Quality of Service Performance Analysis based on Network Calculus

Vom Fachbereich 18
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

Dipl.-Ing. Krishna Pandit
geboren am 14. 3. 1978 in Berlin

Erstreferent: Prof. Dr.-Ing. Ralf Steinmetz
Korreferent: Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn
Tag der Einreichung: 25. 4. 2006
Tag der Disputation: 14. 7. 2006

D 17
Darmstädter Dissertation

Foreword

I am happy that the day has come when I can write these lines. This dissertation is about packets traversing communication networks. I am blessed to be surrounded by a great network and this is my stage to send out some packets (read: messages).

To Prof. Ralf Steinmetz: Thank you for giving me the opportunity to pursue my PhD and for creating an environment in which I could grow professionally and personally. It is an honor to work with you.

To Prof. Paul J. Kühn: Thank you for taking on the role of co-adviser.

To my wife Shruthi: Thank you for your love, support and understanding. You are the best. I love you.

To my parents: Thank you for always being there for me and your unfailing encouragement.

To my brother: Thank you for being you.

To Andrew Brzezinski, Dr. Andreas Faatz, Dr. Markus Fidler, Dr. Holger Jäkel, Claus Kirchner, Prof. Jens Schmitt: Thank you for the comments on the manuscript.

To my students Julian Eckert, Ian Hubbertz, Claus Kirchner, Thomas Pfeiffer and Matthias Priebe: Thank you for entrusting the supervision of your theses to me. It was great working with you.

To my colleagues at KOM: Thank you for the pleasant environment. A special mention goes to Andreas Faatz. It is great when colleagues become friends.

To Prof. Eytan Modiano, Prof. Arogyaswami Paulraj and Prof. Harsha Sirisena: Thank you for hosting me at MIT, Stanford and University of Canterbury. Those were memorable experiences.

I dedicate this work to my beloved grandparents:

- Sundari Isvaran (1913 - 2000)
- V. Isvaran (1908 - 2000)
- Radha Bai Kapoor (1908 - 1996)
- C. Padmanabha Rao (1901 - 2006)

Darmstadt, April 24, 2006

Summary (English)

Data flows belonging to multimedia applications are gaining importance in the Internet. A key characteristic of such data flows is that they require Quality of Service (QoS). An Internet populated with data flows requiring QoS constitutes a paradigm change from the Internet in its early days. This has been accounted for in many research endeavors proposing new architectures, algorithms and protocols. However, one area that has been relatively underexposed is the development of new models for QoS. Hence, the vision that has inspired this dissertation is the development of a unified model for the performance analysis of QoS in the Internet. The potential benefits of such a model can be observed in other fields: Linear system theory is widely used in the analysis of communication and control systems.

In this dissertation, contributions are made towards developing a unified model for the performance analysis of QoS in the Internet. The basis of the work is network calculus. Network calculus is a system theory for deterministic queuing systems, which was developed in the 1990s. The underlying rationale is that deterministic QoS guarantees can be obtained by traffic regulation, deterministic scheduling and admission control. Beyond that, this work integrates elements of system theory and queuing theory. The latter has been the method of choice for modeling data flows in the Internet since its infancy.

The main requirements of the envisioned model are that it should give insight on relevant characteristics, should have a wide range of applicability and should be transparent and easy to use. Recent research results in network calculus which address these requirements are presented. These include statistical network calculus and transforms. Further, some open issues are identified, which are then dealt with in this dissertation.

A network calculus analysis is conducted for dynamically reconfigurable networks. First, the network architecture which can be found in optical networking is presented. The key feature here is that packet forwarding is not only influenced by the routing, but also by the reconfiguration. It is shown how service curves can be determined for different reconfiguration schemes, thus enabling a QoS analysis. On a more general footing, in this chapter it is illustrated how current networking research issues can be translated into network calculus models.

The next contribution is the development of a new transform for network calculus and its application. With the new transform the min-plus convolution, which is an important operation in network calculus, obtains a graphical interpretation and thus becomes easier to use. Based on the transform, theorems on the computation of the min-plus convolution are set up. These theorems are then applied to network design using service curves, with an emphasis on bandwidth/delay decoupled scheduling.

Furthermore, network calculus and queuing theory are brought together. While network calculus focuses on the worst case analysis, queuing theory deals mainly with average behavior. It is examined whether the best of both worlds can be combined to achieve better models. First, analytical approaches are presented, which are then followed by a simulation.

Finally, the achieved progress is summarized and some conclusions drawn.

Summary (German)

Datenströme, welche auf Multimediaanwendungen zurückzuführen sind, gewinnen im Internet zunehmend an Bedeutung. Eine wichtige Eigenschaft solcher Datenströme ist, dass sie eine gewisse Dienstgüte erfordern. Dem wurde in der Forschung bereits Rechnung getragen. Ein Aspekt der hierbei jedoch vernachlässigt wurde, ist die Entwicklung geeigneter Methoden zur Modellierung, die Analyse und Entwurf von Kommunikationsnetzen mit Dienstgüte erfordernden Datenströmen erleichtern. Daher ist die Vision hinter dieser Dissertation die Entwicklung eines universellen Modells für Dienstgüte im Internet. Der Nutzen eines solchen Modells wird bei der Betrachtung anderer Gebiete ersichtlich: das lineare, zeitinvariante Modell im Rahmen der klassischen Systemtheorie ist weit verbreitet bei der Analyse von Fragestellungen der Nachrichten- und Regelungstechnik.

In dieser Dissertation werden einige Beiträge zur Entwicklung eines solchen universellen Modells gegeben. Grundstein ist hierbei der Netzwerkkalkül (engl. Network Calculus). Der Netzwerkkalkül ist eine Systemtheorie für deterministische Warteschlangen und wurde in den 1990er Jahren entwickelt. Die zu Grunde liegende Idee ist, dass deterministische Dienstgütegarantien durch Verkehrsregulierung, Scheduling und Zugangskontrolle gegeben werden können. Des Weiteren baut diese Arbeit auf Elemente der System- und Warteschlangentheorie, welche seit jeher bei der Modellierung von Datenströmen im Internet eingesetzt wird.

Die Hauptmerkmale des angestrebten Modells sind, dass sie Einblick in die relevanten Charakteristiken bieten, möglichst vielfältig anwendbar sind und einfach zu verwenden sind. Die neueren Entwicklungen des Netzwerkkalküls, welche diese Punkte adressieren, sind erläutert. Hierzu zählen unter anderem der stochastische Netzwerkkalkül und Transformationen.

Eine Analyse im Rahmen des Netzwerkkalküls ist für dynamisch rekonfigurierbare Netze durchgeführt. Zunächst wird die Netzwerkarchitektur, welche bei optischen Netzen vorzufinden ist, erläutert. Die entscheidende Eigenschaft solcher Netze ist, dass die Paketweiterleitung nicht nur durch die Wegefindung, sondern auch durch die Rekonfigurationen beeinflusst wird. Es wird gezeigt, wie verschiedene Rekonfigurationsverfahren als Funktionen des Netzwerkkalküls beschrieben werden können, welche eine Dienstgüteanalyse ermöglichen.

Der nächste Beitrag ist die Entwicklung und Bereitstellung einer Transformation für den Netzwerkkalkül. Mit Hilfe der Transformation kann die Mini-Plus Faltung, welche eine wichtige Operation des Netzwerkkalküls ist, graphisch dargestellt und intuitiv veranschaulicht werden. Basierend auf dieser Transformation werden Sätze hergeleitet, welche die Berechnung der Mini-Plus Faltung erleichtern. Diese Sätze werden auf den Netzwerkentwurf durch Dienstkurven angewendet, wobei das Hauptaugenmerk auf Schedulern liegt, bei denen die Bandbreite und Verzögerung entkoppelt sind.

Des Weiteren werden Netzwerkkalkül und Wartenschlangentheorie zusammengebracht. Während der Netzwerkkalkül die Berechnung des ungünstigsten Falles ermöglicht, behandelt die Warteschlangentheorie meist Durchschnittsverhalten. Es wird untersucht, ob durch eine Kombination die Vorteile beider Modelle ausgeschöpft werden können.

Abschliessend werden die erzielten Ergebnisse und einige Konklusionen dargestellt.

Contents

Foreword iii

Summary (English) v

Summary (German) vii

1 Introduction 1
1.1 Motivation 1
1.2 History 1
1.3 Vision and Goals 2
1.4 Approach 3
1.5 Contribution and Outline 4

2 Background 7
2.1 Fundamentals of Communication Networks 7
2.2 Quality of Service in IP Networks 10
2.2.1 Integrated Services 11
2.2.2 Differentiated Services 11
2.2.3 Lightweight Approaches 12
2.3 System Theory 12
2.3.1 Linear Time-invariant Systems 14
2.3.2 Transforms 14
2.4 Queuing Theory 15
2.4.1 Poisson Process 16
2.4.2 M/M/1 and M/M/1/N Queue 17
2.4.3 Queuing Theory and QoS 18
2.5 Network Calculus 18
2.5.1 Min-plus Algebra 19
2.5.2 Arrival Curve Concept 21
2.5.3 Service Curve Concept 25
2.5.4 Selected Results 27

3 Building Blocks for QoS Performance Analysis 31
3.1 Requirements of a Model 31
3.2 Insight on Relevant Characteristics 33
3.2.1 Packet Loss 33
3.2.2 Stochastic Extension to Network Calculus 33
3.2.3 Network Calculus and Queuing Theory 34

3.3 Range of Applicability . . . 35
3.3.1 Aggregation of Flows . . . 35
3.3.2 Feedback Systems . . . 36
3.3.3 General Topologies . . . 36
3.3.4 New Network Elements . . . 36
3.3.5 Application to Specific Traffic Types . . . 37
3.3.6 New Network Paradigms . . . 37
3.4 Transparence and Ease of Applicability . . . 37
3.4.1 Transforms . . . 38

4 Modeling Dynamically Reconfigurable Networks with Network Calculus 41
4.1 Reconfigurable Networks . . . 42
4.1.1 Optical Networking . . . 42
4.1.2 Reconfigurable network model . . . 43
4.2 Constructing Time Division Multiplexing (TDM) Schemes . . . 47
4.2.1 Algorithm . . . 47
4.2.2 Numerical Example . . . 48
4.3 Service Curve Representation of Switching Schemes . . . 49
4.4 QoS Trade-offs in Switching Schemes . . . 50
4.5 Comparison with Average Delay Results . . . 52
4.5.1 The Simulation Setup for the Average Delay . . . 53
4.5.2 Worst Case Result . . . 54
4.6 Recipe for Application . . . 56
4.7 Summary . . . 57

5 A Transform for Network Calculus and its Application 59
5.1 Motivation . . . 59
5.2 A Transform for Network Calculus . . . 59
5.2.1 The Fenchel Transform . . . 59
5.2.2 The Continuous $\mathfrak{P}$-transform . . . 60
5.3 Min-plus Convolution in the Context of Network Calculus . . . 65
5.4 Optimal Network Service Curve . . . 75
5.5 Allocation of Service Curves along a Path . . . 76
5.5.1 Determining Optimal Node Service Curves . . . 76
5.5.2 Numerical Example . . . 77
5.6 Reallocation of Service Curves in Nodes . . . 78
5.6.1 Local Reallocation . . . 78
5.6.2 Global Reallocation . . . 79
5.7 Recipe for Application . . . 80
5.7.1 Min-plus Convolution . . . 80
5.7.2 Service Curve Based Admission Control . . . 81
5.7.3 Allocating Service Curves in Nodes . . . 82
5.8 Summary . . . 83

6 Network Calculus Assisted Queueing Theory **85**
6.1 System Model 86
6.1.1 Shaper 86
6.1.2 Service Curve Enforcer 87
6.2 Analytical approaches 88
6.2.1 Brute Force Method 88
6.2.2 Random Walk Method 89
6.3 Simulations 91
6.3.1 Methodology 91
6.3.2 Results 92
6.4 Recipe for Application 95
6.5 Summary 95

7 Conclusions **99**

Bibliography **100**

A Rebalance Theorem **111**

List of Tables **113**

List of Figures **115**

List of Acronyms **117**

List of Symbols **119**

Curriculum Vitae **123**

1 Introduction

1.1 Motivation

In today's Internet, the share of data flows belonging to multimedia applications [104] is continuously increasing. The key characteristic of multimedia applications is that they have stringent requirements on the throughput, loss and jitter [103]. These parameters, among others, are called Quality of Service (QoS) parameters. We adopt the definition of QoS as the "well-defined, controllable behavior of a system with respect to quantitative parameters" [92]. Examples of QoS-sensitive multimedia applications are Video-on-demand [113] and Voice over IP (VoIP) [2]. VoIP flows require their packets to have an end-to-end delay in the magnitude of milliseconds. This is due to the fact the VoIP systems are competing with the traditional telephone system, and will only be accepted by users if the quality is matched. One advantage of VoIP is that a multitude of services can be efficiently implemented [39]. Furthermore, it is cost effective to integrate many services into one network as opposed to let several networks coexist. Pursuing this thought to the end, this opens the vision of having just one network to cater to all communication: telephone, data, TV, radio, etc. From an economical perspective, the integration of all networks into one would certainly be beneficial. Even if all networks continue to coexist for a long time, the Internet is gaining market share of them. Skype is used instead of the phone, and radio and TV stations broadcast over the Internet. Therefore, the importance of QoS in the Internet will increase in future. Due to its history, which is reviewed in the next section, the transition of the Internet to an integrated network providing a specified QoS is not trivial. However, effective models and mathematical tools which would expedite such a transition are yet to be developed in the field of computer networking [37]. This also holds true for the performance analysis of QoS in the Internet, as to date no widely used model exists. The benefits of a generic model can be observed in related fields. In communication engineering and control systems a variety of problems of analysis and design are solved using linear system theory.

1.2 History

In 1969, University of California Los Angeles, Stanford University, University of California Santa Barbara and University of Utah were connected to a four node packet switched network. This network, the Arpanet, produced the legacy protocols Transmission Control Protocol (TCP) and Internet Protocol (IP) and grew to become what we know as the Internet. The model on which the Arpanet is founded is queuing theory [62]. The first applications of the Arpanet and early Internet were Email and File Transfer Protocol (FTP). Of course, such applications have radically different requirements on the network than today's multimedia traffic. They do not have stringent QoS requirements, as it does not matter whether their packets reach the destination 30 seconds sooner or later. In this spirit, i.e., without giving throughput or delay guarantees, the protocols of the Internet were designed. A prominent example is TCP, which ensures that each user gets a fair share of the bandwidth, but this can be arbitrarily low when many users are contending for the link.

However, the notion of QoS and even that of the integrated network haunted the minds of researchers. A milestone was the Integrated Services (IntServ) architecture from the Internet Engineering Task Force (IETF), in the early 1990s. The heart of IntServ is its Guaranteed Service, which allows flows to exactly specify their required bandwidth and delay. However, this requires each node of the path to keep the state of each flow. The rapid growth of the Internet, which was powered by the emergence of the World Wide Web (WWW) to its killer application, made this unfeasible and prevented IntServ from being widely deployed. IntServ and Asynchronous Transfer Mode (ATM), which was popular in those days, gave birth to network calculus, a theory for deterministic queuing systems [23, 24, 13]. It was developed and advanced in the 1990s and is therefore a fairly young theory. The spirit of Guaranteed Service is clearly visible in network calculus.

The next proposed architecture was the Differentiated Services (DiffServ) architecture, also from the IETF, in the late 1990s. DiffServ is a rather loosely defined toolbox of methods to assign different priorities for flows. It is deployed, however, the lacking of strictly specified services leads to heterogeneity among Internet Service Providers. This was the last major attempt for a unified framework for QoS in the Internet.

As described above, in today's Internet diverse multimedia applications requiring QoS are imminent. Therefore, the expectation of the Internet is to enable such applications despite it clearly being intended for something else. However, QoS is not the only issue in this endeavor. There is the administration of the network, which consists of issues such as pricing, accounting and the interconnection of Internet Service Providers [45]. Further, there is the wide area of security issues [99], which also have some points of contact with QoS [89]. These issues lie outside the scope of this thesis and its focus is on QoS.

1.3 Vision and Goals

The vision that stimulated this dissertation is a unified model for QoS in packet switched networks. This model is to be the basis of a theory akin to linear system theory for communication engineering and control systems. Underlying the model is the paradigm of a system characterised by an input and an output. An input function is then mapped to an output function according to a mapping rule. In the simplest case, this mapping rule is specified by means of a look-up table. Of course this is not feasible as the number of possible input functions can be very large. It is beneficial, if a simple operator can be found which specifies the mapping rule. In classical linear system theory, this operator is the convolution with the impulse response. This is best illustrated by an example from communication engineering. Assume the input function to be the signal at one end of a wireless channel. We are now interested in the signal that arrives at the other end of the wireless channel. The brute force method to obtain it would be to apply every possible input signal to the channel and record the corresponding output signals in a look-up table. However, if the impulse response of the channel is known, the output signal can be obtained by convolving the input signal with the impulse response. Another useful property of system theory is that the system can be broken down into partial systems. In this way complex systems can be broken down into tractable subsystems. In the other direction, subsystems can be concatenated to model complex systems.

Applied to networks, the following system model is obtained. The input function corresponds to a data flow that an edge node desires to send. An intuitive representation is the number of packets the edge node sends at each instant of time. The system is the network. The output function is the data flow arriving at the receiver, from which throughput, delay, and other QoS parameters can be extracted. The subsystems of the network are the nodes along the path. Each node itself can be treated as a system whose input is the output of the preceding node and whose output is the input of the succeeding node. By this means, complex networks can be depicted as the concatenation of tractable network elements, provided the elements do not affect each other mutually.

We believe that a unified model for performance analysis of packet switched networks comprises methods from the three aforementioned theories, queuing theory, system theory and network calculus. Of course, due to the heterogeneity of the Internet, setting up such a unified model is a major task. It will take many years for the model to reach a level of influence akin to linear system theory in relation to communication or control systems. Therefore, the goal now is to work towards setting up the unified model by developing suitable modeling techniques and illustrating their applications. This will ease the process of developing mechanisms to provide QoS in the Internet. Finally, the results are to be obtained and presented in such a way that practising engineers can use them without having to acquire an in-depth knowledge for setting up the model.

1.4 Approach

The goals are achieved by bringing together the best of existing theories as well as extending them. In this thesis, we present the state-of-the-art of this endeavor and contribute to it. Aspects of linear system theory are considered as it is the method of choice in related fields.

Queuing theory is also utilized, as it has been successfully employed as the method of choice for modeling packet switched networks for many years. However, there are two reasons why it is not optimally suited for QoS networks. First, the Poisson process, which is the foundation of queuing theory, does not always describe the multimedia traffic flows adequately. Secondly, queuing theory mainly deals with the average behavior of a queuing system, but for QoS guarantees, many times the worst case is required. Both drawbacks can be overcome by suitably applying queuing theory, but this requires in-depth knowledge and therefore hinders its widespread use.

The underlying paradigm of network calculus is that traffic regulation, scheduling and admission control yield deterministic guarantees. It has similarities with classical system theory. The input, called the arrival curve, denotes the data that the sender will send in the worst case. The impulse response, called the service curve, usually denotes the minimum service that a network element will offer. The output is then the data that leaves the network element. Since IntServ and ATM are not in the research focus anymore, it is a common criticism that network calculus has a too limited range of applicability. However, since network calculus allows for a worst case performance analysis, it is suited for modeling QoS.

The calculations of network calculus are carried out in min-plus algebra. The reason for switching to min-plus algebra is illustrated by the following line of thought. Consider a node with a given capacity. The rate at which packets depart from the node, is always the minimum of the maximum capacity and the rate at which the node wants to send, which depends on the data that enters the node as well as the data in the buffer. The min-plus convolution, which connects the input, impulse response and output plays a fundamental role. This is another factor preventing the wide spread use of network calculus as min-plus algebra is commonly considered unintuitive and tedious. Despite its drawbacks, network calculus is the foundation of the unified model.

Theoretical models are not the only approach to analyze the performance of packet switched networks. Alternative approaches are simulations and experiments [53, 46]. Due to the lack of a theoretical model, currently simulation is mostly the method of choice when analyzing the performance of networks. While being well-suited for the task in many cases, simulations also have severe drawbacks. Setting up the simulation can be tedious. A large number of input parameters in complex systems often leads to combinatorial explosion. Further, changing some parameters can require repetition of an entire simulation.

1.5 Contribution and Outline

In Chapter 2, the background of this dissertation is presented. This comprises the fundamentals of computer networks and QoS, as well as the underlying theories, which are system theory, queuing theory and network calculus. The emphasis is on network calculus, as it is the cornerstone of the following chapters. Chapter 3 consists of the building blocks for the performance analysis of QoS in the Internet. Among the topics covered in this chapter, three areas which are relatively underexposed were identified. Novel results in these areas are derived in the following chapters. Chapter 4 deals with modeling new network architectures in the framework of network calculus. Its applicability is increased by pointing out how the performance of a dynamically reconfigurable network can be analyzed by translating it into a network calculus system. To our knowledge this is unprecedented. In Chapter 5 the min-plus convolution, the network calculus counterpart of the convolution in linear system theory, is studied. A novel transform for it is developed which increases the lucidity of that operation. Beyond that, theorems which simplify the calculation of the min-plus convolution are derived. In Chapter 6 the question is examined whether network calculus and queuing theory can be combined in such a way that the best of both worlds is obtained. To our knowledge this is the first study of its kind. Finally, in Chapter 7 some concluding remarks are made.

2 Background

Communication networks are motivated by the desire to share information over distances. This manifests itself throughout all its incarnations. A few centuries ago, a popular way to communicate over long distances was to sit on top of a hill and send smoke signals. To take a more modern example, the goal of the telephone network is to enable talking to someone who is outside the reach of the natural sonic signal. The radio network also transmits acoustic signals over large distances, the difference to the telephone being that a broadcast takes place. Similarly, the goal of the television network is to enable seeing something which is not in direct sight. And the latest addition to influential networks has been the Internet, which deals with the transport of a multitude of data types.

2.1 Fundamentals of Communication Networks

A key characteristic that all examples above have in common is that the sender and the recipient of the information are too far apart to be communicating directly. They rely on intermediate entities to pass on the information. In the case of the smoke signals, the intermediate entity is another person. For the telephone, radio and television network, as well as the the Internet, the intermediate entity can be abstracted as computers. The intermediate entity can have several neighbors from whom information comes in and to whom it goes out. Hence, an additional task of the intermediate entity is to assign each incoming information the appropriate neighbor to pass it on to. This is referred to as *switching*. Due to this nature, the intuitive visual representation of a network is a graph with *nodes* and *links*. A node denotes an entity that processes information and the links denote a communication channel between two nodes. Communication channel here means a medium over which a signal is transmitted. There are several types: smoke over air, electrical signal over copper wire, wave over air or light over optical fiber. The network model is depicted in Figure 2.1. Nodes belonging to the same group are called a *domain*. An example network is given in Figure 2.1. A cloud around nodes indicates a domain.

Leaving the smoke signals behind, the field is demarcated as computer networking. Excellent textbooks dealing with computer networking are the books by Kurose and Ross [68], Tanenbaum [107], Patterson and Davie [86] and Kumar et al. [66].

The set of nodes that the information passes from the sender to the receiver is called the *path*. One possibility is to establish the path at the beginning of a session. A *session* is the time in which a complete communication task is performed. Establishing a

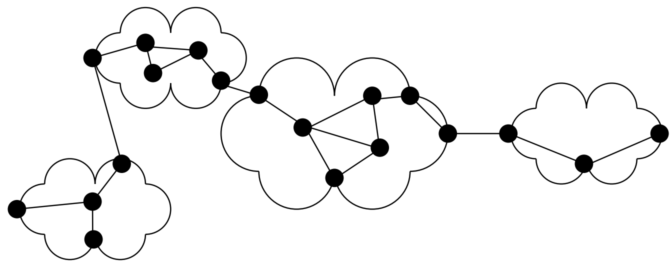

Figure 2.1: Network

fixed path for the communication is known as *circuit switching*. In a circuit switched connection, the intermediate nodes set up a physical connection between the incoming and outgoing link. Therefore, a physical connection between the sender and receiver is set up. The advantage of this set up is that it is well-defined as to which resources are allocated to the connection. Circuit switching is the method of choice for the telephone network, also known as Plain Old Telephone System (POTS). The bandwidth allocated to a connection is high enough to have an acceptable quality for the conversation. The disadvantage of circuit switching is that the resource allocation is not very dynamic. In order to set up the connection it has to be checked on each link along the path whether sufficient resources are available, which takes time.

An alternative referred to as *packet switching* is to organize the information as data packets. A *packet* is a sequence of bits which carries the data, as well as additional information such as its destination address and protocol information. A *flow* is a set of associated packets. While in circuit switching all information takes the same path, this need not be the case for packet switching. There it is left open whether all packets take the same path or not. The process of finding a path of a packet to its destination is called *routing*. In a packet switched network, the task of an intermediate node is to collect the data packets from the incoming links and put them onto the corresponding outgoing links. This includes two tasks, running a routing algorithm to determine the outgoing link and actually forwarding the packet on it. Therefore, each packet must contain the source and destination address and find its path through the network independently of other packets. Each node has a *buffer*, where the packets that arrive and can not be served immediately are stored. The advantage of packet switching is that resources can be allocated dynamically. If one flow pauses and does not send packets for a while, this can be utilized immediately by all other flows sending more packets. This is a useful property for data traffic as it is *bursty*. Bursty means that the ratio of the peak rate to the average rate is high. The burstiness of data traffic is higher than that of telephone traffic. The disadvantage of packet switching is that the resource allocated to each flow can not be clearly determined, as it depends on the behavior of other flows. One flow sending enough packets to fill the buffers along the path can cause other flows to lose packets even if they send at a minimal rate. The focus of this dissertation is to develop analyzing tools for packet switched networks, which help to overcome this weakness.

A further type of switching are so-called virtual circuits. The data is organized in packets, however the connection is established before the actual data is sent. Therefore, a packet does not have to contain the destination address, but an identifier stating which virtual circuit it belongs to is sufficient. An example where virtual circuits are used is ATM.

Generally, if a connection is established before the actual data is transported, the communication is called *connection-oriented*. If this is not the case, it is called *connectionless*. An excellent up-to-date work on connection-oriented and connectionless mechanisms in the Internet is the dissertation by Lorang (in German) [75].

Up to now we have tacitly assumed that the neighboring nodes speak the same language. In technical terms this is referred to as a *protocol*. The fact that different, possibly independent, entities are communicating with each other can make this issue rather tricky. Naturally, each entity has its own idea as to what the best protocol is, especially when they are controlled by companies which are trying to market their proprietary protocols. In order to get a grip on this, the International Standards Organisation (ISO) early set a reference model, the Open Systems Interconnection (OSI) model [38], which governs the exchange of information among computers. The ISO-OSI model is organized in 7 layers. The lowest layer is the physical layer, which deals with the actual transmission of bits over a link. The second layer is the data link layer, which is responsible for a reliable communication between adjacent stations. Redundancy is added to the bit streams for error correction and flow control is performed. When the ISO-OSI model was introduced, point-to-point communication was dominant. A new paradigm of several hosts sharing a medium, such as the Ethernet protocol [76] then rose, which made a new layer necessary: the Medium Access Control (MAC) layer. The MAC layer is considered as part of the data link layer with its main task being contention algorithms for shared mediums. The basic problem hereby is that there are several nodes which desire to send but the medium does not have the capacity to allow all to send at the same time. Sharing a medium is referred to as multiplexing. In the following the main multiplexing techniques are described. In Time Division Multiplexing (TDM) the channel is shared such that the senders get different time slots in which they can send. In Frequency Division Multiplexing (FDM) they all send at the same time, but each with a different frequency. A modification is Wavelength Division Multiplexing (WDM), which is employed in optical networks and the flows are separated by using different wavelengths of the light signal. This is described in more depth in Chapter 4. For the sake of completeness, Code Division Multiplexing (CDM) and Space Division Multiplexing (SDM) are also mentioned but not elaborated further as they have no relevance for this dissertation.

The next layer is the network layer. Its task is to ensure the packet delivery from one end system to another end system. This includes among others routing and congestion control. The dominating protocol on this layer is IP. The only service that the network layer in the Internet offers is *best effort*. That means, anyone can inject packets at anytime, and the network tries to deliver them as efficiently as possible. The network layer will be the layer around which this dissertation evolves.

The transport layer takes care of the communication between processes on the end systems. Its most prominent protocol is TCP. TCP was introduced when the Arpanet became congested. Again, in those times the Internet in its current form was not anticipated. The idea of TCP is to give each flow a fair share of the bandwidth. Upon arrival of a packet, the receiver sends an acknowledgment to the sender. Then the sender additively increases the sending rate whenever an acknowledgment arrives. If a packet is not acknowledged within a certain time, the sender multiplicatively decreases the rate. Therefore, the increase is always slower than the decrease, which ensures an equilibrium. The session layer, which is responsible for administering sessions and the presentation layer, which is responsible for the data formats, are not relevant in the present consideration. Finally, the application layer is home to the applications and the interface to the user.

One questions that naturally arises is the effectivity of maintaining four networks, namely the Internet, radio, television and telephone network, simultaneously. Its answer has a political and a technical dimension. While the political one is out of scope of this dissertation, the focus is set on the technical aspects. Bringing everything together in the radio or television network is not possible because those lack up-link channels. The telephone network is also not well suited for two reasons. Being a circuit switched network it is not suited for bursty traffic. Further, its rather complex connection establishment is unfeasible for broadcasts. Therefore, if all networks are to be migrated into one, only the Internet is left. The IP protocol allows the flexibility to do so. As a matter of fact, this process is in the midst of happening, with VoIP gaining share and the number of Internet based television and radio stations increasing. A consequence from this development is that the Internet has to evolve to a network that offers QoS.

2.2 Quality of Service in IP Networks

In this section, the focus is on the criteria which decide whether the service offered by the network meets the quality demands of the user. In packet switched network such as the Internet, this is referred to as *Quality of Service (QoS)*. In this dissertation, the QoS definition of Schmitt [92] is used. "QoS is the well-defined and controllable behavior of a system with respect to quantifiable parameters". The typical parameters for QoS are throughput, delay, loss and jitter. The *throughput* is given in amount of data per time unit and is an indication of the amount of information that can be transported across the network. The *delay* denotes the time that a packet takes from an origin to a destination node. If the origin node is the sender, and the destination node the receiver, then one talks about end-to-end delay. The *loss* denotes the packets that are dropped along the path. Depending on the application, the consequences of packet loss can be more or less severe. The *jitter* denotes the variation in delay. However, with buffering in the receiver, the jitter usually can be combated.

