable and stable deflating subspaces of the pencil $\lambda E - G$ leads to the construction of the matrices U and V defined as in (6.22). The computational algorithm can be summarized step-by-step as follows:

1. Define the matrices G, E, T as in (6.18).

2. Define $Z = (E+G)^{-1}(G-E)$, $\underline{\gamma}$, $\underline{\mu}$ as in (6.35) and Z_e as in (6.34). Then find $S = sign(Z_e)$ by the quadratically convergent iteration shown in Figure 6.2.

3. - Construct matrix Q_r in the rank-revealing QR decomposition $S + I = Q_r R_r \Pi_r$ and define
   $$V_1 = \textit{leading } m_u \textit{ columns of } Q_r.$$
   - Construct matrix Q_l in the rank-revealing QR decomposition $S - I = Q_l R_l \Pi_l$ and define
   $$V_2 = \textit{leading } m_s \textit{ columns of } Q_l.$$
   - Construct matrix $\widehat{Q}_r$ in the rank-revealing QR decomposition $[\ EV_1 \quad GV_1\ ] = \widehat{Q}_r \widehat{R}_r \widehat{\Pi}_r$ and define
   $$U_1 = \textit{leading } m_u \textit{ columns of } \widehat{Q}_r.$$
   - Construct matrix $\widehat{Q}_l$ in the rank-revealing QR decomposition $[\ EV_2 \quad GV_2\ ] = \widehat{Q}_l \widehat{R}_l \widehat{\Pi}_l$ and define
   $$U_2 = \textit{leading } m_s \textit{ columns of } \widehat{Q}_l.$$

4. Define U, V as in (6.22) and let E_{22} and G_{22} be the lower right $m_s \times m_s$ blocks of $U^{-1}EV$ and $U^{-1}GV$ as in (6.23), respectively. Then define F as in (6.28).

5. Solve the set of linearly independent equations for $\underline{v}_j$, $0 \le j \le M - y_1 - 1$ and $\underline{w}_0$.

6. Define L_2 as in (6.29) and also
   $$\underline{g} = \underline{w}_0 U_2 \text{ and } H = L_2 T$$
   to obtain the matrix-geometric expression $\underline{v}_{k+(M-y_1)} = \underline{g}F^k H$ for $k \ge 0$.

6.2.3 Time and space complexity of the proposed method

The time complexity of the computational algorithm presented in Section 6.2.2.2 can be evaluated step by step as follows.

In step 2, the computation of matrix Z requires $\frac{4}{3}m^3$ FLOPs, whereas the computation of Z_e requires about $\frac{1}{3}(N+1)^3+y(N+1)$ FLOPs for defining vector π and computing the product $\underline{\gamma\mu}$. The computation of the matrix sign function S requires further $\frac{4}{3}m^3$ FLOPs in each iteration. Step 3 contains four rank revealing operations. The Householder QR with column pivoting algorithm used for the rank revealing of an arbitrary matrix R of size $n \times l$ has a complexity of $4nlr - 2r^2(n+l) + \frac{4}{3}r^3$ FLOPs, where $r = rank(R)$ [33]. Using this estimation and after some simple manipulations, the complexity of step 3 is obtained as $4m^3+2m(m_u^2+m_s^2)-\frac{4}{3}(m_u^3+m_s^3)$ FLOPs. The complexity of step 4 is constituted by the two multiplications $U^{-1}EV$ and $U^{-1}GV$, which is totally $\frac{13}{3}m^3$ FLOPs.

Thus, until the point when we are ready to solve the set of linearly independent equations to get the level probabilities vectors, the overall complexity is $(N+1)^3\left[\frac{29}{3}y^3+2yy_1^2+2yy_2^2+\frac{1}{3}-\frac{4}{3}(y_1^3+y_2^3)\right]+y(N+1)+\frac{4}{3}y^3(N+1)^3I_{GIS}$, where I_{GIS} is the number of needed iterations in the iterative procedure computing the matrix sign function.

The space requirement of the method is $(y_1^2+y_1+yy_1)(N+1)^2$, since matrices F, L_2T and U_2 have to be stored.

6.3 Conclusions and contributions

This chapter has proposed two computational methods for the steady state solution of QBD-M processes. Both of them have iterative feature. The first method is expected to be fast and space saving. Its application range, however, is limited by the assumption that the upper bound of arrival batches must be greater than the upper bound of departure batches. The second method does not suffer any limitation and has quadratical convergence. Moreover, it provides closed form expression for the level probability vectors.

Although the methods have been presented in discrete time domain, we emphasize that their applicability is not restricted to this case for the following reason. Given a continuous time Markov chain with generator matrix Q, it is well-known that an embedded discrete-time Markov chain can be produced with transition probability matrix $P = Q/q+I$, where $q = max_{i,j}|Q(i,j)|$, I is the identity matrix with the appropriate dimension and the division means division of all entry of the matrix. This embedded Markov chain has the same steady state distribution with the original continuous chain [59]. This distribution now can be determined applying our methods without any difficulties.

In the next chapter performance capability of each method proposed here will be tested and

examined. Numerical comparison related to these new methods and well-known ones will be presented through practical case studies. Based on numerical results, advantages and vulnerabilities of each method will be explained and exposed in details.

Part of this chapter has been published in our papers [39, 40, 42, 44, 45].

Chapter 7

Performance comparison of computational methods via some applications of the QBD-M process

In the previous chapters, well-known and recently developed numerical methods for the steady state solution of QBD-M processes have been discussed. In this chapter we construct analytical models suitable for the performance analysis of two practical telecommunications phenomena. We show that both the models can typically be formulated as QBD-M processes. The aim of this chapter is twofold. On the one hand, we intend to demonstrate the application range of QBD-M processes in performance evaluation and system analysis. On the other hand, through these case studies the performance capability of the numerical methods will be compared to each other from several computational aspects (e.g. execution time, numerical accuracy) as a function of system parameters (e.g. system load, maximum batch size, size of repeating level).

7.1 Modelling a node performing multipath routing with load balancing

7.1.1 System description

The first case study we take now is an analysis of a node performing multipath routing with load balancing in MPLS (Multiprotocol Label Switched) networks. Here, our main goal is a comparative study of the computational methods, therefore the performance of the implemented computational methods receives attention, rather than performance results concerning to the application study itself.

We consider the case where multipath routing is realized between border (ingress and egress) routers of the MPLS domain [71] as shown in Figure 7.1. When load balancing is applied for

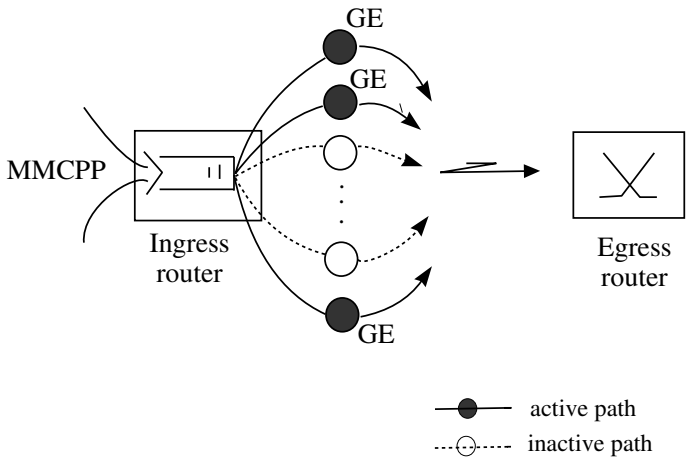


Figure 7.1: System modell for multipath routing

routing traffic between the two routers, multiple label switched paths (LSPs) may be established or released in response to the dynamics of traffic flows. The establishment and release of paths is initiated by the load balancing algorithm in order to avoid overloads and evenly distribute traffic over the paths. To achieve this aim the load balancing algorithm implicitly takes into account the total offered traffic of the network.

Supposed that the ingress router can establish maximum N paths for carrying IP traffic. Packets are arriving at the ingress router according to an MMCPP (Markov Modulated Composed Poisson Process). We opt the MMCPP modell because it can capture both the most important features, the burstiness and the auto-correlations, of real traffic [17]. The service time of each path is assumed to have GE (Generalized Exponential) distribution. The choice of the GE distribution is motivated by two facts. Firstly, it has been shown in [51] that the GE distribution is a robust two-moment approximation for any service time distribution. Secondly, use of the GE distribution assures the analytical tractability of the system. Further on, assume that single and independent establishment (release) of paths, as well as global simultaneous establishment (release) are possible. Release and establish times are assumed to be exponentially distributed. The buffer for packets in the ingress router is assumed to have an infinite capacity.

This system is modelled by a continuous time, two dimensional Markov process. At any time t, the state of the system is denoted by a couple of integers $I(t), J(t)$ in the following way:

- $I(t)$ is the operative state of the system, representing the number of established paths at time t, $I(t) = 1, \ldots, N$.
- $J(t)$ is the number of packets in the system at time t, $J(t) = 0, 1, \ldots$

First of all, we construct the generator matrix of this process. Similarly to the discrete case, let us introduce the following transition matrices

- A_j^*: purely phase transitions – From state (i,j) to state (k,j) $(0 \leq i,k \leq N; i \neq k; j = 0,1,\ldots)$

- $B_{j,s}^*$: bounded $s-$step upward transitions – From state (i,j) to state $(k,j+s)$ $(0 \leq i,k \leq N; 1 \leq s \leq y_1; y_1 \geq 1; j = 0,1,\ldots)$

- $C_{j,s}^*$: bounded $s-$step downward transitions – From state (i,j) to state $(k,j-s)$ $(0 \leq i,k \leq N; s \leq j; 1 \leq s \leq y_2; y_2 \geq 1; j = 0,1,\ldots)$. If $j < s$ then $C_{j,s} = 0$.

As one can see, we have assumed that the upward and downward transitions are not only one step, but may be up to y_1 and y_2 steps. Note that there will be a boundary M, above which the transition matrices become level-independent. In addition, let us assume that when a new batch arrives or when a completed batch departs from the system, the operative state doesn't change, unless there is an independent coincidence towards such a change. Hence, change in the operative state of the system is reflected only in the matrices A^* and A_j^*.

The task now is to build up the transition matrices $A_\cdot^*$, $B_{\cdot,\cdot}^*$ and $C_{\cdot,\cdot}^*$.

7.1.1.1 Building up the $A_\cdot^*$ matrices

As we mentioned before, there are two kinds of possible release and establishment.

- The individual paths are released independently at rate ξ and are established independently at rate η.

- The global simultaneous release of all currently operative paths occur at rate ξ_0 and the global simultaneous establishment of all currently inoperative paths occur at rate η_N.

The A_j^* matrices are j-independent and are given by

$$A^* = A_j^*(j = 0,1,\ldots) = \begin{bmatrix} 0 & N\eta & & & \eta_N \\ \xi_0 + \xi & 0 & (N-1)\eta & & \eta_N \\ \xi_0 & 2\xi & 0 & & \eta_N \\ & & & \ddots & \\ & & & & \eta + \eta_N \\ \xi_0 & & & N\xi & 0 \end{bmatrix}. \tag{7.1}$$

7.1.1.2 Building up the $B_{\cdot,\cdot}^*$ matrices

The arrival process is assumed to be an MMCPP with N phases. When the process is in phase i, the inter-arrival time distribution is GE with parameters (σ_i, θ_i). Due to the interpretation of the GE distribution (see [58]), the arrival point process is Poisson with batches arriving at each point having geometric size. The probability that a batch has size s in phase i is given by $(1-\theta_i){\theta_i}^{s-1}$.

Furthermore, suppose that when the operative state $I(t) = i$, the arrival process is in phase i. The $B^*_{.,.}$ matrices are now ready to be written

$$B^*_{j,s} = diag[\sigma_0, \sigma_1, \ldots, \sigma_N] * P(\text{arrival batch size } = s). \tag{7.2}$$

Using the GE distribution and limiting the maximum batch size to y_1, the distribution of batch size in phase i is defined by

$$P(\text{arrival batch size } = s) = \begin{cases} (1-\theta_i)\theta_i^{s-1} & 1 \le s \le y_1 - 1 \\ \theta_i^{y_1-1} & s = y_1 \end{cases} \tag{7.3}$$

It follows

$$B^*_{j,s} = diag[(1-\theta_0)\theta_0^{s-1}\sigma_0, (1-\theta_1)\theta_1^{s-1}\sigma_1, \ldots, (1-\theta_N)\theta_N^{s-1}\sigma_N] \quad if \quad 1 \le s \le y_1 - 1, \tag{7.4}$$

and

$$B^*_{j,s} = diag[\theta_0^{y_1-1}\sigma_0, \theta_1^{y_1-1}\sigma_1, \ldots, \theta_N^{y_1-1}\sigma_N] \quad if \quad s = y_1. \tag{7.5}$$

For the sake of simple control, we set all $\sigma_i = \sigma$ during the numerical test.

7.1.1.3 Building up the $C^*_{.,.}$ matrices

Let us assume that each operative path has GE service time with parameters (μ, ϕ). The departure rate depends on the current state $I(t) = i$, $J(t) = j$ and is given by the entry $C^*_{j,s}(i,i)$ of matrix $C^*_{j,s}$. Similar to [17], we postulate that the batch size associated with a service completion is bounded by one more than the number of packets waiting to commence service at the departure instant. Therefore

- if $j > i$ (j = number of packets, i = number of operative paths), the maximum batch size is $j - i + 1$. Combined with the assumption that the maximum of departure batch size is y_2, we get the form

$$P(\text{departure batch size } = s) = \begin{cases} (1-\phi)\phi^{s-1} & 1 \le s \le min(j-i+1, y_2) - 1 \\ \phi^{min(j-i+1,y_2)-1} & s = min(j-i+1, y_2) \end{cases} \tag{7.6}$$

  Furthermore, in this case the service rate related to batch departure is $i\mu$.

- For $j \le i$, since there is no packet waiting to be serviced the departing batch has size 1 with probability one. That is

$$P(\text{departure batch size } = s) = \begin{cases} 1 & \text{if } s = 1 \\ 0 & \text{otherwise} \end{cases} \tag{7.7}$$

  In this case, the service rate related to batch departure (with size one) is $j\mu$.

Taking all the aforementioned aspects into consideration we arrive at

$$C^*_{j,s} = diag[0, min(j,1).\mu, \ldots, min(j,N).\mu] * P(\text{departure batch size } = s). \qquad (7.8)$$

Note that $C^*_{j,s}(i,i)$ does not depend on j if

- $j \geq i$ (note that the maximum value of i is N),
- the term $P(\text{departure batch size } = s)$ does not depend on j, which is fulfilled when $j - i + 1 \geq y_2$.

The first condition is fulfilled if $j \geq N$, the later one is true when $j \geq y_2 + N - 1$. That is for $j \geq N + y_2 - 1$

$$C^*_{j,s} = C^*_s = diag[0, \mu, \ldots, N.\mu] * P(\text{departure batch size } = s), \quad 1 \leq s \leq y_2 \qquad (7.9)$$

is $j-$independent. Taking this point into account, it follows that the threshold M is given by $M = N + y_2 - 1$.

7.1.2 Comparative performance study

Through the model of a node performing multipath routing described above, a comparison between the numerical methods for QBD-M processes has been performed. The comparison aspects are mainly focused in the question how fast each method can perform with respect to effects of different system parameters. Five computational ways were implemented as follows:

- *ITE method*: the new iterative method presented in Section 6.1;
- *NA after BL method*: the method of Naomouv et al. [57] applied after block-size-enlargement (re-blocking);
- *SE after BL method*: spectral expansion method [16] applied after block-size-enlargement;
- *SE without BL method*: direct application of the spectral expansion method [16] without re-blocking;
- *GIS method*: the generalised invariant subspace based method presented in Section 6.2.

All the methods have been implemented in C using the Meschach library for matrix operations. All program running and the CPU time measurements have been performed on IBM RISC system/6000 570 station. In all scenarios, the accuracy for stopping the iterative procedures was set to be 10^{-9}. The computation time was measured until the point at which the program is able to compute the level probability vectors.

First we focus attention on the effect of the relation between y_1 and y_2 on the computation time of the above methods. The normalized system load is being kept at a constant level of

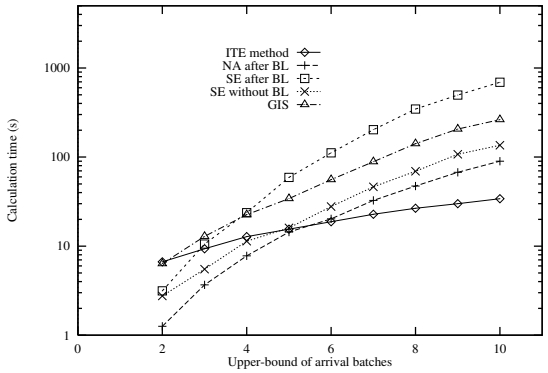


Figure 7.2: Computation time versus the maximum size of arrival batches ($N = 30$, $y_2 = 2$, $load = 0.6$)

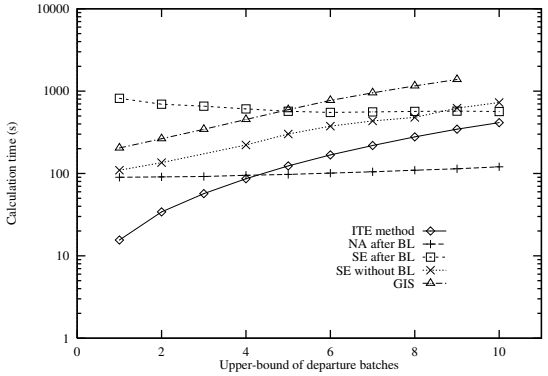


Figure 7.3: Computation time versus the maximum size of departure batches ($N = 30$, $y_1 = 10$, $load = 0.6$)

0.6. Figure 7.2 depicts the relation between computation time and the maximum arrival batch size y_1. The maximum size y_2 of departure batches is kept constant. At first sight, one can see that all the methods take longer time to finish as the maximum size of arrival batches is getting larger. It can be observed that there is a cross-over value of y_1 above which the ITE method outperforms other ones in terms of computation time. Compared to the constant maximum size y_2 of departure batches, the greater the maximum size y_1 of arrival batches is, the more likely the ITE method is the best one. The same trend is observable in Figure 7.3 when y_2 is varying and y_1 is fixed. The ITE method outperforms *all* the other ones if y_2 is small enough compared to y_1.

With a number of other numerical experiences we conclude that the ITE method seems to be the best in case the proportion between the upper bound of arrival batches and departure batches is large enough (practically greater than 2). However, recall that the system load was fixed at 0.6. Therefore, the next question to be answered is how this foregoing tendency is influenced by the system load.

In Figure 7.4, the impact of system load on the computation time has been shown. The maximum batch size was fixed to be $y_1 = 7$, $y_2 = 2$. As one can see, the ITE method requires the least time to accomplish the iterative procedure for a certain range of the system load. The gain in computation time of the ITE method is offered when its advantage in complexity (that is related to *each iteration* and was pointed out in Subsection 6.1.3) prevails. Numerical experiences also show (see Figure 7.4) that the computation time of the GIS, SE after BL and SE without BL methods is independent of system load. At the same time, the computation time of the NA after BL and the ITE method exhibits slightly and strong load-dependent nature, respectively. The time of the ITE method does increase with system load and becomes larger compared to other methods when the saturation is approached ($system\ load \to 1$). The explanation for this is that if iterations of the ITE method are numerous (particularly much more than iterations

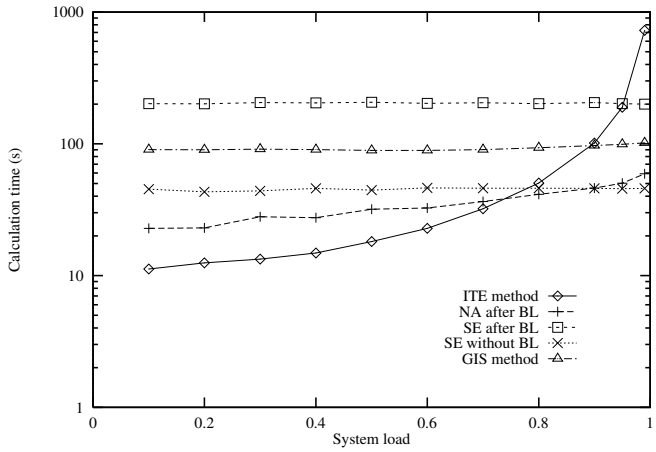


Figure 7.4: Computation time versus system load ($N = 30$, $y_1 = 7$, $y_2 = 2$)

of Naoumov's method), then the advantage in complexity per iteration mentioned above is no longer dominant. The computation time of the ITE method then exceeds that of Naoumov's method and the other methods, as well. An illustration of this is shown in Table 7.1, where the number of necessary iterations of both the two iterative methods (ITE and NA after BL) is reported.

System load	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	0.99
# of iterations (new method)	153	167	182	203	239	306	432	682	1353	2528	9716
# of iterations (Naoumov's method)	5	5	6	6	7	7	8	9	10	11	13

Table 7.1: Number of iterations versus system load

In Figure 7.5, the computation time is considered as a function of system dimension (N). An obvious observation is that the computation time of all the methods of interest increases with the system dimension. In addition, due to the fact that systems (with different dimension) are not heavily loaded and the proportion between y_1 and y_2 is properly large, the ITE method is again proved to be the best in the sense of calculation time.

One can observe from the comparative figures presented so far that the GIS method has larger computation time compared to that of the NA after BL and SE without BL methods in all reported cases, but in general it is faster than the SE after BL and than the ITE method as the saturation point is approached (i.e. the system load tends to 1).

For a deeper insight into the operation of the GIS method, we take a closer look at the execution time of each step of the step-by-step algorithm presented in Section 6.2.2.2. It is

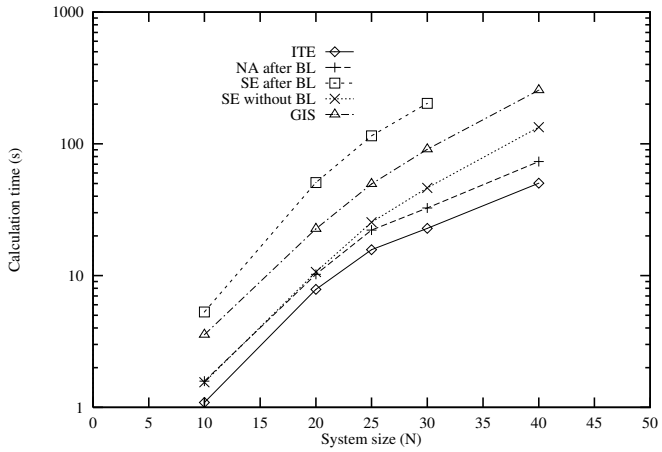


Figure 7.5: Computation time versus system size ($y_1 = 7, y_2 = 2, load = 0.6$)

expected that the major computation time will be devoted to the iterative procedure in step 2 and four QR rank revealing operations in step 3, plus the time of matrix multiplications and inversion in step 4. In fact, numerical results show that the iterative procedure for the matrix sign function converges quite fast after a few iterations. The time required for the iterative computation in step 2 is nearly the same as the time needed for QR operations in step 3. This tendency is valid over all the set of system parameters. This situation is demonstrated in Figure 7.6, where the individual time of each step is depicted for different system loads. The figures also give explanation about the load-insensitive nature of the GIS method. It is observable from the figures that the system load only influences the time of step 2, but that effect is indeed very small. Step 3 and step 4 are not affected by the system load at all. The time of step 4 in all cases only takes about 10 percent of the total amount of the needed time.

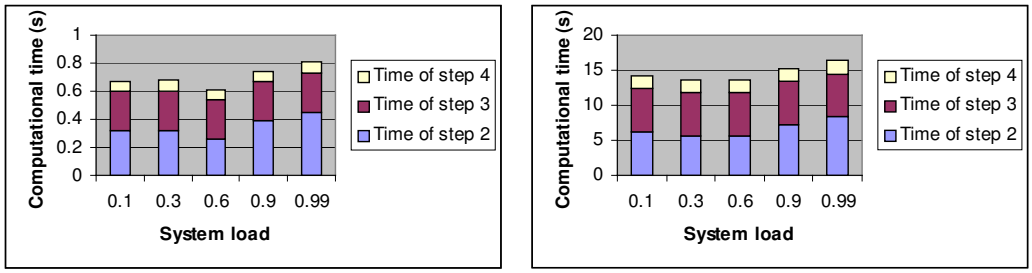


Figure 7.6: Contribution of each step in the GIS method in the total computational time ($y_1 = 3$, $y_2 = 2$, $N = 10, 30$)

It is also noteworthy that while in the case of QBD processes, spectral expansion performs with nearly the same efficiency as Naoumov's method according to the results of Chapter 4,

being applied after re-blocking it becomes much slower in the case of QBD-M processes. The reason for this is lying behind re-blocking. Given the parameters y_1, y_2 and N, re-blocking yields blocks of quite large size $max(y_1, y_2)(N+1)$ (may be in order of several hundreds). The computing process of eigenvalues and eigenvectors then results in a polynomial eigenvalue/eigenvector problem of degree $2max(y_1, y_2)$, which might be very time-consuming. This fact also explains why solving a QBD-M process directly with the spectral expansion method (including a polynomial eigenvalue/eigenvector problem of degree $y_1 + y_2$) is in general faster than using spectral expansion after re-blocking.

7.2 Modelling an ATM concentrator

In this section, we demonstrate the use of a discrete QBD-M process through the analysis of an ATM concentrator. In ATM technology, ATM concentrators [18] are important devices and have received much research attention related to their design as well as their performance evaluation. Queueing models that can be applied to the concentrator problem have been studied in several previous work, such as in [48, 68, 73]. In the context of our present work, the problem is seen from a new point of view. Namely, it will be firmly shown that an appropriate problem formalization, in fact, leads to the application of the QBD-M queueing model. Exploiting this observation, a comparative performance study of the computational methods reviewed and developed in Chapter 5 and Chapter 6 is carried out. In addition, upper bounds for performance parameters of the ATM concentrator itself will be derived to point out the usefulness of the steady state analysis even in case of an infinite buffer.

7.2.1 System description

Consider an ATM concentrator in which several low speed lines are connected to higher speed ones (see Figure 7.7 and Figure 7.8). For example, links at speed 150Mbps are grouped into lines at speed 300Mbps, 620Mbps and so on. Let the ATM concentrator have N input links and output links with speed S times faster than that of input links.

The cell-based feature of ATM links approves the use of a discrete time queueing model, which is described as follows. The input lines are modelled by N independent On-Off sources. Each source alternates between active and idle periods. The distribution of On and Off periods is assumed to be geometric with the mean $1/\alpha$ and $1/\beta$, respectively. An active source continuously generates cells at the beginning of slot times. The arrival process is then driven by the transition probability matrix Q. The transition between phase i (meaning that there are i active sources) and phase j (meaning that there are j active sources) is defined by the entry $q_{i,j}$ of matrix Q

$$q_{i,j} = \sum_{k=max(0,i-j)}^{min(i,N-j)} \begin{pmatrix} i \\ k \end{pmatrix} \alpha^k (1-\alpha)^{i-k} \begin{pmatrix} N-i \\ k+j-i \end{pmatrix} \beta^{k+j-i} (1-\beta)^{N-k-j}$$

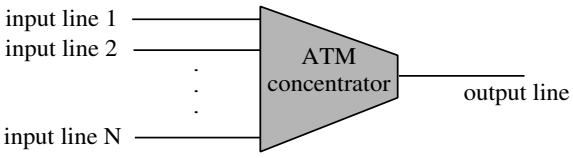


Figure 7.7: An ATM concentrator with N input lines

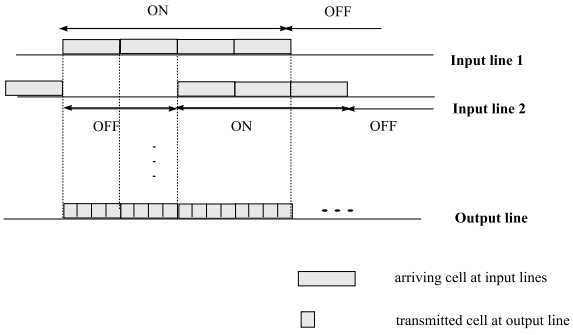


Figure 7.8: An ATM concentrator modell: ON-OFF input lines and an S times faster output line ($S = 4$)

for all $0 \leq i, j \leq N$ pair.

Further on, due to the S-times faster speed of the output line, S servers are considered. The servers are synchronized and the service time is considered to be deterministic and equal to the time unit. Cells are delivered on the output line at the end of each time slot. Cells waiting for transmission are stored in a buffer that has infinite capacity. The service discipline is First-Come-First-Served. The system observed at the beginning of each time slot n (just before the occurrent phase change) can be described by states composed of

- I_n: number of active sources at the beginning of the time slot. Note that I_n takes a value from the set $\{0, 1, \ldots, N\}$. This variable represents the phase dimension of the system.
- J_n: number of cells in the buffer. Since an infinite case is considered, J_n take a value from the infinite set $\{0, 1, \ldots\}$. This variable represents the level dimension of the system.

Let P be a transition probability matrix of the process and define D_k $(0 \leq k \leq N)$ matrices of size $(N+1) \times (N+1)$ with the following meaning. The entry (i, l) of matrix D_k is defined as $D_k(i, l) = Pr(I_{n+1} = l,\ k\ \textit{cells arrive in batch} | I_n = i)$ where $0 \leq i, k, l \leq N$. In other words, this is the probability that upon phase transition (from phase i to phase l) at the beginning of the time slot, the arrival process generates k cells. In our case, it is clear that $D_k(i, l) = q_{i,k}$ if $l = k$, otherwise $D_k(i, l) = 0$. That is for $k = 0, 1, \ldots, N$

$$D_k = \begin{bmatrix} 0 & \ldots & q_{0,k} & \ldots & 0 \\ 0 & \ldots & q_{1,k} & \ldots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \ldots & q_{N,k} & \ldots & 0 \end{bmatrix}. \tag{7.10}$$

The transition probability matrix P of the process then has the form shown in Figure 7.9. This transition probability matrix has the first $(S+1)(N+1)$ identical rows. Regarding to the

$$P = \left[\begin{array}{cccccc|cccc} D_0 & D_1 & \dots & D_S & \dots & D_N & & & & \\ D_0 & D_1 & \dots & D_S & \dots & D_N & & & & \\ \vdots & & \vdots & & & & & & & \\ D_0 & D_1 & \dots & D_S & \dots & D_N & & & & \\ & D_0 & D_1 & \dots & D_S & \dots & D_N & & & \\ & & D_0 & D_1 & \dots & D_S & \dots & D_N & & \\ & & & D_0 & D_1 & \dots & D_S & \dots & D_N & \\ & & & & \ddots & & & \ddots & & \ddots \end{array}\right]$$

Figure 7.9: The block structure of the transition probability matrix of a process

form of the transition probability matrix, it can be seen that its regular part refers to a system having batch arrivals and batch departures. In other words, we obtain a QBD-M process with parameters $y_1 = N-S$ and $y_2 = S$ concerning the upper-bounds of arrival and departure batches, respectively[1]. Note that matrices D_k $(0 \leq k \leq N)$ act the part of the transition probability matrices $A_{\cdot}$, $B_{\cdot,\cdot}$ and $C_{\cdot,\cdot}$ introduced in the standard description of QBD-M processes presented in Section 5.2 of Chapter 5 and it is not difficult to give an exact correspondence between them. The threshold M from which the regular part with repeating columns in the structure of the transition matrix P is observed is given by $M = N + 1$. That means the balance equations written for all level $j \geq M$ have the same level-independent matrix coefficients. Note that the inequality $M \geq y_1$ is obviously fulfilled. The total offered load is defined as $\dfrac{N\beta}{\alpha+\beta}$. The utilization factor (system load) of the considered system is $\dfrac{N\beta}{S(\alpha+\beta)}$ and this can also be referred to as the load per server. Since the system's buffer has infinite size, the system load should be kept under 1 to ensure the stability of the system.