The early applications of the Internet were not very sensitive to QoS. For instance, it does not matter whether an email takes 3 or 30 seconds to arrive. However, multimedia applications, such as VoIP have very stringent QoS requirements. And if the Internet is to replace the radio, television and telephone networks, a sophisticated solution for QoS is required.

Providing QoS in packet switched networks relies on three conditions. The first is traffic regulation. One should be able to control how much a sender is allowed to send. Secondly, scheduling is important. The behavior of the intermediate nodes in terms of how many packets they can store and process has to be known. The final point is admission control. The number of flows allowed into the network has to be limited. If one or more of these conditions are not fulfilled, reliable QoS guarantees can not be given. In the following the most important QoS mechanisms are introduced. An elaborate survey of up-to-date QoS mechanisms is presented by Firoiu et al. [36].

2.2.1 Integrated Services

The IntServ architecture was developed in the early 1990s and became an IETF standard in 1994 [14]. This was a time in which the tremendous growth of the Internet was not foreseeable. IntServ comprises two types of service, the Guaranteed Service [100] and the Controlled Load Service [111]. The Guaranteed Service is a strictly defined class. Each flow is described by a set of parameters, the so-called Traffic Specification (TSpec). These are the peak rate, sustained rate, maximum packet size and burst size. For each flow the resources are reserved at each node of its path through the entire network. Therefore, it is a connection-oriented service. As a signaling protocol the Resource ReSerVation Protocol (RSVP) [15] is used. An implementation and performance analysis of RSVP can be found in work led by Karsten [57, 60]. In contrast to the Guaranteed Service, the Controlled Load Service is rather loosely defined as a service which is better than the current best effort. However, IntServ failed to become widely spread in IP networks for several reasons. Firstly, at the time IntServ was developed, the Internet was still rather small in terms of the number of flows. IntServ requires each node to keep track of all its flows, which was still feasible in those times, but became impossible soon due to the increase of Internet traffic. Further, IntServ requires all participants of the network to adhere to its standard, in the sense use the same parameters to describe flows and use RSVP. Given the heterogeneity of the Internet, this is no longer realistic.

2.2.2 Differentiated Services

The DiffServ architecture [9] is a response to IntServ. Instead of a sophisticated per flow end-to-end service like the Guaranteed Service of IntServ, the approach DiffServ takes is to consider traffic classes. This considerably reduces the state which has to be kept in the nodes. A so-called Per Hop Behavior (PHB) is described, which leads to a Per Domain Behavior (PDB). A PHB describes the service that a flow receives in a node. The standardized PHBs are the Expedited Forwarding [48] and Assured Forwarding [51] services. The former corresponds to the Guaranteed Service, while the latter corresponds to Controlled Load Service. The rationale is that if the PHBs of each node in a domain is known, then its PDB can be obtained. The absence of strictly defined service classes has let DiffServ become a toolbox with a vast amount of incarnations. Today, many Internet Service Providers (ISPs) use DiffServ, each in their own way.

A further characteristic of DiffServ is that it allows for relative QoS. An up-to-date overview of relative QoS is given in the dissertation by Bodamer [10] (in German).

2.2.3 Lightweight Approaches

Beyond the two major IETF approaches there are numerous ideas to provide QoS in a not complex manner. Hurley et al. [50] propose two equal service classes with solely a throughput/delay trade-off. A further approach is via the Load Control Gateways [58, 59, 79]. There a stateless admission control mechanism is derived based on binary packet marking. A method to implement a guaranteed service with a stateless core network is presented by Stoica et al. [105].

There are voices in the community who believe that sufficient QoS can be achieved by over-provisioning. Over-provisioning means offering a vast amount of capacity and not taking any measures beyond that. It has been studied by Breslau and Shenker [16] and revisited by Heckmann and Schmitt [47]. The advantage of over-provisioning is that it is simple and easy to implement. However, its drawback is that once the demand exceeds the capacity, it takes some time to adapt to it and provide more capacity.

2.3 System Theory

System theory is used as a tool for analysis in a vast amount of different areas. All systems have in common that an *input signal* is mapped to an *output signal*. In the different areas the input and output signal have different meanings. One area where system theory is intensively used is the field of communication systems, which are based on electric circuits. The input is an electric signal which carries the information. The system is the medium, wired or wireless, over which this signal is transmitted. The output is the signal that arrives at the receiver, which is usually distorted by noise signals in the medium. In the ISO-OSI model for computer networks, this is the subject of the physical layer. System theory for communication systems was developed by Küpfmüller and published in his legacy book (in German) in 1952 [67]. The reference for this section is the book by Oppenheim et al. [77]. The signals are in the form of voltages varying over time. These can be discrete, which are called sequences, or continuous, which are called functions. In this dissertation, $\mathbf{x}[k]$ denotes an input sequence, and $\mathbf{x}(t)$ denotes an input function. Accordingly, $\mathbf{y}[k]$ and $\mathbf{y}(t)$ denote an output sequence and output function, respectively. The mapping rule is defined by the operator $\mathbf{S}\{\}$, i.e.,

$$\mathbf{y}(t) = \mathbf{S}\{\mathbf{x}(t)\} \tag{2.1}$$

and

$$\mathbf{y}[k] = \mathbf{S}\{\mathbf{x}[k]\}. \tag{2.2}$$

An important signal is the unit impulse. In the discrete case it is given by

$$\mathbf{d}[k] = \begin{cases} 0 & k \neq 0 \\ 1 & k = 0 \end{cases} . \tag{2.3}$$

The step sequence in the discrete case is

$$\mathbf{u}[k] = \sum_{i=-\infty}^{k} \mathbf{d}[i] = \begin{cases} 0 & k < 0 \\ 1 & k \geq 0 \end{cases} . \tag{2.4}$$

The continuous setting is more tedious. The step function is given by

$$\mathbf{u}(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} . \tag{2.5}$$

The impulse function is the derivative of the step function and given by the expression

$$\mathbf{d}(t) = \frac{d\mathbf{u}(t)}{dt} = \begin{cases} 0 & t < 0 \\ \frac{1}{\tau} & 0 \leq t \leq \tau \\ 0 & t > \tau \end{cases} , \tag{2.6}$$

where $\tau \to 0$.

The system alters the input signal, and the question then is whether the information from the input signal can be retrieved from the output signal. For this it is essential to know how the system maps an input to an output. The brute force method to find this out would be to implement or simulate the system, and feed it every possible input signal and record the corresponding output signal in a look-up table. However, since the amount of possible input signals can be large, this is not a feasible method.

Whether there are more tractable ways to map inputs to outputs depends on the properties of the system. The fundamental properties are reviewed in the following. A system is called *memoryless*, if the output only depends on the current input. If each input has a distinct output, i.e., if one can assign each output an unique input, then the system is *invertible*. A *causal* system is one where the output depends only on the current and past inputs, but not on the future. In a *time-invariant* system, an input yields the same output, regardless at which time it is processed. Finally, a system is deemed *linear*, if the following two properties hold true.
1. The additivity property:

$$\mathbf{S}\{\mathbf{x}_1(t) + \mathbf{x}_2(t)\} = \mathbf{S}\{\mathbf{x}_1(t)\} + \mathbf{S}\{\mathbf{x}_2(t)\} \quad \text{for all } \mathbf{x}_1(t), \mathbf{x}_2(t). \tag{2.7}$$

2. The scaling propety, where a is a constant:

$$\mathbf{S}\{a\mathbf{x}(t)\} = a\mathbf{S}\{\mathbf{x}(t)\} \quad \text{for all } \mathbf{x}_1(t), \mathbf{x}_2(t). \tag{2.8}$$

2.3.1 Linear Time-invariant Systems

Systems that are linear and time-invariant possess properties which make them easy to handle. They allow a system to be described by its *impulse response*. The impulse response is the output of the system when the input is the impulse function and is denoted by $\mathbf{h}(t)$. The significance of the impulse response is pointed out by the following property. In a continuous, linear, time-invariant system, the output for any input can be computed by the convolution of the input function with the impulse response,

$$\mathbf{y}(t) = (\mathbf{x} * \mathbf{h})(t) = \int_{-\infty}^{t} \mathbf{x}(\tau)\mathbf{h}(t-\tau)d\tau. \tag{2.9}$$

Accordingly, in the discrete system the output is computed by the convolution sum

$$\mathbf{y}[t] = [\mathbf{x} * \mathbf{h}][t] = \sum_{l=-\infty}^{k} \mathbf{x}[l]\mathbf{h}[k-l]. \tag{2.10}$$

The output of two systems with impulse responses $\mathbf{h}_1(t)$ and $\mathbf{h}_2(t)$ in sequel can be obtained by consecutively convolving the input signal with the two impulse responses. However, an alternative way to obtain the output is to first convolute the two impulse responses and then convolute the input signal with it. This is a useful property, as it makes it possible to break down complex systems into tractable subsystems.

2.3.2 Transforms

Considering the voltage variations over time is only one of several ways to represent a signal. A common way to represent signals is by superposition of basic functions. Carrying over a signal from one representation to another is called a *transform*. Even though the information is the same, with transforms many times certain properties are made clearer. E.g., the spectrum of a signal can be seen, i.e., the signal is transformed from the time domain into the frequency domain. A fundamental transform of communication systems is the Fourier transform. The underlying basic functions are sinusoidal functions. Therefore, the signal is represented as a superposition of sinusoidal functions. The Fourier transform is obtained by the following equation

$$\mathbf{X}(j\omega) = \mathfrak{F}\{\mathbf{x}(t)\} = \int_{-\infty}^{\infty} \mathbf{x}(t)e^{-j\omega t}dt. \tag{2.11}$$

The inverse Fourier transform is

$$\mathbf{x}(t) = \mathfrak{F}^{-1}\{\mathbf{X}(j\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{X}(j\omega)e^{j\omega t}d\omega. \tag{2.12}$$

The Fourier transform answers the following question. Given an infinitely large number of generators of sinusoidal functions with the parameters amplitude and phase, how do they have to be configured, so that the desired signal is obtained?

The Fourier transform is only defined for a limited set of functions, namely the ones for which the integral converges. The Laplace transform allows for a broader set of functions by introducing a damping term in the exponent. The Z-transform is the discrete equivalent of the Laplace transform. It is given by

$$\mathbf{X}[z] = \mathfrak{Z}\{\mathbf{x}[k]\} = \sum_{k=-\infty}^{\infty} \mathbf{x}[k] z^k. \tag{2.13}$$

The inverse Z-Transform is then given by

$$\mathbf{x}[k] = \mathfrak{Z}^{-1}\{\mathbf{X}[z]\} = \frac{1}{2\pi j} \oint \mathbf{X}[z] z^{k-1} dz. \tag{2.14}$$

A beneficial property that these transforms share is that the convolution in the time domain corresponds to the multiplication in the frequency domain. Using the Fourier transform as an example, one obtains

$$(\mathbf{x} * \mathbf{h})(t) = \mathfrak{F}^{-1}\{\mathfrak{F}\{\mathbf{x}(t)\}\mathfrak{F}\{\mathbf{h}(t)\}\}. \tag{2.15}$$

This detour of computing the convolution is often used in practice. Especially for numerically computing convolution sums, efficient algorithms have been developed, such as the Fast Fourier Transform (FFT) [87].

In order to avoid confusion with min-plus system theory, the term conventional system theory is used in the course of this text.

2.4 Queuing Theory

Now the focus is turned back to the subject of this dissertation, viz., an analytical model for packet switched networks. The first tool that comes to mind is queuing theory.

Queuing theory is the oldest model for packet switched networks and has been accompanying the Internet development since its Arpanet days. The first link between packet switched networks and queuing theory dates back to Kleinrock's PhD thesis proposal of 1961 [62], which also holds an extensive bibliography of early works on queuing theory. This is viewed as the birth of packet switched networks. Kleinrock played a key role in the development of the Arpanet, which is one of the reasons why queuing theory is imminent in today's networking research.

The basics of queuing theory are reviewed in this section, with special reference to the textbooks by Kleinrock [63, 64] and Robertazzi [88]. The paradigm of a queuing system is that there are *customers* who seek service from a *server*. There may be one, multiple, or an infinite number of servers. The service provided to the customer takes time, and therefore it can happen that that all servers are busy upon arrival of a customer. In this case, the customer must wait in a *queue* until a server becomes available. The queue can be limited in size or modeled as infinitely large. This paradigm holds in a vast amount of settings. A popular example is a grocery store, where the customers wait in the line (the queue), for a cashier (the server). In the setting of computer networks, the customers are the data packets that arrive at a network node. The server is the entity that places the incoming packets on to the corresponding outgoing link. The queue is the memory that buffers incoming packets. Usually, and throughout this dissertation, the network nodes are modeled as single-server systems.

Also here the system view can be taken. The input is the traffic that enters the node, the system describes the server processing the packets, and the output is the traffic that leaves the node. Again, the computation of the output is of interest. By comparing the input and the output traffic, QoS parameters such as throughput, delay and loss can be determined.

Of course, the description of the incoming traffic and behavior of the server is of utmost importance, not only for the quality of the results but also the complexity to obtain them. In queuing theory, the method of choice for these descriptions are stochastic processes. For an elaborate study of stochastic processes the reader is referred to [84].

The arrivals are modeled by assuming that their inter-arrival times, that are the times between two arrivals, follow a stochastic distribution. The server is modeled by assuming a distribution for the time a packet needs to be served. Another parameter concerning the server is the scheduling discipline. In a First In First Out (FIFO) scheduler the packet that arrived first is also the one to be served first. Other basic scheduling disciplines are Last In First Out (LIFO) and Earliest Deadline First (EDF). Some results of queuing theory require the server to be *work-conserving*. A server is deemed work-conserving, if it is not idle, whenever there are packets in the system.

Therefore, there are five main parameters which describe a queuing system:

- distribution of the inter-arrival times (A)
- distribution of the service times (B)
- number of servers (C)
- buffer size (D)
- scheduling discipline

A popular notation for queuing systems is Kendall's notation A/B/C/D, where each of the letters denotes the parameters as given in the list above. Only the scheduling discipline is to be given separately.

2.4.1 Poisson Process

As mentioned above, the modeling of networks with queuing theory has two objectives. The first is that the model should be accurate and the second is that it should be tractable. These two aspects stand in competition with one another and often as a good trade-off the *Poisson process* is used.

A Poisson process is a random process that is built upon the following three axioms. Let $\tau \rightarrow 0$ and λ be the mean arrival rate.
1. There is at most 1 arrival in τ.
2. The probability of an arrival in τ is proportional to its length, i.e.,

$$\mathbf{P}(\text{exactly 1 arrival in } [t, t+\tau]) = \lambda\tau. \tag{2.16}$$

3. Arrivals in two intervals $[t_1, t_2]$ and $[t_3, t_4]$, where

$$[t_1, t_2] \cap [t_3, t_4] = \varnothing,$$

i.e., the intervals are not overlapping, are independent.

The probability, that there are n arrivals in a time interval t for a Poisson process of rate λ is given by

$$\mathbf{P}(n) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}. \tag{2.17}$$

The Poisson process has certain desirable properties. It can be shown that its inter-arrival times are exponentially distributed.

Further, the arrivals are independent of each other and memoryless. The probability of k arrivals in an interval t is independent of the number of arrivals in any previous interval.

2.4.2 M/M/1 and M/M/1/N Queue

The simplest queuing system is the M/M/1 queue. In Kendall's notation, the first "M" denotes that the arrivals follow a Poisson process. Its mean arrival rate is usually denoted by λ. The second "M" is the service times which are exponentially distributed. The server rate is usually denoted by μ. The "1" indicates that there is one server. By convention, the missing fourth parameter denotes an infinitely large buffer. With the mean arrival rate being λ and the mean service time $\frac{1}{\mu}$, the utilization of a queuing system is given by $\rho = \frac{\lambda}{\mu}$. In a M/M/1 queue, the probability of being in state k is geometrically distributed and given by the equation

$$\mathbf{P}(k) = \rho^k(1-\rho). \tag{2.18}$$

The output of the M/M/1 queue is again a Poisson process.

The assumptions of the M/M/1/N queue differ from those of the M/M/1 queue only in the buffer being finite. Here the buffer size is denoted by n. In a M/M/1/N queue, the probability of being in state k is given by the equation

$$\mathbf{P}(k) = \frac{1-\rho}{1-\rho^{n+1}}(\rho^k). \tag{2.19}$$

If $\rho \ll 1$, the M/M/1 queue is a good approximation of the M/M/1/N queue.

2.4.3 Queuing Theory and QoS

Up to now the obtained results only give insight on the buffer occupancy distribution of the queue. In this section it is pointed out how QoS can be modeled with queuing theory. The QoS parameters of interest are throughput, delay and loss. The throughput of a queuing system is trivial as it can be directly obtained from the server rate.

In order to compute the delay of a queuing system, there is a simple yet powerful result connecting the average delay, average buffer length and arrival rate. It is known as Little's law and states that the average delay $E(\delta_L)$ equals the average queue length $E(k)$ divided by the arrival rate λ_L,

$$E(\delta_L) = \frac{E(k)}{\lambda_L}. \tag{2.20}$$

The arrival rate λ_L denotes the rate of arrivals that actually enter the queuing system. In the M/M/1 case $\lambda_L = \lambda$, but when packets are dropped, e.g., in the M/M/1/N case, then $\lambda_L < \lambda$.

The paradigm of the M/M/1/N queuing system allows for the computation of loss. A packet loss occurs when arrival sees a full queue, i.e., in state n. This can be computed by using Equation 2.19,

$$\mathbf{P}(n) = \frac{1-\rho}{1-\rho^{n+1}}(\rho^n). \tag{2.21}$$

The average rate of packet losses is then $\lambda\mathbf{P}(k)$. Here a popular approximation is to use the Equation 2.18 of the M/M/1 queue instead.

What has been presented here are the basics of queuing theory. Once the assumptions on the arrivals and service times are relaxed, tractability is quickly lost. This applies especially for the assumption that the arrivals follow a Poisson process. However, beginning with [70], it has been shown several times that this is not necessarily a realistic assumption for Internet traffic. This can be combated by shaping the traffic such that even though it does not follow a Poisson process, it has its stochastic properties. Abendroth and Killat [1] develop a method to do so.

2.5 Network Calculus

Network calculus was first published by Cruz in 1991 [23, 24]. It was then elaborated by Le Boudec [13] and Chang [18], who both pointed out the relationship to conventional system theory. Network Calculus is a theory for deterministic queuing systems. The paradigm of a system with an input and an output is maintained. The system is a network or a part of the network. This is significant, as it implies that the entities analyzable with network calculus range from single nodes over several nodes up to entire networks. As pointed out before, it is an important property of system theory that complex systems can be broken down into tractable subsystems. This property is carried over to network calculus. The mathematical background is introduced first, as with it the nature of input and output functions becomes clearer. The underlying mathematical structure of network calculus is the min-plus dioid. A comprehensive description of the max-plus algebra is given in the book by Baccelli et al. [6]. Min-plus algebra and max-plus algebra are identical in the sense that they share all properties. The core application that Baccelli et al. deal with are Petri Nets. Le Boudec [13] dedicates a chapter to the review of min-plus algebra.

The reason for min-plus algebra being the structure of choice for the analysis of networks becomes clear with the following line of thought. Assume a link with a fixed capacity emerging from a sender. The rate on this link will always be the minimum of the link capacity and the rate at which the sender desires to send. I.e., if the sender desires to send more than the capacity, then the rate on the link will be the capacity itself. Otherwise, the rate will be the actual sending rate. A similar rationale applies for the analysis of a buffer limited in size. The number of packets in the buffer is the minimum of the difference between all arrivals and departures and the size of the buffer.

Before proceeding, two operators need to be introduced, the *infimum* and *supremum* operator.

Definition 2.1 (Infimum operator) *The infimum of a set M, denoted by* $\inf\{M\}$, *is the greatest lower bound of the set.*

Note that in contrast to the minimum operator, the result need not be in the set. An example where the infimum is not in the set is

$$\inf_{t \in \mathbb{R}^+} \{1/t\} = 0.$$

If the minimum is in the set, then the minimum and infimum operators are identical,

$$\inf\{2,3\} = \min\{2,3\} = 2.$$

The supremum operator, denoted by ”sup”, is the equivalent for the maximum operator.

2.5.1 Min-plus Algebra

We next review the properties of min-plus algebra. As mentioned before, this is extensively presented in the books by Baccelli [6] and Le Boudec [13]. Recall that traditional algebra uses the set of real numbers. The addition operator is $+$ and the multiplication operator is $\times$. This structure $(\mathbb{R}, +, \times)$ is called a commutative field. In contrast, min-plus algebra is based on the structure $(\mathbb{R} \cup +\infty, \min, +)$, i.e., the set of real numbers with $+\infty$ endowed with the minimum and sum operator. Therefore, the ”addition” operator is min and the ”multiplication” operator is $+$. The quotation marks indicate that this is a confusing nomenclature, as one is used to associating addition and $+$, as well as multiplication and $\times$. In this thesis, only the terms ”plus” or ”sum” are used for the operator +, and "product" for the operator ×. The terms "addition" and "multiplication" are reserved for the abstracted operators on sets.

According to Baccelli ([6], page 154), the following properties make the structure $(\mathbb{R} \cup +\infty, \min, +)$ a commutative dioid.
Associativity of addition: For all $a, b, c \in \mathbb{R} \cup +\infty$,
$\min\{\min\{a, b\}, c\} = \min\{a, \min\{b, c\}\}$.
Commutativity of addition: For all $a, b \in \mathbb{R} \cup +\infty$, $\min\{a, b\} = \min\{b, a\}$.
Associativity of multiplication: For all $a, b, c \in \mathbb{R} \cup +\infty$, $(a + b) + c = a + (b + c)$.
Commutativity of multiplication: For all $a, b \in \mathbb{R} \cup +\infty$, $a + b = b + a$.
Distributivity of multiplication with respect to addition: For $a, b, c \in \mathbb{R} \cup +\infty$, $\min\{a, b\} + c = \min\{a + c, b + c\}$.
Existence of a zero element: For $a \in \mathbb{R} \cup +\infty$, there is some $\epsilon = +\infty \in \mathbb{R} \cup +\infty$, such that $\min\{\epsilon, a\} = a$.
Absorbing zero element: For $a \in \mathbb{R} \cup +\infty$, $a + \epsilon = \epsilon$.
Existence of an identity element: For $a \in \mathbb{R} \cup +\infty$, there is some $e = 0 \in \mathbb{R} \cup +\infty$, such that $e + a = a$.
Idempotency of addition: For $a \in \mathbb{R} \cup +\infty$, $\min\{a, a\} = a$.
Le Boudec additionally points out two further properties.
Closure of addition: For all $a, b \in \mathbb{R} \cup +\infty$, $\min\{a, b\} \in \mathbb{R} \cup +\infty$.
Closure of multiplication: For all $a, b \in \mathbb{R} \cup +\infty$, $a + b \in \mathbb{R} \cup +\infty$.

A difference between a commutative dioid and a commutative field is that in the latter the idempotency of addition is replaced by the existence of a cancellation element of addition. In the commutative field $(\mathbb{R}, +, \times)$ there exists for $a \in \mathbb{R}$, a cancellation element $-a$ such that $a + (-a) = 0$. This has consequences as it enables unique solutions when equalities of the type $a + t = b$ are to be solved for t. Similarly, from $a + b = a + c$ one can conclude that $b = c$. Solving the equality $\min\{a, t\} = b$ for t does not always yield a unique solution and $\min\{a, b\} = \min\{a, c\}$ does not imply $b = c$.

In accordance with the notation of Le Boudec [13], we define $\mathcal{G}$ and $\mathcal{G}_d$ as the spaces of wide-sense increasing functions and sequences, respectively,

$$\mathcal{G} := \{\mathbf{f} : \mathbb{R} \to \mathbb{R}_0^+ \mid \mathbf{f}(t_1) \leq \mathbf{f}(t_2) \text{ for } t_1 \leq t_2\} \tag{2.22}$$

and

$$\mathcal{G}_d := \{\mathbf{f} : \mathbb{Z} \to \mathbb{R}_0^+ \mid \mathbf{f}[k_1] \leq \mathbf{f}[k_2] \text{ for } k_1 \leq k_2\}. \tag{2.23}$$

Usually, it is further required that $\mathbf{f}(t) = 0$ for $t \leq 0$. The set of functions $\mathcal{F}$, as well as the set of sequences $\mathcal{F}_d$ are defined as

$$\mathcal{F} := \{\mathbf{f} : \mathbb{R} \to \mathbb{R}_0^+ \mid \mathbf{f}(t) = 0 \text{ for } t \leq 0, \mathbf{f}(t_1) \leq \mathbf{f}(t_2) \text{ for } t_1 \leq t_2\} \tag{2.24}$$

and

$$\mathcal{F}_d := \{\mathbf{f} : \mathbb{Z} \to \mathbb{R}_0^+ \mid \mathbf{f}[t] = 0 \text{ for } t \leq 0, \mathbf{f}[k_1] \leq \mathbf{f}[k_2] \text{ for } k_1 \leq k_2\}. \tag{2.25}$$

A key operation is the min-plus convolution. The min-plus convolution with the impulse response is the operator which maps the input to the output.

Definition 2.2 (Continuous min-plus convolution) *Let* $\mathbf{f}(t)$ *and* $\mathbf{g}(t)$ *be two functions of* $\mathcal{F}$*. The min-plus convolution of* $\mathbf{f}(t)$ *and* $\mathbf{g}(t)$ *is the function*

$$(\mathbf{f} \star \mathbf{g})(t) = \min_{0 \le \tau \le t} \{\mathbf{f}(t-\tau) + \mathbf{g}(\tau)\}. \tag{2.26}$$

Definition 2.3 (Discrete min-plus convolution) *Let* $\mathbf{f}[k]$ *and* $\mathbf{g}[k]$ *be two functions of* $\mathcal{F}_d$*. The min-plus convolution of* $\mathbf{f}[k]$ *and* $\mathbf{g}[k]$ *is the function*

$$[\mathbf{f} \star \mathbf{g}][k] = \min_{0 \le l \le k} \{\mathbf{f}[k-l] + \mathbf{g}[l]\}. \tag{2.27}$$

Some results, the most prominent example being the output bound, rely on the min-plus deconvolution.

Definition 2.4 (Continuous min-plus deconvolution) *Let* $\mathbf{f}(t)$ *and* $\mathbf{g}(t)$ *be two functions of* $\mathcal{F}$*. The min-plus deconvolution of* $\mathbf{f}(t)$ *and* $\mathbf{g}(t)$ *is the function*

$$(\mathbf{f} \hat{\star} \mathbf{g})(t) = \sup_{0 \le \tau \le t} \{\mathbf{f}(t+\tau) - \mathbf{g}(\tau)\}. \tag{2.28}$$

An example of a min-plus deconvolution is given in Section 2.5.4. The result of the min-plus deconvolution of two functions of $\mathcal{F}$ is not necessarily in $\mathcal{F}$. However, since in practice only values for $t \ge 0$ are relevant, such functions can be brought back into $\mathcal{F}$ by setting them to 0 for $t \le 0$.

The min-plus convolution appears in the construction of the input and output functions.

2.5.2 Arrival Curve Concept

Intuitively, if a network is to be described as a system, the input is the data that the sender injects into the network, and the output is the data that arrives at the receiver. Due to the property that the network can be broken down into several subnetworks, the output of a network can be the input of another network. Therefore it is essential that the input and output belong to the same space. Let us take a step back and at first consider a "network" with only one element. As mentioned before, the input function must represent the traffic that enters the network. The most intuitive way to describe data traffic as a function is to have a time axis t and assign to each time instant a value $\mathbf{r}(t)$ which denotes the instantaneous rate. This can be done statistically, which results in traffic models such as the Markov modulated fluid or an on/off source. Many statistical models rely on stationarity of the traffic. I.e., a rate $\mathbf{r}(t)$ does not depend on any rate $\mathbf{r}(t-\tau)$ before or after. A tractable extension to this is the Markov modulated fluid, where the rate depends on the previous state, but explicitly no state prior to that. The disadvantage of statistical traffic descriptions is that it is hard to give deterministic guarantees. An alternative is a deterministic traffic description. This is realized by the *arrival curve* concept, which denotes the worst case traffic. It is advantageous to consider the integral of the instantaneous rates, i.e., the cumulated amount of data that has been injected into the network up to the time t. Since $\mathbf{r}(t) \geq 0$ for all t this yields a wide-sense increasing function $\mathbf{x}(t)$, which is beneficial for operations in the min-plus algebra,

$$\mathbf{x}(t) = \int_{-\infty}^{t} \mathbf{r}(\tau) d\tau. \tag{2.29}$$

Definition 2.5 (Arrival Curve) *Given a wide-sense increasing function $\boldsymbol{\alpha}(t) \in \mathcal{F}$, a flow $\mathbf{x}(t)$ is constrained by $\boldsymbol{\alpha}(t)$ if and only if for all $\tau \leq t$*

$$\mathbf{x}(t) - \mathbf{x}(\tau) \leq \boldsymbol{\alpha}(t - \tau). \tag{2.30}$$

Further terminology is stating that $\mathbf{x}(t)$ has $\boldsymbol{\alpha}(t)$ as an arrival curve, or also that $\mathbf{x}(t)$ is $\boldsymbol{\alpha}(t)$-smooth.

It is important to note that in an arrival curve $\boldsymbol{\alpha}(t)$ the argument t is not to be interpreted as an instant of time but as a time interval. Therefore, a value $\boldsymbol{\alpha}(t)$ denotes the amount of data that maximally is allowed to be sent in any time interval t. The arrival curve can be viewed as an an abstraction of traffic regulation algorithms. An arrival curve can also be discrete. It is then denoted by $\boldsymbol{\alpha}_k[]$.

The most popular traffic regulation algorithm is the token bucket, also known as leaky bucket [109]. The token bucket has two parameters, the token rate r_γ and the bucket depth b_γ. As the name implies the token bucket can be described pictorially. Tokens flow into the bucket at a constant rate. The tokens are collected in the bucket, until the bucket depth is reached. Tokens entering then are discarded. Whenever data is sent, the corresponding amount of tokens are removed from the bucket. It is to be noted, that in this model only tokens enter, stay in and leave the bucket, but not the actual data. This is depicted in Figure 2.2. There is an input into the buffer.