7.2.2 Comparative performance study

In this subsection the goal is again to give a comparison related to the capability of the computational methods of QBD-M processes. The environment and conditions of the implementation in which comparison results have been produced is similar to the case reported in the Section 7.1.2. Basically, we take four methods into investigation:

- *ITE method*: the new iterative method presenred in Section 6.1;
- *NA after BL method*: the method of Naomouv et al. [57] applied after block-size-enlargement (re-blocking);

[1]The assumption that $N > S$ is adopted

- *SE after BL method*: the spectral expansion method [16] applied after block-size-enlargement;
- *GIS method*: the generalised invariant subspace based method presented in Section 6.2.

At this time, the direct SE method is not involved, since we deliberately avoid reporting results of cases in which numerical problems were experienced with this method.

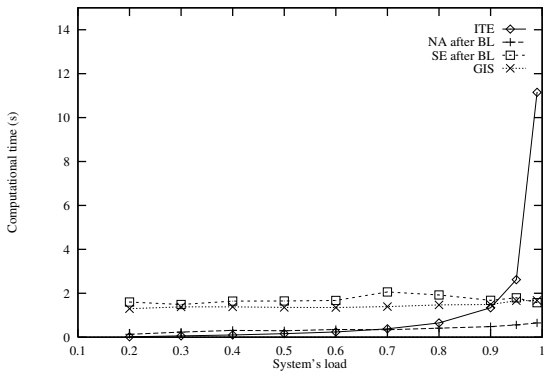


Figure 7.10: Computational time versus system load ($N = 8$)

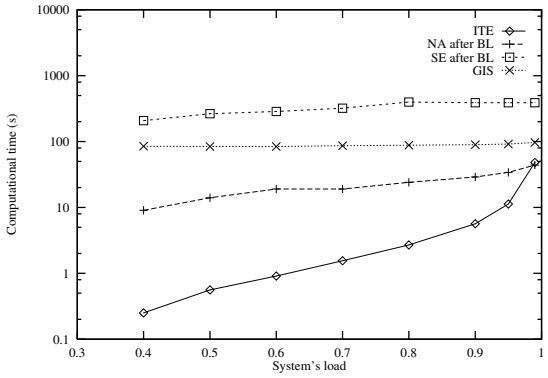


Figure 7.11: Computational time versus system load ($N = 16$)

Being inspired by the conclusions of the previous section, the performance comparison between the computational methods is done by examining effects of the system load (system utilization) and effects of the relation between the upper bounds of arrival and departure batches. In Figure 7.10 and Figure 7.11 the system utilization is varying, meanwhile other parameters are fixed ($S = 2$, $\beta = 0.05$, $N = 8, 16$). Focusing on the GIS and ITE methods, one can observe that the computational time of the GIS method is not influenced by the system load and remains almost constant. From this point of view, the GIS method can be ranked among the best methods ranging with the spectral expansion method. However, the GIS method is better to use compared to the SE due to two aspects. First, the GIS method is faster than the SE after BL method, as shown in the two above figures. Secondly, the GIS method avoids the procedure of computing all relevant eigenvalues and associated eigenvectors, which is compulsory to be done in the case of the SE after BL method. Therefore, numerical problems related to ill-conditions such as too close eigenvalues and the computational difficulty deriving from the presence of complex eigenvalues and eigenvectors are no longer imposed.

In contrast to the GIS method, the computational time of the ITE method is increasing with the system load and may become quite large at the close neighbourhood of the saturation point. It is important to note that below a certain value of the system load (this value is about 0.7 if $N = 8$ and 0.99 if $N = 16$), the ITE is the fastest method. The time benefit of the ITE method especially prevails in the case of large system size (i.e the parameter N is large). Meanwhile in Figure 7.10, the gain of execution time of the ITE method compared to the NA after BL method

is quite slight, this difference is considerably observable in Figure 7.11 and practically for nearly *all* the ranges of the system load, the ITE method performs fastest.

The GIS method seems to be faster than the SE after BL but slower than the NA after BL. It only outperforms the ITE method regarding to computational time if the aforementioned threshold of the system load is overstepped.

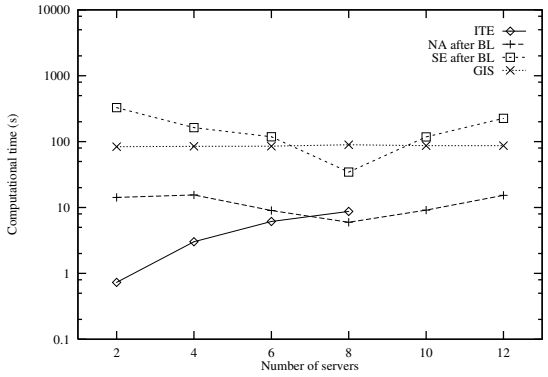


Figure 7.12: Effect of relation between arrival and departure bounds on computational time ($N = 8$, $load = 0.75$)

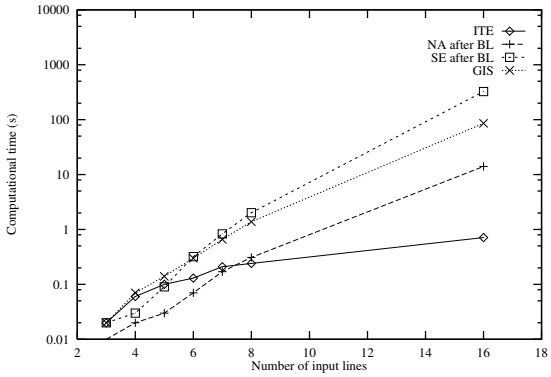


Figure 7.13: Effect of relation between arrival and departure bounds on computational time ($S = 2$, $load = 0.75$)

The next figures illustrate the effect of the relation between the arrival and departure bounds on the computational time. Recall that in this ATM concentrator example the relation between the parameters is $y_1 = N - S$ and $y_2 = S$. Figure 7.12 demonstrates the situation in which the number of servers (S) is increased and the number of input lines is kept unchanged $N = 16$. Note that increasing S is roughly equivalent with decreasing the proportion y_1/y_2. One finds from Figure 7.12 that while the proportion y_1/y_2 is large enough (greater than 1.66 corresponding to $S = 6$) the ITE method is the best in terms of execution time. The GIS method, in most cases, performs faster than the SE after BL, but slower than the NA after BL method. Since the ITE method only works for $y_1/y_2 \geq 1$ cases, for $S \geq 8$ it can not be applied. This is a drawback of the ITE method which the GIS method does not suffer.

When the proportion y_1/y_2 increases due to the increase of input lines N (with fixed $S = 2$), the computational time of the methods varies in the way shown in Figure 7.13. Note that enlarging N increases the system size and the upper bound of arrival batches at once. When $N \geq 8$ (i.e. $y_1/y_2 \geq 3$), the ITE method is definitely the best due to its fastest execution time. The GIS method shows the regular tendency, i.e. it is faster than the SE after BL but slower than the NA after BL method.

With a number of other numerical tests, we conclude that the ITE method proves the fastest method if the system load is not too close to saturation point and the proportion between the upper bounds of arrival and departure batches is large enough, practically greater than 2.

Numerical accuracy of the computational methods is checked by examining the residual error in the following way. We calculate $||\underline{v} - \underline{v}P||_\infty$, the infinity norm of the residual vector. Here, $\underline{v} = (\underline{v}_1, \underline{v}_2, \ldots \underline{v}_M, \ldots)$ is the steady state solution vector, P denotes the transition probability matrix of the process and $\underline{v}P = (\underline{x}_1, \ldots, \underline{x}_M, \ldots)$ is computed by equation (5.2). Since both $\underline{v}$ and $\underline{v}P$ row vectors have infinite size, we only take their first M subvectors into account and compute $max_{\{j,0\leq j\leq M\}}||\underline{v}_j - \underline{x}_j||_\infty$.

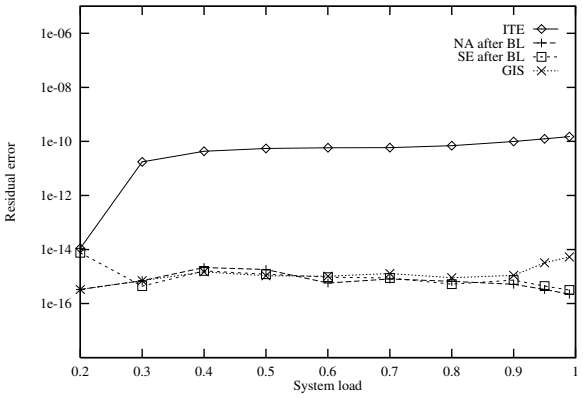


Figure 7.14: Residual error versus system load

Figure 7.14 shows the residual error produced for different values of the system load. Other system parameters are $N = 8$, $S = 2$, $\beta = 0.05$. It turns out that the system load has negligible effects on the computation error of the GIS, NA after BL and SE after BL methods. Moreover, computation errors of the three aforementioned methods fall into the same range $(10^{-16} - 10^{-15})$ and they are about 5-orders of magnitude better than the error of the ITE method. Even though, note that the residual error produced by the ITE method is still acceptably small (in range of 10^{-11}) from the practical point of view.

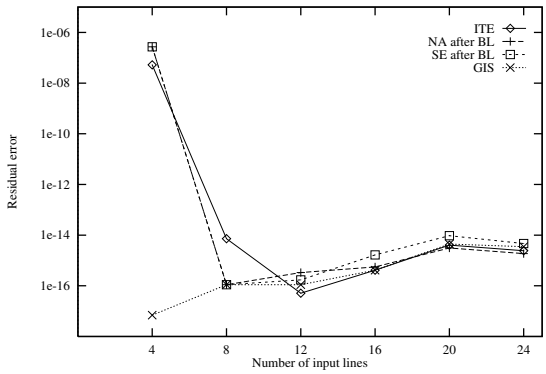


Figure 7.15: Residual error versus system size ($load = 0.2$)

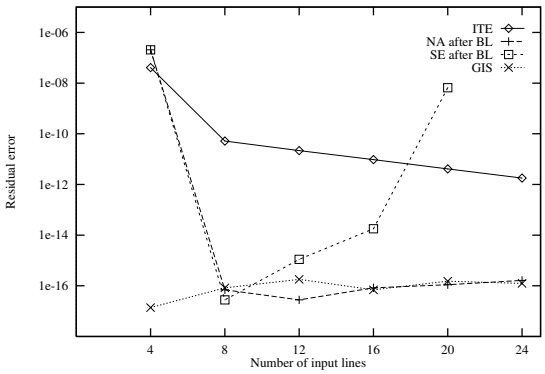


Figure 7.16: Residual error versus system size ($load = 0.8$)

In Figure 7.15 and Figure 7.16, the impact of the increasing system size N on the numerical accuracy is illustrated. The system load is kept unchanged at 0.2 and 0.8 for different values of N, other parameters are $S = 2$, $\beta = 0.01$. The curves show that the residual error of the GIS method reflects the smallest fluctuation with respect to the system size and it always remains in the range of $10^{-16}-10^{-15}$. This is valid for both two levels of the system load. Significant changes are observable mainly in case of the SE after BL method (see Figure 7.16), when its residual error exhibits an increasing tendency with system size at the load level 0.8. Another observation is that at the system load 0.2 the goodness of all the methods regarding to numerical accuracy is commensurate, since differences between their residual errors are negligible over different system sizes. However, when the system load is 0.8, this statement is valid only for the GIS and NA after BL methods. The ITE method, with its error in range of 10^{-11}, proves about 5 orders of magnitude less accurate in the final solution, which confirms the conclusion drawn from Figure 7.14.

Nevertheless, we note that the results of performance measures, such as the mean number of cells in the system, computed by the different methods do not show any significant bias and always have up to five coincident digits after the floating point compared to the numerical results of each other.

7.2.3 Estimation of performance parameters of the ATM concentrator

Although the buffer capacity in reality is always finite, the distribution of the infinite buffer case still possesses useful information. Namely, using the steady state distribution obtained with an infinite buffer, one can give an estimation on performance parameters for the finite buffer case. In the remainders of this section we shall give upper-bounds for the two most important measures, namely, the cell delay and cell loss probability. By some probabilistic derivations, simple formulas are obtainable for the worst case cell loss probability and cell delay.

7.2.3.1 Elementary probabilities

First of all, let us introduce the probability vector

$$\underline{v} = \sum_{j=0}^{\infty} \underline{v}_j = (p_0, p_1, \ldots, p_N)$$

and note that $\sum_{k=0}^{N} D_k \underline{e} = \underline{e}$ and therefore $\sum_{k=1}^{N} D_k \underline{e} = (I - D_0)\underline{e}$.

Some principle probabilities are calculated as follows:

$$\begin{aligned} & P(\text{a batch of size } k \text{ arrives} | I_n = i) \\ = & \sum_{l=0}^{N} P(I_{n+1} = l, \text{a batch of size } k \text{ arrives} | I_n = i) \end{aligned}$$

$$= \sum_{l=0}^{N} D_k(i,l) \tag{7.11}$$

P(batch of size k arrives)

$$\begin{aligned} &= \sum_{i=1}^{N} P(\text{batch of size } k \text{ arrives}|I_n = i)P(I_n = i) \\ &= \sum_{i=0}^{N} p_i \left(\sum_{l=0}^{N} D_k(i,l) \right) \\ &= \underline{v} D_k \underline{e} \end{aligned} \tag{7.12}$$

P(a batch of size k arrives, $J_n = j$)

$$\begin{aligned} &= \sum_{i=0}^{N} P(\text{a batch of size } k \text{ arrives }, J_n = j, I_n = i) \\ &= \sum_{i=0}^{N} P(\text{a batch of size } k \text{ arrives } |I_n = i, J_n = j)P(I_n = i, J_n = j) \\ &= \sum_{i=0}^{N} P(\text{a batch of size } k \text{ arrives } |I_n = i)p_{i,j} \\ &= \sum_{i=0}^{N} p_{i,j} \left(\sum_{l=0}^{N} D_k(i,l) \right) \\ &= \underline{v}_j D_k \underline{e} \end{aligned} \tag{7.13}$$

P(a cell arrives)

$$\begin{aligned} &= \sum_{k=1}^{N} P(\text{a batch of size } k \text{ arrives}) \\ &= \sum_{k=1}^{N} \sum_{i=0}^{N} p_i \left(\sum_{l=0}^{N} D_k(i,l) \right) \\ &= \sum_{k=1}^{N} \underline{v} D_k \underline{e} \\ &= \underline{v}(I - D_0)\underline{e} \end{aligned} \tag{7.14}$$

$P(J_n = j,$ size of arriving batch $= k|$a cell arrives)

$$\begin{aligned} &= \frac{P(J_n = j, \text{ size of arriving batch} = k, \text{ a cell arrives})}{P(\text{a cell arrives})} \\ &= \frac{P(J_n = j, \text{ a batch of size } k \text{ arrives})}{P(\text{a cell arrives})} \end{aligned}$$

$$= \frac{\underline{v}_j D_k \underline{e}}{\underline{v}(I - D_0)\underline{e}} \tag{7.15}$$

7.2.3.2 Upperbound for the cell delay

Let us assume that a cell arrives at the beginning of a time slot. The cell is in a batch of size k ($1 \leq k \leq N$), which is accepted into the system and sees j cells being in the system. Note that if $j \leq S$ then all the cells in the system are being in service. For the tagged cell, the worst case is when it will be served as the last cell of the batch. Since the service discipline is FIFO, the worst case delay that the cell may encounter (measured in slots) is given by $\left[\frac{k}{S+1}\right]$ if $j \leq S$ and $\left[\frac{k+j-S}{S+1}\right]$ for the case $j > S$ ($[x]$ denotes the integer part of x). Mathematically, a nicer formula for the worst case delay is written as $\left[\frac{k+max(0,j-S)}{S+1}\right]$. Consequently, the mean cell delay in worst case is

$$\sum_{j=0}^{\infty}\sum_{k=1}^{N}\left[\frac{k+max(0,j-S)}{S+1}\right] P(J_n = j,\ \text{size of arriving batch} = k|\text{a cell arrives})$$

This result is stated in the following proposition

Proposition 7.2.1 *The mean cell delay in worst case is calculable as*

$$E_{delay} = \sum_{j=0}^{\infty}\sum_{k=1}^{N}\left[\frac{k+max(0,j-S)}{S+1}\right]\frac{\underline{v}_j D_k \underline{e}}{\underline{v}(I - D_0)\underline{e}}$$

and this is obviously an upperbound of the cell delay encountered in the finite case.