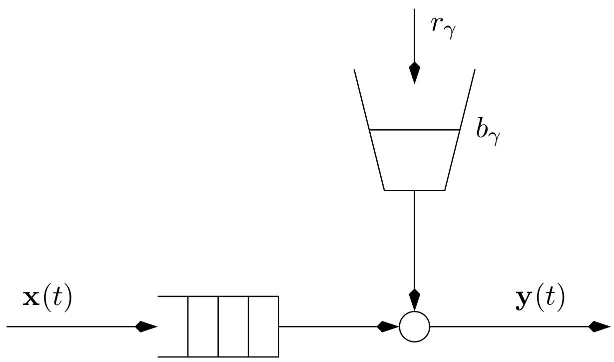

Figure 2.2: Token bucket

The token bucket controls when the data is released. This can range from infinitely fast if tokens are available to the data having to wait for new tokens to be generated. It is important to note that the token bucket is independent of the buffer, i.e., the token bucket does not give any indication as to how to deal with data arriving when no tokens are available. Options are to assume an infinite buffer which stores all data until tokens are available, a fixed size buffer as well as discarding the data immediately.

Again the relationship between min-plus algebra and the analysis of packet-switched networks becomes clear. The amount of data that is released is the minimum of what the sender desires and what the token bucket allows to be emitted. The arrival curve of the token bucket is given by the following equation.

$$\boldsymbol{\alpha}(t) = \begin{cases} 0 & t \leq 0 \\ \boldsymbol{\gamma}(t) = b_\gamma + r_\gamma t & t > 0 \end{cases} . \tag{2.31}$$

Another wide spread arrival curve is the Traffic Specification (TSpec) from IntServ (cf. Section 2.2). It is defined as the concatenation of two token buckets and therefore has four parameters: the maximum packet size $b^{\zeta,1}$, the burst parameter $b^{\zeta,2}$, the peak rate $r^{\zeta,1}$ and the sustained rate $r^{\zeta,2}$. The TSpec arrival curve is given by the equation

$$\boldsymbol{\zeta}(t) = \min\{b^{\zeta,1} + r^{\zeta,1}t, b^{\zeta,2} + r^{\zeta,2}t\}. \tag{2.32}$$

Figures 2.3 to 2.6 illustrate the relationship between the different types of functions to describe traffic as well as the arrival curve. In Figure 2.3 a rate function $\mathbf{r}(t)$ in

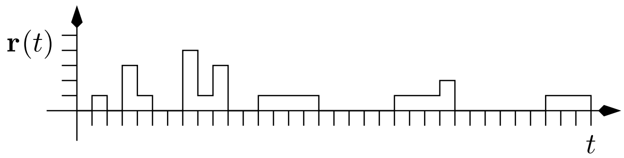

Figure 2.3: Rate function

the classical sense is shown, i.e., each time t is given with its instantaneous rate $\mathbf{r}(t)$. Accumulating the rates according to Equation 2.29 yields the function of Figure 2.4.

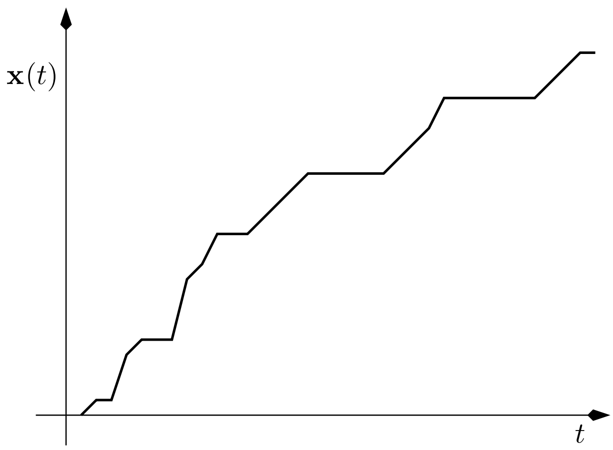

Figure 2.4: Cumulated rate function

Here each value $\mathbf{x}(t)$ denotes the amount of data that has been sent until time t. The token bucket $\boldsymbol{\gamma}_a(t)$ shown in Figure 2.5 is not an arrival curve for the underlying cumulated rate function. In the interval $[t_1, t_2]$ more data is sent than allowed by

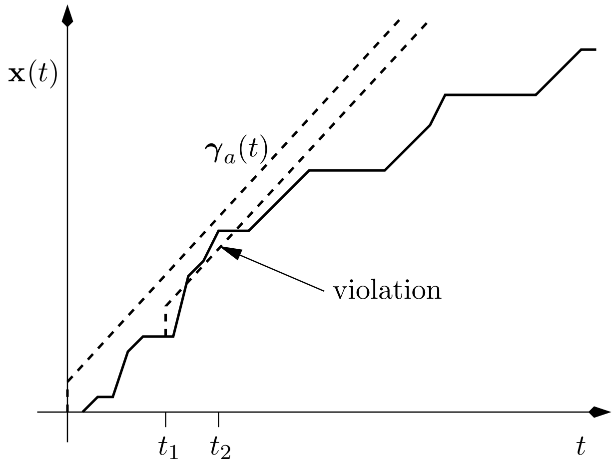

Figure 2.5: Arrival curve with violation

the arrival curve. With this example it becomes clear that it is not sufficient that the cumulated rate function is below the arrival curve starting at $t = 0$. It has to be below the arrival curve even if the arrival curve is placed on it at any point. In

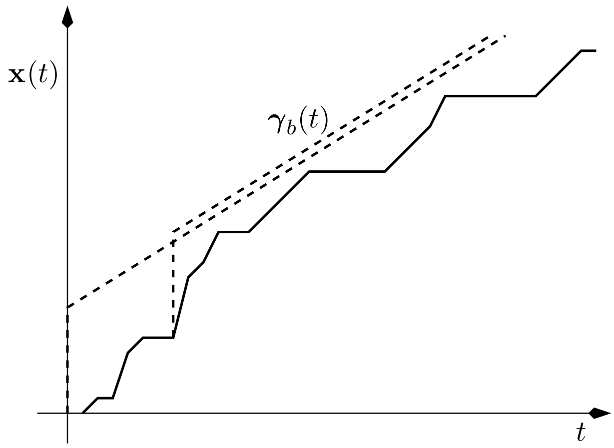

Figure 2.6: Arrival curve

Figure 2.6 a token bucket $\boldsymbol{\gamma}_b(t)$, which is an arrival curve for the underlying cumulated rate function, is depicted.

The description of the token bucket left open, how packets that are not conform with the arrival curve are dealt with. One option is to store the non-conforming packets in a buffer and release them once enough tokens have be generated for them to be conform again. This option, i.e., how a non-conform traffic flow can be made conform, is now highlighted further. This process is called *traffic shaping*. The greedy shaper ([13], page 37) shapes a flow such that each bit is emitted at the earliest time possible while conforming to the underlying arrival curve.

Theorem 2.6 (Greedy shaper) *Let* $\mathbf{x}(t) \in \mathcal{F}$ *be a traffic stream and* $\boldsymbol{\alpha}(t) \in \mathcal{F}$ *be a* *sub-additive arrival curve, i.e.,* $\boldsymbol{\alpha}(t+\tau) \leq \boldsymbol{\alpha}(t) + \boldsymbol{\alpha}(\tau)$ *for all* t, τ*. The shaped traffic stream* $\mathbf{y}(t)$ *which conforms to the arrival curve* $\boldsymbol{\alpha}(t)$ *such that each bit is released at the earliest time possible, is*

$$\mathbf{y}(t) = (\mathbf{x} \star \boldsymbol{\alpha})(t). \tag{2.33}$$

Proof The proof is constructed by combining Theorem 1.5.1 and Corollary 3.1.1 of the book by Le Boudec [13]. The former states that the output is the min-plus convolution with the sub-additive closure of the shaping curve. The latter states that a sub-additive function is its own sub-additive closure. ■

In order to give an example of the greedy shaper, consider the input function

$$\mathbf{x}_a(t) = \begin{cases} 0 & t \leq 5 \\ 10 & t > 5 \end{cases}. \tag{2.34}$$

I.e., after 5 time units, a burst of 10 data units arrives. Assume as a shaper a token bucket with rate 1 and depth 3

$$\boldsymbol{\alpha}_a(t) = \begin{cases} 0 & t \leq 0 \\ 3+t & t > 0 \end{cases}. \tag{2.35}$$

The output of the shaper is then

$$\mathbf{y}_a(t) = (\mathbf{x}_a \star \boldsymbol{\alpha}_a)(t) = \begin{cases} 0 & t \leq 5 \\ 3 + (t-5) & 5 < t \leq 12 \\ 10 & t > 12 \end{cases}. \tag{2.36}$$

Interpreting this result, it can be seen that the shaper is first idle for 5 time units, which is obvious as no traffic arrives at it until then. Then it immediately emits 3 data units, as this is the bucket depth, and then continues to emit 1 data unit per time unit until all 10 data unites are serviced.

2.5.3 Service Curve Concept

This part deals with how packets are manipulated within the network or network element. For this it is essential to recall the task of a node. The elementary task of a node in a packet-switched network is to collect the packets from the incoming links and put them on the outgoing links.

The service curve quantifies the amount of service that a node offers. It is a function $\boldsymbol{\beta}(t)$ and denotes, how long after an arrival a packet must be serviced. Upon arrival of an infinitely large burst at time t_0, at time t_0+t, $\boldsymbol{\beta}(t)$ data units must be serviced. This corresponds to the following definition, which is a slight modification of the definition of Le Boudec ([13], page 23).

Definition 2.7 (Service Curve) *Let* $\mathbf{x}(t)$ *be the input function and* $\mathbf{y}(t)$ *its corresponding output function of a system. The system offers to the flow a service curve* $\boldsymbol{\beta}(t)$ *if and only if* $\boldsymbol{\beta}(t) \in \mathcal{F}$ *and for all* $\mathbf{x}(t)$*,*

$$\mathbf{y}(t) \geq (\mathbf{x} \star \boldsymbol{\beta})(t). \tag{2.37}$$

Le Boudec [13] describes four types of service curves. A minimum service curve denotes the service that a system offers in the worst case. Analogously, a maximum service denotes the service that a system offers in the best case. However, this is only required in exceptional cases. The strict service curve and adaptive service curve have no further relevance in this thesis. Therefore, the term service curve here denotes the minimum service curve. A service curve can also be discrete. It is then denoted by $\boldsymbol{\beta}[k]$.

The most prominent service curve is the *latency rate service curve.*

Definition 2.8 (Latency rate function) *The service curve of a latency rate scheduler is given by*

$$\boldsymbol{\phi}(t) = \begin{cases} 0 & t \leq l_\phi \\ r_\phi(t - l_\phi) & t > l_\phi \end{cases} . \tag{2.38}$$

The parameters are r_ϕ, which denotes the rate and l_ϕ, which denotes the latency.

The latency rate service curve is depicted in Figure 2.7. It is the service curve for the Generalized Processor Sharing (GPS) scheduler [85].

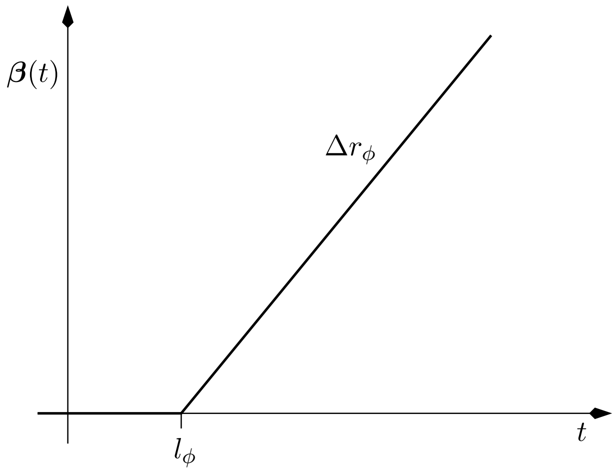

Figure 2.7: Latency rate service curve

To illustrate the concept of the service curve, reconsider the input function $\mathbf{x}_a(t)$. I.e., 10 data units arrive at $t = 5$. Further, let it traverse a system which offers the service curve $\boldsymbol{\beta}_{\boldsymbol{a}}(t)$,

$$\boldsymbol{\beta}_{\boldsymbol{a}}(t) = \begin{cases} 0 & t \leq 3 \\ 2(t-3) & t > 3 \end{cases} . \tag{2.39}$$

The worst case output function $\mathbf{y}_a(t)$ is then computed by the min-plus convolution

$$\mathbf{y}_b(t) = (\mathbf{x}_a \star \boldsymbol{\beta}_{\boldsymbol{a}})(t) = \begin{cases} 0 & t \leq 8 \\ 2(t-8) & 8 < t \leq 13 \\ 10 & t > 13 \end{cases} . \tag{2.40}$$

This result can also be obtained by the following line of thought. Note that solely the worst case is considered. The service curve indicates that the first data has to be served 3 time units after its arrival with a rate of 2 thereafter. The serving rate is maximal as there is backlog. It takes $10/2 + 3 = 8$ time units to serve the burst. Hence, the output function $\mathbf{y}_b(t)$ starts at $t = 8$, rises with rate 2 until $t = 13$ and stays constant thereafter.

2.5.4 Selected Results

In this section some basic, yet powerful, results of network calculus, which are dealt with in-depth in the book by Le Boudec [13], are reviewed. The concatenation of service curves provides a method to represent several network elements as one system. It states that a flow traversing service curves in sequel experiences the service curve that is obtained from the min-plus convolution of the service curves.

Theorem 2.9 *Assume a system consisting of a path of n network elements each offering a service curve $\boldsymbol{\beta_i}(t)$. Then the system offers the service curve*

$$\boldsymbol{\beta}(t) = (\boldsymbol{\beta_1} \star \boldsymbol{\beta_2} \star \ldots \star \boldsymbol{\beta_n})(t). \tag{2.41}$$

Proof The proof relies on the associativity of the min-plus convolution. Let $\mathbf{y}_1(t) = \mathbf{x}_2(t)$ be the output of the first element which is the input of the second one,

$$\mathbf{y}_1(t) \geq (\mathbf{x}_1 \star \boldsymbol{\beta_1})(t). \tag{2.42}$$

Similarly,

$$\begin{aligned} \mathbf{y}_n(t) &\geq (\mathbf{x}_n \star \boldsymbol{\beta_n})(t) && (2.43)\\ &\geq ((\mathbf{x}_{n-1} \star \boldsymbol{\beta_{n-1}}) \star \boldsymbol{\beta_n})(t) && (2.44)\\ &\geq (((\mathbf{x}_1 \star \boldsymbol{\beta_1}) \star \boldsymbol{\beta_2}) \star \ldots \star \boldsymbol{\beta_n})(t) && (2.45)\\ &= (\mathbf{x}_1 \star (\boldsymbol{\beta_1} \star \boldsymbol{\beta_2} \star \ldots \star \boldsymbol{\beta_n}))(t). && (2.46) \end{aligned}$$

■

Next, three important bounds are reviewed. The backlog $\boldsymbol{\upsilon}(t)$ is the amount of data that is in the system at time t. When $\mathbf{x}(t)$ is the input function and $\mathbf{y}(t)$ the output function, it is given by

$$\boldsymbol{\upsilon}(t) = \mathbf{x}(t) - \mathbf{y}(t). \tag{2.47}$$

If the system consists of only one queue, then the backlog denotes the amount of data that is in the queue. The backlog bound determines the maximum amount of data that is in the system at any instant.

Theorem 2.10 (Backlog bound) *Let $\mathbf{x}(t)$ be the input function and $\mathbf{y}(t)$ the output function of a system, which offers the service curve $\boldsymbol{\beta}(t)$. Let $\boldsymbol{\alpha}(t)$ be the arrival curve that constrains the input $\mathbf{x}(t)$. Then, the backlog bound, i.e., the maximum amount of data in the system, is given by*

$$\upsilon \leq \sup_{t>0}\{\boldsymbol{\alpha}(t) - \boldsymbol{\beta}(t)\}. \tag{2.48}$$

Proof It is to be shown that

$$v \leq \sup_{t>0}\{\mathbf{x}(t) - \mathbf{y}(t)\}. \tag{2.49}$$

The following inequalities hold,

$$\mathbf{x}(t) - \mathbf{y}(t) \leq \mathbf{x}(t) - \inf_{0\leq\tau\leq t}\{\mathbf{x}(t-\tau) + \boldsymbol{\beta}(\tau)\} \tag{2.50}$$

$$= \sup_{0\leq\tau\leq t}\{\mathbf{x}(t) - \mathbf{x}(t-\tau) - \boldsymbol{\beta}(\tau)\} \tag{2.51}$$

$$\leq \sup_{0\leq\tau\leq t}\{\boldsymbol{\alpha}(\tau) - \boldsymbol{\beta}(\tau)\}. \tag{2.52}$$

In Inequality 2.50 the Definition 2.7 of the service curve, i.e., Inequality 2.37 is substituted. Equation 2.51 follows from taking all terms into the inf-expression and changing the sign. Inequality 2.52 uses Equation 2.30 of Definition 2.5 of the arrival curve. Since in Inequality 2.52 the supremum is not dependent on t, we have Equation 2.48. ■

The second important bound is the delay bound. The delay bound determines maximal time that the data will need to traverse the system.

Theorem 2.11 *Let a flow constrained by $\boldsymbol{\alpha}(t)$ enter a system that offers the service curve $\boldsymbol{\beta}(t)$. Then the maximum delay that the data will experience in the system is given by*

$$\delta = \sup_{t\geq 0}\{\inf_{\tau\geq 0}\{\boldsymbol{\alpha}(t) \leq \boldsymbol{\beta}(t+\tau)\}\}. \tag{2.53}$$

We omit the proof and refer to Page 28 of [13] for a proof of this theorem. The inner expression means the minimumτ such that $\boldsymbol{\alpha}(t) \leq \boldsymbol{\beta}(t+\tau)$. While this expression seems complicated, it merely denotes the maximal horizontal deviation of two curves. The backlog bound and delay bound are depicted in Figure 2.8.

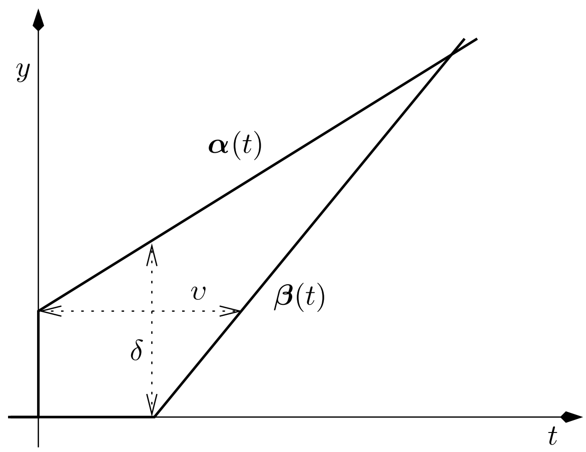

Figure 2.8: Backlog and delay bound

Another basic result is the output bound, which gives an upper bound of the output traffic of a system.

Theorem 2.12 *Let $\mathbf{x}(t)$ be the input function and $\mathbf{y}(t)$ the output function of a system, which offers the service curve $\boldsymbol{\beta}(t)$. Further, let $\mathbf{x}(t)$ be constrained by the arrival curve $\boldsymbol{\alpha}(t)$. Then the output function $\mathbf{y}(t)$ is bounded by the min-plus deconvolution of the arrival curve with the service curve,*

$$\boldsymbol{\psi}(t) = \sup_{0 \leq \tau \leq t} \{\boldsymbol{\alpha}(t+\tau) - \boldsymbol{\beta}(\tau)\}. \tag{2.54}$$

PROOF The proof is given on page 29 in [13]. ∎

An example of the output bound given in Figure 2.9. It can be seen that $\boldsymbol{\psi}(0) \neq 0$.

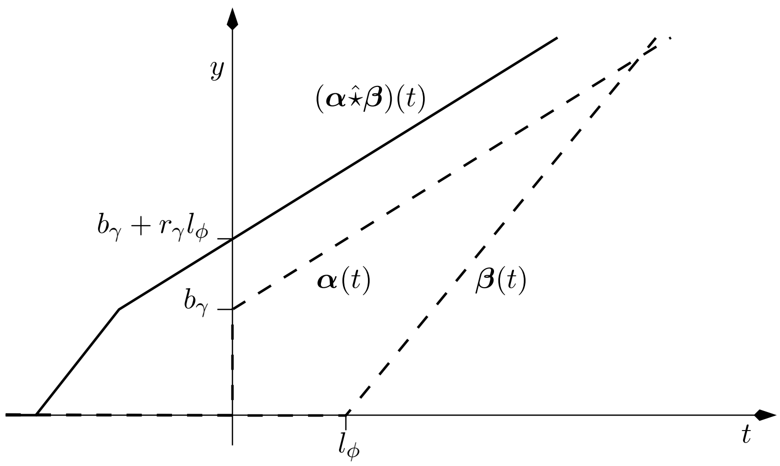

Figure 2.9: Output bound

Since the values for $t < 0$ have no practical purpose, the function is usually set to 0 there which puts it in $\mathcal{F}$. The output bound can also be explained by the following rationale. Assume the traffic to arrive according to the worst case. I.e., first a burst of size b_γ arrives and from then onwards rate r_γ. The maximum burstiness in the output flow is achieved if the server is idle for l_ϕ time units, and then serves all backlogged traffic at once. The backlogged traffic at that point is the burst plus whatever has accumulated during the latency period, i.e., $b_\gamma + r_\gamma l_\phi$.

A result which stems from applying some of the basic theorems is the *Pay bursts only once principle*. Recall the token bucket $\boldsymbol{\alpha}_a(t)$ from Equation 2.35 and the latency rate service curve $\boldsymbol{\beta}_{\mathbf{a}}(t)$ from Equation 2.39. Assume a system consisting of two network elements in sequel offering the service curves $\boldsymbol{\beta}_{\mathbf{1}}(t)$ and $\boldsymbol{\beta}_{\mathbf{2}}(t)$, respectively, with

$$\boldsymbol{\beta}_{\mathbf{1}}(t) = \boldsymbol{\beta}_{\mathbf{2}}(t) = \boldsymbol{\beta}_{\mathbf{a}}(t). \tag{2.55}$$

Further, let the input to this system, i.e., into the first network element, be constrained by an arrival curve $\boldsymbol{\alpha}_1(t) = \boldsymbol{\alpha}_a(t)$. There are two ways to obtain the maximum delay of the system. One is to compute the delay bound for each network element independently and add them. For the first network element, the delay bound is given by the maximum horizontal deviation between $\boldsymbol{\alpha}_1(t)$ and $\boldsymbol{\beta}_{\mathbf{1}}(t)$, which is $\delta_1 = 4$. In order to determine the delay bound for the second network element, the arrival curve for the second network element has to be computed using the output bound,

$$\boldsymbol{\alpha}_2(t) = \begin{cases} 0 & t \leq 0 \\ 5 + t & t > 0 \end{cases} . \tag{2.56}$$

The delay bound for the second network element for this arrival curve is $\delta_2 = 5$. The total delay bound as computed by iteration is then $\delta_{iter} = \delta_1 + \delta_2 = 9$.

The alternative method is to compute the system service curve by concatenation of $\boldsymbol{\beta_1}(t)$ and $\boldsymbol{\beta_2}(t)$ and then determine the delay bound. The concatenation yields

$$\boldsymbol{\beta_b}(t) = (\boldsymbol{\beta_1} \star \boldsymbol{\beta_2})(t) = \begin{cases} 0 & t \leq 0 \\ 2(t-6) & t > 6 \end{cases} . \tag{2.57}$$

The delay bound is then $\delta_{conc} = 8$. It is to be noted that $\delta_{conc} < \delta_{iter}$. The reason for this is, that the arrival curve from the output bound relies on a traffic bulk bigger than the burst size being served at once. This implies, that certain data, explicitly the data arriving during the latency period, was served prior to its deadline set by the service curve. Therefore, the additional delay in the second element only applies to data which was expedited in the first network element. When computing the delay bound with the iterative method, this knowledge is not used, and therefore the bound deteriorates.

3 Building Blocks for QoS Performance Analysis

3.1 Requirements of a Model

The motivation of this work is a unified model for QoS performance analysis. In this section we highlight the characteristics of such a model. Different ways of studying a system are brought into relation in the book by Law and Kelton [69], and revisited in the dissertation by Bodamer [10] (in German) in the context of QoS. This is shown in Figure 3.1. One way is to study the actual system. In our case this means studying the

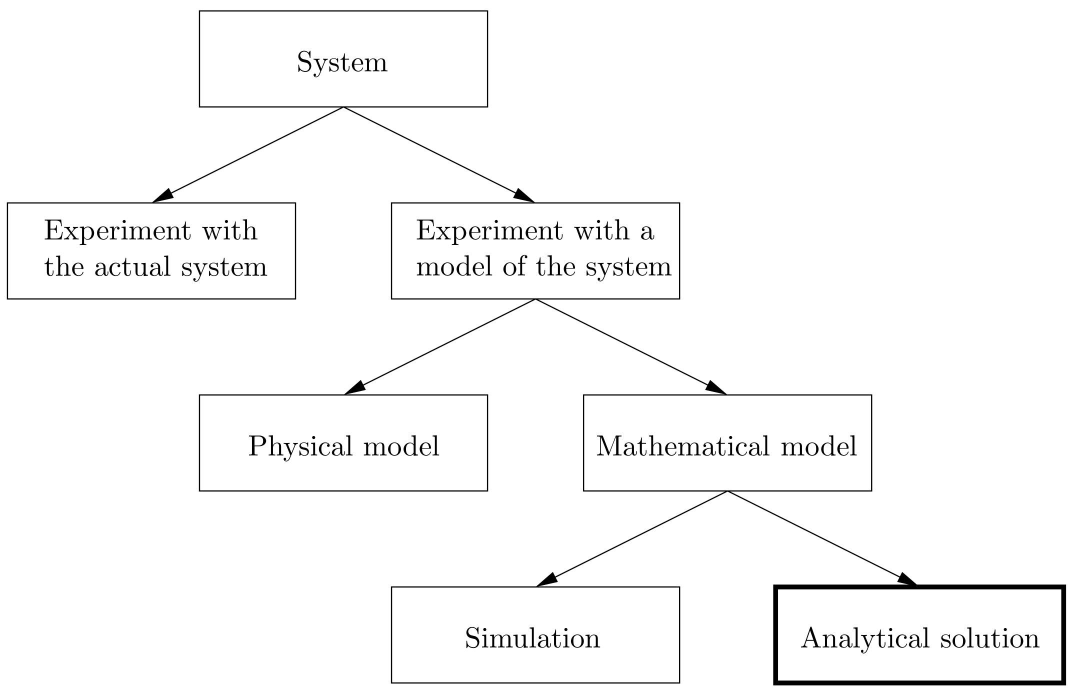

Figure 3.1: Relations of model types from Law and Kelton [69]

Internet. The drawbacks of this method are obvious. First of all, no changes can be studied as it is not possible to globally change parameters such as routing or congestion control of the Internet. Secondly, it is impossible to incorporate predictions on the future. E.g., it can not be studied how the Internet would perform with a larger fraction of multimedia traffic. Therefore, models are beneficial. One can distinguish between physical models and mathematical models. A physical model is when a network is built in a lab environment and actual packets are sent through it. A mathematical model is further divided into analytical models and simulation. Simulations are very popular for the study of networks, with *ns2* being the front runner simulation tool. However, the subject of this dissertation is an analytical model.

Generally, the purpose of a model is to simplify a real world scenario by approximating it. The biggest challenge thereby is to find a balance between being close enough to reality so that the desired characteristics can be extracted, yet simplifying enough to make the model tractable. The requirements of a model are stated in the following list and elaborated thereafter.

- Insight on the relevant characteristics
- Wide range of applicability
- Transparence and ease of applicability

Regarding the first point it goes without saying that a model is worthless if it does not provide insight on the desired characteristics. In this case, the desired characteristic is QoS performance. We strive for a model which gives insight on the QoS parameters of data flows that traverse networks. Combining network calculus and queuing theory in Chapter 6 addresses this point. A common criticism of network calculus is that the worst case bounds are too pessimistic, while a common criticism of queuing theory is that the average behavior is not meaningful for QoS. We analyze whether the best of both worlds can be combined.

A limited range of applicability is often encountered in networking research. This is due to the fact that most researchers focus on a protocol or standard and derive locally optimized methods. Network calculus started out in that spirit as a model for ATM and IntServ networks. One of the goals of this dissertation is to extend the range of applicability of network calculus. This is the subject of Chapter 4 where a method to apply network calculus to a novel type of network is described.

In order for a model to be successful, it has to be easy to use. Only then will it be exposed to practical problems and developed further. On the contrary, if it takes deep knowledge before a model can be applied, it will only exist in the minds of a few researchers. In this issue network calculus has room for improvement, mainly because min-plus algebra is not widely taught. It is addressed in Chapter 5, where the emphasis is on a graphical representation of the min-plus convolution, which is a key operation in network calculus.

Finally, it is beneficial if the model is transparent, i.e., not black box behavior. Neural Networks [44] are an example of a model type which is not very transparent. Throughout this dissertation the theoretical results are accompanied by describing what they actually mean to packets traversing the network.

The main contributions of this dissertation are novel results of network calculus. In the remainder of this chapter, the state of the art of network calculus is presented. Thereby gaps are identified, which are bridged by the contributions in the following chapters.

3.2 Insight on Relevant Characteristics

Recall that one requirement of the model is to offer an analysis of the desired characteristics of the system. In reality, networks often do not behave the way one would ideally want them to, which makes their analysis tedious. In this section two such issues are highlighted and their network calculus solutions pointed out.

3.2.1 Packet Loss

Packet loss is one of the key QoS parameters in packet switched networks. Hence, analyzing packet loss has been a widely studied research topic in the Internet. Its importance is increased by the fact that TCP, the most widely spread transport protocol of the Internet, controls the rate of the data flows over the lost packets. Generally, analyzing packet loss is relatively underexposed in network calculus. This is because historically network calculus was intended for ATM and IntServ networks, where packet loss is not a core issue. Network calculus offers several approaches to deal with packet loss. One is the clipper by Cruz and Taneja [25]. The clipper is a deterministic network element, which subtracts packets according to a given function. This way, e.g., a congested router can be modeled by a lossless network element describing its scheduling behavior in conjunction with a clipper. In this style, Agharebparast and Leung use network calculus building blocks to model a wireless channel [3]. These are useful to model wireless channels, where the service offered varies over time.

Alternative approaches that allow an analysis of packet loss are presented by Agrawal et al. [4], who describe a time varying service, and Chang et al. [20], who describe dynamic service guarantees.

3.2.2 Stochastic Extension to Network Calculus

The advantage of packet switching over circuit switching is that it allows for a link to be easily be shared by several data flows. This is known as statistical multiplexing and especially of interest when the traffic is bursty, which is a property often found in multimedia traffic. The underlying rationale is that flows do not want to send all the time. Therefore, it is not necessary that resources are provided for all flows to send at the same time. The amount of resources to be provided generally depends on the statistical properties of the flows and the accepted probability of QoS not being achieved. However, determining the appropriate amount of resources is usually tricky because the nature of statistical guarantees is that at certain times they are not given. On the other hand, the major advantage of statistical multiplexing is that the resources are utilized better.

There are several approaches to extending network calculus to a stochastic setting. A stochastic extension potentially has two fields of application. The first is to model packet loss. The second is to deal with alternative traffic and service descriptions.

This is of utmost importance as the requirement to be able to grasp all traffic and scheduling by arrival and service curves, respectively, is a major constraint.

A starting point is to assume stochastic traffic. There are a multitude of stochastic traffic descriptions that bound burstiness. There is much work on multiplexing such flows. This falls under the category of effective bandwidth approaches, to which a chapter is dedicated in the book by Chang [18]. An overview of statistical admission control schemes is given in the article by Knightly and Shroff [65]. Prominent stochastic traffic descriptions are the Exponential Bounded Burstiness [112] and Stochastically Bounded Burstiness [102] approaches. The underlying rationale is that the traffic flow is not described by its average rate and variance, but by its entire moment generating function. Chang [18] also establishes a connection between these models and network calculus, which is extended by Fidler [30].

One approach is to assume that the flows are not strictly constrained by an arrival curve, but allow a violation with a given probability. Similarly, a server can be defined that does not serve strictly according to its service curve but violates it with a given probability. This is approach followed by Jiang et al. [54, 55, 56, 74], who generalizes the arrival curve and service curve to a stochastic setting and derives analogies for some basic deterministic bounds and results. They also give an overview of stochastic extensions to network calculus [74].

Boorstyn et al. [11, 12] show how multiplexing gains can be achieved at a node for flows adhering to arrival curves by effective envelopes. Liebeherr et al. [73] extend these results to the end-to-end case. In the same framework, Liebeherr et al. [72] develop an effective service curve and derive end-to-end statistical guarantees for it. The same issue is revisited by Ciucu et al. [22]. Li et al. [71] connect this approach to effective bandwidth.

Vojnovic and Le Boudec [110] derive bounds for flows adhering to deterministic arrival curves being multiplexed at a node described by a service curve.

3.2.3 Network Calculus and Queuing Theory

A completely different approach to extend network calculus to a stochastic setting is combining it with queuing theory. The potential benefits become obvious, when comparing the two with regard to utilization and the certainty of QoS guarantees. Since network calculus considers the worst case, many times a low network utilization is achieved. However, this ensures that each flow receives the service that was assigned to it. Queuing theory on the other hand does not give any deterministic guarantees, but its stochastic nature ensures a good network utilization.

Schmitt studies priority queuing in a network calculus setting [94, 95]. Priority Queuing is a simple way to provide service differentiation to flows. The major drawback of priority queuing is that flows of the lower services classes might be starved. This happens when the flows of the higher classes send continuously. The rationale of this work is that the flows of the higher classes can be constrained by arrival curves which gives the lower classes a deterministic service.

In work by Pandit et al. [83] a novel approach to combining network calculus and queuing theory is taken by analyzing a M/M/1 queuing system bounded by network calculus elements. This is elaborated in Chapter 6.

3.3 Range of Applicability

The second requirement of a unified model for QoS performance analysis is it should be applicable to a wide range of networks. This point is closely related to the point of gaining insight on the relevant characteristics. However, here the focus is on the type of network and its properties. Historically, network calculus was developed in the early 1990s and intended for ATM and IntServ networks. A dominating characteristic of these networks is that service guarantees are given on a per flow basis. However, in some aspects the Internet did not develop as was anticipated in those days. The tremendous growth of the Internet and the number of flows that routers deal with was unexpected. Therefore, ATM and IntServ networks do not play an important role in today's networking research. A criticism that network calculus often encounters is that it is a model for outdated network architectures . Hence, from its introduction until today much network calculus research has gone into applying it to new network architectures. Of course, especially considering the older results, the border line between "basics" and "application to new architectures" is a gradual one. Here two types of research have to be distinguished. One is the development of tools which enable the study of new networks. This is closely related to the previous section regarding the insight on new characteristics. As representatives of this type, the aggregation of flows, feedback systems, application to general topologies and data scaling are discussed. The second group of research that falls in this category is applications to specific traffic and network types. Examples of such networks types are sensor networks and reconfigurable networks. The latter is further extensively described in Chapter 4. Beyond the topics presented in the following, network calculus is also used in the context of entirely different issues of networking. E.g., Thiele et al. [108] apply network calculus to the design of network processors.

3.3.1 Aggregation of Flows

It is not realistic in today's Internet to track the state on a per flow basis in routers. As discussed in Section 2.2, it has turned out that many times traffic flows are grouped into classes. Schmitt [96] points out a mechanism for obtaining the deterministic arrival curve of an aggregate of flows. There is much work on statistical aggregation, e.g., by Pandit et al. [82]. An arrival curve and service for the aggregate can easily be determined and performance bounds computed. One question that arises now is what service a flow in the aggregate experiences. A useful result is the worst case service curve after blind multiplexing, which is described on page 208 in the book of Le Boudec and Thiran [13]. Assume an aggregate of n flows, each constrained by an arrival curve $\boldsymbol{\alpha}_i(t), i = 1, \ldots n$ to share a link which offers the service curve $\boldsymbol{\beta}(t)$. Then the service that a flow $\boldsymbol{\alpha}_t(i)$ experiences in the worst case is the service curve $\boldsymbol{\beta_i}(t)$, which is given by

$$\boldsymbol{\beta_i}(t) = \max\{0, \boldsymbol{\beta}(t) - \sum_{j=1, j \neq i}^{n} \boldsymbol{\alpha}(i)\}. \tag{3.1}$$

A more sophisticated approach to determine delay bounds in aggregate scheduling is presented in work by Charny and Le Boudec [21]. Fidler extends the Pay Bursts Only Once principle (see Section 2.5.4) to aggregate scheduling [28]. Fidler also studies the impact of traffic shaping in aggregate scheduling networks [29].

3.3.2 Feedback Systems

Feedback systems originally stem from control theory. The rationale in control theory is that a parameter should be tuned such that a goal is achieved under the constraint that there is some sort of noise. A feedback element indicates that the decision how to tune the parameter is influenced by the current state of the system. A common application to networking is flow control. The tunable parameter is the sender rate, the goal is to ensure a fair yet sufficient rate for the flow and the noise is the behavior of other flows. Agrawal et al. [5] derive performance bounds for general flow control protocols using feedback systems. The most widespread flow control algorithm is undoubtedly TCP which has been studied in the network calculus framework. Baccelli and Hong [7] take an analytical approach and show that TCP is max plus linear. Kim and Hou [61] compare network calculus based simulations to packet level simulations of TCP.

3.3.3 General Topologies

Certain results, such as the aggregate scheduling, require feed forward networks. A feed forward network is one where all nodes in the network can be numbered in such a way that all routes consists of nodes with increasing numbers. However, in reality networks are not necessarily feed forward. Starobinski et al. [101] derive a mechanism called turn-prohibition with which arbitrary networks can be made into feed forward networks. The basic idea is that certain links are not allowed to be used for routing packets. Since this implies a lower utilization of the network, their mechanism ensures that the least possible number of links are removed, which are at the most $\frac{1}{3}$ of the links. Fidler et al. [31] ensure the independence of flows in general topologies.

3.3.4 New Network Elements

Network elements such as firewalls, proxies or media gateways conduct a vast variety of transformation processes on the flows such as coding, decoding and compression.

The challenge is that these operations take place on different layers but influence the packet flow on the network layer. A generic method to incorporate them into a network calculus analysis is data scaling, which is proposed by Fidler and Schmitt [35].

3.3.5 Application to Specific Traffic Types

Originally network calculus uses a very abstract traffic model, namely the upper bound. This has to be brought into relation with the real traffic that is sent over networks. An interesting traffic type to look at in this context is video traffic, as it is the most challenging for the network in the sense that its rate and burstiness, which stems from the compression, are high. Further, depending on the application, it is often subject to real time constraints, such as video conferencing. Graf analyzes network calculus based traffic shaping for compressed video data [40, 41].

3.3.6 New Network Paradigms

Another frontier in extending the range of applicability is enabling a network calculus analysis of new network architectures and paradigms. In recent work Schmitt and Roedig [97, 98] apply network calculus to sensor networks. They address two properties that sensor networks often have. Their topology is uncertain and they require a deterministically predictable network behavior. Especially the latter makes a network calculus analysis destined for this issue. Schmitt and Roedig build a framework for analyzing sensor networks with network calculus. They derive the worst case topology that the sensor network can take and compute bounds for throughput and delay for that case.

A further new network paradigm is reconfigurable networking. Those are networks, where the logical topology can change over time in a controlled manner. One area where reconfigurable networks appear is optical networking. In a nutshell, the nodes are connected to each other physically through optical fiber and by assigning wavelengths, different logical topologies can be created. In Chapter 4 of this dissertation a network calculus analysis of such a network is derived. Beyond that it is shown how new network paradigms can generally be translated into network calculus issues. The underlying network architecture is described by Baldine and Rouskas [8], as well as Brzezinski and Modiano [17], the latter being the work Chapter 4 builds upon. Related work is the network calculus analysis of a switch serving flows with a credit based system [27]. Previous work on QoS in optical networks include work by Guan et al. [43], but also the dissertation of Dolzer [26].

3.4 Transparence and Ease of Applicability

The last property of the envisioned model is that it is transparent and easy to use. It is especially of use if the results are accompanied by intuitive graphical representa-

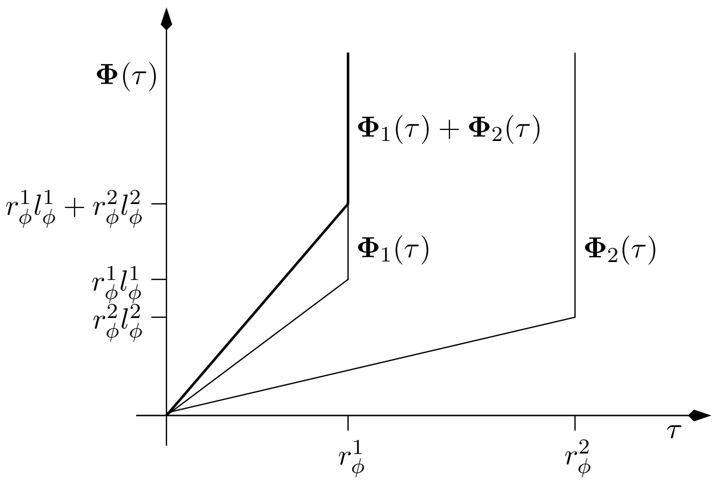

Figure 3.2: Sum of transformed service curves in the Legendre domain

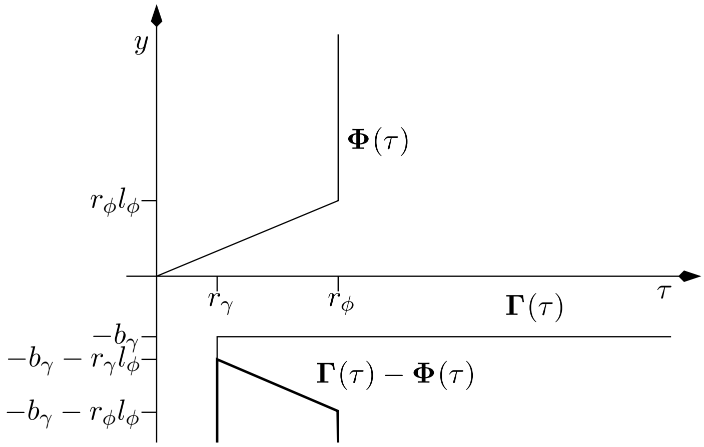

Figure 3.3: Output bound in the Legendre domain

tions. These are properties that the transforms offer. Therefore, the development of transforms for network calculus is of interest.

3.4.1 Transforms

As pointed out in Section 2.3.2, transforms are a popular tool in conventional system theory. Transforming a signal essentially means representing it in a different way. There are two main efforts to develop a transform for network calculus. One is by Pandit et al. [78, 81, 80] and the other by Fidler and Recker [33, 32, 34]. The first is described in detail in Chapter 5 of this dissertation. Both were developed independently and use the Fenchel transform as a starting point. The Fenchel transform is described in the book by Bacelli et al. [6]. It states that the convolution in the time domain corresponds to taking the sum in the Fenchel domain. This is analogous to transforms such as the Fourier, Laplace or Z transform in conventional system theory,

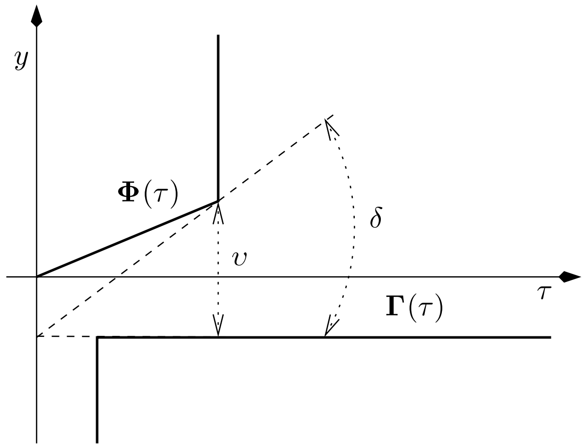

Figure 3.4: Delay and backlog bound in the Legendre domain

where the convolution in the time domain corresponds to taking the product in the frequency domain. However, a drawback of the Fenchel transform is it only applies to convex functions. Fidler and Recker introduce the Legendre transform, which extends the Fenchel transform in the sense that it can be applied functions which are either convex or concave. They derive graphical representations of the output, delay and backlog bounds as well as the concatenation of service curves. In contrast, Pandit et al. focus on the development of a transform for general functions, i.e., functions which are neither convex nor concave. This is, as mentioned above, the subject of Chapter 5.

In the remainder of this section the results from Fidler and Recker are presented descriptively. The mathematically inclined reader is referred to their paper [34].

Convex functions are transformed into the Legendre domain by the same operator as the Fenchel transform. The definition and equation of the Fenchel transform is given in Section 5.2.1, here we limit ourselves to a graphical description.

Two latency rate service curves in the Legendre domain

$$\boldsymbol{\Phi}_1(\tau) = \mathcal{D}\{\boldsymbol{\phi}_1(t)\} \tag{3.2}$$

and

$$\boldsymbol{\Phi}_2(\tau) = \mathcal{D}\{\boldsymbol{\phi}_2(t)\} \tag{3.3}$$

and their sum are shown in Figure 3.2. Underlying are two latency rate service curves $\boldsymbol{\phi}_1(t)$ and $\boldsymbol{\phi}_2(t)$. Their latencies are l_ϕ^1 and l_ϕ^2, and their rates r_ϕ^1 and r_ϕ^2, respectively. The latency l_ϕ is the slope of the Legendre transform $\mathcal{D}\{\boldsymbol{\phi}_1(t)\}$ and the rate r_ϕ is where it jumps to $+\infty$. Therefore, if two such transforms are added in the Legendre domain, the result is a function with the added slopes and which jumps to $+\infty$ at the smaller of the two rates. Transforming this back yields a rate latency curve with the summed latency and a rate which is the lower of the two rates. Note that the rates are mapped to the τ-axis. This explains why this transform is only suited for convex functions. Only increasing rates can be handled.

In a similar fashion, the output bound of an arrival curve and service curve can be determined by subtracting the service curve from the arrival curve in the Legendre domain. This is shown in Figure 3.3.

The min-plus deconvolution corresponds to the subtraction $\mathbf{\Gamma}(\tau) - \mathbf{\Phi}(\tau)$ in the Legendre domain. The resulting functions jumps from $-\infty$ to $-b_\gamma - r_\gamma l_\phi$ when $\tau = r_\gamma$. Then it decreases until it reaches $-b_\gamma - r_\phi l_\phi$ at $\tau = r_\phi$ and is $-\infty$ thereafter. The relationship of the parameters of the output bound in the time domain and Legendre domain can be seen by comparing this resulit to Figure 2.9 in Section 2.5.4.

The Legendre domain further offers representations of the delay and backlog bound (cf. Section 2.5.4). This is shown in Figure 3.4.

The backlog bound is the minimum vertical distance of the arrival curve and service curve in the Legendre domain. The delay bound is the difference of the slopes of the tangents that intersect in the ordinate.

Therefore, the Legendre transform offers some intuitive graphical representations of certain network calculus results. As mentioned above, the drawback is that only functions that are strictly concave or strictly convex can be handled.

4 Modeling Dynamically Reconfigurable Networks with Network Calculus

The vision of this dissertation is a model with which networks can be described as systems with input and output. The traffic which arrives at a network can then be mapped to traffic which leaves the network. This enables networks to be analyzed elegantly and without running expensive simulations. As pointed out in Section 2.3 the system can be described by the impulse response. The output is then computed by the min-plus convolution or, depending on the context, min-plus de-convolution, of the input function and the impulse response. Therefore, it is easily possible to obtain the corresponding outputs for different inputs.

One requirement to be met in order to develop network calculus into a universal system theory for networks is to describe real-world networks as network calculus systems. The task is to specify how network architectures and algorithms translate into network calculus systems. Translating a network into a network calculus system means finding a service curve, which describes the worst case behaviour of the network. Recall that the value of a service curve $\boldsymbol{\beta}(t)$ denotes the amount of bits that must have passed through the network after time t (cf. Definition 2.7). Therefore, one way to derive the service curve is to determine from the network properties the number of packets that are served in the worst case for each t. This is our method of choice.

We apply it to dynamically reconfigurable IP-over-WDM networks in the following. With the emergence of optical Wavelength Division Multiplexing (WDM) networks, their interplay with IP networks are of crucial research interest. In work by Brzezinski and Modiano [17] dynamic reconfiguration algorithms for IP-over-WDM networks have been devised. We analyse such a network from the viewpoint of QoS-sensitive traffic, which is undoubtedly gaining a bigger share in IP networks. The key characteristic of multimedia traffic is that it requires deterministic QoS. Therefore, the subject of this chapter is to get insight on the QoS a IP-over-WDM network can provide. The QoS parameters of interest are the throughput and delay.

The remainder of this chapter is outlined as follows. In the next section we describe the reconfigurable network model. We then in Section 4.2 show how TDM schemes are constructed. In Section 4.3 a service curve representation of switching schemes is given. We discuss QoS tradeoffs in switching schemes in Section 4.4. The results are compared to the average results with stochastic traffic in Section 4.5. In Section 4.6 directions how to apply the results obtained in this chapter are presented. Section 4.7 summarizes this chapter.

4.1 Reconfigurable Networks

4.1.1 Optical Networking

As described in Section 2.1, a network is a set of nodes which are connected via links, on which information can be sent from one node to another. In optical networks, the links that interconnect nodes consist of optical fiber and the information is exchanged via optical signals. By using different wavelengths, different streams can be multiplexed over the same fiber without interfering with each other. This is known as WDM. The widely used expression IP-over-WDM is slightly misleading, as WDM is the multiplexing technique for optical networking but not a protocol of the physical or data link layer. For a detailed treatment of optical networking, the reader is referred to the dissertation of Dolzer [26].

A node in an optical network is equipped with four basic components beyond the IP router functionality. A transceiver sends and receives optical signals. The number of incoming and outgoing links of each node is determined by the number of transceivers. An add/drop multiplexer converts optical signals into electronic signals, i.e., IP packets, and vice versa. An optical cross-connect, also known as optical switch, moves optical signals of one fiber to another, without affecting the IP layer. If IP packets are to be sent to a neighbouring node, they are converted by the add/drop multiplexer to optical signals, sent over the optical fiber by the transceiver, picked up by the transceiver of the destination node and converted by the add/drop multiplexer of the destination back to IP packets. If the sending and receiving node are not neighboring, the optical cross-connects of the nodes along the path can be configured such that the optical signal is switched through without being seen or affected by the IP layers of any of the nodes along the path. From the viewpoint of the IP layer, the sending node and receiving node are logically neighbors, even though not physically. Hence, a vast variety of logical topologies can be fixed by configuring lightpath interconnections to create logical links between nodes. The limitations of such a network are given by the number of different wavelengths that can be assigned and the number of transceivers.

A network endowed with this property of being able to change the logical topology is called a *reconfigurable network*. The distinguishing property to conventional networks is that packet forwarding is not solely done by routing but by a combination of routing and reconfiguration.

The intervals at which the reconfiguration takes place ("granularity")can vary. Reconfiguration lasting for an entire session is referred to as optical circuit switching, which is further divided into optical label switching and optical flow switching. Before reconfiguring, networks employing optical circuit switching need reserve resources along the entire path. Since this requires waiting for acknowledgements of the reservation being successful, this takes a relatively long time. One reimports all problems of circuit switching, the foremost being that it is it not suited for bursty data. On the other extreme, when reconfigurations made on a packet basis, it is referred to as optical packet switching. However, today's technology does not allow for a widespread deployment of optical packet switching. A compromise is optical burst switching. In optical burst switching, the reconfigurations are such that some data is allowed to collect at the IP layer and then the data is served as a burst. The sender announces by a control packet that it is about to send, thus the time-consuming end-to-end reservation is avoided. However, collisions are possible. In the following, optical networks which employ a frame based scheduling, i.e., do not require reconfigurations on a packet-by-packet basis, are of primary interest.

4.1.2 Reconfigurable network model

In this section we describe the network that we will analyse with network calculus methods. Mainly the system model from [17] is adopted. In the following, we make the following restrictions. We assume a line network with the number of nodes being $n = 3$. The traffic can be described by a traffic matrix, where each entry (i, j) denotes the traffic from node i to node j. The nodes can be configured to any topology, There are 3 wavelengths per fiber and each node has at most 1 incoming and 1 outgoing link. The service rate of each logical link is normalized to 1. There are $3! = 6$ maximal configurations, that are configurations where all wavelengths are assigned. The configurations $\underline{C}_i$ are given by $n \times n$ square matrices filled with 0's and 1's, where a 1 denotes that the link from the node indexed by the row number to the node indexed by the column number is fixed,

$$\underline{C}_i = \begin{pmatrix} c_{1,1} & \cdots & c_{1,n} \\ \vdots & \ddots & \vdots \\ c_{n,1} & \cdots & c_{n,n} \end{pmatrix}, \begin{cases} c_{i,j} = 1 & \text{, if a logical link between } i \text{ and } j \text{ exists} \\ c_{i,j} = 0 & \text{, if no logical link between } i \text{ and } j \text{ exists} \end{cases}. \tag{4.1}$$

Therefore, in each row and in each column there is exactly one entry 1. Since a node can not be connected to itself, an entry 1 in the diagonal indicates that that node is left idle. Matrices in the form as given in Equation 4.1 are also called *control matrices.* Two of them are the ring topologies $\underline{C}_1 : 1 \to 2 \to 3 \to 1$ and $\underline{C}_2 : 1 \to 3 \to 2 \to 1$. They are depicted in Figure 4.1. The dashed lines denote the physical toplogy.

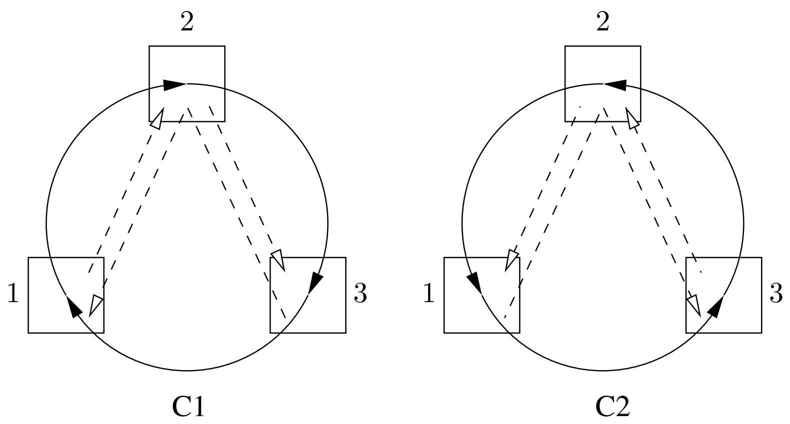

Figure 4.1: Ring topologies

Consider the topology $\underline{C}_1$ on the left. 3 wavelengths are assigned. Each is responsible for one logical link. The transceiver in node 1 is configured such that its IP packets are converted to the wavelength connecting nodes 1 and 2. Further, it picks up the wavelength which receives packets from node 3. This wavelength is assigned for both physical links between nodes 3 and 2 and nodes 2 and 1, where the optical cross-connect in node 2 switches it through. The transceiver in node 3 transmits packets on the wavelength connecting it with node 1 and picks up the wavelength which receives packets from node 2. Therefore, node 2 uses all 3 possible wavelengths, while the other nodes only require two.

The control matrices of the ring topologies $\underline{C}_1$ and $\underline{C}_2$ are the following, respectively,

$$\underline{C}_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \tag{4.2}$$

and

$$\underline{C}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{4.3}$$

Further, there are three possibilities to connect two nodes and leaving one idle, e.g., $\underline{C}_3$ connecting the links $1 \to 2$ and $2 \to 1$, while leaving node 3 idle. This is depicted in Figure 4.2. The control matrices for $\underline{C}_3$, $\underline{C}_4$ and $\underline{C}_5$ are

$$\underline{C}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{4.4}$$

$$\underline{C}_4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{4.5}$$

and

$$\underline{C}_5 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.6}$$

We call topologies with idle nodes *incomplete topologies*. The sixth topology is the pathological case of all three nodes being idle.

Each node is equipped with $(n-1)$ virtual output queues, where each virtual output queue, denoted by $V_{i,j}$, contains the backlogged data from node i to j. While Brzezinski and Modiano assume the virtual output queues to be infinitely large, we propose a method to dimension them. Time is assumed to be slotted and data units are in the form of fixed length packets, each requiring one single slot for transmission. Fragmentation can be used to hanlde variable length packets. The network allows a maximum of one packet to be transmitted over a link in one time slot. The network may initiate a logical topology reconfiguration, under which existing links are torn

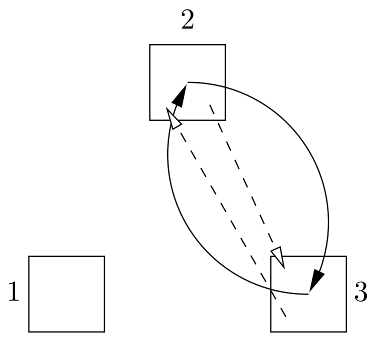

Figure 4.2: Topology with one idle node

down and re-established to form a new logical topology. The affected links are forced to be idle for the certain time called the *tuning latency*, which is denoted by D and given in time slots. Unaffected links may continue to service traffic during the reconfiguration. We call the switch between two topologies that have no links in common a *disjunctive reconfiguration*. The queue occupancy process $\{\underline{\mathbf{X}}[k]\}_{k=0}^{\infty}$ is defined as an infinite sequence of matrices, where $\underline{\mathbf{X}}[k]$ is the queue backlog matrix at time k and $\mathbf{X}_{i,j}[k]$ is the number of packets at node i destined for node j at time k. This process evolves according to the matrix equation

$$\underline{\mathbf{X}}[k+1] = \max\{0, \underline{\mathbf{X}}[k] - \underline{C}_I[k+1]\} + \underline{I}[k+1] \tag{4.7}$$

for $k > 0$. In Equation 4.7, $\underline{C}_I[k]$ denotes the instantaneous control matrix, i.e., the topology switched at time instant k. Accordingly, $\underline{I}[k]$ denotes the instantneous arrival matrix, i.e., the arrivals at time instant k. Without loss of generality we assume $\underline{\mathbf{X}}[0]$ to be 0. Regarding the arrivals our model differs from the cited work [17]. They assume the arrivals to be a stochastic process with a rate, where the only further constraint is that the process satisfies a strong law of large numbers. In contrast we assume a deterministic traffic model for the arrivals. The reason for this is that in order to guarantee deterministic QoS for multimedia flows, there must be a strict upper bound on the sent traffic. The advantage of this model over stochastic traffic is that the QoS characteristics of the multimedia traffic can be determined elegantly. Therefore, we assume the traffic to satisfy arrival curves from network calculus. The arrival matrices are of the form

$$\underline{\boldsymbol{\alpha}}[k] = \begin{pmatrix} \boldsymbol{\alpha}_{1,1}[k] & \boldsymbol{\alpha}_{1,2}[k] & \boldsymbol{\alpha}_{1,3}[k] \\ \boldsymbol{\alpha}_{2,1}[k] & \boldsymbol{\alpha}_{2,2}[k] & \boldsymbol{\alpha}_{2,3}[k] \\ \boldsymbol{\alpha}_{3,1}[k] & \boldsymbol{\alpha}_{3,2}[k] & \boldsymbol{\alpha}_{3,3}[k] \end{pmatrix}. \tag{4.8}$$

Each $\boldsymbol{\alpha}_{i,j}[k]$ denotes the arrival curve of the QoS critical traffic aggregate from node i to j. Our results do not depend on specific arrival curves to be used. Note the difference between the arrival matrix defined in Equation 4.8 and the instantaneous arrival matrix from Equation 4.7. The entries of the former are arrival curves, whereas $\underline{I}[k]$ denotes the number of packets that arrive in the time slot k.

The attractiveness of reconfiguration is that it allows traffic matrices to be served that can not be served with fixed topologies. Consider an arrival matrix with rate-limited flows

$$\underline{\boldsymbol{\alpha}}_1[k] = \begin{pmatrix} 0 & 0.2k & 0.4k \\ 0.3k & 0 & 0.5k \\ 0.1k & 0.7k & 0 \end{pmatrix}. \tag{4.9}$$

If a topology were to be fixed permanently, say $\underline{C}_1$, then the traffic for the links not connected logically would have to be routed over two hops. E.g., traffic from node 1 to node 3 would have to be routed via node 2, therefore the link from node 2 would hold the traffic from node 1 to node 2 as well as from node 1 to node 3, which exceeds the link capacity. This implies the arrival matrix

$$\underline{\boldsymbol{\alpha}}_2[k] = \begin{pmatrix} 0 & 1.3k & 0k \\ 0k & 0 & 1.2k \\ 1.1k & 0k & 0 \end{pmatrix}. \tag{4.10}$$

It can not be served as the sustained rates exceed the service rates. Managing the reconfigurations such that a large class of arrival matrices can be served is the subject of Section 4.2.

Due to the reconfigurations and their associated tuning latencies it can happen that the traffic entering the network exceeds the traffic leaving it. The *stability* of the queues is of crucial interest.