7.2.3.3 Upperbound for the cell loss ratio

Now assuming the finite buffer of size B, we can give an upper bound of the cell loss ratio based on the steady state probabilities computed in the corresponding infinite system. The estimation we use here is

$$P_{loss} \leq \frac{P_\infty(\text{a cell arrives and is dropped})}{P_\infty(\text{a cell arrives})},$$

where the probability P_∞(a cell arrives and is dropped) is calculated by using the distribution of the infinite system and under the cell dropping criteria inheriting from the finite case.

In the finite system, an arriving cell in a batch of size k is lost if upon arrival it sees $J_n = j$ ($j \leq B + S$) cells in the system and the whole batch can not be put into the buffer (i.e. $j + k > B + S$). Assuming equality of all active input lines, the cell will be discarded with probability $\frac{(k+j)-(B+S)}{k}$ under the PBA (Partial Batch Accepting) scheme and with probability 1 under the WBA (Whole Batch Accepting) scheme.

Let us define the quantity

$$t_{j,k,B} = \begin{cases} \frac{(k+j)-(B+S)}{k} & \textit{if } B+S-k < j \leq B+S, \textit{ PBA scheme} \\ 1 & \textit{if } B+S-k < j \leq B+S, \textit{ WBA scheme} \\ 0 & \textit{otherwise} \end{cases}$$

Thus, the criteria of dropping a cell in the arriving batch of size k is defined by two factors: the event $j > B+S-k$ and the probability $t_{j,k,B}$. Denote $T_{k,B} = max(0, B+S-k+1)$ and the event that an arriving batch of size k sees j cells in the system by $\omega_{j,k}$. Note that the result of Subsection 7.2.3.1 indicates that the probability $P(\omega_{j,k})$ equals to $\underline{v}_j D_k \underline{e}$. Using the distribution of the infinite case, the probability P_∞(a cell arrives and is dropped) is calculable as follows:

$$\begin{aligned} & P_\infty(\text{a cell arrives and is dropped}) \\ &= \sum_{k=1}^{N} P_\infty(\text{a cell in a batch of size } k \text{ arrives and is dropped}) \\ &= \sum_{k=1}^{N} \left(\sum_{j=T_{k,B}}^{\infty} P_\infty(\text{a cell in a batch of size } k \text{ arrives, sees } j \text{ cells and is dropped}) \right) \\ &= \sum_{k=1}^{N} \left(\sum_{j=T_{k,B}}^{\infty} P_\infty\left(\text{a cell in batch is dropped} \mid \omega_{j,k}\right) P_\infty(\omega_{j,k}) \right) \\ &= \sum_{k=1}^{N} \left(\sum_{j=T_{k,B}}^{\infty} t_{j,k,B} \underline{v}_j D_k \underline{e} \right) \end{aligned} \tag{7.16}$$

Proposition 7.2.2 *For a concentrator having a buffer of size B, the following inequality holds*

$$P_{loss} \leq \frac{\sum_{k=1}^{N} \sum_{j=T_{k,B}}^{\infty} t_{j,k,B} \underline{v}_j D_k \underline{e}}{\underline{v}(I - D_0)\underline{e}}.$$

Some exemplary numerical results regarding to the cell loss ratio are depicted in Figure 7.17 and Figure 7.18. Figure 7.17 shows the upper bound of the cell loss probability for different buffer size, when the system load decreases by getting more servers (i.e. using higher speed output line). As expected, the more servers are present the lower the cell loss probability is due to the decrease of the load per server. Notice that to reach the negligible cell loss probability, more number of servers implies quite small buffer capacity.

Figure 7.18 shows that increasing the number of input lines results in the increase of the cell loss probability. This is because the load per server becomes larger and, additionally, arrival batches with larger size are also in present. Note that in the experiments of Figure 7.17 and Figure 7.18, the parameter pair α, β of each source has been chosen in a way that it always assures the stability of the system.

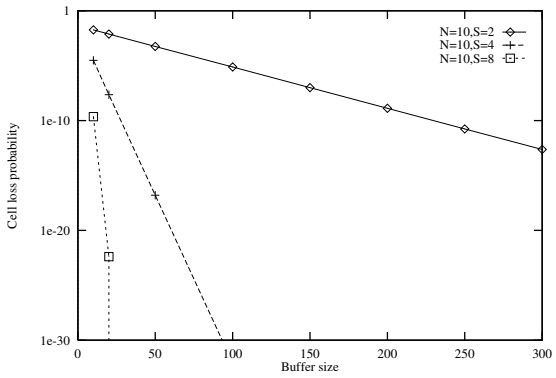


Figure 7.17: Cell loss probability for different number of servers ($\alpha = 0.45$, $\beta = 0.1$)

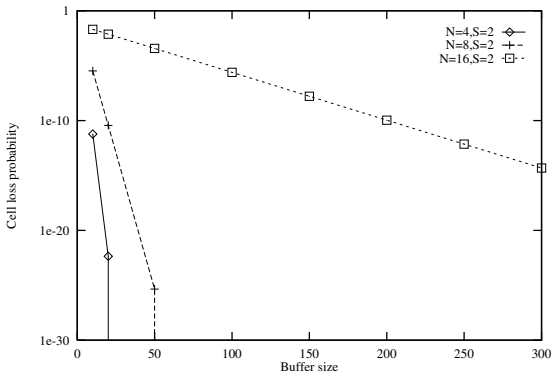


Figure 7.18: Cell loss probability for different number of input lines ($\alpha = 0.75$, $\beta = 0.1$)

7.3 Conclusions and contributions

The contribution of this chapter is twofold. On the one hand, an in-depth performance comparison of computational methods for QBD-M processes has been provided. By examining different case studies and taking both the well-known and recently developed methods into consideration, we have been able to highlight the strong and vulnerable behaviors of each of them, especially in case of the newly developed methods.

The new iterative method seems more reasonable to use when the system load is not too heavy and the proportion between upper bounds of arrival and departure batches is large enough. Note that in the later condition, the term "enough" is considered fulfilled if the proportion is not less than 2. This is typically the case of systems in practice. Thus, this fast and storage-thrift method is one of potential and appealing candidate tools for solving QBD-M processes.

The method based on the theory of generalised invariant subspaces, though is not considered as the fastest one, also has promising application merits. It works with the smallest residual error and exhibits robustness to system load and seems well stable, hence applicable to any set of system parameters. In addition, as we can see later in Chapter 8, this method retains its applicability for finite QBD-M processes as well.

On the other hand, this chapter has also pictured the versatile application range of the QBD-M queueing model in performance evaluation and system analysis. Specifically, we have built up a new analytical model for a node performing multipath routing in MPLS networks with the use of a QBD-M process. Moreover, we have shown that a properly constructed model for an ATM concentrator also leads to a QBD-M one.

Parts of this chapter have been published in our papers [39, 40, 43, 45].

Chapter 8

Analysis of a queueing model with priority classes

In the previous chapter, two applications of the QBD-M process were demonstrated. However, the emphasis was mainly put on the comparative performance assessment of the available numerical methods. In this chapter, we present another applicability of the QBD-M model. This time, the aim is not to compare the performability of the numerical methods. Therefore, after choosing an appropriate numerical method to solve the model, we will focus our attention on the analysis of the system.

8.1 Introduction

The pure best-effort nature of services provided over the worldwide Internet today seems to be insufficient in many cases, as the demand for getting guaranteed quality increasingly emerges. To meet the challenge of guaranteeing quality of services (QoS), the research community is moving towards the development of viable QoS architectures, for example IntServ [14], DPS [70] and DiffServ [26].

Up to now, it turns out that DiffServ is the most prominent QoS architecture due to its salient feature of scalability [77]. The basic idea of the DiffServ concept is to allow core routers treating only with a few number of aggregate traffic classes, rather than with each individual micro traffic flow. The QoS then is delivered to each predefined class by handling them in an adequate way. One of important mechanisms to handle the different traffic classes with regards to their QoS is to deploy proper scheduling schemes in core routers, among which priority queueing may play a key role.

These attendances have motivated the research work presented in this chapter. We construct and analyse a flow-level priority queueing model that has a close abstraction to the beforementioned scheduling scheme. It should be mentioned that queueing models with priority classes

has been an important research topic over the last decade and treated in earlier works for the performance evaluation of telecommunications systems (see e.g. [28, 31, 64, 65]). In our work, we supplement the priority queueing system with several more realistic conditions, hence significantly extend its applicability merit. Firstly, the realistic assumption of *finite buffers* for *all* traffic classes is added. Secondly, the bursty nature of traffic is captured by batch arrival processes. Thirdly, in order to avoid congestion, a two-level feedback control mechanism is introduced for low priority traffic class. Once the number of low priority flows in the system reaches the first control threshold, the arrival rate of low priority flows is forced to reduce to a guaranteed arrival rate. If the second control threshold is passed, each arriving batch of low priority flows will be discarded with a certain dropping probability.

With two separate queues and the introduced dropping scheme, both of the two basic router mechanisms known as priority scheduling and threshold dropping [67] in the DiffServ architecture are implicitly imposed. By means of this combination of router mechanisms, up to three classes of traffic can be handled at DiffServ-capable routers. The highest priority traffic are put into the high priority buffer, meanwhile traffic belonging to the other two classes are stored in the low priority buffer. Once a dropping level in the low priority buffer has reached, incoming traffic of the lowest priority class is discarded. Thus, the dropping probability in this case is interpreted as the fraction of the lowest priority traffic in the traffic coming into the low priority buffer. In the context DiffServ architecture, the highest priority class corresponds to Premium Service, the lowest priority class is represented by Best-Effort Service and the middle priority class is Assured Service.

It will be shown that the queueing system of interest falls into the category of finite QBD-M processes with multiple boundaries. The steady state analysis of the system is performed with a novel approach based on the theory of generalized invariant subspaces. Unlike other possible solutions such as spectral expansion or the matrix geometric with re-blocking, this method avoids the calculation of eigenvalues and eigenvectors and provides a direct closed expression for steady state probabilities. Based on the steady state distribution, important performance parameters are derived. Specifically, closed forms are given for the mean number of customers of each class, the batch and customer loss probability of each class, the mean system time of customers and the system time distributions.

The organisation of this chapter is as follows. In Section 8.2 a queueing model with two priority classes is discussed in details. Section 8.3 presents a model of a general finite QBD-M process and proposes a computational procedure for its steady state analysis. In Section 8.4 we show that the queueing model with two priority classes can be considered as a finite QBD-M process. In Section 8.5 important performance measures of the queueing model are derived from steady state probabilities computed in the previous section. Finally, numerical results illustrating the operation of the system are presented in Section 8.6.

8.2 A queueing model with two priority classes

The system consists of a server and two finite queues. The service capacity of the server is assumed to be composed of N service units. Each customer[1] of the high priority class requires a single service unit.We assume that the arrival processes of both high priority and low priority customers are Poisson processes with arrival rate λ_H and $\lambda_{L,j}$ (see later for the subindex's interpretation), respectively. Moreover, arrival batches are possible at each arrival instant. The batch size is upper-bounded and the distribution of the batch size is arbitrary. For the high priority class, the upper bound of arrival batches is H and a batch of size i occurs with probability h_i ($\sum_{i=1}^{H} h_i = 1$). The corresponding parameters of the low priority class are L and l_i ($\sum_{i=1}^{L} l_i = 1$). The service discipline for high priority and low priority customers is FIFO (First In First Out) and PS (Processor Sharing), respectively. Preemption is granted to the high priority traffic class. This allows high priority customers to push out low priority customers being in service upon their arrival. Consequently, whenever a batch of high priority customers arrives, a part of them will get service immediately if there are less than N service units occupied by other high priority customers. The other customers in the batch are put into a buffer of size K_H. If all the N service units are busy and serving high priority customers, then the batch is put into the buffer if enough room is offered, otherwise the whole batch (or a part of it) is lost. Since the service discipline of low priority customers is processor sharing, a low priority batch will get service immediately if there is service capacity not used by high priority customers. Otherwise the batch is put into a waiting room of size K_L. Recall again that low priority customers under service may be pushed back into the waiting room by high priority customers. It is highly recommended that once a customer has been accepted into the system, it should not be lost because of the aforementioned push-back mechanism. Therefore, at any time we require that no more than K_L of low priority customers (either being in service or waiting in the queue) are in the system.

The dropping rule for arriving batches of both classes may be WBA (Whole Batch Acceptance) or PBA (Partial Batch Acceptance). WBA implies that the whole batch will be dropped if the system is not able to accept the whole batch, while PBA means that only a part of the batch is dropped for which available room can not be assured.

Service time of high customers is assumed to be exponentially distributed with mean $1/\mu_H$. Service requirement of low priority customers is also exponentially distributed with mean $1/\mu_L$, which is independent of everything else. Based on the service discipline described before, it is easy to see that if there are i high priority and j low priority customers in the system ($0 \leq i \leq N + K_H$, $1 \leq j \leq K_L$), then each of the low priority customers receives service at rate

[1] In order to stay closely with the queueing context, we use the term "customer" interchangeably with the term "flow".

$\frac{max(N-i,0)}{j}\mu_L$.

This queueing system is a two dimensional Markov process in continuous time domain. At any time t, the state of system is described by a couple of variables $(I(t), J(t)) = (i, j)$ where

- $I(t)$ denotes the number of high priority customers in the system. This variable represents the phase dimension of the system. Note that $0 \leq I(t) \leq N + K_H$.
- $J(t)$ denotes the number of low priority customers in the system. This variable represents the level dimension of the system. Note that $0 \leq J(t) \leq K_L$.

In order to avoid congestion and speed up service of the low priority class, a two-level feedback control mechanism is introduced. Firstly, we exert control on the arrival rate of low priority customers as follows. Assuming that in general we have level-dependent arrival rates, i.e. at level j $(0 \leq j < m_1)$ (batch) arrivals occur at rate $\lambda_{L,j}$. When the number of low priority customers in the system reaches the level m_1, its (batch) arrival rate is forced to reduce from $\lambda_{L,j}$ to the guaranteed value λ_L[2]. Secondly, we use an early dropping mechanism implying the second control level $j = K_L - m_2$. That means whenever the difference between the number of low priority customers and the waiting room's capacity drops below m_2, each low priority arrival batch will be discarded with the pre-defined probability p_{dropp}.

In the rest of the analysis we adopt the Whole Batch Acceptance rule for both classes of customers. Nevertheless, the analysis presented in this chapter is also valid for the Partial Batch Acceptance principle. The change of the acceptance rule only requires some slight modifications in the construction of the generator matrix of the process and in the computation of performance parameters. For the sake of easy reading, parameters of the queueing model are summarised in Table 8.2.

λ_H	arrival rate of high priority batches
λ_L	guaranteed arrival rate of low priority batches
H	upper bound of high priority batches
L	upper bound of low priority batches
K_H	buffer size for high priority class
K_L	buffer size for low priority class
m_1	the first control level for low priority class
m_2	$K_L - m_2$ gives the second control level
p_{dropp}	dropping probability used for overload control

Table 8.1: Parameters of the considered queueing model

Note that in the queueing system described above, analysis related to the high priority class can be done using $M^{[X]}/M/m/N$ model, independently of the presence of the low priority class

[2]From the aspect of computation, it does not matter what rate batches arrive, but from the context of congestion control it is natural to have $\lambda_{L,0} \geq \lambda_{L,1} \geq \ldots \geq \lambda_L$.

due to the preemption. The two dimensional queueing model has been considered in order to capture the influence of the high priority class on the service capacity reserved for the low priority customers.