Definition 4.1 *A queue as defined in Equation 4.7 is deemed stable if the long-term arrivals do not exceed the packets that can be served.*

The only parameter of the arrival curve that is relevant for stability is the sustained rate. We label the sustained rate of each arrival curve $r^s_{i,j}$. Since the long-term arrivals of any queue will not exceed its sustained rate, serving the queue with that rate is sufficient. However, the other properties of the arrival curve, e.g., the burst size, influence the buffer needed for that queue. This is discussed in detail in Section 4.3. The challenge in achieving stability is that the frequency of reconfiguations has to be taken care of as they cause idle time slots. Brzezinski and Modiano show [17] that stability can be achieved despite arbitrarily large tuning latencies. Their result states the contraints for switching topologies. We call the sequence of topologies a *time-division multiplex (TDM) scheme.* A TDM scheme is given by a *period*, i.e., the layout is repeated over and over again. Since often TDM schemes constitute the same topology to be fixed for several consecutive time slots, we define *frames.*

Definition 4.2 (Frame) *We define a frame* w_i^f *as a sequence of* f *time slots in which one topology* $\underline{C}_i$ *is fixed. The variable* f *is called the frame size.*

In the following section an algorithm to obtain TDM schemes guaranteeing stability is devised. Note that this algorithm works regardless of deterministic or stochastic arrivals. This is due to the fact that the properties of deterministic traffic are subjected to more stringent limitations than the properties of the underlying stochastic traffic.

4.2 Constructing TDM Schemes

4.2.1 Algorithm

In this section we apply the results from Brzezinski and Modiano [17] to obtain possible TDM schemes from an arrival matrix $\underline{\boldsymbol{\alpha}}[k]$. As mentioned, for constructing stable TDM schemes only the sustained rates of the arrival curves are relevant. We first define the value σ_A as

$$\sigma_A = 1 - max\{max_i \sum_j r^s_{i,j}, max_j \sum_i r^s_{i,j}\}. \tag{4.11}$$

The TDM scheme is best found by decomposing the arrival matrix according to the Birkhoff-van Neumann method [19]. Applying this method requires the sum of all rows and columns of the arrival matrix to be equal. Thereby, we require the notion of a dominating arrival matrix.

Definition 4.3 *An arrival matrix $\underline{\boldsymbol{\alpha}}^*[k]$ is called a dominating arrival matrix of $\underline{\boldsymbol{\alpha}}[k]$ if $\boldsymbol{\alpha}^*_{i,j}[k] \geq \boldsymbol{\alpha}_{i,j}[k]$ for all i, j, k.*

Next a dominating matrix $\underline{\boldsymbol{\alpha}}^*[k]$ is to be found such that the sustained rates of all rows and columns sum up to σ_A, while leaving the other parameters unaltered. Obviously any TDM scheme that services this sufficiently services the original arrival matrix. The matrix $\underline{\boldsymbol{\alpha}}^*[k]$ can be decomposed by the Birkhoff-van Neumann method. The result is a sum in the form

$$\underline{\boldsymbol{\alpha}}^*[k] = x_1 \cdot \underline{C}_1 + x_2 \cdot \underline{C}_2 + \ldots + x_n \cdot \underline{C}_n. \tag{4.12}$$

The coefficients x_i denote the least value of the fractions that the respective topology $\underline{C}_i$ has to be fixed. It can be shown that

$$\sigma_A = 1 - \sum_i x_i. \tag{4.13}$$

Which topology is fixed for the remaining fraction σ_A does not matter if $D = 0$. Tuning latencies $D > 0$ are overcome by cleverly making use of the remaining fraction. This is done by constructing suitable frames. To handle the tuning latency, a key result is that

$$f > \frac{D}{\sigma_A}. \tag{4.14}$$

Note that while Inequality 4.14 always holds, in certain cases even less than equality is sufficient. By constructing frames of a sufficiently large size the stability criterion is taken care of. This is regardless of the layout of the frames. In the worst case, in one frame $(f - D)$ packets of each link can be served. The worst case applies when there is a disjunctive reconfiguration. Recall that the links that are unaffected by the reconfiguration continue to service packets. Obviously, if two frames of the same topology are employed consecutively, there are no time slots lost due to tuning latency, and therefore the throughput is high. On the other hand, the more frames of the same topology are fixed, the higher is the worst case delay of all other topologies. This trade-off, which determines the QoS, will be studied in the following sections.

Equation 4.12 states the fractions of slots at which the topologies must be switched. There are several switching schemes that achieve this. They can be classified as periodic and aperiodic switching schemes. In the periodic switching scheme, the same layout of frames is repeated over and over again. It has to be ensured that the appropriate fractions are maintained in the period. In an aperiodic scheme, it is decided on the basis of the current state, which topology is switched. The most prominent example of an aperiodic scheme is Maximum Weight Matching (MWM). In MWM, at each slot the topology is switched according to the rule such that the sum over all queues that the switched links serve is maximized. Even though this is stable, it can happen that queues are starved arbitrarily long. Therefore, MWM based algorithms are not suited when targeting deterministic QoS and in this work only periodic schemes are considered.

4.2.2 Numerical Example

A prototypical example illustrates the above algorithm. Recall the arrival matrix $\underline{\boldsymbol{\alpha}}_1[k]$ consisting of rate limited flows,

$$\underline{\boldsymbol{\alpha}}_1[k] = \begin{pmatrix} 0 & 0.2k & 0.4k \\ 0.3k & 0 & 0.5k \\ 0.1k & 0.7k & 0 \end{pmatrix}. \tag{4.15}$$

A dominating matrix is then

$$\underline{\boldsymbol{\alpha}_1}^*[k] = \begin{pmatrix} 0.3k & 0.2k & 0.4k \\ 0.4k & 0 & 0.5k \\ 0.2k & 0.7k & 0 \end{pmatrix}. \tag{4.16}$$

This can be decomposed into

$$\underline{\boldsymbol{\alpha}_1}^*[k] = 0.2k \cdot \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} + 0.4k \cdot \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} + 0.3k \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \tag{4.17}$$

Therefore,

$$\underline{\boldsymbol{\alpha}_1}^*[k] = 0.2 \cdot \underline{C}_1 + 0.4 \cdot \underline{C}_2 + 0.3 \cdot \underline{C}_3. \tag{4.18}$$

Set the tuning latency to $D = 1$. With Inequality 4.14 we obtain $f > 10$. It will follow that this is one of the cases where equality in Inequality 4.14 is sufficient. Setting $f = 10$, we find that if 1 time slot is needed for reconfiguring, 9 packets are transmitted in a frame. In the worst case, reconfiguration of every link takes place before every frame, which allows for 9 packets to be sent on each link per frame. The decomposition yields that 20% of the time slots must be of topology $\underline{C}_1$, 40% of $\underline{C}_2$ and 30% of $\underline{C}_3$. The number of packets served on each link meets the requirement from the arrival matrix. E.g., considering the link $3 \rightarrow 2$ with $r^s_{3,2} = 0.7k$, in the worst case in one period $0.7 \cdot 10 \cdot 9 = 63$ packets arrive. Link $3 \rightarrow 2$ is served in topologies $\underline{C}_2$ and $\underline{C}_3$, therefore in one period maximally $(4+3) \cdot 9 = 63$ can be served, which fulfills the requirement. Recall there is no tuning latency for the links that remain unchanged when re-tuning. A similar calculation can be done for each virtual output queue $V_{i,j}$. Obviously there are several different layouts in which the frames within one period can be fixed. One is to bulk all frames of each topology. This is labeled as L_1, where

$$L_1 =< w_1^{10}, w_1^{10}, w_2^{10}, w_2^{10}, w_2^{10}, w_2^{10}, w_3^{10}, w_3^{10}, w_3^{10} > . \tag{4.19}$$

The layout L_1 is depicted in Figure 4.3. The dotted lines indicate the beginning and the end of a frame. The dashed lines indicate the beginning and the end of a period. Note that it is arbitrary where the limits of the period are set. Each frame consists of 10 time slots where a grey slot indicates an idle time slot. Another is to alternate the

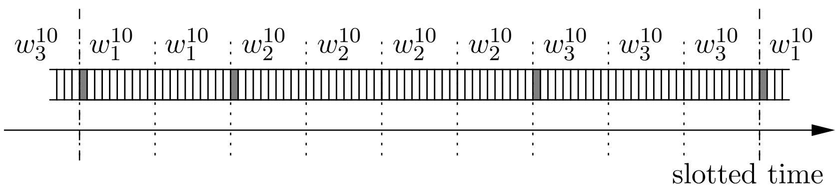

Figure 4.3: Layout L_1

frames as much as possible. This is labeled as L_2, where

$$L_2 =< w_2^{10}, w_3^{10}, w_2^{10}, w_1^{10}, w_2^{10}, w_3^{10}, w_2^{10}, w_1^{10}, w_3^{10} > \tag{4.20}$$

and depicted in Figure 4.4. In the next section we show how service curves are derived

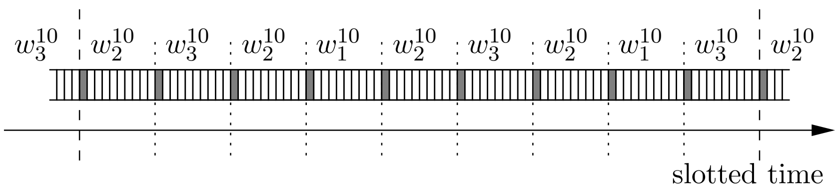

Figure 4.4: Layout L_2

from TDM schemes.

4.3 Service Curve Representation of Switching Schemes

A TDM scheme can be represented by a Service Curve. Recall that a service curve $\boldsymbol{\beta}[k]$ denotes how many packets are minimally served when k time slots have elapsed. Note that each queue $V_{i,j}$ has its own service curve depending on which configurations it is served in. The service curve is constructed, by computing the number of packets that are served within k time slots in the worst case, after an infinitely large number of packets arrive at an empty buffer. All topologies serving the queue of interest are to be considered. To illustrate this we revisit the example of Subsection 4.2.2. Consider the virtual output queue $V_{1,3}$. It is only served when topology $\underline{C}_2$ is fixed. In the layout L_1, the worst case packets from $V_{1,3}$ can arrive is in the time slot right after the switch from $\underline{C}_2$ to $\underline{C}_3$. In this case they have to wait for 5 frames plus 1 slot tuning latency until it can be served. Then $40 - 1 = 39$ packets are served consecutively. After this the period repeats, 5 frames plus 1 slot tuning latency idle time are followed by 39 packets served. This is denoted $\boldsymbol{\beta_1}[k]$ and depicted in Figure 4.5. This is a new type of service curve which we call *periodic service curve.*

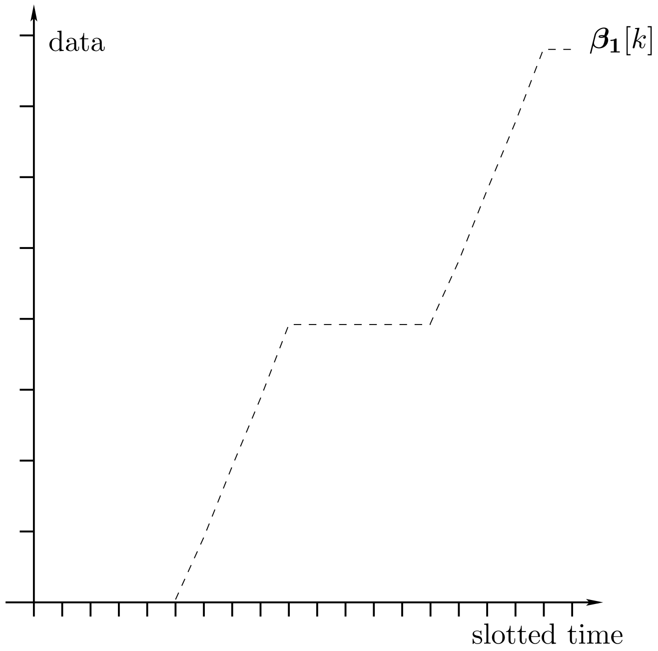

Figure 4.5: Service curve of L_1

In contrast, L_2 is the layout where frames are alternated as much as possible. Constructing its service curve is slightly more complex. The worst case for packets to arrive is the time slot, where a frame w_1^{10} is followed by another frame w_1^{10}. These are the last two frames in Equation 4.20. They have to wait for 2 frames plus 1 slot tuning latency until $10 - 1 = 9$ of them are served, which is by the first w_2^{10} in Equation 4.20. Then follow always one idle frame and one $\underline{C}_2$ frame, until the period repeats. The service curve $\boldsymbol{\beta_2}[k]$ of L_2 is depicted in Figure 4.6. Note that such service curves can be integrated into any network calculus analysis of networks with several elements.

4.4 QoS Trade-offs in Switching Schemes

Intuitively, one could assume that the service curve of $\boldsymbol{\beta_2}[k]$ is always superior to $\boldsymbol{\beta_1}[k]$. However, this is not the case. Since there are more reconfigurations needed in L_2, the number of idle time slots is higher which results in lower throughput. In our example the difference is nominal but it increases in settings with a higher ratio of $\frac{D}{f}$. Besides

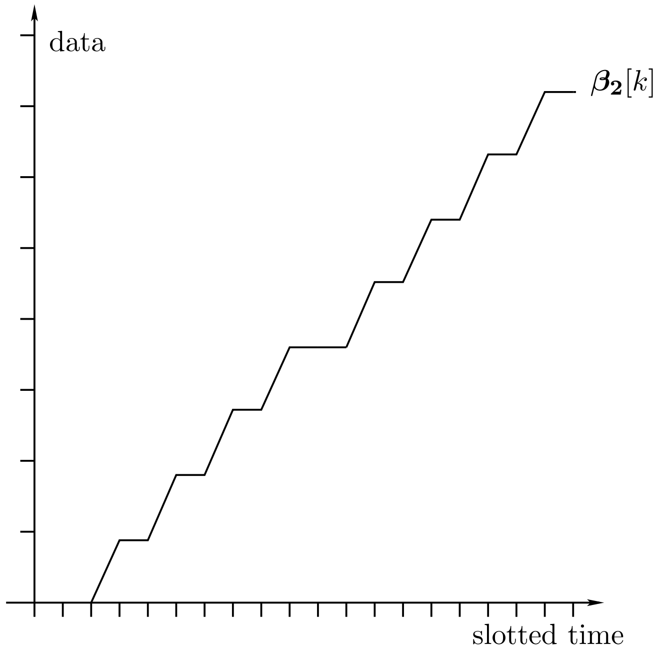

Figure 4.6: Service curve of L_2

the service curves $\boldsymbol{\beta_1}[k]$ and $\boldsymbol{\beta_2}[k]$, in Figure 4.7 also selected token bucket arrival curves, labeled $\boldsymbol{\alpha}_1[k]$ and $\boldsymbol{\alpha}_2[k]$, are shown. For the arrival curve $\boldsymbol{\alpha}_1[k]$ the layout L_1 results in a lower worst case delay, which is denoted by δ_1. In contrast, layout L_2 yields a lower worst case delay, denoted by δ_2, for arrival curve $\boldsymbol{\alpha}_2[k]$. A useful heuristic is to consider the relationship of the burst size to the period. A TDM scheme such as L_2 is mostly better suited if the burst can be served within one period. Whereas if the burst is so large that it needs several periods to be served, many times TDM schemes in the form of L_1 perform better, as the increase in throughput pays off. Above a certain burst size, L_1 is always superior as the increased throughput causes its service curve $\boldsymbol{\beta_1}[k]$ to be always left of $\boldsymbol{\beta_2}[k]$.

Next we consider the case where the rate of the arrival curve is the rate that the underlying queue requires. This is depicted in Figure 4.8. The arrival curves with the same burst sizes as in Figure 4.7 are shown. It can be observed that in this particular case $\boldsymbol{\beta_2}[k]$ yields a lower delay bound, denoted by δ_3 and δ_4, for both arrival curves, $\boldsymbol{\alpha}_3[k]$ and $\boldsymbol{\alpha}_4[k]$, respectively. However, since $\boldsymbol{\beta_1}[k]$ serves a rate higher than required, it will eventually exceed $\boldsymbol{\beta_2}[k]$. In this example with a relatively small tuning latency this effect is hardly visible, nevertheless there can be scenarios where it is more dominant. The thin line indicates, that when the long term arrival rate and long term service rate are equal, replacing the service curve with a rate-latency service curve yields the same delay bound.

The backlog of each virtual output queue is bounded and therefore the buffer requirement can be determined. Recall that the backlog bound, introduced in Theorem 2.10, is given by the maximum vertical deviation between the arrival curve and service curve.

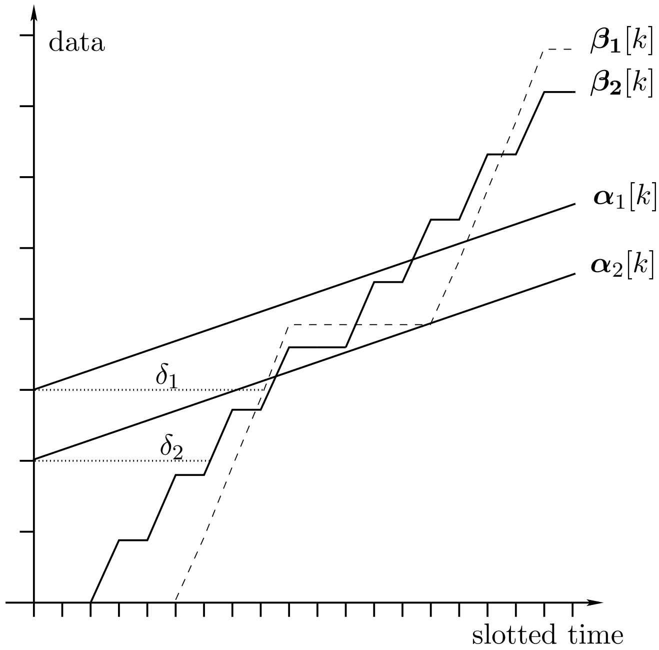

Figure 4.7: Service curves of L_1 and L_2

In Figure 4.9 the maximum backlog of a flow with arrival curve $\boldsymbol{\alpha}_1[k]$ is depicted for the layouts L_1 and L_2. Taking the sum over all flows of a node yields the total buffer requirement of this node.

The bottom line is that there is no layout that is universally superior to all others, especially when taking into account that we so far only considered a manipulating fraction of the potential turning knobs. One turning knob is to extend the analysis from 1 virtual output queue in the 3 node case to the n node case with all, i.e., $n \times (n-1)$, virtual output queues. In order to consider all virtual output queues, an optimization problem has to be stated which ensures that in all virtual output queues the delay requirement is met. Obviously the complexity of this optimization problem increases with the number of nodes, of which we assume approximately 20 in a realistic setting. Further, there is some degree of freedom to obtain the used topologies and their fractions. First the choice of dominating matrix is flexible. Then several decompositions according to Equation 4.12 are possible. Finally, many permutations of frames within a period are possible.

4.5 Comparison with Average Delay Results

In this section the worst case results obtained above are compared to the average results from Brzezinski and Modiano [17]. For this it is useful to take a closer look at their results. In a nutshell, the core of their work is the numerical evaluation of several reconfiguration algorithms. All the considered reconfiguration algorithms are based on MWM and are therefore dynamic. As mentioned above, such dynamic algorithms have

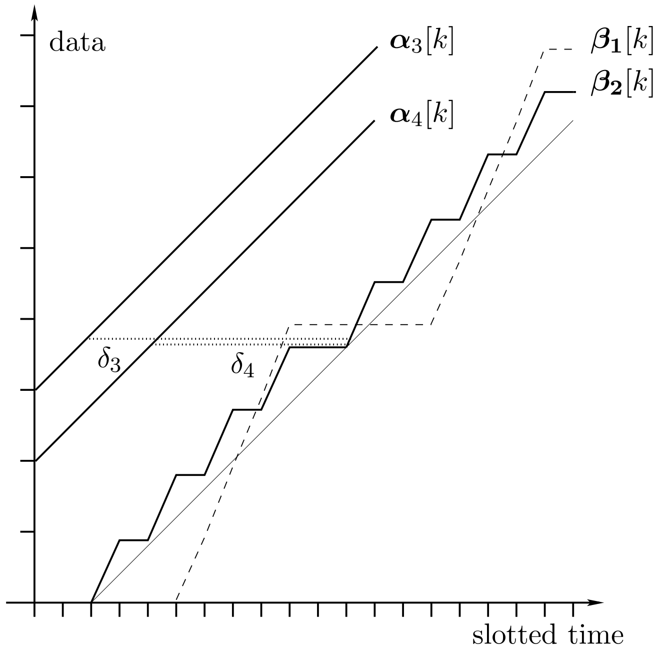

Figure 4.8: Service curves of L_1 and L_2

the drawback of allowing pathological cases with arbitrary high delays for some packets. Further, multi-hopping is considered. So far an assumption has been that packets in a queue $V_{i,j}$ stay in the queue until a link between the nodes i and j is switched. Multi-hopping is an alternative to this, where packets can be routed over several hops. However, multi-hopping is excluded from the network calculus considerations.

4.5.1 The Simulation Setup for the Average Delay

In Figure 7 of the motivating work [17] the simulation results of the average delay for several throughput levels are given. This is the simulation to which we compare our results, hence its setup is outlined in the following. In this simulation MWM and three other algorithms are evaluated. We only consider the MWM simulation as the other algorithms are variations, e.g., allowing multi-hopping, of it. The goal is not to have a competitive comparison. This would anyway be inappropriate, as we compute a worst case delay bound based on deterministic traffic as opposed to the average delay based on stochastic traffic. The goal is rather to have a qualitative comparison of the results. The number of nodes is set to $n = 6$ and the tuning latency to $D = 1000$ slots. For each throughput level, random arrival matrices are constructed and normalized to the desired throughput level. Several runs are conducted for each of these arrival matrices using Bernoulli arrivals. The entire range of throughput levels between 0 and 1 is simulated and the results for MWM are in the magnitude of 10^3 to 10^5 slots.

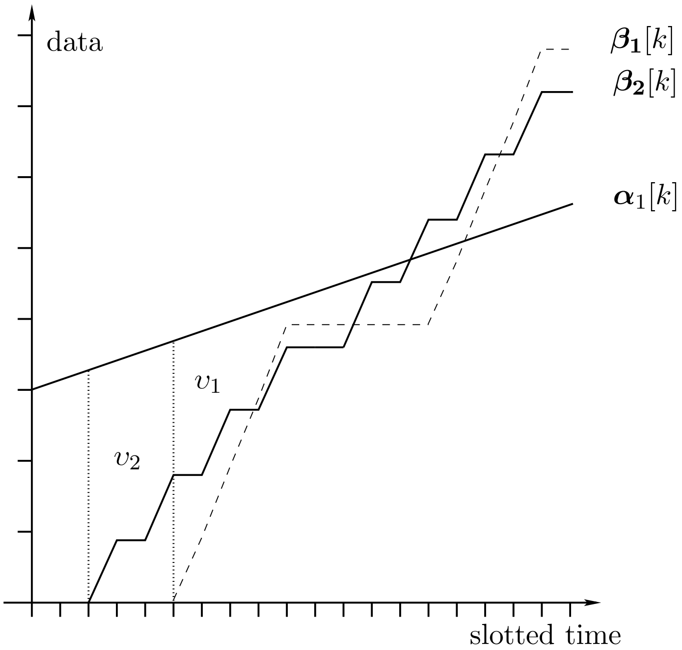

Figure 4.9: Buffer requirement for $\boldsymbol{\alpha}_1[k]$ under L_1 and L_2

4.5.2 Worst Case Result

The number of nodes is set to $n = 6$ and the tuning latency is set to $D = 1000$ slots. In constrast to the work by Brzezinski and Modiano [17], we consider only equally distributed arrival matrices. Each arrival curve is token bucket regulated with the burst size b_γ. The throughput level is q and since this is distributed equally over the arrival curves, each one has a rate of

$$r_\gamma = \frac{q}{5}. \tag{4.21}$$

Therefore our arrival matrix is

$$\underline{\boldsymbol{\alpha}}_u[k] = \begin{pmatrix} 0 & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k \\ b_\gamma + r_\gamma k & 0 & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k \\ b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & 0 & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k \\ b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & 0 & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k \\ b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & 0 & b_\gamma + r_\gamma k \\ b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & b_\gamma + r_\gamma k & 0 \end{pmatrix}. \tag{4.22}$$

This allows for a straightforward TDM scheme. The 5 topologies as given in Table 4.1 are used.

We set the highest level of interleaving and obtain the layout

$$L =< w_1^f, w_2^f, w_3^f, w_4^f, w_5^f > . \tag{4.23}$$

Topology	Links
$\underline{C}_1$	Ring $1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 1$
$\underline{C}_2$	Ring $1 \to 6 \to 5 \to 4 \to 3 \to 2 \to 1$
$\underline{C}_3$	Two rings, $1 \to 3 \to 5 \to 1$ and $2 \to 4 \to 6 \to 2$
$\underline{C}_4$	Two rings, $1 \to 5 \to 3 \to 1$ and $2 \to 6 \to 4 \to 2$.
$\underline{C}_5$	Three rings $1 \to 4 \to 1$, $2 \to 5 \to 2$ and $3 \to 6 \to 3$

Table 4.1: Topologies

In this setting, the worst case delay is the same for each queue. Each queue is served by exactly one topology, and the worst case is obtained if its packets arrive directly after the last slot in which it was served. Given the parameters we can derive an equation for the worst case delay. The minimum frame size is given by Inequality 4.14. Let $D = 1000$ and $\sigma_A = 1 - q$, which is the currently considered throughput level. Again, since we are dealing with deterministic arrivals, the equality

$$f = \lceil \frac{1000}{1-q} \rceil \tag{4.24}$$

holds. The worst case for a queue to wait until its packets start to get served is 4 frames plus the tuning latency. After that it is served with the rate 1 for $f - D$ slots, yielding a long term rate of r_γ. As pointed out in the discussion of Figure 4.8, the service curve can be seen as a latency-rate service curve with the latency $4f + 1000$ and the rate r_γ.

The delay bound δ_q in this case is the delay of a token bucket arrival curve and the latency-rate service curve. The delay bound is the time, after which the last packet of the burst is served in the worst case. For that, the equation

$$b_\gamma = r_\gamma(\delta_q - (4f + 1000)) \tag{4.25}$$

holds.

Rearranging and substituting Equations 4.24 and 4.21 yields the delay bound

$$\delta_q = \frac{5b_\gamma}{q} + 4\lceil \frac{1000}{1-q} \rceil + 1000. \tag{4.26}$$

In Figure 4.10 the worst case delay bound is depicted for some burst sizes.

What is striking is that if the burst size is fixed, the worst case delay decreases as the rate increases. This at first seems counter intuitive but it is explained by the rigid way that switching schemes are constructed. For low rates the frame size is also low. Therefore, it takes many periods to serve the burst which causes the high delay. As the rates get higher the frames also get larger, therefore it takes fewer periods to serve the burst. However, if the rate is high it takes a longer time until the first packet is served. After a certain rate this effect dominates and the worst case delay increases.

Consider the case where the burst size is $b_\gamma = 0$. The delay bound is then the time at which the first packet is served. The worst case delay bound always increases with the

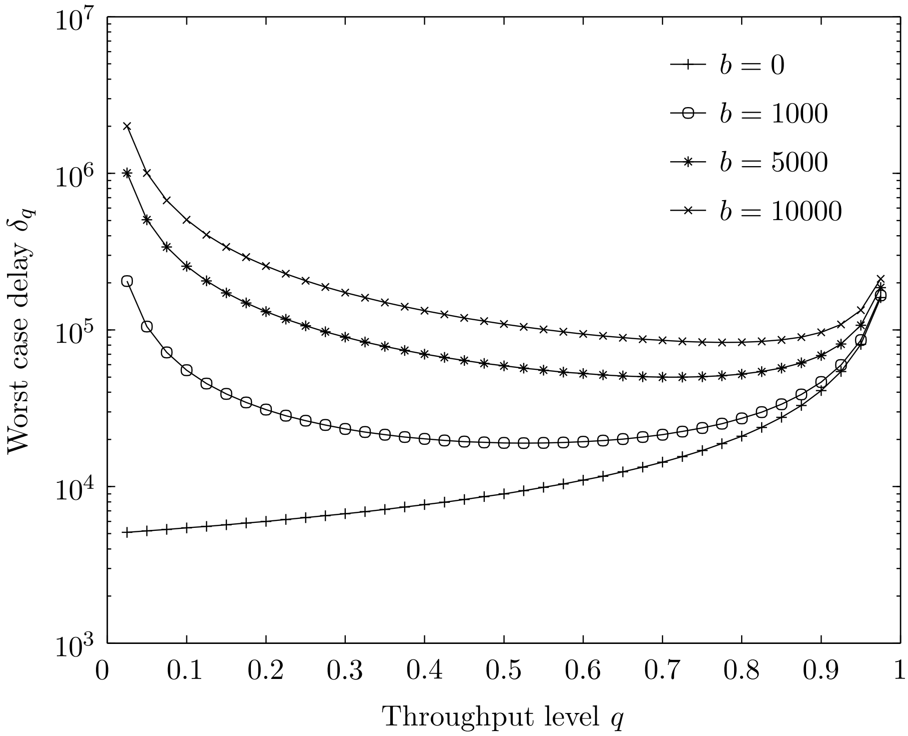

Figure 4.10: Worst case delay bound for different burst sizes

rate, since a higher rate indicates a larger frame size and a packet might have to wait until frames of other queues are served.

Next the worst case delay results are compared to the average delay results from Brzezinski and Modiano. As explained above, we only consider the MWM results of Figure 7 of their article [17]. The average delay is in the magnitude of 10^3 to 10^5 depending on the arrival rate. The worst case delay when $b_\gamma = 0$ is only marginally above the average delay with MWM. However, one has to keep in mind that our constraints on the traffic are much more strict than theirs. Especially in the case of $b_\gamma = 0$ a rate-limited flow is compared to a Bernoulli process. For high rates the gap between the average and worst case delay decreases.

Further conclusions can not be drawn due to the different assumptions of the two setups.

4.6 Recipe for Application

In this section the recipe for engineers to employ the results of this chapter are presented. Generally, the four steps of Figure 4.11 have to be performed for a network calculus analysis of a system. First the incoming traffic has to be examined to cast it into the arrival curve concept. In Step 2 the service curve of the system is to be determined. Optionally, the service curve can be convolved with service curves of other

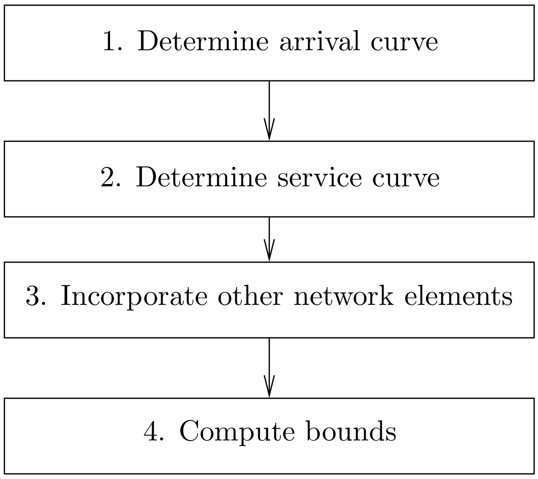

Figure 4.11: Four steps for the analysis of network calculus systems

network elements to conduct an end-to-end analysis. Once the arrival curve and service curve are obtained, the bounds, such as delay, backlog and output bound, can be computed.