Instead of moving ahead directly to the steady state analysis of the queueing system, we take a closer look at the steady state analysis of a finite QBD-M process in the next section. The reason for this is that the queueing model under the circumstances we consider in fact belongs to a class of finite QBD-M processes with multiple boundary, as verified later. Therefore, having an analysis tool for a finite QBD-M process with multiple boundary naturally makes the analysis of the queueing system in question simple and straightforward.

8.3 A finite QBD-M process with multiple boundaries and its steady state analysis

A finite QBD-M process is a two dimensional Markov process with finite state space. In the continuous time domain, at each time t the process is defined by a state $(I(t), J(t)) = (i, j)$ $(0 \leq i \leq \mathcal{N},\ 0 \leq j \leq \mathcal{K})$. The integer variables i and j are referred to as the phase and the level of the process, respectively.

Let us define the following transition matrices of size $(\mathcal{N}+1) \times (\mathcal{N}+1)$

- A_j: purely phase transitions – From state (i, j) to state (k, j) $(0 \leq i, k \leq \mathcal{N}; j = 0, 1, \ldots \mathcal{K})$.
- $B_{j,s}$: bounded $s-$step upward transitions – From state (i, j) to state $(k, j+s)$ $(0 \leq i, k \leq \mathcal{N}; 1 \leq s \leq y_1; y_1 \geq 1; j = 0, 1, \ldots \mathcal{K})$.
- $C_{j,s}$: bounded $s-$step downward transitions – From state (i, j) to state $(k, j-s)$ $(0 \leq i, k \leq \mathcal{N}; s \leq j; 1 \leq s \leq y_2; y_2 \geq 1; j = 0, 1, \ldots \mathcal{K})$. If $j < s$ then $C_{j,s} = 0$.
- $\mathcal{A}_j$: For $0 \leq j \leq \mathcal{K}$ this matrix is defined as follows

$$\mathcal{A}_j(i,k) = \begin{cases} A_j(i,k) & \text{for } 0 \leq i,k \leq \mathcal{N} \text{ and } i \neq k \\ A_j(i,i) - \sum_{ii=0}^{\mathcal{N}} A_j(i,ii) - \sum_{s=1}^{y_1} \sum_{ii=0}^{\mathcal{N}} B_{j,s}(i,ii) - \sum_{s=1}^{y_2} \sum_{ii=0}^{\mathcal{N}} C_{j,s}(i,ii) & \text{for } i = k \end{cases} \tag{8.1}$$

We are interested in determining the steady state probabilities $p_{i,j} = \lim_{t\to\infty} P(I(t) = i, J(t) = j)$ or in other words the steady state vectors

$$\underline{v}_j = (p_{0,j}, p_{1,j}, \ldots, p_{\mathcal{N},j}) \text{ for } j = 0, 1, \ldots \mathcal{K}. \tag{8.2}$$

As usual, the incipient point is the basic balance and normalizing equations of the system that can be written as

$$\sum_{s=1}^{y_1} \underline{v}_{j-s} B_{j-s,s} + \underline{v}_j \mathcal{A}_j + \sum_{s=1}^{y_2} \underline{v}_{j+s} C_{j+s,s} = 0, \tag{8.3}$$

(it is assumed $\underline{v}_{j-s} = \underline{0}$ if $j < s$ and $\underline{v}_{j+s} = \underline{0}$ if $j + s > \mathcal{K}$), and

$$\sum_{j=0}^{\mathcal{K}} \underline{v}_j \underline{e} = 1. \tag{8.4}$$

Let us assume that there are values M_1 and M_2 so that for $M_1 \leq j \leq M_2$ the equations (8.3) include only $j-$independent coefficient matrices and have a common form

$$\sum_{s=1}^{y_1} \underline{v}_{j-s} B_s + \underline{v}_j \mathcal{A} + \sum_{s=1}^{y_2} \underline{v}_{j+s} C_s = 0. \tag{8.5}$$

The corresponding part in the generator matrix has a nice structure of repeating columns. As a matter of fact, this is a version of the QBD-M queueing process considered in Chapter 5. The slight difference is the finite state space and the existence of multiple boundaries from two directions (i.e. multiple boundary levels occur both at the beginning and the end of the level range).

So far, our attention has been mainly focused on the steady state analysis of infinite QBD-M processes. For the finite case, the solution becomes more complicated. Although all of the aforementioned approaches for infinite QBD-M processes may still be applicable, they must be used twice this time (once for the corresponding original infinite process and once again for its reverse process). For example, in case of the spectral expansion method all the eigenvalues inside the unit disk and the corresponding eigenvectors of both (the original and the reverse) processes have to be computed. This may increase the possibility of numerical problems, which was encountered even in the infinite case. On the other hand, the reblocking approach does not provide a direct expression for the original level probability vectors, but just only for the probability vectors of augmented size.

The approach we take now is different from those mentioned above. Namely, we make use of the theory of generalised invariant subspaces that was introduced in [10] and has been shown in Section 6.2 of Chapter 6 to be a good solution method for the steady state analysis of infinite QBD-M processes. Recall that with the theory of generalised invariant subspaces we are able to compute matrices $U = [\ U_1 \quad U_2\ ]$ and $V = [\ V_1 \quad V_2\ ]$, which makes the following decomposition possible

$$U^{-1} E V = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix} \text{ and } U^{-1} G V = \begin{bmatrix} G_{11} & 0 \\ 0 & G_{22} \end{bmatrix}, \tag{8.6}$$

where matrices E and G are constructed from the QBD-M process as follows

$$G = \begin{bmatrix} 0 & 0 & \dots & \dots & -D_0 \\ I & 0 & \dots & \dots & -D_1 \\ 0 & I & \dots & \dots & -D_2 \\ \vdots & \vdots & \ddots & & \\ 0 & 0 & \dots & I & -D_{y-1} \end{bmatrix}, \quad E = \begin{bmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & D_y \end{bmatrix}. \tag{8.7}$$

Here $y = y_1 + y_2$ and matrix D_i $(i = 0, 1, \dots y)$ is the coefficient matrix of $\underline{v}_{j-y_1+i}$ in the balance equation (8.5).

Let m_s and m_u denote the number of singularities of matrix pencil $\lambda E - G$, which lie inside and outside the open unit disk, respectively. Note that the singularities of the matrix pencil $\lambda E - G$, i.e. the roots of $det(\lambda E - G) = 0$ are exactly the roots of the equation $det(D(\lambda)) = 0$, where $D(\lambda) = D_0 + D_1\lambda + \ldots + D_y\lambda^y$. As a consequence, we have $m_s + m_u = m = y(\mathcal{N} + 1)$. Matrices U_1, V_1 are of size $m \times m_u$, matrices U_2, V_2 are of size $m \times m_s$, matrices E_{11}, G_{11} are of size $m_u \times m_u$ and matrices E_{22}, G_{22} are of size $m_s \times m_s$, respectively.

Let us define the following partition and matrices

$$U^{-1} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}, \quad F_1 = E_{11}G_{11}^{-1}, \quad F_2 = G_{22}E_{22}^{-1}, \quad T_n = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} 0 \\ \vdots \\ (n-1)\text{-st} \\ n\text{-th} \\ (n+1)\text{-st} \\ \vdots \\ (y-1)\text{-st} \end{matrix} \quad \text{for } 0 \le n < y. \tag{8.8}$$

Matrices F_1 and F_2 are of size $m_u \times m_u$ and $m_s \times m_s$, respectively. Let us define the grouping

$$\underline{w}_k = [\ \underline{v}_{M_1-y_1+k} \quad \cdots \quad \underline{v}_{M_1-y_1+k+(y-1)}\ ] \tag{8.9}$$

for $0 \le k \le K = M_2 - M_1 + 1$ and

$$[\ \underline{p}_k \quad \underline{q}_k\ ] = \underline{w}_k[\ U_1 \quad U_2\ ]. \tag{8.10}$$

The vectors $\underline{p}_k$ and $\underline{q}_k$ $(0 \le k \le K)$ are of size m_u and m_s, respectively. It can be easily checked that

$$\underline{w}_{k+1}E = \underline{w}_k G, \quad 0 \le k \le K. \tag{8.11}$$

By post-multiplying the equation (8.11) by matrix V and combining with equations (8.6), we obtain two un-couple generalized difference equations for $\underline{p}_k$ and $\underline{q}_k$

$$\underline{p}_{k+1}E_{11} = \underline{p}_k G_{11}, \quad 0 \le k < K, \tag{8.12}$$

$$\underline{q}_{k+1}E_{22} = \underline{q}_k G_{22}, \quad 0 \le k < K. \tag{8.13}$$

This follows that the next two expression hold

$$\underline{p}_k = \underline{p}_K F_1^{K-k} \text{ and } \underline{q}_k = \underline{q}_0 F_2^k. \tag{8.14}$$

We are now able to express the level probability vectors as

$$\underline{v}_{M_1-y_1+k} = \underline{w}_k T_0 = \underline{p}_K F_1^{K-k} L_1 T_0 + \underline{q}_0 F_2^k L_2 T_0 \text{ for } 0 \le k \le K, \tag{8.15}$$

and

$$\underline{v}_{M_1-y_1+K+n} = \underline{w}_K T_n = \underline{p}_K L_1 T_n + \underline{q}_0 F_2^K L_2 T_n \text{ for } 0 \le n \le y-1. \tag{8.16}$$

With equations (8.15) and (8.16) the unknown probability vectors that we have to compute now are $\underline{v}_0, \ldots, \underline{v}_{M_1-y_1-1}, \underline{p}_K, \underline{q}_0$ and $\underline{v}_{M_2+y_2+1}, \ldots, \underline{v}_{\mathcal{K}}$, i.e. there are $(M_1 - y_1) + y + (\mathcal{K} - M_2 - y_2)$ $= M_1 + \mathcal{K} - M_2$ unknown vectors, each of size $\mathcal{N}+1$. To get these unknown vectors, we have vector equations (8.3) for $j = 0, \ldots, M_1 - 1$ (the first M_1 equations) and for $j = M_2 + 1, \ldots, \mathcal{K}$ (the last $\mathcal{K} - M_2$ equations). However, only $M_1 + \mathcal{K} - M_2 - 1$ of these equations are linearly independent due to the fact that the generator matrix of the Markov process is singular. This means that one equation must be replaced by the normalizing equation (8.4). Using equations (8.15) and (8.16), the equivalent form of (8.4) is

$$\begin{aligned} &\sum_{j=0}^{M_1-y_1-1} \underline{v}_j \underline{e} + \sum_{j=M_2+y_2+1}^{\mathcal{K}} \underline{v}_j \underline{e} + \underline{p}_K \left[\sum_{k=0}^{K} {F_1}^{K-k} L_1 T_0 + \sum_{n=1}^{y-1} L_1 T_n \right] \underline{e} + \\ &+ \underline{q}_0 \left[\sum_{k=0}^{K} F_2^k L_2 T_0 + \sum_{n=1}^{y-1} F_2^K L_2 T_n \right] \underline{e} = 1. \end{aligned} \tag{8.17}$$

What remains to be clarified is how to determine the values of m_u and m_s. For this aim we take a corresponding infinite QBD-M process, which possesses the same balance equations (8.5) with the same level-independent coefficient matrices $\mathcal{A}$, B_s $(1 \le s \le y_1)$ and C_s $(1 \le s \le y_2)$.

In accordance with the interpretation of the stability criteria formulated in previous works (see [16, 19]) we stress that this infinite process is stable as long as, for large values of j, the forward drift from level j is less than the backward drift from it. For large values of j, the conditional probability of being in state (i, j), given that we are in level j, tends to the entry v_i of vector $\underline{v}$ which is of size $\mathcal{N}+1$ and is the unique solution of the equations

$$\underline{v} \left(\mathcal{A} + \sum_{s=1}^{y_1} B_s + \sum_{s=1}^{y_2} C_s \right) = 0, \quad \underline{v}\underline{e} = 1. \tag{8.18}$$

In other words, $\underline{v}$ is the stationary solution of the Markov chain having $\mathcal{A} + \sum_{s=1}^{y_1} B_s + \sum_{s=1}^{y_2} C_s$ as the infinitesimal generator matrix which is assumed to be irreducible. The relation between the forward and the backward drifts for the stability then is

$$\underline{v} \left(\sum_{s=1}^{y_1} s B_s \right) \underline{e} < \underline{v} \left(\sum_{s=1}^{y_2} s C_s \right) \underline{e}. \tag{8.19}$$

If the inequality (8.19) is fulfilled, then it is known from [16] that the $det(D(\lambda)) = 0$ equation has exactly $y_1(\mathcal{N}+1)$ roots inside the unit disk. In this case we have $m_u = y_2(\mathcal{N}+1)$ and $m_s = y_1(\mathcal{N}+1)$. Also from [16], the case in which the inequality does not hold implies that $m_u = y_2(\mathcal{N}+1)+1$ and $m_s = y_1(\mathcal{N}+1)-1$.

8.4 Fitting the queueing system into a QBD-M process

As it has been pointed out in Section 8.2, a queueing model with two priority classes is a two dimensional Markov process. Due to the operational features of the queues (WBA rule, two level feedback control, preemptive service), the generator matrix of this process will have the form shown in Figure 8.1.

$$
Q = \left[\begin{array}{ccccc|ccccc|ccccc}
\mathcal{A}_0 & B_{0,1} & B_{0,2} & & & & & & & & & & & & \\
C_1 & \mathcal{A}_1 & B_{1,1} & B_{1,2} & & & & & & & & & & & \\
 & C_1 & \mathcal{A}_2 & B_{2,1} & B_{2,2} & & & & & & & & & & \\
\hline
 & & C_1 & \mathcal{A} & B_1 & B_2 & & & & & & & & & \\
 & & & C_1 & \mathcal{A} & B_1 & B_2 & & & & & & & & \\
 & & & & C_1 & \mathcal{A} & B_1 & B_2 & & & & & & & \\
 & & & & & C_1 & \mathcal{A} & B_1 & B_2 & & & & & & \\
 & & & & & & \ddots & & \ddots & & & & & & \\
 & & & & & & & & C_1 & \mathcal{A} & B_1 & B_2 & & & \\
\hline
 & & & & & & & & & C_1 & \mathcal{A}_{K_L-3} & B_{K_L-3,1} & B_{K_L-3,2} & & \\
 & & & & & & & & & & C_1 & \mathcal{A}_{K_L-2} & B_{K_L-2,1} & B_{K_L-2,2} & \\
 & & & & & & & & & & & C_1 & \mathcal{A}_{K_L-1} & B_{K_L-1,1} & \\
 & & & & & & & & & & & & C_1 & \mathcal{A}_{K_L} &
\end{array}\right]
$$

Figure 8.1: The block structure of the generator matrix of the queueing system ($y_1 = 2$, $m_1 = 3, m_2 = 4$)

It is not difficult to show now that this queueing model is a finite QBD-M process with parameters $\mathcal{N} = N + K_H$, $y_1 = L$, $y_2 = 1$, $\mathcal{K} = K_L$, $M_1 = m_1 + y_1$ and $M_2 = min(K_L - y_2, K_L - m_2)$. The generator matrix is constructed by building up the fundamental transition matrices A_j, $B_{j,s}$ and $C_{j,s}$, each of size $(N + K_H) \times (N + K_H)$. Given the parameters and operational principle delineated in Section 8.2, one can easily write the basic matrices as in Figure 8.2.

Note that expression (8.20) is valid for the case $K_H \leq y_1$. If $K_H > y_1$, the matrix A_j still has the same form, except that row i will appear after row N. Also note that in expression (8.21) $\lambda_{L,j} = \lambda_L$ if $m_1 \leq j \leq K_L - m_2$ and $\lambda_{L,j} = (1 - p_{dropp})\lambda_L$ for $K_L - m_2 < j \leq K_L$. Since in our case $y_2 = 1$, in expression (8.22) only $C_{j,1}$ is considered.

$$A_j = A = \begin{bmatrix} 0 & h_1\lambda_H & h_2\lambda_H & \dots & h_H\lambda_H & & & & & 0 \\ \mu_H & 0 & h_1\lambda_H & \dots & \dots & h_H\lambda_H & & & & 1 \\ & \ddots & \ddots & \ddots & & & & & & \vdots \\ & & i\mu_H & 0 & h_1\lambda_H & \dots & & & h_H\lambda_H & i \\ & & & & \ddots & \ddots & \ddots & & & \vdots \\ & & & & N\mu_H & 0 & & & h_{K_H}\lambda_{K_H} & N \\ & & & & & \ddots & & \ddots & \vdots & \\ & & & & & & & & h_1\lambda_H & \vdots \\ & & & & & & & N\mu_H & 0 & N+K_H \end{bmatrix} \quad (8.20)$$

$$B_{j,s} = \begin{bmatrix} l_s\lambda_{L,j} & & & \\ & l_s\lambda_{L,j} & & \\ & & \ddots & \\ & & & l_s\lambda_{L,j} \end{bmatrix} \text{ for } 0 \le j \le K_L, 1 \le s \le L \quad (8.21)$$

$$C_{j,1} = C_1 = \begin{bmatrix} N\mu_L & & & & & & & 0 \\ & \ddots & & & & & & \vdots \\ & & 2\mu_L & & & & & N-2 \\ & & & \mu_L & & & & N-1 \\ & & & & 0 & & & N \\ & & & & & \ddots & & \vdots \\ & & & & & & 0 & N+K_H \end{bmatrix} \quad (8.22)$$

Figure 8.2: Basic transition matrices

Once mapping the queueing system of interest into a finite QBD-M process is done, we are now ready to compute the steady state distribution of the queueing system by the straightforward application of the analysis method described in Section 8.3.

8.5 Derivation of performance measures

In this section some important performance measures are derived from the steady state probabilities.

8.5.1 Average number of customers

Denote $\underline{\pi} = \sum_{j=0}^{K_L} \underline{v}_j$ and let π_i be the i-th element of vector $\underline{\pi}$.

For the high priority class, the average number of customers in the queue is computed as

$$E_{high} = \sum_{i=N}^{N+K_H} (i-N)\pi_i. \tag{8.23}$$

The average number of low priority customers in the system is calculated as

$$E_{low} = \sum_{j=0}^{K_L} j\underline{v}_j\underline{e}. \tag{8.24}$$

8.5.2 Batch loss probability

This parameter is defined as the probability that an arriving batch is rejected due to the lack of storage and/or due to the effect of the dropping control level. For the high priority class, a batch of size k is rejected only if upon its arrival, the number of high priority customers in the system is more than $(N + K_H) - k$. Using the PASTA (Poisson Arrivals See Time Average) property, the loss probability for the high priority batch of size k is

$$P(\text{a hp. batch of size } k \text{ is lost}) = \sum_{i=(N+K_H-k+1)^+}^{N+K_H} \pi_i, \tag{8.25}$$

where $(x)^+$ gives the maximum of x and 0.

By unconditioning, the batch loss priority for the high priority class is

$$P(\text{hp. batch loss}) = \sum_{k=1}^{H} h_k \left(\sum_{i=(N+K_H-k+1)^+}^{N+K_H} \pi_i \right). \tag{8.26}$$

For the low priority class, a batch of size k is always dropped with probability p_{dropp} if there are more than $K_L - m_2$ low priority customers in the system. Moreover, this batch is also dropped if it sees more than $K_L - k$ customers in the system. Therefore, two kinds of batch loss probabilities can be introduced. The first one, referred to as the *blocking probability*, is caused merely by the effect of the finite buffer and is derived in a similar way we have done for the high priority class

$$P^{(1)}(\text{a lp. batch of size } k \text{ is lost}) = \sum_{j=(K_L-k+1)^+}^{K_L} \underline{v}_j\underline{e}, \tag{8.27}$$

$$P^{(1)}(\text{lp. batch loss}) = \sum_{k=1}^{L} l_k \left(\sum_{j=(K_L-k+1)^+}^{K_L} \underline{v}_j\underline{e} \right). \tag{8.28}$$

The second kind of loss probability is caused by the effect of both the control feedback level and the finite buffer. This is referred to as the *overall batch loss probability* and is computed as

$$P^{(2)}(\text{a lp. batch of size } k \text{ is lost}) = \sum_{j=(K_L-k+1)^+}^{K_L} \underline{v}_j\underline{e} + \Omega_{(k\le m_2-1)} \cdot p_{dropp} \left(\sum_{j=K_L-m_2+1}^{K_L-k} \underline{v}_j\underline{e} \right), \tag{8.29}$$

$$P^{(2)}(\text{lp. batch loss}) = \sum_{k=1}^{L} l_k \left[\sum_{j=(K_L-k+1)^+}^{K_L} \underline{v}_j \underline{e} + \Omega_{(k \le m_2 - 1)} p_{dropp} \left(\sum_{j=K_L-m_2+1}^{K_L-k} \underline{v}_j \underline{e} \right) \right], \quad (8.30)$$

where $\Omega_{\mathcal{X}}$ denotes the indicator function, which is 0 if event $\mathcal{X}$ is false and 1 otherwise.

8.5.3 Customer loss probability

Let the average arrival batch size of the high priority and the low priority class be $\overline{h}$ and $\overline{l}$, respectively. It is clear that $\overline{h} = \sum_{i=1}^{H} i h_i$ and $\overline{l} = \sum_{i=1}^{L} i l_i$. The probability that a customer arrives in a batch with size k is equal to $kh_k/\overline{h}$ (high priority case) and $kl_k/\overline{l}$ (low priority case). When the WBA rule is applied, the probability that the customer in a batch of size k is dropped is equal to the probability that the batch itself is dropped. Therefore, it follows that

$$P(\text{customer loss}) = \sum_{k=1}^{b_{max}} \frac{k r_k}{b} P(\text{a batch of size } k \text{ is lost}). \quad (8.31)$$

The expression above is valid for customers of any class. For the high priority class we have to make $b_{max} = H, b = \overline{h}, r_k = h_k$ and use the equation (8.25). For the two kinds of low priority losses, $b_{max} = L, b = \overline{l}, r_k = l_k$ substitutions and equations (8.27) or (8.29) are necessary.

8.5.4 Average queueing and system times

Applying Little's formula, the mean waiting time of a high priority customer is written as

$$W_{high} = \frac{E_{high}}{\overline{h}\lambda_H (1 - P(\text{hp. customer loss}))}. \quad (8.32)$$

For the low priority customer, the mean system time is calculated again by Little's theorem

$$S_{low} = \frac{E_{low}}{\overline{l} \left(\sum_{j=0}^{K_L} \lambda_{L,j} \underline{v}_j \underline{e} \right) \left(1 - P^{(2)}(\text{lp. customer loss}) \right)}. \quad (8.33)$$

8.5.5 System time distribution

For the high priority class, define the Laplace transform

$$S_i(s) = \begin{cases} \left(\dfrac{N\mu_H}{s + N\mu_H} \right)^{i-N} \left(\dfrac{\mu_H}{s + \mu_H} \right) & \text{for } N < i \le N + K_H \\ \dfrac{\mu_H}{s + \mu_H} & \text{for } i \le N \end{cases} \quad (8.34)$$

If a high priority customer arrives in position k of a batch of size r and sees i other high priority customers in the system (provided that a batch is accepted), then the Laplace transform

of the system time distribution of that customer is given by $S_{i+k}(s)$. Since the probability of being in position k in a batch of size r is $\xi_k = \frac{1}{r}$ and the distribution of the batch size is known, the total probability rule yields

$$S_{hp}(s) = \sum_{i=0}^{N+K_H} \pi_i \left(\sum_{r=1}^{H} \Omega_{\{r+i \leq N+K_H\}} h_r \left(\sum_{k=1}^{r} \xi_k S_{i+k} \right) \right). \tag{8.35}$$

The derivation of system time distribution for low priority customers is done in two steps. Firstly, for the sake of convenient interpretation, we renumber the states of the QBD-M process, i.e. each (i, j) state now corresponds to a state with assigned index $j(\mathcal{N}+1)+i$. With this "transformation", each state of the process can be accessed not by a couple of index numbers, but only one. Given a state having index k, the original couple of indexes are

$$j = \left[\frac{k}{\mathcal{N}+1} \right], \tag{8.36}$$

$$i = k - j(\mathcal{N}+1), \tag{8.37}$$

where $[x]$ denotes the greatest integer, which is less than or equal to x. Let us denote the steady state probability vector of the process by $\underline{\gamma}$, which is in fact a rewritten version of all vectors $\underline{v}_j$ calculated in the previous section.

Secondly, the technique of the Markov driven workload process is utilized to determine the system time distribution. Recall that the service capacity for low priority customers is evenly shared between them and it depends on the number of high priority customers. In other words, one can consider a workload process controlled by our original Markov process. When the driving Markov process is in state k, the workload process accomplishes service at rate $r_k = \frac{max(N-i,0)}{j}$ where i and j are determined from k as in expressions (8.36) and (8.37)

When a low priority customer arrives, it brings its service requirement x. This customer will leave the system when the workload process defined above accomplishes a work amount of x. The fact that the customer arrives alone or in a batch with other ones does not matter at this point, because the essence here is that upon an arrival a workload process changes its service rate according to the state change of the driving Markov process. That is, the information of possible batch arrivals is indeed included in the generator matrix of the driving process.

Since the service requirement x is exponentially distributed with parameter μ_L, the Laplace transform of the density function of system time can be derived in a similar way to what is described in [29] and appears to correspond to a phase time distribution

$$S_{lp}(s) = \underline{p}_R [sI - (Q - \mu_L R)]^{-1} R \mu_L \underline{e}, \tag{8.38}$$

where Q is the generator matrix of the driving Markov process (defined by expression (8.1)), R is a diagonal matrix, in which the entry (k, k) gives the service rate in state k, and $\underline{p}_R = \frac{\underline{\gamma} R}{\underline{\gamma} R \underline{e}}$ is a

row vector of initial probabilities seen by the entering customer. The phase type distribution has the initial probability $\underline{p}_R$, transient matrix $Q - \mu_L R$ and vector $R\mu_L \underline{e}$ of rates to the absorbing state.

8.6 Numerical results

Since the system has a quite wide set of input parameters, in this section we only give some representative numerical examples concerning the operation of the system. In what follows, unless otherwise stated, the parameter set $N = 10, \mu_L = 1, \lambda_L = 2, \mu_H = 2, \lambda_H = 4, K_L = 50, H = L = y_1$ is used. The distribution of arrival batch size is assumed to be uniform.

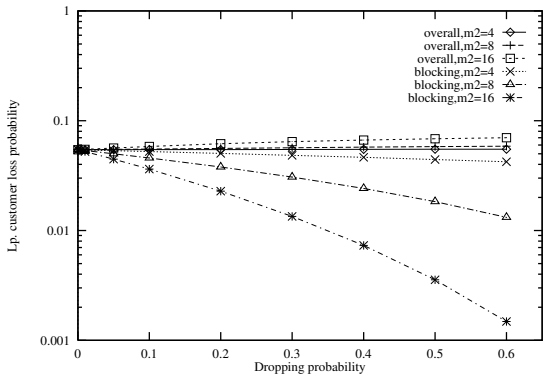


Figure 8.3: Lp. customer loss probability versus dropping probability

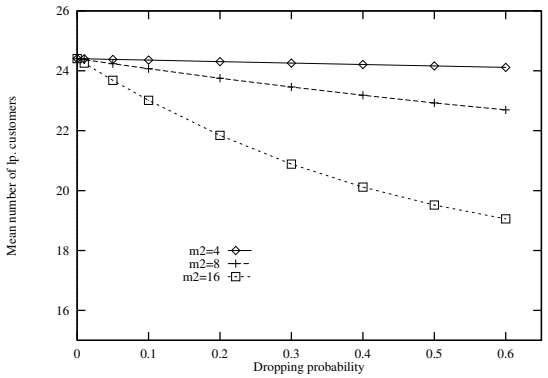


Figure 8.4: Mean queue length of lp. customers versus dropping probability

First, we discuss the motivation of introducing the early dropping scheme through the effect of the dropping probability and the dropping threshold level. The impact of dropping probability (p_{dropp}) on performance measures related to the low priority class is shown in Figure 8.3 and Figure 8.4. The rest of the system parameters is set to $m_1 = 3, y_1 = 4, K_H = 10$. In Figure 8.3, both the blocking and the overall loss probabilities are plotted as a function of the dropping probability. The first noteworthy observation is that increasing the dropping probability of arriving batches increases slightly the overall and at the same time reduces significantly the blocking probability. The latter performance measure is in connection with the mean queue length, which decreases with the dropping probability as shown in Figure 8.4. Thus, we obtain better performance with regard to the mean system time for low priority customers. Of course, the trade-off between the increase of the overall customer loss probability and the reduction of the mean queue length (and through it, the mean system time) must be kept in mind for the aim of dimensioning.

The second observation from the figures is that the smaller the control level $K_L - m_2$ is chosen, the better performance is obtained from the aspect of the mean queue length and the blocking probability, at the same time the worse the performance becomes from the aspect of

the overall loss probability. Again, the trade-off between the increase of the overall customer loss probability and the reduction of the mean queue length (and through it, the mean system time) should be carefully considered when choosing an appropriate control level $K_L - m_2$. Thus, by choosing appropriately the control level and the predefined dropping probability we are able to prevent the system from congestion.

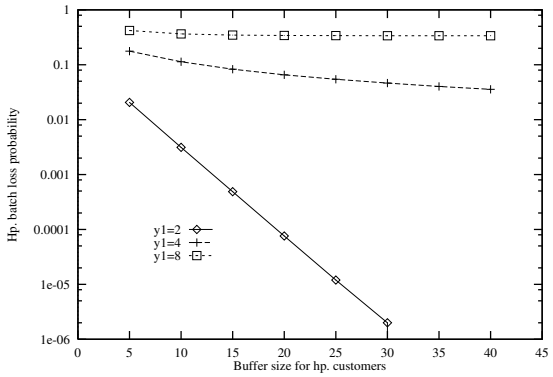


Figure 8.5: Hp. batch loss probability versus buffer size

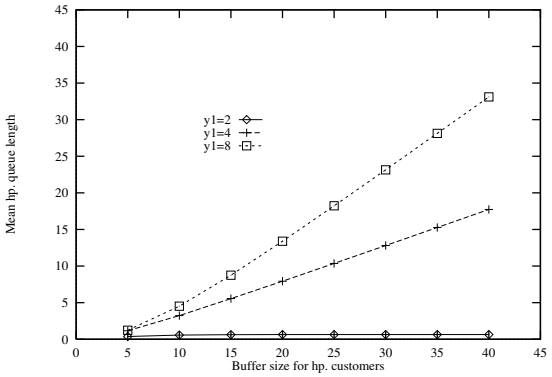


Figure 8.6: Mean queue length of hp. customer versus buffer size

In Figure 8.5 and Figure 8.6 we check the impact of buffer size (K_H) for customers of the high priority class on the queueing performance. The system parameters are $N = 5, m_1 = 5, m_2 = 8, p_{dropp} = 0.01$. The maximum size of arriving batches y_1 is chosen in sequence to be $2, 4, 8$. Obviously, increasing the buffer size reduces the batch loss probability as well as the customer loss probability, as shown in Figure 8.5. The decreasing rate is quite fast when the maximum batch size is small (corresponding to small offered traffic) and seems to slow down when the maximum batch size (so as the offered traffic) becomes greater. Note that if the reduction of the batch (customer) loss probability is achieved by increasing the storage requirement then it also implies the increased the mean queue length and through it the mean queueing delay, especially in case of heavy traffic load. This situation is depicted in Figure 8.6.