The focus of this chapter is on Step 2, the determination of the service curve. A method is developed with which service curves of reconfigurable networks can be identified. The seven steps for this are shown in Figure 4.12. The arrival matrix is given, whose elements are the arrival curve for each queue. For this, a dominating arrival matrix has to be found. Next, the dominating arrival matrix is decomposed according to the Birkhoff-von Neumann method [19]. The minumum frame size is computed with Inequality 4.14, which restates the result from Brzezinski and Modiano [17]. Then the frames are aligned to a TDM scheme. The next step is the service curve identification. The service curve is the worst case output if an infinite burst of packets is injected into the network. It can be either done experimentally, or by deliberation as in this chapter. Each queue has to be considered separately. With the service curves and the arrival curves of the first step, the bounds can be computed.

4.7 Summary

In this chapter we demonstrated that applying network calculus to dynamically reconfigurable IP-over-WDM networks enables a throughput and worst case delay analysis. This is of crucial interest when designing networks which comprise QoS-sensitive traffic. We developed a framework in which the throughput and worst case delay analysis can be performed efficiently. Further, the buffer requirement can be determined. Subject of future work is to deploy this framework to obtain design decisions for optical networks. This consists of formulating an optimization problem which finds the optimal layout for all virtual output queues subject to the dominating matrix, the decomposition and the

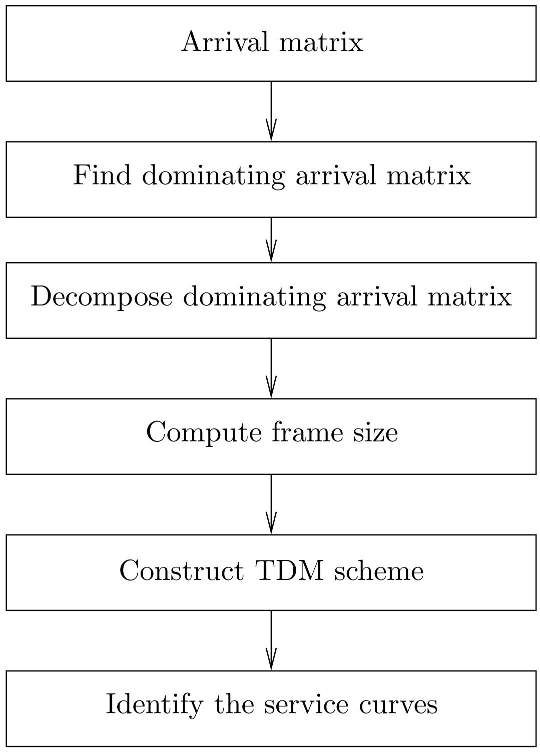

Figure 4.12: Six steps for determining service curves of recofigurable networks

frame size. Beyond that, this framework serves as a network calculus building block for an end-to-end QoS analysis of traffic flows which traverse heterogeneous networks.

5 A Transform for Network Calculus and its Application

5.1 Motivation

This chapter deals with the computation of the output of a system employing the min-plus convolution. It has a similar role as the conventional convolution in conventional system theory. Recall that our goal remains to enable the analysis of a network as a system with input, transfer function and output. While the output computation is rather straightforward in conventional system theory, it requires more attention for network calculus systems as the properties of the min-plus algebra have to be considered. In conventional system theory, computation algorithms for the convolution often rely on transforms, since the convolution in the time domain corresponds to the product in the frequency domain. The indirect way of computing the convolution of two functions by transforming them, multiplying the transforms and taking the inverse transform of this product can be more efficient than convolving directly. This especially holds true when the task is to be done by computers, i.e., numerically in a discrete setting. According to the legacy book by Press et al. [87], an efficient numerical algorithm to compute the conventional convolution is via the Fast Fourier Transform (FFT).

This chapter consists of three parts. In the first part we discuss different transforms for network calculus. We then derive the $\mathfrak{P}$-transform and point out a graphical representation of the convolution computation. In the second part we derive some theorems which utilize properties of the min-plus convolution and simplify its computation. Applications of our theorems are illustrated in the third part. One of these is to derive the optimal service curves for the nodes along a path. Further, admission control can be improved when designing networks by service curves. Considering one node, reallocating the service curves leads to admitting more flows. Finally we point out scenarios where sub-optimal allocation of service curves in a node can increase the number of admitted flows to the network.

5.2 A Transform for Network Calculus

5.2.1 The Fenchel Transform

When looking for a transform for network calculus, the first thing that comes to mind is the *Fenchel transform*, also known as the convex conjugate function. It is also the starting point of the Legendre transform, which was the subject of Section 3.4. In the book by Baccelli et al. [6] it is briefly pointed out that the Fenchel transform carries over the min-plus convolution in one domain to a sum in another domain. In the book by Hiriart-Urruty and Lemarechal [49] one finds the following definition for the convex conjugate function: Let $\mathbf{f} : \mathbb{R} \to \mathbb{R}$. Then

$$dom\,\mathbf{f} = \{t \in \mathbb{R} \mid |\mathbf{f}(t)| < \infty\}. \tag{5.1}$$

Definition 5.1 *For a function* $\mathbf{f}(t)$ *its Fenchel transform, or convex conjugate function,* $\mathbf{F}(\tau)$ *is given by*

$$\mathfrak{C}\{\mathbf{f}(t)\} = \mathbf{F}(\tau) = \sup\{\tau t - \mathbf{f}(t) \mid t \in dom\,\mathbf{f}\}. \tag{5.2}$$

I.e., the supremum of the term is taken for all t, where $\mathbf{f}(t)$ is defined. For some functions the Fenchel transform is a self-dual transform, which means the inverse transform is the same operation as the transform. Therefore, we have for a function $\mathbf{F}(\tau)$ in the Fenchel domain

$$\mathfrak{C}^{-1}\{\mathbf{F}(\tau)\} = \mathbf{f}(t) = \sup\{\tau t - \mathbf{F}(\tau) \mid t \in dom\,\mathbf{f}\}. \tag{5.3}$$

Unfortunately, a shortcoming of the Fenchel transform is that it only works well for convex functions. For piecewise linear, convex functions and slightly more general ones, an efficient algorithm to obtain the min-plus convolution is outlined in Chapter 3 in the book by Le Boudec [13], where it is pointed out that the min-plus convolution of two piecewise linear, convex functions is obtained by simply sorting the slopes of the individual functions. In general, the bi-conjugate of a function yields the convex closure of the function. For closed convex functions $\mathbf{f}_c(t)$ (e.g. the piecewise linear, convex functions in [13] fall into this category) we have

$$\mathfrak{C}^{-1}\{\mathfrak{C}\{\mathbf{f}_c(t)\}\} = \mathbf{f}(t). \tag{5.4}$$

But in general, i.e., for non-convex functions, we can not hope for equality. Therefore, the Fenchel transform is not optimally suited in the context of network calculus. Applications and problems of the Fenchel transform in the context of network calculus are described in the technical report by Pandit et al. [78]. Furthermore, sample computations are provided in that work. The Fenchel transform is closely related to the *Legendre* transform (cf. Section 3.4). The Legendre transform and its applications within network calculus have been elaborated by Fidler and Recker [33, 32, 34]. The Legendre transform is defined independently for convex and concave functions. It therefore allows for the analysis of systems where all functions are either convex or concave. Fidler and Recker give elegant representations of the backlog bound, delay bound, as well as the output bound by computing them in the Legendre domain.

5.2.2 The Continuous $\mathfrak{P}$-transform

In reconfigurable networks (cf. Chapter 4) but also in bandwidth/delay decoupled scheduling (cf. Section 5.4 and [93]) service curves which are neither convex nor concave appear. For this reason we develop a transform in the following that does not require the underlying functions to be strictly convex or concave. Baccelli et al. [6] introduce the discrete $\mathfrak{P}$-transform, which is elaborated in the dissertation of Holger Jäkel (in German) [52]. Note that Jäkel [52] labels this ΓΔ-transform, while Baccelli et al. [6] label it "γ-transform in the 2-dimensional domain". We first describe the discrete $\mathfrak{P}$-transform and then extend it to the continuous case. Here we limit ourselves to a descriptive discussion on this transform, the mathematically inclined reader is referred to the aforementioned two texts. As our starting point are sequences $\mathbf{x}[k], k \in \mathbb{Z}$, we first look for a transform analogous to the Z-transform. For this a formal shift operator $\hat{\gamma}$ is defined as

$$\hat{\gamma}^m \mathbf{x}[k] := \mathbf{x}[k-m], m \in \mathbb{Z}. \tag{5.5}$$

We now introduce the $\mathfrak{G}$-transform, which Jäkel [52] labels as the Γ-transform and Baccelli et al. [6] as γ-transform. Utilizing the operator $\hat{\gamma}$, the $\mathfrak{G}$-transform is given by

$$\mathbf{X}[\hat{\gamma}] := \mathfrak{G}\{\mathbf{x}[k]\} := \inf_{k \in \mathbb{Z}} \{\mathbf{x}[k]\hat{\gamma}^k\}. \tag{5.6}$$

In order to find a viable interpretation of such a transform, it is helpful to introduce a second shift operator $\hat{\delta}$. This leads us to the $\mathfrak{P}$-transform. Formally, the transform is defined by

$$\mathbf{X}[\hat{\gamma}, \hat{\delta}] := \mathfrak{P}\{\mathbf{x}[k]\} := \inf_{k \in \mathbb{Z}} \{\hat{\gamma}^k \hat{\delta}^{\mathbf{x}[k]}\}. \tag{5.7}$$

$\mathbf{X}[\hat{\gamma}, \hat{\delta}]$ is a formal power series with Boolean coefficients. The coefficients are either the zero element of the min-plus dioid $+\infty$ or the one element of the min-plus dioid 0. The coefficient takes the value 0 for every point which is shifted by k on the abscissa and $\mathbf{x}[k]$ on the ordinate. For all other points the coefficient is $+\infty$.

However, it is the graphical interpretation that makes this transform useful. To depict this, the sequence is mapped to a "cloud" of information points,

$$\mathbf{s}_x[k,l] = \begin{cases} 0 & \mathbf{x}[k] = l \\ +\infty & \mathbf{x}[k] \neq l \end{cases}. \tag{5.8}$$

The interpretation of the information points depends on the context. In the original setting, the max-plus algebra describing Petri nets, it indicates at which time instant k the event number l can be triggered. When $\mathbf{x}[k]$ denotes a service curve and we are interested in the worst case consideration, which is our scenario, an information point indicates that at time instant k the packet l can be served. However, if there are 2 information points e.g., $(k,l) = (3,7)$ and $(k,l) = (2,7)$, then the latter information point is redundant. The information "the 7th packet can be served at latest after 2 time units" is redundant if the information "the 7th packet can be served after 3 time units" is available. Clearly all information points which are above or left ("north-west") of another information point are redundant.

In the following two examples of such clouds are given. Set $\mathbf{x}_1[k]$ to

$$\mathbf{x}_1[k] = \begin{cases} 0 & k = 0 \\ 2+k & 1 \leq k \leq 3 \\ +\infty & \text{otherwise} \end{cases}. \tag{5.9}$$

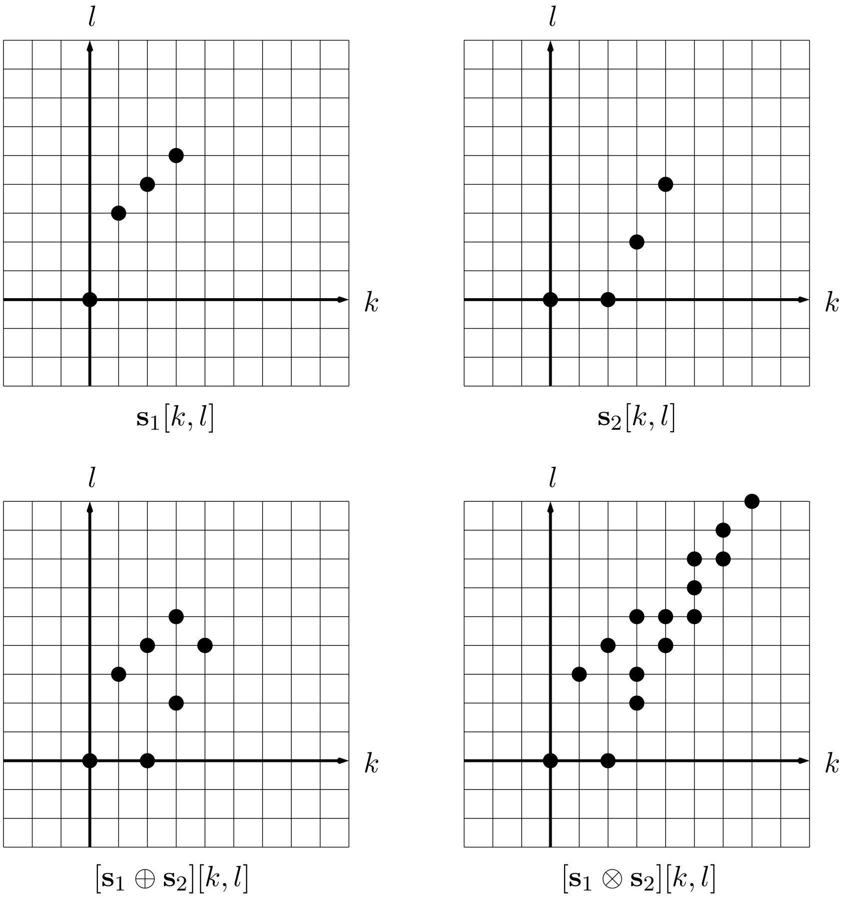

Figure 5.1: Point cloud operators

The corresponding point cloud $\mathbf{s}_1[k,l]$ is depicted in the top left grid of Figure 5.1 and contains the points

$$\mathbf{s}_1[k,l] = \{(0,0);(1,3);(2,4);(3,5)\}. \tag{5.10}$$

Further, set $\mathbf{x}_2[k]$ to

$$\mathbf{x}_2[k] = \begin{cases} 0 & k=0 \\ 2(k-2) & 2 \le k \le 4 \\ +\infty & \text{otherwise} \end{cases}. \tag{5.11}$$

Its corresponding point cloud $\mathbf{s}_2[k,l]$ is

$$\mathbf{s}_2[k,l] = \{(0,0);(2,0);(3,2);(4,4)\} \tag{5.12}$$

and depicted in the top right grid of Figure 5.1. We next introduce the addition and multiplication operators on these clouds. For two point clouds their addition (in the min-plus sense) is defined component-wise as the minimum operation of their coefficients. Let $\mathbf{s}_i[k,l], i=1,2$ be two point clouds. Then the addition of the two point clouds is given by

$$[\mathbf{s}_1 \oplus \mathbf{s}_2][k,l] = \min\{\mathbf{s}_1[k,l], \mathbf{s}_2[k,l]\}. \tag{5.13}$$

The graphical interpretation of the addition operation is simply to take all points of the underlying point clouds, choose the minimum of those and fix it as a new point for the point cloud of the sum. Since we are dealing with Boolean coefficients, this turns out to be the sum being the point cloud which holds all points that are in at least one of the point clouds. The addition of $[\mathbf{s}_1 \oplus \mathbf{s}_2][k,l]$ is depicted in the bottom left grid of Figure 5.1.

The multiplication is more tedious. Again, let $\mathbf{s}_i[k,l], i = 1,2$ denote two point clouds. Then, the multiplication of the two point clouds is given by

$$[\mathbf{s}_1 \otimes \mathbf{s}_2][k,l] = \min_{i_1+i_2=k, j_1+j_2=l} \{\mathbf{s}_1[i_1,j_1] + \mathbf{s}_2[i_2,j_2]\}. \tag{5.14}$$

Therefore, the multiplication operation is to take the vectorial sum of all points of one point cloud with all points of the other one. The multiplication is depicted in the bottom right grid of Figure 5.1.

In Figure 5.2 the area holding redundant points is depicted.

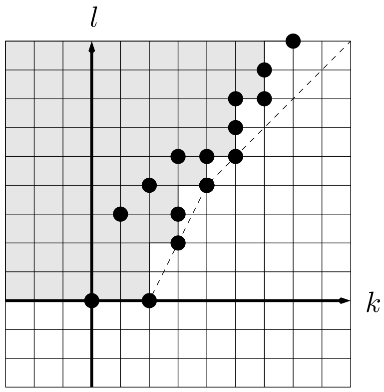

Figure 5.2: Point cloud with redundant area

The inverse transform $\mathfrak{P}^{-1}$ is defined by transforming the border of this area back to a sequence of one variable,

$$\mathbf{x}[k] = \mathfrak{P}^{-1}\{\mathbf{X}[\hat{\gamma},\hat{\delta}]\} = \begin{cases} \min_i l_i & \mathbf{X}[k,l_i] = 0 \\ +\infty & \text{otherwise} \end{cases}. \tag{5.15}$$

In our example, the inverse transform of the multiplication of $[\mathbf{s}_1 \otimes \mathbf{s}_2][k,l]$ yields

$$\mathbf{y}[k] = \mathfrak{P}^{-1}\{[\mathbf{s}_1 \otimes \mathbf{s}_2][k,l]\} = \begin{cases} 0 & 0 \le k < 2 \\ 2(k-2) & 2 \le k < 4 \\ 4+k & 4 \le k \le 5 \\ 5+2(k-5) & 5 < k \le 7 \\ +\infty & \text{otherwise} \end{cases}. \tag{5.16}$$

The sequence $\mathbf{x}_1[k]$ can be interpreted as a sample of the continuous token bucket arrival curve

$$\boldsymbol{\alpha}_x(t) = \begin{cases} 0 & t \leq 0 \\ 2+t & t > 0 \end{cases} . \tag{5.17}$$

Similarly, $\mathbf{x}_2[k]$ can be interpreted as a sample of the continuous latency rate service curve

$$\boldsymbol{\beta}_{\boldsymbol{x}}(t) = \begin{cases} 0 & t \leq 2 \\ 2(t-2) & t > 2 \end{cases} . \tag{5.18}$$

We observe that $\mathbf{y}[k]$ in the interval $k = [0,5]$ is the min-plus convolution of the samples of $\boldsymbol{\alpha}_x(t)$ and $\boldsymbol{\beta}_{\boldsymbol{x}}(t)$. However, for $k > 5$ this is not the case anymore. The correct min-plus convolution is indicated by the dashed line in Figure 5.2. The reason is the discretization or sampling procedure. The accuracy of the computation depends strongly on the amount of discrete points we choose. It is also not possible to find some characteristic points which describe the min-plus convolution of piecewise linear functions [78]. However, the more points we take into account, the closer the result approaches the min-plus convolution. Taking this to the limit, we end up at the *continuous* $\mathfrak{P}$*-transform*. If an infinite number of points is considered, the exact min-plus convolution can be obtained.

Consider again the arrival curve sampled by $\mathbf{x}_1[k]$ and the service curve sampled by $\mathbf{x}_2[k]$. A function, such as an arrival curve or service curve, can be viewed as an infinite set of points. We proceed with them as with the points in the point clouds. I.e., every point of one function is summed to every point of the other one. It can be easily verified that the border of this denotes the min-plus convolution. This is depicted in Figure 5.3.

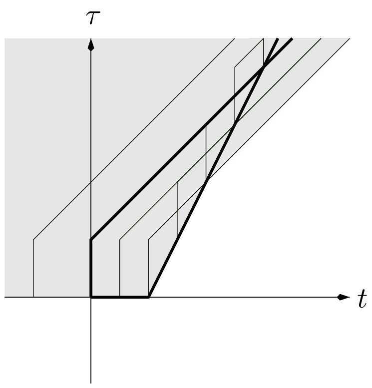

Figure 5.3: Continuous $\mathfrak{P}$-transform

With this in mind, we bring the attention back to Equation 5.14 and realize that this is merely another representation of the min-plus convolution. Despite giving an intuitive graphical representation, the complexity of the computation by this method remains the same. Therefore, the $\mathfrak{P}$-transform per se is not very helpful to simplify the general computation of the min-plus convolution.

Taking a close look at this graphical representation, we obtain a theorem which is given in the following section.

5.3 Min-plus Convolution in the Context of Network Calculus

Recall that our goal is to give insight and facilitate the computation of the min-plus convolution. While the $\mathfrak{P}$-transform gives insight in the form of an intuitive graphical interpretation, it does not directly help with its computation. In this section we develop theorems which simplify the min-plus convolution for application in network calculus. First we focus the service curve of the bandwidth/delay decoupled scheduler (L2R service curve), which in the following sections enables us to extend the work of Schmitt [93]. This work is reviewed in Section 5.4. We require the following definition.

Definition 5.2 (Convex and concave inflection points) *A function* $\mathbf{f}(t)$ *has a convex inflection point at* T*, if and only if the slope to its right is greater than the slope to its left. I.e., a convex inflection point at* T^f *has the property*

$$\frac{\mathbf{f}(T^f+\epsilon)-\mathbf{f}(T^f)}{\epsilon} > \frac{\mathbf{f}(T^f)-\mathbf{f}(T^f-\epsilon)}{\epsilon}. \tag{5.19}$$

Analogously, we call an inflection point where the slope to its right is less than the slope its left a concave inflection point.

The L2R service curve first has a latency until some time instant T_0, then has a peak rate until some time instant T_1, at which it switches to a sustained rate. In other words, the L2R service curve has exactly one convex inflection point and exactly one concave inflection point. However, we consider functions with exactly one convex inflection point and an arbitrary number of concave inflection points (cf. Figure 5.4). These are more general and the reason for choosing them will become clear shortly. The functions $\mathbf{f}(t)$ and $\mathbf{g}(t)$ are of the above described type. Let $i = 0, \ldots, m$ and $j = 0, \ldots, n$. Further, $r > 0$ and $T > 0$ for all r and T. Then we have

$$\mathbf{f}(t) = \begin{cases} 0 & & t \leq T_0^f \\ r_0^f(t-T_0^f) & T_0^f < & t \leq T_1^f \\ r_0^f(T_1^f-T_0^f)+r_1^f(t-T_1^f) & T_1^f < & t \leq T_2^f \\ \vdots & & \\ \sum_{i=0}^{m-2} r_i^f(T_{i+1}^f-T_i^f)+r_{m-1}^f(t-T_{m-1}^f) & T_{m-1}^f < & t \leq T_m^f \\ \sum_{i=0}^{m-1} r_i^f(T_{i+1}^f-T_i^f)+r_m^f(t-T_m^f) & & t \geq T_m^f \end{cases} \tag{5.20}$$

and

$$\mathbf{g}(t) = \begin{cases} 0 & & t \leq T_0^g \\ r_0^g(t - T_0^g) & T_0^g < & t \leq T_1^g \\ r_0^g(T_1^g - T_0^g) + r_1^g(t - T_1^g) & T_1^g < & t \leq T_2^g \\ \vdots & & \\ \sum_{j=0}^{n-2} r_j^g(T_{j+1}^g - T_j^g) + r_{n-1}^g(t - T_{n-1}^g) & T_{n-1}^g < & t \leq T_n^g \\ \sum_{j=0}^{n-1} r_j^g(T_{j+1}^g - T_j^g) + r_n^g(t - T_n^g) & & t \geq T_n^g \end{cases} . \qquad (5.21)$$

Now let the functions $\mathbf{f}(t)$ and $\mathbf{g}(t)$ be such that they have exactly one convex inflection point each at T_0^f and T_0^g, respectively, and only concave inflection points thereafter. I.e., $r_m^f < r_{m-1}^f < \ldots < r_0^f$ and $r_n^g < r_{n-1}^g < \ldots < r_0^g$. In the style of L2R functions, we call them *LnR functions*. The shape of $\mathbf{f}(t)$ and $\mathbf{g}(t)$ is given in Figure 5.4. Note that $\mathbf{f}(t)$ and $\mathbf{g}(t)$ belong to the class of wide-sense increasing functions as defined by Le Boudec [13].

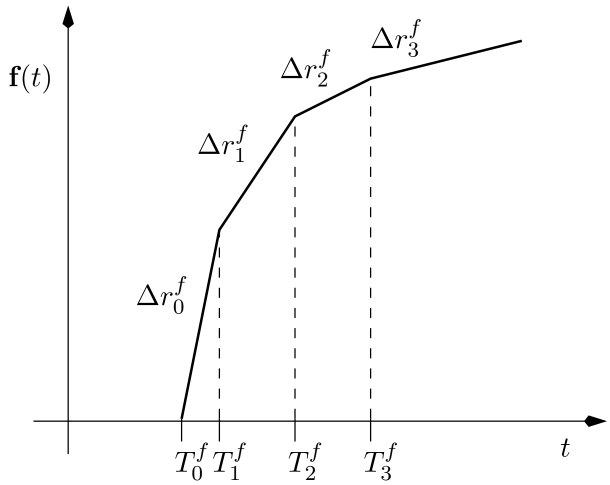

Figure 5.4: Function with one convex inflection point

The computation of the convolution of such functions via the $\mathfrak{P}$-transform is shown in Figure 5.5. Here the functions $\mathbf{f}(t)$ and $\mathbf{g}(t)$ are set to L2R functions, i.e., each with one convex and one concave inflection point. As described in the previous section, the gray area is obtained by interpreting both functions as an infinite number of points and taking their vectorial sum. The convolution is given by the border of the gray area. There are several observations to be made here. The resulting convolution is also piecewise linear. It has one convex and two concave inflection points. Further, the convolution is obtained by shifting the functions such that both have their respective convex inflection point at $T_0^f + T_0^g$, and then taking the minimum of these shifted functions.

In the remainder of this section we set up a few theorems and remarks that generalize these observations. The first theorem states that the convolution is obtained by taking the minimum of the two functions shifted such that their convex inflection points are at $T_0^f + T_0^g$. It is depicted in Figure 5.6.

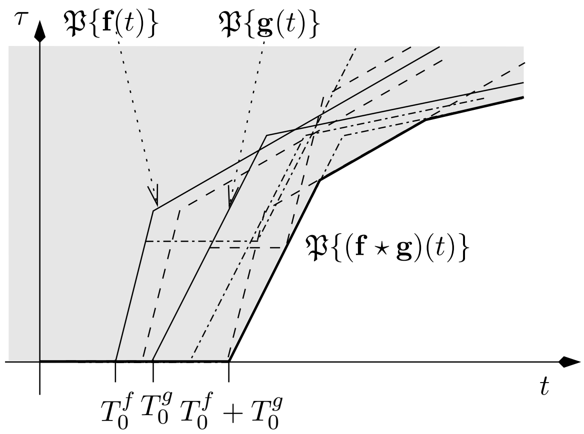

Figure 5.5: 𝔓-transform of L2R functions

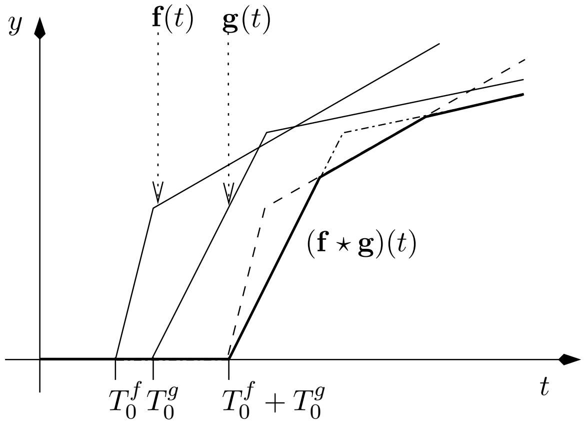

Figure 5.6: Min-plus convolution of L2R functions

Theorem 5.3 *The min-plus convolution of two functions with one convex inflection point and only concave inflection points thereafter,* $\mathbf{f}(t)$ *and* $\mathbf{g}(t)$*, is given by*

$$(\mathbf{f} \star \mathbf{g})(t) = \inf_{0 \le \tau \le t} \{\mathbf{f}(t-\tau) + \mathbf{g}(\tau)\} = \min\{\mathbf{f}(t - T_0^g), \mathbf{g}(t - T_0^f)\}. \tag{5.22}$$

Proof The rationale of the proof is given first by graphical means which is then followed by the complete proof. As pointed out in the previous section, the min-plus convolution can be obtained by using the continuous 𝔓-transform. The continuous 𝔓-transform of two functions is given by summing all points of one function to all points of the other function. The inverse transform, i.e., taking the minimum over all t, then yields the min-plus convolution. We show now that the minimum of the shifted functions is always the overall minimum. In Figure 5.7, all points of $\mathbf{g}(t)$ are summed to different points of $\mathbf{f}(t)$, namely Q_1, Q_2, and Q_3. At time instant t_x, the lowest point is Q_0, which is obtained when all points of $\mathbf{g}(t)$ are summed to the point Q_1. The value at Q_0 is

$$\mathbf{f}((t_x - T_0^g) + \mathbf{g}(T_0^g) = \mathbf{f}(t_x - T_0^g). \tag{5.23}$$

It can be seen that $\mathbf{f}(t_x - T_0^g - \epsilon) + \mathbf{g}(T_0^g + \epsilon)$ yields a higher value at t_x for any $\epsilon > 0$.

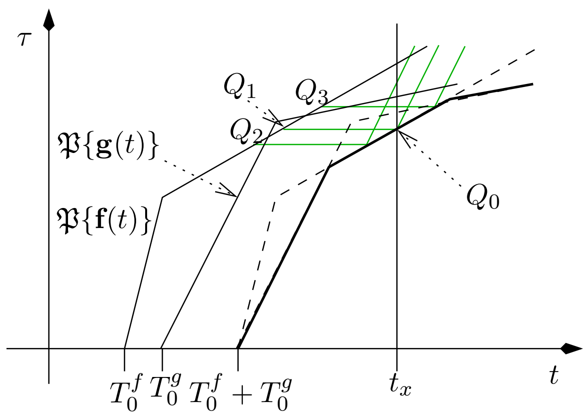

Figure 5.7: Proof of Theorem 5.3

The reason is that the slope of $\mathbf{g}(t)$ between T_0^g and $T_0^g + \epsilon$ is greater than the slope of $\mathbf{f}(t)$ between $t_x - T_0^g$ and $t_x - T_0^g - \epsilon$. This is depicted where the points of $\mathbf{g}(t)$ are summed to point Q_2. Similarly, $\mathbf{f}(t_x - T_0^g + \epsilon) + \mathbf{g}(T_0^g - \epsilon)$ always yields a higher value at t_x for any $\epsilon > 0$. The value for $\mathbf{f}(t_x - T_0^g + \epsilon)$ is greater than $\mathbf{f}(t_x - T_0^g)$ and $\mathbf{g}(T_0^g - \epsilon)$ is always 0. This is depicted where the points of $\mathbf{g}(t)$ are summed to point Q_3. Therefore, moving the function $\mathbf{g}(t)$ in both directions on $\mathbf{f}(t)$ always yields a higher value at t_x.