We now discuss the influence of the offered traffic on the queueing performance with respect to the interaction between the two classes. The system parameters are $m_1 = 5, m_2 = 8, K_H = 10, y_1 = 4, p_{dropp} = 0.01, \mu_L = 1$. The offered traffic is increased by varying the parameter $ratio = \lambda_H/\mu_H$ meanwhile the parameter $r_L = \lambda_L/\mu_L$ is fixed. Figure 8.7 and Figure 8.8 clearly indicate the effect of the preemptive principle. As high priority customers have a right to occupy the service capacity over low priority customers, the offered traffic of the high priority class firstly forces more customers of the low priority class to disclaim their service capacity and return to waiting condition. Until a certain value of *ratio*, there is no customer of the high priority class in the queue and no high priority customer loss at all. Further increase of the high priority offered traffic causes the occupancy of buffers and initiates loss events related to both classes. However, due to the preemptive service discipline, the mean number of low priority customers

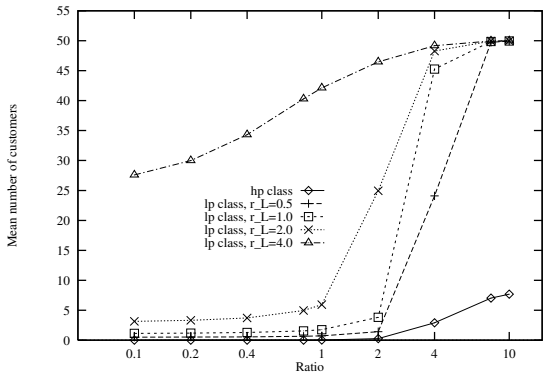


Figure 8.7: Mean queue length versus offered traffic

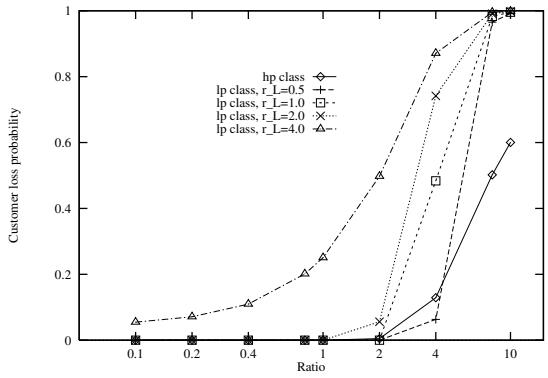


Figure 8.8: Customers loss probability versus offered traffic

in the queue, as well as the customer loss probability of the low priority class are increasing at a rate greater than that of the high priority class. It is also noteworthy that the offered load of the low priority class (expressed by parameter r_L) does not have any effect on how the performance parameters of the high priority class shapes. That is why for different values of r_L the curve indicating the mean number of customers (the customer loss probability) of the high priority class remains the same as plotted in Figure 8.7 (Figure 8.8).

8.7 Conclusions and contributions

In this chapter, a priority queueing model has been constructed in order to enable the performance analysis of scheduling mechanisms possibly applied to the DiffServ architecture. Compared to previous works on priority queues, some extensions have been considered such as the finite capacity of buffers, the batch arrival condition and the level-dependent feedback control mechanism.

We have shown that the system belongs to a class of finite QBD-M processes, for the steady state analysis of which a novel calculation method based on the theory of generalised invariant subspaces has been proposed. By means of the steady state distribution some important performance parameters have been derived and numerically evaluated.

Future works may include the investigation of some interesting issues such as transient analysis of the queueing model, or giving suggestions on how to provide an optimal choice of the dropping probability and the congestion feedback level.

Part of this chapter was published in our paper [41].

Part V

Conclusions

Chapter 9

Conclusions

The dissertation has dealt with some algorithmic and numerical methods emerging in network planning and performance evaluation. The first part of the thesis has presented a practical methodology for the topology optimisation of an ATM network to be deployed on top of the existing SDH infrastructure. The optimisation task has been formulated with exact mathematical tools. From the practical point of view, the appealing and salient feature of the proposed planning method is that the whole task is divided into subtasks, whose solution can be efficiently obtained in sequence with the reuse of heuristic algorithms. The proposed algorithmic method has been implemented and integrated into the platform of the XPLANET network planning tool, which indeed has raised its availability and applicability to realistic optimisation tasks.

The rest parts of the thesis aim to find and choose efficient numerical methods capable for solving some classes of Markov chains. Emphasis has been put to two versatile models called QBD and QBD-M processes. In the second part, we have summarized the most popular numerical methods developed for the steady state solution of QBD processes and have compared the performance of these methods via a concrete application example. The comparative results contribute to the choosing policy of when, why and what method should be used for a QBD process with given parameters.

The third part of the thesis has been devoted to QBD-M processes and its application in performance evaluation and system analysis. We have given a general description of QBD-M processes and a review of numerical methods for their steady state solution as well. Furthermore, we have presented two novel proposals for determining the steady state solution of QBD-M processes. Our in-depth comparative study through some chosen systems has clearly indicated the advantages of each proposed method over the existing ones. We have constructed three queueing models for the performance analysis of a node performing multipath routing in MPLS networks, of an ATM concentrator and of a scheduling mechanism in DifServ architecture. We have shown that with adequate formulations, all the above three models lead to the application of QBD-M processes.

The scope of the dissertation raises a lot of interesting research directions. For example, the performance assessment of the network obtained as the result of the proposed planning method developed in the first part of the dissertation deserves further efforts because it provides useful feedbacks to improve the planning process. Moreover, the optimisation method may be extended to cases where protection techniques are involved during the planning process.

Another one of the most tangibly emerging challenges is to complete the applicability range of the new iterative method proposed in Chapter 6. That is, to extend the method's capability to such cases, when the upperbound of departure batches (y_2) is greater than the upperbound of arrival batches (y_1). We feel that somewhat the same train of thought delineated in Section 6.1 should lead to the expected result, although we have not managed to find out an appropriate derivation until now. Thus, further investigations and efforts are needed, indeed.

It may come to view that throughout the dissertation only the steady state analysis has been touched. Thus, the transient behaviour of the considered queueing systems should also receive attention. This is another direction to move ahead and is certainly of interest.

Our belief is that the modelling capability of QBD and QBD-M queueing models is much more powerful than what has been demonstrated in this dissertation. Their modelling power may be utilized in a variety of today's favorite research areas which covers investigation of TCP, analysis of mobile cellular system and wireless CDMA networks with regards to quality of services, evaluation of QoS and traffic engineering architectures (DiffServ, MPLS). Among them, modelling and analysis of multiple handoff phenomenon in cellular mobile network and performance analysis of nodes performing multipath routing in MPLS networks especially attract my interest. At present, this kind of research work is underway. The related preliminary results have been published in [46].

Appendix A

Proofs of the theorems

A.1 Proof of theorem 6.1.2

Recall that after reblocking, we obtain the transformed QBD process which has the minimal non-negative solution for the rate matrix $\mathbf{R}$. Consequently, all the inner matrices $T_{i,j}$ $(0 \leq i \leq y_1 - 1, 0 \leq k \leq y_2 - 1)$ are also minimal non-negative solution. In order to see that they fulfill the given matrix equations, we first consider the following two remarks:

Remark 1 *Denote* $S = \sum_{k=1}^{y_2} \underline{v}_{j+k} C_k$*, then applying equation (6.3) leads to*

$$\begin{aligned}
S &= \sum_{k=1}^{y_2} \left(\sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i+1} T_{i,k-1} \right) C_k \\
&= \sum_{k=1}^{y_2} \left(\sum_{i=0}^{y_1-2} \underline{v}_{j-y_1+i+1} T_{i,k-1} + \underline{v}_j T_{y_1-1,k-1} \right) C_k \\
&= \sum_{k=1}^{y_2} \sum_{i=0}^{y_1-2} \underline{v}_{j-y_1+i+1} T_{i,k-1} C_k + \sum_{k=1}^{y_2} \underline{v}_j T_{y_1-1,k-1} C_k \\
&= \sum_{k=1}^{y_2} \sum_{i=1}^{y_1-1} \underline{v}_{j-y_1+i} T_{i-1,k-1} C_k + \underline{v}_j \sum_{k=1}^{y_2} T_{y_1-1,k-1} C_k \\
S &= \sum_{i=1}^{y_1-1} \underline{v}_{j-y_1+i} \left(\sum_{k=1}^{y_2} T_{i-1,k-1} C_k \right) + \underline{v}_j \sum_{k=1}^{y_2} T_{y_1-1,k-1} C_k
\end{aligned}$$

Remark 2 *Due to equation (6.2)*

$$\underline{v}_j = \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i} T_{i,0} = \underline{v}_{j-y_1} T_{0,0} + \sum_{i=1}^{y_1-1} \underline{v}_{j-y_1+i} T_{i,0}$$

Now, using the result of Remark 1, equation (5.3) can be rewritten as

$$\begin{aligned}
\underline{v}_j &= \sum_{i=1}^{y_1} \underline{v}_{j-i} B_i + \underline{v}_j A + \sum_{k=1}^{y_2} \underline{v}_{j+k} C_k \\
&= \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i} B_{y_1-i} + \underline{v}_j A + S \\
&= \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i} B_{y_1-i} + \underline{v}_j \left(A + \sum_{k=1}^{y_2} T_{y_1-1,k-1} C_k \right) + \sum_{i=1}^{y_1-1} \underline{v}_{j-y_1+i} \left(\sum_{k=1}^{y_2} T_{i-1,k-1} C_k \right)
\end{aligned}$$

Using Remark 2 leads to

$$\begin{aligned}
\underline{v}_j = \underline{v}_{j-y_1} &\left[B_{y_1} + T_{0,0} \left(A + \sum_{k=1}^{y_2} T_{y_1-1,k-1} C_k \right) \right] \\
&+ \sum_{i=1}^{y_1-1} \underline{v}_{j-y_1+i} \left[B_{y_1-i} + T_{i,0} \left(A + \sum_{k=1}^{y_2} T_{y_1-1,k-1} C_k \right) \right. \\
&\left. + \sum_{k=1}^{y_2} T_{i-1,k-1} C_k \right]
\end{aligned}$$

Comparing this equation with equation (6.2) of Theorem 6.1.1

$$\underline{v}_j = \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i} T_{i,0}, \quad \forall j \geq M$$

and making the coefficients of $\underline{v}_{j-y_1+i},\ i = 0, 1, \ldots, y_1 - 1$ equal, we arrive at (6.6) and (6.7).

In order to justify (6.8) and (6.9), we use equation (6.2) in Theorem 6.1.1 and substitute subindex j by $j + k$, $1 \leq k \leq y_2 \leq y_1$

$$\begin{aligned}
\underline{v}_{j+k} &= \sum_{ii=0}^{y_1-1} \underline{v}_{j-y_1+ii+k} T_{ii,0} \\
&= \sum_{ii=0}^{y_1-k} \underline{v}_{j-y_1+ii+k} T_{ii,0} + \sum_{ii=y_1-k+1}^{y_1-1} \underline{v}_{j-y_1+ii+k} T_{ii,0} \\
&= \sum_{ii=k-1}^{y_1-1} \underline{v}_{j-y_1+ii+1} T_{ii-(k-1),0} + \sum_{ii=1}^{k-1} \underline{v}_{j+ii} T_{ii+y_1-k,0} \qquad \text{(A.1)}
\end{aligned}$$

Denote the second sum of the above expression by SS. Applying equation (6.3) in Theorem 6.1.1, we have

$$\begin{aligned}
SS &= \sum_{ii=1}^{k-1} \underline{v}_{j+ii} T_{ii+y_1-k,0} \\
&= \sum_{ii=1}^{k-1} \left(\sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i+1} T_{i,ii-1} \right) T_{ii+y_1-k,0}
\end{aligned}$$

$$= \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i+1} \left(\sum_{ii=1}^{k-1} T_{i,ii-1} T_{ii+y_1-k,0} \right)$$

Substituting back the expression of SS into (A.1) we get

$$\underline{v}_{j+k} = \sum_{i=0}^{k-2} \underline{v}_{j-y_1+i+1} \left(\sum_{ii=1}^{k-1} T_{i,ii-1} T_{ii+y_1-k,0} \right) + \sum_{i=k-1}^{y_1-1} \underline{v}_{j-y_1+i+1} \left(\sum_{ii=1}^{k-1} T_{i,ii-1} T_{ii+y_1-k,0} + T_{i-(k-1),0} \right)$$

Recall that according to equation (6.3) of Theorem 6.1.1

$$\underline{v}_{j+k} = \sum_{i=0}^{y_1-1} \underline{v}_{j-y_1+i+1} T_{i,k-1}$$

therefore (6.8) and (6.9) come directly from the equalities of coefficients of $\underline{v}_{j-y_1+i+1}$, $0 \leq i \leq y_1 - 1$. □

A.2 Proof of theorem 6.1.3

Two steps are performed.

Step 1

In this step we show that the sequences of $X_{i,k}^{(n)}$, $0 \leq i \leq y_1 - 1, 0 \leq k \leq y_2 - 1$ are entry-wise non-decreasing. The induction technique is used for this aim.

Since matrices $B_1, B_2, \ldots, B_{y_1}$ contain non-negative elements, $0 = X_{i,k}^{(0)} \leq X_{i,k}^{(1)}$ for all $0 \leq i \leq y_1 - 1$ and $0 \leq k \leq y_2 - 1$. Assuming that the inequality holds for n then

$$\begin{aligned} X_{0,0}^{(n+1)} &= B_{y_1} + X_{0,0}^{(n)} \left(A + \sum_{k=1}^{y_2} X_{y_1-1,k-1}^{(n)} C_k \right) \\ &\geq B_{y_1} + X_{0,0}^{(n-1)} \left(A + \sum_{k=1}^{y_2} X_{y_1-1,k-1}^{(n-1)} C_k \right) \\ &= X_{0,0}^{n} \end{aligned}$$

$$\begin{aligned} X_{i,0}^{(n+1)} &= B_{y_1-i} + X_{i,0}^{(n)} \left(A + \sum_{k=1}^{y_2} X_{y_1-1,k-1}^{(n)} C_k \right) + \sum_{k=1}^{y_2} X_{i-1,k-1}^{(n)} C_k \\ &\geq B_{y_1-i} + X_{i,0}^{(n-1)} \left(A + \sum_{k=1}^{y_2} X_{y_1-1,k-1}^{(n-1)} C_k \right) + \sum_{k=1}^{y_2} X_{i-1,k-1}^{(n-1)} C_k \\ &= X_{i,0}^{(n)} \text{ for } i = 1, \ldots, y_1 - 1 \end{aligned}$$

$$
\begin{aligned}
X_{i,k}^{(n+1)} &= \sum_{ii=1}^{k} X_{i,ii-1}^{(n)} X_{ii+y_1-(k+1),0}^{(n)} \\
&\geq \sum_{ii=1}^{k} X_{i,ii-1}^{(n-1)} X_{ii+y_1-(k+1),0}^{(n-1)} \\
&= X_{i,k}^{(n)} \text{ for } 1 \leq k \leq y_2 - 1 \text{ and } 0 \leq i \leq k-1
\end{aligned}
$$

$$
\begin{aligned}
X_{i,k}^{(n+1)} &= \sum_{ii=1}^{k} X_{i,ii-1}^{(n)} X_{ii+y_1-(k+1),0}^{(n)} + X_{i-k,0}^{(n)} \\
&\geq \sum_{ii=1}^{k} X_{i,ii-1}^{(n-1)} X_{ii+y_1-(k+1),0}^{(n-1)} + X_{i-k,0}^{(n-1)} \\
&= X_{i,k}^{(n)} \text{ for } 1 \leq k \leq y_2 - 1 \text{ and } k \leq i \leq y_1 - 1
\end{aligned}
$$

Step 2

Now, we show that the sequences of $X_{i,k}^{(n)}$, $0 \leq i \leq y_1 - 1, 0 \leq k \leq y_2 - 1$ are upper-bounded. This assertion is verified again by induction. For $n = 0$, $0 = X_{i,k}^{(0)} \leq T_{i,k}$, $0 \leq i \leq y_1 - 1, 0 \leq k \leq y_2 - 1$. Assuming that $X_{i,k}^{(l)} \leq T_{i,k}$ holds for all $l \leq n$, then

$$
\begin{aligned}
X_{0,0}^{(n+1)} &= B_{y_1} + X_{0,0}^{(n)} \left(A + \sum_{k=1}^{y_2} X_{y_1-1,k-1}^{(n)} C_k \right) \\
&\leq B_{y_1} + T_{0,0} \left(A + \sum_{k=1}^{y_2} T_{y_1-1,k-1} C_k \right) \\
&= T_{0,0}
\end{aligned}
$$

$$
\begin{aligned}
X_{i,0}^{(n+1)} &= B_{y_1-i} + X_{i,0}^{(n)} \left(A + \sum_{k=1}^{y_2} X_{y_1-1,k-1}^{(n)} C_k \right) + \sum_{k=1}^{y_2} X_{i-1,k-1}^{(n)} C_k \\
&\leq B_{y_1-i} + T_{i,0} \left(A + \sum_{k=1}^{y_2} T_{y_1-1,k-1} C_k \right) + \sum_{k=1}^{y_2} T_{i-1,k-1} C_k \\
&= T_{i,0} \text{ for } i = 1, \ldots, y_1 - 1
\end{aligned}
$$

$$
\begin{aligned}
X_{i,k}^{(n+1)} &= \sum_{ii=1}^{k} X_{i,ii-1}^{(n)} X_{ii+y_1-(k+1),0}^{(n)} \\
&\leq \sum_{ii=1}^{k} T_{i,ii-1} T_{ii+y_1-(k+1),0} \\
&= T_{i,k} \text{ for } 1 \leq k \leq y_2 - 1 \text{ and } 0 \leq i \leq k-1
\end{aligned}
$$

$$
\begin{aligned}
X_{i,k}^{(n+1)} &= \sum_{ii=1}^{k} X_{i,ii-1}^{(n)} X_{ii+y_1-(k+1),0}^{(n)} + X_{i-k,0}^{(n)} \\
&\leq \sum_{ii=1}^{k} T_{i,ii-1} T_{ii+y_1-(k+1),0} + T_{i-k,0} \\
&= T_{i,k} \text{ for } 1 \leq k \leq y_2 - 1 \text{ and } k \leq i \leq y_1 - 1
\end{aligned}
$$

In summary, the sequences of $X_{i,k}^{(n)}$, $0 \leq i \leq y_1 - 1, 0 \leq k \leq y_2 - 1$ are monotonously increasing and upper-bounded, hence they are convergent. In addition, the limit matrices fulfill the equations (6.6), (6.7), (6.8), (6.9) and they are not greater than the minimal non-negative solutions. Thus the sequences of $X_{i,k}^{(n)}$, $0 \leq i \leq y_1 - 1, 0 \leq k \leq y_2 - 1$ converge on the minimal non-negative solutions of equations (6.6), (6.7), (6.8), (6.9). □

Bibliography

[1] http://ls4-www.informatik.uni-dormund.de/tools.html.

[2] http://www4.informatik.uni-erlangen.de/Projects.

[3] http://www.ee.duke.edu/~kst/tools.html.

[4] The ATM Forum: Wide Area ATM Deployment in Europe. http://www.atmforum.com/atmforum/library/widearea_atm_deploy_eu.html.

[5] N. Akar, N. C. Oguz, and K. Sohraby. Matrix-geometric Solution of M/G/1 Type Markov Chains: A Unifying Genealized State-space Approach. *IEEE JSAC*, 16:626–639, 1998.

[6] N. Akar, N. C. Oguz, and K. Sohraby. A Novel Computational Method for Solving Finite QBD processes. *Communications in Statistics-Stochastic Models*, 16:273–312, 2000.

[7] N. Akar and K. Sohraby. On Computational Aspects of the Invariant Subspace Approach to Teletraffic Problems and Comparisons. Technical report, Univ. Missouri-Kansas City, 1995.

[8] N. Akar and K. Sohraby. An Invariant Subspace Approach in M/G/1 and G/M/1 Type Markov Chains. *Communications in Statistics-Stochastic Models*, 13:381–416, 1997.

[9] N. Akar and K. Sohraby. Finite and Infinite QBD Chains: A Simple and Unifying Algorithmic Approach. In *Proceedings of IEEE INFOCOM*, pages 1105–1113, 1997.

[10] N. Akar and K. Sohraby. Matrix-geometric Solutions in M/G/1 type Markov Chains with Multiple Boundaries: A Generalized State-space Approach. *International Teletraffic Congress*, March, 1997.

[11] F.-R. Bartsch and E. Auer. Lessons Learned from Multimedia Field Trials in Germany. *IEEE Communications Magazine*, 35:40–45, October 1997.

[12] D. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17:906–926, 1996.

[13] D. Bini and B. Meini. Improved Cyclic Reduction for Solving Queueing Problems. *Numerical Algorithms*, 15:57–74, 1997.

[14] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC1633, June 1994.

[15] J. Cauchy. Atm: The strategic telecommunications service for global corporations. http://www.global-one.net/en/press/atm_article.html.

[16] R. Chakka. *Performance and Reliability Modelling of Computing System Using Spectral Expansion*. PhD thesis, University of Newcastle upon Tyne, 1995.

[17] R. Chakka and P. G. Harrison. The MMCPP/GE/c Queue. *Queueing Systems*, 38(3):307–326, 2001.

[18] M. Chien and Y. Oruc. High performance concentrators and superconcentrators using multiplexing schemes. *IEEE Transactions on Communications*, 42:3045–3050, 1994.

[19] G. Ciardo and E. Smirni. ETAQA: An Efficient Technique for the Analysis of QBD-processes by Aggregation. *Performance Evaluation*, 36-37:71–93, 1999.

[20] J. Daigle. *Queueing Theory for Telecommunications*. Addison Wesley, 1992.

[21] T. V. Do, G. Kalvach, **H. T. Tran**, T. T. Nguyen, and B. Varga. Topology Optimization of ATM Networks for LAN/MAN Interworking in the Top of an SDH Infrastructure. In *Proceedings of the 8th International Network Planning Symposium, Networks'98*, pages 141–146, Sorrento, Italy, 1998.

[22] T. V. Do and T. T. Nguyen. New Algorithm for Mesh Network Optimisation. In *Proceedings of the 8th IFIP Workshop on Performance Modelling and Evaluation of ATM&IP Networks*, pages 20/1–12, Ilkley, UK, 2000.

[23] T. V. Do, T. T. Nguyen, **H. T. Tran**, G. Kalvach, and B. Varga. Topology Optimization of an Overlay ATM Network in an SDH Infrastructure. *Computer Networks*, 34:199–210, 2000.

[24] T. V. Do, T. T. Nguyen, **H. T. Tran**, G. Kalvach, and B. Varga. Topology Optimization of an Overlay ATM Network in an SDH Infrastructure. In *Proceedings of the 6th Workshop on Performance Modelling and Evaluation of ATM Networks*, pages 89/1–15, Ilkley, UK, 1998.

[25] T. V. Do, T. T. Nguyen, **H. T. Tran**, G. I. Rozsa, G. Kalvach, and B. Varga. Practical Approach for the Topology Optimization of an Overlay ATM Network . In *Proceedings of the 4th European Summer School, EUNICE'98*, Munic, Germany, 1998.

[26] S. Blake et al. An Architecture for Differentiated Services. RFC2745, Dec 1998.

[27] R. V. Evans. Geometric Distribution in some Two-dimensional Queueing Systems . *Operations Research*, 15:830–846, 1967.

[28] G. Falin, Z. Khalil, and D. A. Stanford. Performance analysis of a hybrid switching system where voice messages can be queued. *Queueing Systems*, 16:51–65, 1994.

[29] G. Fodor. *Call Level Modeling and Performance Analysis of Some Resources Allocation Techniques in Multiservice Networks*. PhD thesis, Technical University of Budapest, 1998.

[30] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Boyd & Fraser publishing company, 1993.

[31] H. R. Gail, S. L. Hantler, and B. A. Taylor. On a preemptive Markovian queue with multiple servers and two priority classes. *Mathematics of Operation Research*, 17:365–391, 1992.

[32] D. P. Gaver, P. A. Jacobs, and G. Latouche. Finite birth-and-death models in randomly changing environments. *Advanced Applied Probability*, 16:715–731, 1984.

[33] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.

[34] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE JSAC*, 9:968–981, September 1991.

[35] B. Hajek. Birth-and-Death processes on the integers with phases and general boundaries. *Journal of Applied Probability*, 19:488–499, 1982.

[36] B. Haverkort and A. Ost. Steady State Analysis of Infinite Stochastic Petri Nets: A Comparing between the Spectral Expansion and the Matrix Geometric Method. In *Proceedings of the 7th International Workshop on Petri Nets and Performance Models*, pages 335–346, 1997.

[37] **H. T. Tran** and T. V. Do. Comparison of computational methods for QBD processes. In *Proceedings of the 4th International Conference on Applied Informatics*, pages 299–311, September 1999, Hungary.

[38] **H. T. Tran** and T. V. Do. Computational Aspects for Steady State Analysis of QBD Processes. *Periodica Polytechnica, Ser. El. Eng.*, 44:179–200, 2000.

[39] **H. T. Tran** and T. V. Do. Generalised invariant Subspace based Method for Steady State Analysis of QBD-M Proceses. *Periodica Polytechnica, Ser. El. Eng.*, 44:159–178, 2000.

[40] **H. T. Tran** and T. V. Do. An iterative method for queueing systems with batch arrivals and batch departures. In *Proceedings of the 8th IFIP Workshop on Performance Modelling and Evaluation of ATM&IP Networks*, pages 80/1–13, Ilkley, UK, 2000.

[41] **H. T. Tran** and T. V. Do. Application of Finite QBD-M Process to Analysis of a Queue with two Priority Classes. In *Proceedings of the 4th International Workshop on Queueing Networks with Finite Capacity, QNETs*, pages 39/1–12, Ilkley, UK, 2000.

[42] **H. T. Tran** and T. V. Do. A new iterative method for systems with batch arrivals and batch departures. Technical Report 257TD(99)42, September Larnaca, 1999.

[43] **H. T. Tran** and T. V. Do. Comparison of some Numerical Methods for QBD-M Processes via Analysis of an ATM Concentrator. In *Proceedings of 20th IEEE International Performance, Computing and Communications Conference, IPCCC 2001*, pages 253–254, Pheonix, USA, 2001.

[44] **H. T. Tran** and T. V. Do. A new iterative method for systems with batch arrivals and batch departures. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, CNDS'00*, pages 131–137, San Diego, USA, 2000.

[45] **H. T. Tran** and T. V. Do. Generalised invariant Subspace based Method for Steady State Analysis of QBD-M Proceses. In *Proceedings of Conference of PhD Students in Computer Science, CSCS'00*, July Szeged, Hungary, 2000.

[46] **H. T. Tran** and T. V. Do. An Analytical Model for Multipath Routing Schemes. In *Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunications Systems, SPECTS*, pages 170–178, USA, San diego, 2002.

[47] L. Jereb, T. V. Do, and G. I. Rózsa. Flexible planning of telecommunications networks based on layered network model. In *Proceedings of the 6th International Conference on Telecommunication Systems, Modelling and Analysis, Nashville USA*, 1998.

[48] A. E. Kamal. Efficient Solution of Multiple Server Queues with Application to the Modeling of ATM Concentrator. In *Proceedings of IEEE INFOCOM*, pages 248–254, 1996.

[49] A. Kershenbaum. *Telecommunications Networks Design Algorithms*. McGraw-Hill, 1993.

[50] A. Kershenbaum, P. Kermani, and G. Grover. MENTOR: An Algorithm for Mesh Network Topological Optimization and Routing. *IEEE Transactions on Communications*, 39:503–513, 1991.

[51] D. D. Kouvatsos. MEM for Arbitrary Queueing Networks with Multiple General Servers and Repetitive-Service Blocking. *Performance Evaluation*, 10:169–195, 1989.

[52] U. R. Krieger, V. Naoumov, and D. Wagner. Analysis of a Finite FIFO Buffer in an Advanced Packet-Switched Network. *IEICE Trans. Commun.*, E81-B:937–947, 1998.

[53] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for quasi-birth-death processes. *Journal of Applied Probability*, 30:650–674, 1993.

[54] B. Meini. Solving QBD problems: the cyclic reduction algorithm versus the invariant subspace method. *Advances in Performance Analysis*, 1:215–225, 1998.

[55] M. Meo, E. de Souza e Silva, and M. Ajmone Marsan. A method for calculating successive approximate solutions for a class of block banded M/G/1 type Markovian models. *Performance Evaluation*, 44:97–119, 2001.

[56] I. Mitrani and R. Chakka. Spectral Expansion solution of a class of markov models: Application and comparison with the matrix-geometric method. *Performance Evaluation*, 23:241–260, 1995.

[57] V. Naoumov, U. R. Krieger, and D. Warner. Analysis of a Multi-Server Delay-Loss System With a General Markovian Arrival Process. In S. R. Chakravarthy and A. S. Alfa, editors, *Matrix-analytic methods in stochastic models*, volume 183 of *Lecture Notes in Pure and Applied Mathematics*. Marcel Dekker, September 1996.

[58] K. Natarajan. On the G/GE/1 Queue. Technical report, Universität Trier Mathematic/Informatik, 1993.

[59] R. Nelson. *Probability, stochastic processes and queueing theory*. Springer-Verlag, New York, 1995.

[60] M. F. Neuts. *Matrix Geometric Solutions in Stochastic Model*. Johns Hopkins University Press, Baltimore, 1981.

[61] A. Ost and B. R. Haverkort. Evaluating Computer-Communication Systems using Infinite-State Stochastic Petri Nets. In *Proceedings of 3rd International Conference on Matrix Analytic Method*, pages 295–314, 2000.

[62] G. Paksy. SDH Transport Network Architecture. *Hungarian Telecommunications, Selected Papers in English from the Journal*, 1997.

[63] V. De Nitto Persone and V. Grassi. Solution of Finite QBD Processes. *Applied Probability*, 33:1003–1010, 1996.

[64] R. N. Queija. Steady State Analysis of a Queue with varying Service Rate. Technical Report PNA-R9712, CWI, 1997.

[65] R. N. Queija and O. J. Boxma. Analysis of a Multi-Server Queueing Model of ABR. Technical Report BS-R9613, CWI, 1996.

[66] T.G. Robertazzi. *Computer Networks and Systems: Queuing Theory and Performance Evaluation*. 2nd Edition, Springer-Verlag, 1994.

[67] S. Sahu, D. Towsley, and J. Kurose. A Quantitative Study of Differentiated Services for the Internet. Technical Report CMPSCI Technical Report, UMass, 09 1999.

[68] K. Sohraby and J. Zhang. Spectral decomposition approach for transient analysis of multi-server discrete-time queues. *Performance Evaluation*, 21:131–150, 1994.

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

[70] I. Stoica and H. Zhang. Providing Guaranteed Services without per Flow Management. In *Proceedings of SIGCOMM*, pages 81–94, 1999.

[71] C. Villamizar. MPLS Optimized Multipath. Internet-draft, Feb. 1999.

[72] V. L. Wallace. *The Solution of Quasi Birth and Death Processes Arising from multiple Access Computer Systems*. PhD thesis, University of Michigan, 1969.

[73] Q. Wang and V. S. Frost. A new solution for discrete queueing analysis of ATM systems. In *Proceedings of GLOBECOM*, pages 358–364, 1991.

[74] G. Wolfner. *ATM forgalomformázás teljesítőképességi vizsgálata*. PhD thesis, Technical University of Budapest, 1997.

[75] G. Wolfner and M. Telek. Numerical analysis of queues with batch arrivals. *Performance Evaluation*, 41:179–194, 2000.

[76] K. Wuyts, B. Van Houdt, R. K. Boel, and C. Blondia. Matrix Geometric Analysis of Discrete Time Queues with Batch Arrivals and Batch Departures with Applications to B-ISDN. In *Proceedings of the 16th International Teletrffic Congress ITC-16*, 1999.

[77] X. Xiao and L. M. Ni. Internet QoS: the Big Picture. *IEEE Network*, 13:1–13, 1999.

[78] J. Ye and S. Q. Li. Analysis of Multimedia Traffic Queues with Finite Buffer and Overload Control, Part II: Applications. In *Proceedings of IEEE INFOCOM*, pages 848–859, May 1992.

[79] J. Ye and S. Q. Li. Folding Algorithm: A Computational Method for Finite QBD Processes with Level-Dependent Transitions. *IEEE Trans. Commun.*, 42:625–639, February 1994.