We next give the formal proof. We assume $T_0^f \leq T_0^g$ without loss of generality. For $t \leq T_0^f$ set $\tau = 0$ and we have

$$(\mathbf{f} \star \mathbf{g})(t) = \mathbf{f}(t) + \mathbf{g}(0) = 0.$$

For $T_0^f < t \leq T_0^f + T_0^g$ let $\tau = T_0^f$ which yields

$$(\mathbf{f} \star \mathbf{g})(t) = \mathbf{f}(T_0^f) + \mathbf{g}(t - T_0^f) = 0.$$

Next we consider the case where $t > T_0^f + T_0^g$. The assumption $T_0^f \leq T_0^g$ is dropped. However, we need the assumption that $\mathbf{f}(t - T_0^g) \geq \mathbf{g}(t - T_0^f)$ for this case. This is not a crucial restriction as if it does not hold, the proof can be constructed by interchanging $\mathbf{f}$ and $\mathbf{g}$. The inequality

$$\frac{\mathbf{f}(t - T_0^g) - \mathbf{f}(T_0^f)}{t - T_0^g - T_0^f} \geq \frac{\mathbf{g}(t - T_0^f) - \mathbf{g}(T_0^g)}{t - T_0^g - T_0^f} \tag{5.24}$$

restates this assumption, since $\mathbf{f}(T_0^f) = 0$ and $\mathbf{g}(T_0^g) = 0$. Let t_I be such that $T_0^f + T_0^g + t_I \leq t$. We state the inequality

$$\frac{\mathbf{f}(t - T_0^g) - \mathbf{f}(T_0^f)}{t - T_0^g - T_0^f} \leq \frac{\mathbf{f}(T_0^f + t_I) - \mathbf{f}(T_0^f)}{t_I}. \tag{5.25}$$

Proof of Inequality 5.25: We have $t > T_0^f + T_0^g$. For $t_I \geq 0$ with $t \geq T_0^f + T_0^g + t_I$ we have $T_0^f \leq T_0^f + t_I \leq t - T_0^g$. Hence $T_0^f + t_I$ can be written as unique convex combination of the points T_0^f and $t - T_0^g$. Therefore we need to solve the following:

$$T_0^f + t_I = uT_0^f + (1-u)(t - T_0^g).$$

A simple computation yields

$$u = \frac{T_0^f + t_I - t + T_0^g}{T_0^f - t + T_0^g} = 1 - \frac{t_I}{t - T_0^f - T_0^g} \leq 1. \tag{5.26}$$

Hence we have

$$1 - u = \frac{t_I}{t - T_0^f - T_0^g}.$$

Now we use the property that $\mathbf{f}$ is concave. Recall, that a function $\mathbf{z}(t)$ is concave on $[t_1, t_2]$, if for every $u \in [0,1]$ the following inequality

$$u\mathbf{z}(t_1) + (1-u)\mathbf{z}(t_2) \leq \mathbf{z}(ut_1 + (1-u)t_2)$$

holds.

Setting $t_1 = T_0^f$ and $t_2 = t - T_0^g$ and u as in Equation 5.26, we obtain

$$\begin{aligned} \mathbf{f}(T_0^f + t_I) &\geq u\mathbf{f}(T_0^f) + (1-u)\mathbf{f}(t - T_0^g) \\ = \mathbf{f}(T_0^f) - \frac{t_I}{t - T_0^f - T_0^g}\mathbf{f}(T_0^f) &+ \frac{t_I}{t - T_0^f - T_0^g}\mathbf{f}(t - T_0^g) \\ &= \mathbf{f}(T_0^f) + \frac{t_I}{t - T_0^f - T_0^g}(\mathbf{f}(t - T_0^g) - \mathbf{f}(T_0^f)). \end{aligned}$$

Rearranging yields Inequality 5.25.

Another inequality is required,

$$\frac{\mathbf{g}(t - T_0^f) - \mathbf{g}(t - T_0^f - t_I)}{t_I} \leq \frac{\mathbf{g}(t - T_0^f) - \mathbf{g}(T_0^g)}{t - T_0^g - T_0^f}. \tag{5.27}$$

Proof of Inequality 5.27: We have $t > T_0^f + T_0^g$. For $t_I \geq 0$ with $t \geq T_0^f + T_0^g + t_I$ we have $T_0^g \leq t - T_0^f - t_I \leq t - T_0^f$. Hence $t - T_0^f - t_I$ can be written as a unique convex combination of the points T_0^g and $t - T_0^f$. Therefore we need to solve the following:

$$t - T_0^f - t_I = u \cdot T_0^g + (1-u) \cdot (t - T_0^f).$$

A simple computation yields

$$u = \frac{-t_I}{T_0^g + T_0^f - t} = \frac{t_I}{t - T_0^f - T_0^g} \leq 1.$$

Now we use the property that $\mathbf{g}$ is concave. Recall, that a function $\mathbf{z}(t)$ is concave on $[t_1, t_2]$, if for every $u \in [0,1]$ the following inequality

$$u\mathbf{z}(t_1) + (1-u)\mathbf{z}(t_2) \leq \mathbf{z}(ut_1 + (1-u)t_2)$$

holds. Setting $t_1 = T_0^g$ and $t_2 = t - T_0^f$, we obtain

$$\begin{aligned}\mathbf{g}(t - T_0^f - t_I) &\geq u\mathbf{g}(T_0^g) + (1-u)\mathbf{g}(t - T_0^f) \\ = \frac{t_I}{t - T_0^f - T_0^g}\mathbf{g}(T_0^g) &+ \left(1 - \frac{t_I}{t - T_0^f - T_0^g}\right)\mathbf{g}(t - T_0^f) \\ &= \mathbf{g}(t - T_0^f) + \frac{t_I}{t - T_0^f - T_0^g}(\mathbf{g}(T_0^g) - \mathbf{g}(t - T_0^f)).\end{aligned}$$

Thus we have

$$\frac{\mathbf{g}(t - T_0^f - t_I) - \mathbf{g}(t - T_0^f)}{t_I} \geq \frac{\mathbf{g}(T_0^g) - \mathbf{g}(t - T_0^f)}{t - T_0^f - T_0^g}$$

and multiplying by -1 one obtains Inequality 5.27.

Combining 5.24 and 5.27 yields

$$\frac{\mathbf{g}(t - T_0^f) - \mathbf{g}(t - T_0^f - t_I)}{t_I} \leq \frac{\mathbf{f}(t - T_0^g) - \mathbf{f}(T_0^f)}{t - T_0^g - T_0^f}. \tag{5.28}$$

Further, combining 5.28 and 5.25 one has

$$\frac{\mathbf{g}(t - T_0^f) - \mathbf{g}(t - T_0^f - t_I)}{t_I} \leq \frac{\mathbf{f}(T_0^f + t_I) - \mathbf{f}(T_0^f)}{t_I}. \tag{5.29}$$

Multiplying by $t_I > 0$ and rearranging we obtain

$$\mathbf{g}(t - T_0^f) + \mathbf{f}(T_0^f) \leq \mathbf{f}(T_0^f + t_I) + \mathbf{g}(t - T_0^f - t_I).$$

Setting $\tau := T_0^f + t_I$ and recalling $\mathbf{f}(T_0^f) = 0$ we have

$$\mathbf{g}(t - T_0^f) \leq \mathbf{f}(\tau) + \mathbf{g}(t - \tau).$$

■

In this case, an alternative to compute $(\mathbf{f} \star \mathbf{g})(t)$ is to resolve the function $\mathbf{f}(t)$ and $\mathbf{g}(t)$ into a burst-delay component and a concave component and compute the convolution then. Define the functions

$$\mathbf{f}_a(t) = \begin{cases} 0 & t \leq T_0^f \\ +\infty & t > T_0^f \end{cases}, \tag{5.30}$$

$$\mathbf{f}_b(t) = \begin{cases} +\infty & t < 0 \\ \mathbf{f}(t + T_0^f) & t \geq 0 \end{cases}, \tag{5.31}$$

$$\mathbf{g}_a(t) = \begin{cases} 0 & t \leq T_0^g \\ +\infty & t > T_0^g \end{cases} \tag{5.32}$$

and

$$\mathbf{g}_b(t) = \begin{cases} +\infty & t < 0 \\ \mathbf{f}(t + T_0^g) & t \geq 0 \end{cases}. \tag{5.33}$$

Then $(\mathbf{f} \star \mathbf{g})(t)$ can be computed by

$$(\mathbf{f} \star \mathbf{g})(t) = ((\mathbf{f}_a \star \mathbf{f}_b) \star (\mathbf{g}_a \star \mathbf{g}_b))(t). \tag{5.34}$$

Utilizing the distributivity, we have

$$(\mathbf{f} \star \mathbf{g})(t) = (\mathbf{f}_a \star \mathbf{g}_a \star (\mathbf{f}_b \star \mathbf{g}_b))(t). \tag{5.35}$$

This is the convolution of two concave functions followed by the convolution with burst-delay functions. As pointed out in the book by Le Boudec and Thiran [13] and recapitulated in Section 2.5 these convolutions follow simple rules.

The advantage of the proof derived here is that it is more general and easily extensible to functions with mixed concave and convex inflection points. That will be the subject of the next section. Note that the proof does not require $\mathbf{f}(t)$ and $\mathbf{g}(t)$ to be piecewise linear and works for any function that is wide-sense increasing and has only one convex inflection point. However, we only consider piecewise linear functions as all functions in contemporary network calculus are piecewise linear. The slopes always denote some kind of rate. The piecewise linearity is the subject of the next theorem.

Theorem 5.4 *The convolution of two piecewise linear functions* $\mathbf{f}$ *and* $\mathbf{g}$ *with one convex inflection point each is also piecewise linear and concave from* $T_0^f + T_0^g$ *onwards.*

Proof Since $\mathbf{f}$ and $\mathbf{g}$ are piecewise linear and concave, $\mathbf{f}(t - T_0^g)$ and $\mathbf{g}(t - T_0^f)$ also possess these properties, since they evolve from $\mathbf{f}$ and $\mathbf{g}$, respectively, by shifting along the t-axis. That the convolution is again piecewise linear is clear, since $\mathbf{f}$ and $\mathbf{g}$ are piecewise linear and continuous. It remains to show that the convolution is a concave function. Let $\mathbf{f}$ and $\mathbf{g}$ be concave from a point Q_c onwards. Set $\mathbf{z}(t) := \min\{\mathbf{f}(t), \mathbf{g}(t)\}$. Then $\mathbf{z}(t)$ is concave. We need to show that for any given $t_1, t_2 \in \mathbb{R}$ and $u \in [0, 1]$ we have

$$\mathbf{z}(ut_1 + (1-u)t_2) \geq u\mathbf{z}(t_1) + (1-u)\mathbf{z}(t_2). \tag{5.36}$$

We assume the following

$$\mathbf{f}(ut_1 + (1-u)y) \leq \mathbf{g}(ut_1 + (1-u)y) \tag{5.37}$$

without affecting the reasoning of the proof. Then we have

$$\begin{aligned}
\mathbf{z}(ut_1 + (1-u)t_2) &= \min\{\mathbf{f}(ut_1 + (1-u)t_2), \mathbf{g}(ut_1 + (1-u)t_2)\} && (5.38)\\
&= \mathbf{f}(ut_1 + (1-u)t_2) && (5.39)\\
&\geq u\mathbf{f}(t_1) + (1-u)\mathbf{f}(t_2) && (5.40)\\
&\geq u \min\{\mathbf{f}(t_1), \mathbf{g}(t_1)\} + (1-u) \min\{\mathbf{f}(t_2), \mathbf{g}(t_2)\} && (5.41)\\
&= u\mathbf{z}(t_1) + (1-u)\mathbf{z}(t_2). && (5.42)
\end{aligned}$$

■

Note that Inequality 5.40 holds because of the assumption in Inequality 5.37 and Inequality 5.40 uses the concavity of $\mathbf{f}$.

This leads to the following theorem.

Theorem 5.5 *Given n functions* $\mathbf{f}_1(t), \mathbf{f}_2(t), \ldots, \mathbf{f}_n(t)$*, each with the properties of* $\mathbf{f}$*. The min-plus convolution of all of them is given by*

$$(\mathbf{f}_1 \star \mathbf{f}_2 \star \ldots \star \mathbf{f}_n)(t) = \min_{j=1,2,\ldots,n} (\mathbf{f}_j(t + T_0^j - \sum_{i=1}^{n} T_0^i). \tag{5.43}$$

PROOF Utilizing the distributivity,

$$(\mathbf{f}_1 \star \mathbf{f}_2 \star \ldots \star \mathbf{f}_n)(t) = (((\mathbf{f}_1 \star \mathbf{f}_2) \star \mathbf{f}_3) \star \ldots \star \mathbf{f}_n)(t).$$

This can be computed using Theorem 5.3 recursively, which yields the above term. Equation 5.43 is equivalent to shifting all functions $\mathbf{f}_j$ by the sum of all T_0^i's other than the own one and taking the minimum. ■

The theorems of this chapter so far will be applied to bandwidth/delay decoupled schedulers in the Sections 5.5 and 5.6. Before that, for the sake of completeness, the convolution of general functions is dealt with.

Even though 5.3 covers the majority of functions concurrently used in network calculus, we point out an efficient algorithm to compute the min-plus convolution of arbitrary functions in this section. The starting point is arbitrary piecewise linear functions. Consider again the functions $\mathbf{f}(t)$ and $\mathbf{g}(t)$ from Equations 5.20 and 5.21, respectively, where

$$\mathbf{f}(t) = \begin{cases} 0 & t \leq T_0^f \\ r_0^f(t - T_0^f) & T_0^f < t \leq T_1^f \\ r_0^f(T_1^f - T_0^f) + r_1^f(t - T_1^f) & T_1^f < t \leq T_2^f \\ \vdots & \\ \sum_{i=0}^{m-2} r_i^f(T_{i+1}^f - T_i^f) + r_{m-1}^f(t - T_{m-1}^f) & T_{m-1}^f < t \leq T_m^f \\ \sum_{i=0}^{m-1} r_i^f(T_{i+1}^f - T_i^f) + r_m^f(t - T_m^f) & t \geq T_m^f \end{cases}$$

and

$$\mathbf{g}(t) = \begin{cases} 0 & t \leq T_0^g \\ r_0^g(t - T_0^g) & T_0^g < t \leq T_1^g \\ r_0^g(T_1^g - T_0^g) + r_1^g(t - T_1^g) & T_1^g < t \leq T_2^g \\ \vdots & \\ \sum_{j=0}^{n-2} r_j^g(T_{j+1}^g - T_j^g) + r_{n-1}^g(t - T_{n-1}^g) & T_{n-1}^g < t \leq T_n^g \\ \sum_{j=0}^{n-1} r_j^g(T_{j+1}^g - T_j^g) + r_n^g(t - T_n^g) & t \geq T_n^g \end{cases}.$$

Therefore, the assumptions that $r_m^f < r_{m-1}^f < \ldots < r_0^f$ and $r_n^g < r_{n-1}^g < \ldots < r_0^g$ are dropped. An example function is depicted in Figure 5.8.

Let $\hat{m}$ and $\hat{n}$ denote the number of convex inflection points of $\mathbf{f}(t)$ and $\mathbf{g}(t)$, respectively. Denote by $\hat{T}_i^f$ the ith convex inflection point of function $\mathbf{f}(t)$.

The following theorem expresses the min-plus convolution for arbitrary functions. It is a generalization of Theorem 5.3. The min-plus convolution is obtained by taking the

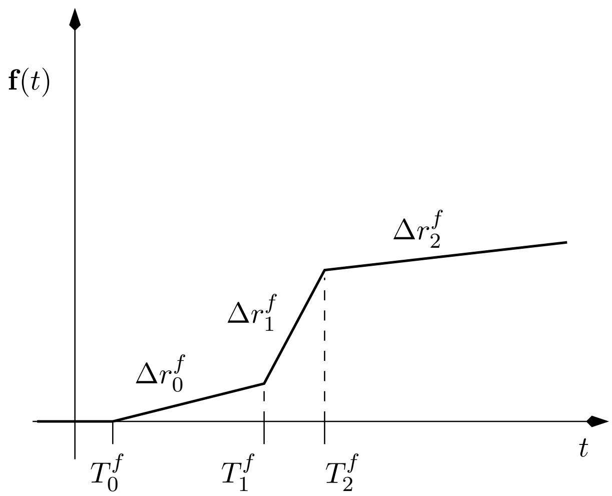

Figure 5.8: Function with arbitrary convex inflection points

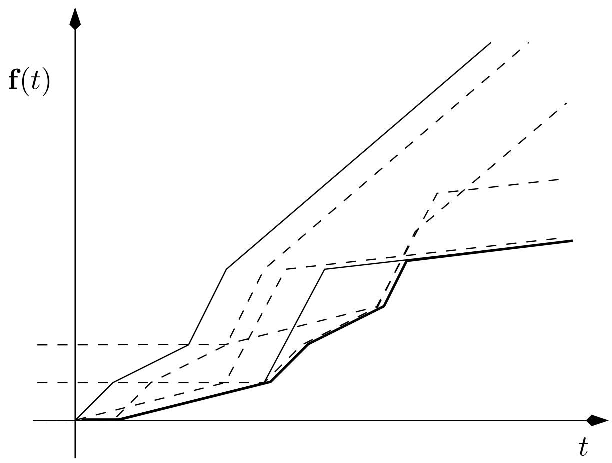

Figure 5.9: Min-plus convolution of arbitrary functions

minimum of all functions which are generated by shifting both functions horizontally and vertically, until the origin of a shifted function lies on all convex inflections of the other one. This is depicted in Figure 5.9.

Theorem 5.6 *The min-plus convolution of arbitrary functions* $\mathbf{f}(t)$ *and* $\mathbf{g}(t)$ *is given by*

$$(\mathbf{f} \star \mathbf{g})(t) = \min\{\min_{i=1,\ldots\hat{m}}\{\mathbf{f}(\hat{T}_i^f) + \mathbf{g}(t - \hat{T}_i^f)\}, \min_{j=1,\ldots\hat{n}}\{\mathbf{g}(\hat{T}_j^g) + \mathbf{f}(t - \hat{T}_j^g)\}\}. \quad (5.44)$$

Proof This proof is given using a geometrical representation. Consider the Figure 5.10 showing selected points of the graph of $\mathfrak{P}\{\mathbf{f}(t)\} \otimes \mathfrak{P}\{\mathbf{g}(t)\}$.

Q_1^* is a point of $\mathfrak{P}\{\mathbf{f}(t)\}$ which is not a convex inflection point. To Q_1^* all points of $\mathfrak{P}\{\mathbf{g}(t)\}$ are added. Of interest is the value of $(\mathbf{f} \star \mathbf{g})(t)$ at time instant t_x. It can be seen that the value of t_x increases if the points of $\mathfrak{P}\{\mathbf{g}(t)\}$ are added above or

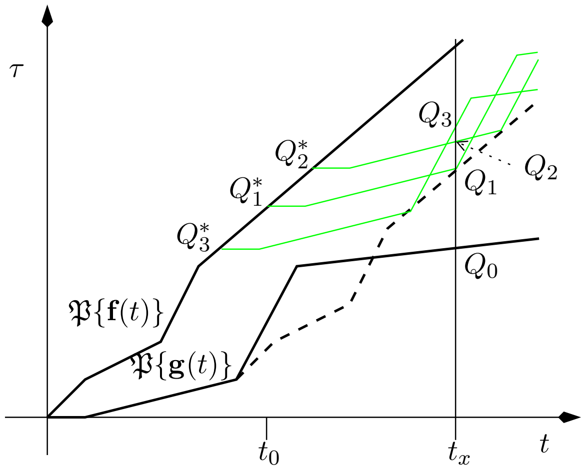

Figure 5.10: Proof of Theorem 5.6

below Q_1^*. Adding those points at Q_2^* and Q_3^* yields the values denoted by Q_2 and Q_3, respectively. It is next examined, under which circumstance adding $\mathfrak{P}\{\mathbf{g}(t)\}$ to Q_1^* yields the minimum at time instant t_x as opposed to adding it to other points of $\mathfrak{P}\{\mathbf{f}(t)\}$ in the vicinity. The value at t_x which is obtained from adding $\mathfrak{P}\{\mathbf{g}(t)\}$ to Q_1^* is $\mathbf{f}(t_0) + \mathbf{g}(t_x - t_0)$. In order for this value to be less than the value obtained from adding $\mathfrak{P}\{\mathbf{g}(t)\}$ to a point just below Q_1^*, i.e.,

$$\mathbf{f}(t_0) + \mathbf{g}(t_x - t_0) < \mathbf{f}(t_0 - \epsilon) + \mathbf{g}(t_x - t_0 + \epsilon), \tag{5.45}$$

the condition on the slopes

$$\frac{\mathbf{g}(t_x - t_0 + \epsilon) - \mathbf{g}(t_x - t_0)}{\epsilon} > \frac{\mathbf{f}(t_0) - \mathbf{f}(t_0 - \epsilon)}{\epsilon} \tag{5.46}$$

is required. Similarly, in order for

$$\mathbf{f}(t_0) + \mathbf{g}(t_x - t_0) < \mathbf{f}(t_0 + \epsilon) + \mathbf{g}(t_x - t_0 - \epsilon) \tag{5.47}$$

to hold, the condition

$$\frac{\mathbf{g}(t_x - t_0) - \mathbf{g}(t_x - t_0 - \epsilon)}{\epsilon} < \frac{\mathbf{f}(t_0 + \epsilon) - \mathbf{f}(t_0)}{\epsilon} \tag{5.48}$$

must hold. Recall the assumption that Q_1^* is not a convex inflection point, i.e.,

$$\frac{\mathbf{f}(t_0 + \epsilon) - \mathbf{f}(t_0)}{\epsilon} \leq \frac{\mathbf{f}(t_0) - \mathbf{f}(t_0) - \epsilon}{\epsilon}. \tag{5.49}$$

Combining Inequalities 5.46 and 5.49 yields

$$\frac{\mathbf{g}(t_x - t_0 + \epsilon) - \mathbf{g}(t_x - t_0)}{\epsilon} > \frac{\mathbf{f}(t_0 + \epsilon) - \mathbf{f}(t_0)}{\epsilon}. \tag{5.50}$$

Finally, combining Inequalities 5.50 and 5.48 yields

$$\frac{\mathbf{g}(t_x - t_0 + \epsilon) - \mathbf{g}(t_x - t_0)}{\epsilon} > \frac{\mathbf{g}(t_x - t_0) - \mathbf{g}(t_x - t_0 - \epsilon)}{\epsilon}. \tag{5.51}$$

Inequality 5.51 states that the slope of $\mathbf{g}(t)$ is greater right of $t_x - t_0$ than to left of it, in other words, $\mathbf{g}(t)$ has a convex inflection point at $t_x - t_0$. Thus, there is a local minimum at t_x if the points of $\mathfrak{P}\{\mathbf{g}(t)\}$ are added at a point t_0 such that $\mathbf{g}(t_x - t_0)$ is a convex inflection point. Of course, this need not be the global minimum, which is denoted by Q_0 in Figure 5.10. Another way to view this is that a local minimum is obtained by adding points of $\mathfrak{P}\{\mathbf{f}(t)\}$ to a convex inflection point of $\mathfrak{P}\{\mathbf{g}(t)\}$. This is denoted by the dashed line in Figure 5.10. The same deliberation holds for adding points of $\mathfrak{P}\{\mathbf{f}(t)\}$ to other convex inflection points of $\mathfrak{P}\{\mathbf{f}(t)\}$, as well as adding the points of $\mathfrak{P}\{\mathbf{g}(t)\}$ to convex inflection points of $\mathfrak{P}\{\mathbf{f}(t)\}$. The global minimum is obtained by taking the minimum of all local minima, i.e., by taking the minimum over all functions obtained by shifting one function such that its origin lies on all convex inflection points of the other and vice versa. ■

5.4 Optimal Network Service Curve

In the remainder of this chapter we apply the theorems derived above. They are of use when analyzing the relationship between a network service curve and node service curves. In particular, we show what inferences can be made on the node service curves given an optimal network service curve.

Schmitt [93] develops the optimal service curve for a bandwidth/delay-decoupled scheduler. The optimal service curve is denoted by $\boldsymbol{\omega}(t)$. The bandwidth/delay-decoupled scheduler is characterized by a L2R service curve, i.e., a service curve that has a latency, and two rates. The latency is determined by factors such as the hardware and the scheduling discipline and is therefore not a design parameter for us. After the latency $l^{\omega,1}$ the flow is served with a rate $r^{\omega,1}$ up to a certain time instant which we refer to as the inflection point $l^{\omega,2}$. After that it switches down to rate $r^{\omega,2}$, which is the sustained rate of the arrival curve. As depicted in Figure 5.11, $\boldsymbol{\omega}(t)$ is given by the 4-tuple $(l^{\omega,1}, l^{\omega,2}, r^{\omega,1}, r^{\omega,2})$. Further, we define $t^{\omega} = l^{\omega,2} - l^{\omega,1}$, which is the time instant that the flow is served at peak rate.

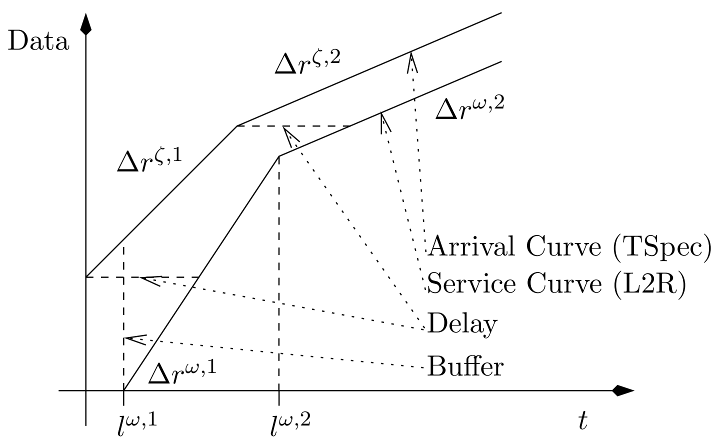

Figure 5.11: Optimal network service curve

In this case, optimality is defined on a per-flow perspective, i.e., minimizing the resource consumption of a single flow. The optimal network service curve for an arrival curve is derived. In this work we extend those results to a wider perspective, in particular we concentrate on two aspects. The implications of an optimal network service curve to the service curves of the nodes along the path are studied as well as the allocation of resources within one node to accommodate as many flows as possible. It turns out that the inflection point is chosen such that the delay bound is met at the point where the arrival curve makes its last rate change to the sustained rate. For the TSpec (cf. Section 2.2 and Equation 2.32), this is the point where it switches from peak rate to sustained rate, as shown in Figure 5.11. Besides deriving the optimal service curve, numerical examples are given which show the benefit of this approach. Schmitt further discusses related work. Most interesting are the deterministic schedulers, as they are the basis for allocating resources with the service curve approach. Sariowan et al. [91] and Stoica et al. [106] describe algorithms which schedule according to service curves.

5.5 Allocation of Service Curves along a Path

In this section and the following one we apply the theorems derived above. The starting point is the bandwidth/delay decoupled scheduler [93]. This section deals with the relationship of the network service curve to the node service curves. To recall, Schmitt derived the optimal network service curve for flows. The network service curve indicates the service the entire network has to provide to a flow for it to meet its QoS requirement. In order for the network to provide deterministic QoS to a flow, all routers which the flow traverses have to provide a deterministic service. In this section the question answered, which service curves the node along the path optimally should provide. The goal is to meet the QoS requirements, in particular the delay bound, of a flow while allocating as little resources to it as possible.

5.5.1 Determining Optimal Node Service Curves

With the theorems of the previous section we have established a relationship between functions and their min-plus convolution. Recall that the network service curve is the concatenation, i.e., the min-plus convolution, of the node service curves (cf. Theorem 2.9). Therefore, we can use our theorem to answer the question, which service curves the nodes on the path must have, so that the concatenation of these service curves yields the network service curve desired for the flow. Assume a network service curve $\boldsymbol{\omega}(t)$ has been obtained as shown in Section 5.4. Further, assume n nodes, where each node has a service curve $\boldsymbol{\beta_j}(t)$, with $j = 1 \dots n$. Applying Theorem 5.5 we obtain

$$\boldsymbol{\omega}(t) = \min_{j=1,2,\dots,n} \{\boldsymbol{\beta_j}(t + l_j^{\omega,1} - \sum_{i=1}^{n} l_i^{\omega,1})\}. \tag{5.52}$$

It can be seen that the latency $l^{\omega,1}$ of the network service curve is the sum of all $l_j^{\omega,1}$'s of all node service curves. To the right of $l^{\omega,1}$, the slope of the network service curve will first be the minimum of all $r^{\omega,1}$'s of all node service curves. In other words, if any node has an $r_j^{\omega,1}$ higher than another one, it is wasted as only the minimum defines the network service curve. This property is depicted in Figure 5.6. The functions $\mathbf{f}(t)$ and $\mathbf{g}(t)$ are to be interpreted as service curves. The area between the dashed lines and the solid line denoting the convolution are wasted resources. The same rationale applies for all times, therefore it is beneficial that all node service curves are equal after their convex inflection point. Therefore, optimal node service curves are such the we have for all $i, j = 1, \ldots, n$

$$t_i^\omega = t_j^\omega, \quad r_i^{\omega,1} = r_j^{\omega,1}, \quad r_i^{\omega,2} = r_j^{\omega,2}. \tag{5.53}$$

Setting the latencies $l_j^{\omega,1}$ is a question of delay distribution along a path, which we will not discuss further. Trivially, if one node offers less resources the delay bound cannot be met anymore without the other nodes raising their resources. However, if one node offered a higher rate $r_j^{\omega,1}$ or $r_j^{\omega,2}$, or a longer peak rate interval $r_j^{\omega,1}$, this would not have any effect on the network service curve and therefore resources would be wasted. This would prevent more flows from being admitted.

5.5.2 Numerical Example

The following numerical examples illustrate these findings. Assume a low bandwidth, short delay flow that satisfies a TSpec with peak rate 9000 bytes/s, sustained rate 1000 bytes/s and buffer 2000 bytes. Without loss of generality we assume a fluid model, i.e., $b^{\zeta,1} = 0$. Hence we have

$$\boldsymbol{\zeta}_1(b^{\zeta,2}, r^{\zeta,2}, r^{\zeta,1}, b^{\zeta,1}) = \min\{(9000 \text{ bytes/s})t, (2000 \text{ bytes}) + (1000 \text{ bytes/s})t\}. \tag{5.54}$$

In other words, a flow can send at its peak rate for 250 ms but then has to reduce the rate to the sustained rate, i.e.,

$$\boldsymbol{\zeta}_1(b^{\zeta,2}, r^{\zeta,2}, r^{\zeta,1}, b^{\zeta,1}) = \begin{cases} 0 & t \le 0 \\ (9000 \text{ bytes/s})t & 0 \le t \le 250 \text{ ms} \\ (2000 \text{ bytes}) + (1000 \text{ bytes/s})t & t > 250 \text{ ms} \end{cases}. \tag{5.55}$$

The flow traverses a path consisting of 5 nodes and has a delay bound of 500 ms. Using the method outlined in Section 5.4 the optimal network service curve is given by

$$\boldsymbol{\omega}(t) = \begin{cases} 0 & t \le 500 \text{ ms} \\ (9000 \text{ bytes/s})(t - 500 \text{ ms}) & 500 \text{ ms} \le t \le 750 \text{ ms} \\ (2250 \text{ bytes}) + (1000 \text{ bytes/s})(t - 750 \text{ ms}) & t > 750 \text{ ms} \end{cases}. \tag{5.56}$$

Assuming an equal delay distribution over all nodes, the optimal node service curve for each node is

$$\boldsymbol{\beta}_1(t) = \begin{cases} 0 & t \le 100 \text{ ms} \\ (9000 \text{ bytes/s})(t - 100 \text{ ms}) & 100 \text{ ms} \le t \le 350 \text{ ms} \\ (2250 \text{ bytes}) + (1000 \text{ bytes/s})(t - 350 \text{ ms}) & t > 350 \text{ ms} \end{cases}. \tag{5.57}$$

5.6 Reallocation of Service Curves in Nodes

5.6.1 Local Reallocation

In this subsection we consider a single node. There are instances where altering the service curves within a node can lead to increasing the number of flows admitted. The rationale is explained by a numerical example and depicted in Figure 5.12. There

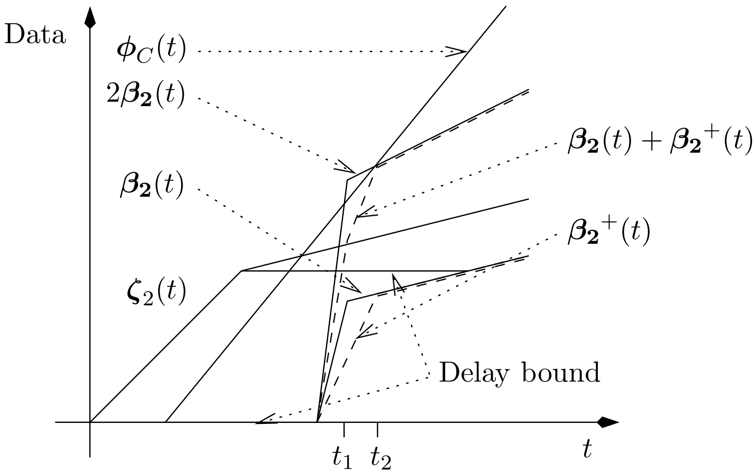

Figure 5.12: Local reallocation

is a maximum service curve which denotes the capacity of a node. Without loss of generality, we assume this to be an LR service curve, which we call the capacity service curve $\phi_C(t)$. Assume a node has two requests of flows, each with the TSpec arrival curve $\boldsymbol{\zeta}_2(t)$, where

$$\boldsymbol{\zeta}_2(t) = \min\{(10000 \text{ bytes/s})t, (1500 \text{ bytes}) + (2500 \text{ bytes/s})t\}. \tag{5.58}$$

A flow can send at its peak rate for 250 ms and then has to reduce the rate to the sustained rate. The alternative representation of this TSpec is

$$\boldsymbol{\zeta}_2(t) = \begin{cases} 0 & t \leq 0 \\ (10000 \text{ bytes/s})t & 0 \leq t \leq 200\text{ms} \\ (1500 \text{ bytes}) + (2500 \text{ bytes/s})t & t > 200\text{ms} \end{cases} . \tag{5.59}$$

Assume a delay requirement $\delta = 200$ ms. The service curve $\boldsymbol{\beta_2}(t)$ of a bandwidth/delay decoupled scheduler, where

$$\boldsymbol{\beta_2}(t) = \begin{cases} 0 & t \leq 300 \text{ ms} \\ (40000 \text{ bytes/s})(t - 300 \text{ ms}) & 300 \text{ ms} \leq t \leq 340 \text{ ms} \\ (1600 \text{ bytes}) + (2500 \text{ bytes/s})(t - 340 \text{ ms}) & t > 340 \text{ ms} \end{cases} , \tag{5.60}$$

ensures that the delay bound is met at the two crucial points, namely, the inflection points of the arrival curve. Per se neither can be accepted as the sum of the service curves $2\boldsymbol{\beta_2}(t)$ exceeds the capacity service curve $\boldsymbol{\phi}_C(t)$ between t_1 and t_2. However, if a service curve $\boldsymbol{\beta_2}^+(t)$ can be found such that the delay bound remains fulfilled at the two crucial points and the sum $\boldsymbol{\beta_2}(t) + \boldsymbol{\beta_2}^+(t)$ never exceeds the capacity service curve $\boldsymbol{\phi}_C(t)$, both flows can be accommodated. For this example, a possible $\boldsymbol{\beta_2}^+(t)$ is given by

$$\boldsymbol{\beta_2}^+(t) = \begin{cases} 0 & t \le 300 \text{ ms} \\ (21250 \text{ bytes/s})(t - 300 \text{ ms}) & 300 \text{ ms} \le t \le 380 \text{ ms} \\ (1720 \text{ bytes}) + (2500 \text{ bytes/s})(t - 380 \text{ ms}) & t > 380 \text{ ms} \end{cases} . \tag{5.61}$$

From Theorem 5.3 we can be sure that the network service curve is not affected if one or more nodes apply this method.

5.6.2 Global Reallocation

We next discuss how deficiencies of one node can be compensated by other nodes. We require the following definition.

Definition 5.7 (Compensated latency) *We define compensated latency l_c as the time that has to be reduced from the initially allocated latency in order to meet the service curve requirement. Depending on the context, the reduction is done either by the shortcoming node itself, or one other node along the path, or a combination of nodes along the path.*

There are 3 ways in which a node can fail to offer the demanded service and therefore be forced to reject a request: latency, peak rate and sustained rate. If a node cannot offer the sustained rate, then the flow cannot be accepted. There is no way that the other nodes can compensate this, as the sustained rate of the network service curve is determined by the smallest sustained rate of all nodes. A node not making the delay requirement can be compensated rather easily. The time that it exceeds the delay requirement has to be made up by one or many of the other nodes guaranteeing a lower latency, such that the sum of the latencies of all nodes remains the required latency. This corresponds directly to the shift of Theorem 5.5. To illustrate this with the example of Section 5.5.2, if one node can only offer a latency of 140 ms, then the network service curve can still be achieved if the other nodes along the path save 40 ms. This can either be done by reducing the latency of one node to 60 ms or by the 4 other nodes having a latency of 90 ms each, or any other combination for which the sum of all latencies is 500 ms. The last possible failure is when a node is not able to offer the demanded peak rate. Here we distinguish whether the node is not able to offer the rate itself or whether it is not able to offer the rate for the required time interval. A crucial quantity is β, which we define as the amount of data that is served at the inflection point. It follows from Theorem 5.3 that having less data served at the inflection point would decrease the minimum of the shifted functions. This would destroy the network service curve as the inflection point is crucial for the delay bound.

Consider a node that can only offer a deficient peak rate r_d. If it can offer it for a time l_d, such that $r_d l_d = \beta$, then the delay bound can be met by reducing the latency. The amount by which the latency has to be reduced, $l_d - t^{\omega}$, is obtained by considering that

$$\beta = r_d l_d = r^{\omega,1} t^{\omega} \tag{5.62}$$

and

$$l_c = l_d - t^{\omega} = \beta(\frac{1}{r_d} - \frac{1}{r^{\omega,1}}). \tag{5.63}$$

Again, it is arbitrary which nodes make up the latency. This case is depicted in Figure 5.13, where Δ again labels the slope. The next case is a node that can offer the

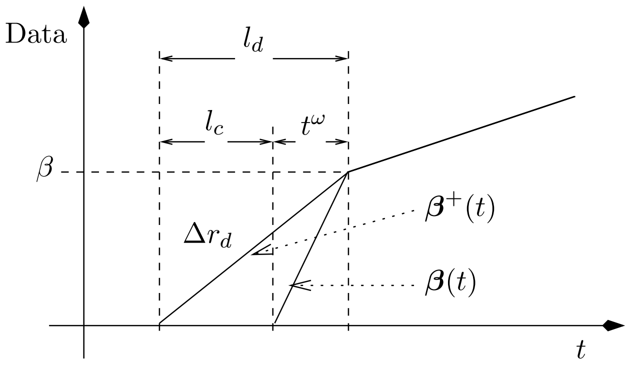

Figure 5.13: Compensated latency with deficient peak rate

peak rate only for a time $l_t < t^{\omega}$. The term l_x is the time needed to reach β after the peak rate stopped. The latency that has to be compensated is then $l_t + l_x - t^{\omega}$, with

$$\beta = r^{\omega,1} l_t + r^{\omega,2} l_x = r^{\omega,1} t^{\omega} \tag{5.64}$$

and

$$l_c = l_t + l_x - t^{\omega} = \frac{(r^{\omega,1} - r^{\omega,2})(\beta - r^{\omega,1} l_t)}{r^{\omega,1} r^{\omega,2}}. \tag{5.65}$$

This case is depicted in Figure 5.14. It can be seen that if the ratio between peak rate $r^{\omega,1}$ and sustained rate $r^{\omega,2}$ is large the compensated latency grows rapidly. Note that the service curves $\boldsymbol{\beta}^{+}(t)$ in this section are not optimal, in the sense that they guarantee a lower delay for the first packets than required. This does not improve the overall performance as the worst case delay, which is at the inflection point, remains untouched. Therefore, some resources are wasted. The bottom line is that no shortcoming of a node can be compensated by any other offering only a higher rate. All compensations require other nodes guaranteeing a lower latency.

5.7 Recipe for Application

5.7.1 Min-plus Convolution

The $\mathfrak{P}$-transform derived in this chapter allows functions to be convolved in the min-plus algebra in an intuitive manner. This method contains three steps as given in

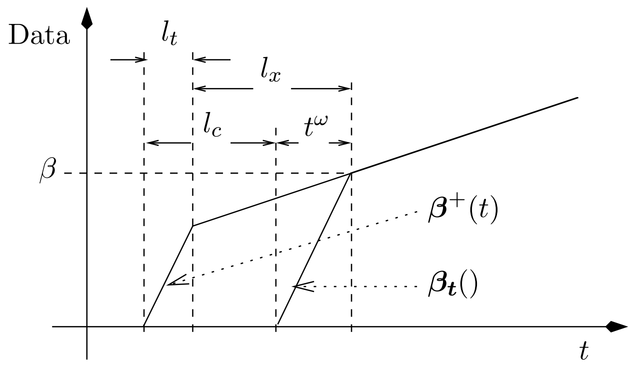

Figure 5.14: Compensated latency with deficient peak rate serving time

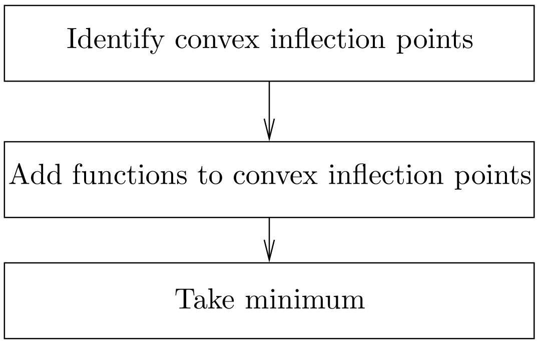

Figure 5.15: Three steps for computing the min-plus convolution

Figure 5.15. The method is based on the $\mathfrak{P}$-transform, which states that the min-plus convolution is given by the lower border of the area obtained by adding all points of one function to all points of the other one. The border is obtained by taking the minimum over all points. A property of the border is that only points added at convex inflection points influence it, therefore, only such points have to be considered. With this method problems involving many convolution computations can be solved efficiently.

5.7.2 Service Curve Based Admission Control

One field where many convolutions are necessary, and therefore our method is destined to be used, is admission control based on service curves. The basic steps for conducting such admission control are outlined in Figure 5.16. Upon a request from a flow, a path for it through the network has to be selected. The network service curve that the flows experience on that path can be obtained by convolving the service curves offered by the nodes along the path. This is the step where our method for convolving functions is useful. By computing the relevant bounds with the arrival curve and the network service curve, it is determined whether the desired QoS performance is achieved. If this is the case, the flow is accepted. If not, an alternative path is selected and the procedure repeated. If no path yields the desired QoS performance the flow is rejected.

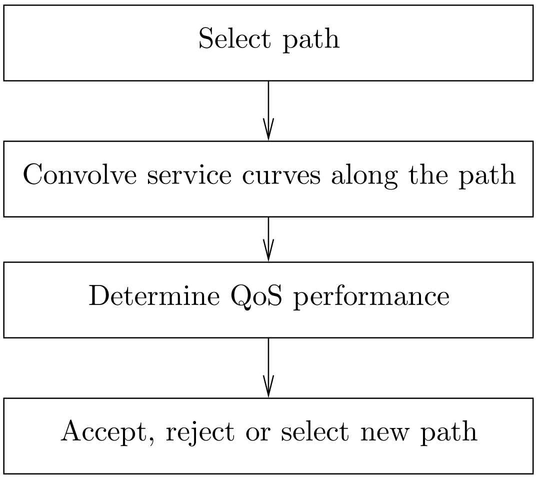

Figure 5.16: Four steps for admission control

5.7.3 Allocating Service Curves in Nodes

Finally, the results from this chapter give insight to optimally allocating resources along a path so that a network service curve is achieved. This is closely related to the admission control procedure, where the question is whether it is possible to accommodate a flow in the network. In contrast this procedure, outlined in Figure 5.17, deals with how to optimally choose the node service curves along the path so that the network service curve is obtained. The first step is to determine the network service

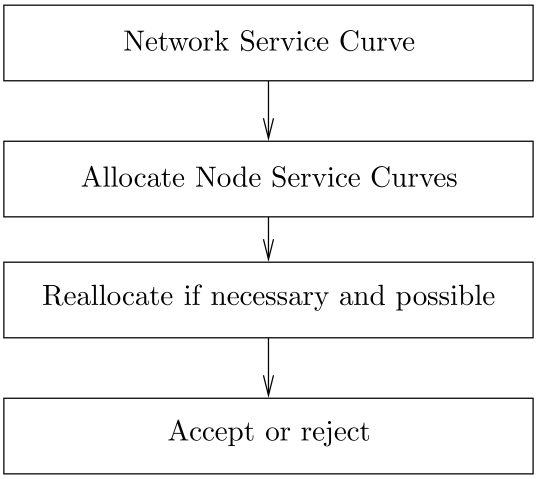

Figure 5.17: Four steps for service curve allocation

curve. The next step is allocating the node service curves according to the rationale of Section 5.5. The service rates and the durations how long they are served should be equal in all nodes, since the network service curve is the minimum over all node service curves. Therefore, if one node has a higher rate or a longer duration a rate is served than other nodes, resources are wasted. However, there are cases where a node is not able to offer the path optimal service curve. Then this procedure becomes an admission control problem, and it has to be checked whether the flow can be accommodated by reallocating flows in the node. An accept or reject decision concludes the procedure.

5.8 Summary

In this chapter we developed the continuous $\mathfrak{P}$-transform and devised a theorem on the computation of the min-plus convolution under network calculus constraints. These results advance the research in network calculus. Beyond that our results were applied to assist admission control decisions in networks with flows described by arrival curves. With the theorem, optimal service curves of single nodes along a path in order to meet a network service curve were determined. Further, we showed how shortcomings of nodes can be compensated locally and globally to improve admission control.

6 Network Calculus Assisted Queueing Theory

In its original form, network calculus is a system theory for deterministic queuing systems. The worst case performance bounds are computed. Historically, network calculus was intended to analyze IntServ and ATM networks. Their nature explains why an emphasis was made on deterministic guarantees (cf. Section 2.2). However, in concurrent network scenarios, many times the worst case performance bounds are too conservative, which leads to a low resource utilization. This calls for probabilistic models. We are interested in network calculus systems with random inputs, and therefore, the input is to be modeled by a stochastic process.

The first thing that comes to mind when considering probabilistic models for packet switched networks is queuing theory. The basics of queuing theory are reviewed in Section 2.4. Queuing theory has been for 40 years the method of choice for modeling packet-switched networks. One reason for its success is that it is very easily tractable and therefore usable for practitioners. Since the motivation of this thesis is to increase the applicability of network calculus, this is a very important factor. Hence, queuing theory will play a prominent role in this chapter. As opposed to network calculus, queuing theory mainly deals with the average behavior of queues. While the worst case analysis from network calculus is often too strict, considering only the average behavior might be too loose. There are queuing theory results which give results beyond average behavior, but unfortunately they are highly complex. We bring together the best of both worlds, and enhance queuing theory with elements of network calculus. However, in order to maintain tractability, we limit ourselves to basic methods from network calculus, namely, the arrival curve and service curve concept, as well as the backlog bound (cf. Section 2.5).

The approach followed in this work is related to the combination of queuing theory and Petri Nets, which is known as Stochastic Petri Nets. A reference for Stochastic Petri Nets is the book by Baccelli et al. [6]. An alternative approach to enhance network calculus with probabilistic methods is presented in Section 3.2.2. There the probability is brought in by assuming that a flow is not strictly constrained by a deterministic arrival curve but violates it with a certain probability. Similarly, the scheduler violates the service curve with a certain probability.

The remainder of this chapter is organized as follows. We first describe our system model. Then we present some analytically obtained results and point out that even simple cases lead to intractability. The main part of this chapter then deals with a simulation based approach. Finally, we offer recipes for practitioners to evaluate the performance of queuing systems.

6.1 System Model

The goal of this chapter is to perform a queuing analysis of a M/M/1 queue enhanced with methods from network calculus. In this section the system model is introduced. The starting point is a traditional queue, consisting of a buffer and server, as shown in Figure 6.1. The input process is given by $\overline{\mathbf{x}}(k)$ and the output process by $\overline{\mathbf{y}}[k]$. For

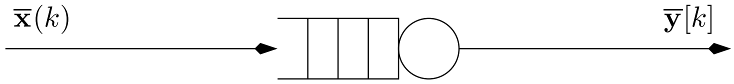

Figure 6.1: Queuing system

the $M/M/1$ case the input and output processes are both Poisson processes. Recall that a Poisson process does not exclude an arbitrarily large number of arrivals in an arbitrarily small (however not infinitesimal) time interval. Similarly, the service times can become arbitrarily large since they are exponentially distributed. Due to this property it is not possible to provide deterministic QoS guarantees when Poisson processes are underlying.

Deterministic QoS guarantees can be provided if the traffic and scheduler are described by an arrival curve and service curve, respectively. Therefore, we modify the M/M/1 queue by shaping the input process and ensuring that the server offers a service curve. I.e., we introduce a shaper that manipulates the arrival times of the packets before they enter the queue and a service curve enforcer, which manipulates the service times. The resulting system, which is referred to as bounded queue, is depicted in Figure 6.2, where "S.C.E." denotes the service curve enforcer. This modified queuing model can

Figure 6.2: Shaped queue

be described in Kendall's notation as a $G_1/G_2/1$ queue. G_1 is a modified arrival process and G_2 a modified service process, e.g., a truncated exponential distribution. In the following the mode of functioning of the shaper and service curve enforcer is elaborated.

6.1.1 Shaper

The shaper ensures that the packets entering the queue are constrained by an arrival curve. Here a greedy shaper is used, whose output $\mathbf{x}'(t)$ is obtained by the min-plus convolution of $\overline{\mathbf{x}}(t)$ and the shaping curve (cf. Theorem 2.6). In other words, the shaper works as follows. When a packet arrives at the shaper, the shaper checks whether there are enough tokens in the shaper to admit the packet. Without loss of generality we assume all packets to be of size 1 throughout this chapter. If that is the case, then the packet traverses the shaper infinitely fast and arrives at the queue. If there are not enough tokens to admit the packet, the packet is held in the shaper until enough tokens have been collected. Therefore, the shaper theoretically has an infinitely large buffer. Since packets are delayed, the shaper might decrease the rate of the process. We define a new arrival rate

$$\lambda' = \frac{n}{T^{obs}}, \tag{6.1}$$

where n is the number of packets observed during a time interval T^{obs}. Note that in our model the shaper is only a conceptual model rather than an actual device holding packets. In many cases, a higher layer such as the application layer ensures that all traffic is constrained by an arrival curve. Even if this is not done explicitly, there are means to determine an arrival curve for a flow. Many times the maximum rate of a flow is known and the maximum burstiness can be obtained from knowledge of the buffer properties along the path.

6.1.2 Service Curve Enforcer

When a packet arrives at the queue, a check is made whether the server is available. If this is the case, the packet receives service immediately, otherwise it waits in a queue until the server becomes available. Once the packet enters the server, an exponentially distributed service time is assigned to it. Note that this can be arbitrarily long. The service curve enforcer then checks whether the service time is less than or equal to the maximum service time allowed by the service curve. If this is the case, the service time remains untouched, otherwise the service time is set to the maximum allowed service time. This corresponds to a truncated exponential distribution function. The service curve enforcer therefore increases the server rate. Accordingly, we define a new server rate

$$\mu' = \frac{\sum_{i=1}^{n} \mu_i^{obs}}{n}, \tag{6.2}$$

where μ_i^{obs} is the actual, observed server rate of the ith packet.

Note that while the shaper can only delay packets, the service curve enforcer releases packets ahead of schedule. Therefore, its placement behind the server seems counter-intuitive as the transfer function mapping $\overline{\mathbf{y}}[k]$ to $\overline{\mathbf{y}}'[k]$ is non-causal.

Like the shaper the service curve enforcer is not an actual device. It is given by the properties of the link. E.g., assume the capacity service curve $\boldsymbol{\phi}_C(t)$ of the link is known. Further, if the sum of the arrival curves of all other flows on the link is known, the service curve can be obtained by using the result from aggregate scheduling (cf. Section 3.3.1).

6.2 Analytical approaches

The main goal of this section is to point out the intractability of the analytical approach to this problem and therefore justify the simulative approach.

6.2.1 Brute Force Method

In order to get a feel for the results, a brute force analysis is a popular first approach. Its methodology is outlined in Figure 6.3. Slotted time is considered. In each slot,

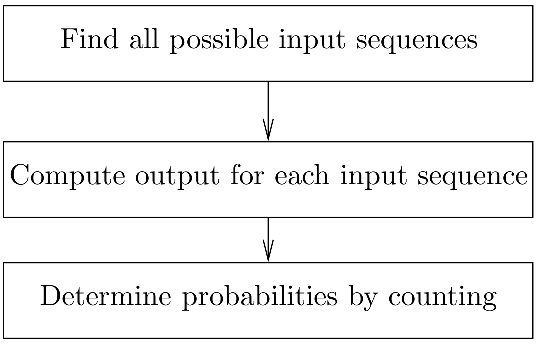

Figure 6.3: Methodology of the brute force analysis

either one packet arrives with a probability p or no packets arrive with $1-p$. The arrival in a slot is independent from any other slot. Therefore, the underlying process is a Bernoulli process $\overline{\mathbf{w}}[k]$. This is known as an on/off source and used as a simple model for e.g., cell arrivals in ATM. This process is shaped with a greedy token bucket shaper according to Theorem 2.6. Upon an arrival a check is made as to whether a token is available. If a token is available, the packet is sent immediately, otherwise, it is delayed until a token becomes available. Of course, the output of the shaper is not a Bernoulli process anymore, as the probability $\mathbf{P}\{\overline{\mathbf{w}}'[k]=1\}$ is not independent of other slots anymore.

For an interval $k=0,\dots n$, the number of possible input functions is 2^n. In the brute force method, the output of the shaper for each possible input function is computed. Here it is done for $n=18$. The probability $\mathbf{P}\{\overline{\mathbf{w}}[k]=1\}$ can be obtained by counting all shaped Bernoulli processes.

We set the bucket depth of the leaky bucket to $b_\gamma=1$ and the token rate $r_\gamma=0.5$. The sequence $\overline{\mathbf{w}}'[k]$ is the output of the greedy token bucket shaper of the input $\overline{\mathbf{w}}[k]$. We set $p=0.5$. With the brute force analysis, we obtain the probability $\mathbf{P}\{\overline{\mathbf{w}}'[k]=1\}$ for $k=1,2,...,15$. Note that $\overline{\mathbf{w}}'[k]=0$ for $k\le 0$. The results are depicted in Figure 6.4.

By carefully observing the pattern, one finds that they follow the equation

$$\mathbf{P}\{\overline{\mathbf{w}}'[k]=1\}=\begin{cases} 0.5 & \text{for odd } k \\ 0.5-\frac{k!}{2^{k+1}\left(\frac{k}{2}!\right)^2} & \text{for even } k \end{cases}. \tag{6.3}$$

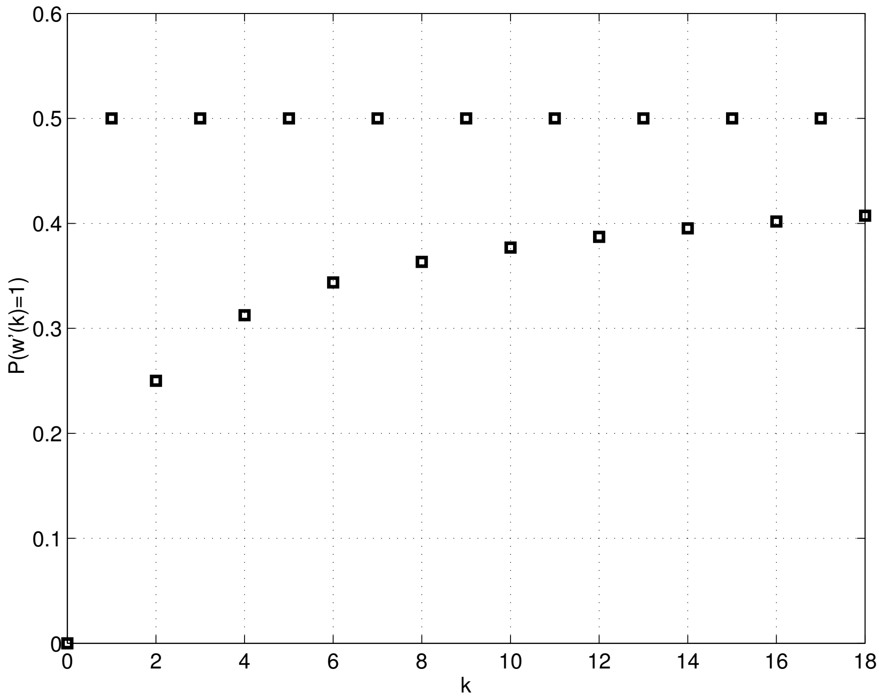

Figure 6.4: Brute force analysis (Equation 6.3)

It can be shown that

$$\lim_{k\to\infty} \mathbf{P}\{\overline{\mathbf{w}}'[k] = 1\} = 0.5. \tag{6.4}$$

The brute force analysis shows that even for a simple set of parameters, the random process at the output of the shaper is rather complex. As expected, the shaped process is not stationary. It also becomes clear that an asymptotic analysis is not useful in this case, as it ignores the burstiness of the flow.

6.2.2 Random Walk Method

Now a different approach to the analytical solution of a shaped process with Bernoulli arrivals is considered. It turns out that shaping a Bernoulli process is related to a random walk process. Again, let $\overline{\mathbf{w}}[k]$ be a Bernoulli process

$$\mathbf{P}\{\overline{\mathbf{w}}'[k]\} = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases} . \tag{6.5}$$

The cumulated process is then

$$\overline{\mathbf{x}}[k] = \sum_{l=1}^{k} \overline{\mathbf{w}}[l]. \tag{6.6}$$

Let $\mathbf{h}[k]$ be the shaping function, a discretized token bucket curve,

$$\mathbf{h}[k] = \begin{cases} 0 & k \leq 0 \\ b_\gamma + r_\gamma k & k > 0 \end{cases}. \tag{6.7}$$

The output of the shaper is given by the min-plus convolution of the input function and the shaping function,

$$\mathbf{x}'[k] = [\overline{\mathbf{x}} \star \mathbf{h}][k] = \min_{0 \leq l \leq k} \{\overline{\mathbf{x}}[l] + \mathbf{h}[k-l]\}. \tag{6.8}$$

The goal is now to characterize the stochastic process at the output of the shaper. The interval over which the minimum is computed can be divided into the three parts,

$$\min_{0 \leq l \leq k} \{\overline{\mathbf{x}}[l] + \mathbf{h}[k-l]\} = \min\{\overline{\mathbf{x}}[0] + \mathbf{h}[k-0], \overline{\mathbf{x}}[k] + \mathbf{h}[k-k], \min_{0<l<k} \{\overline{\mathbf{x}}[l] + \mathbf{h}[k-l]\}\}. \tag{6.9}$$

Considering that $\overline{\mathbf{x}}[0] = \mathbf{h}[0] = 0$ yields

$$\mathbf{x}'[k] = \min\{\overbrace{\mathbf{h}[k]}^{I}, \overbrace{\overline{\mathbf{x}}[k]}^{II}, \overbrace{\min_{0<l<k} \{\overline{\mathbf{x}}[l] + \mathbf{h}[k-l]\}}^{III}\}. \tag{6.10}$$

Evaluating the term III from Equation 6.10 yields an interesting result,

$$\begin{aligned} \min_{0<l<k} \{\overline{\mathbf{x}}[l] + \mathbf{h}[k-l]\} &= \min_{0<l<k} \{\overline{\mathbf{x}}[l] + b_\gamma + r_\gamma[k-l]\} & (6.11) \\ &= b_\gamma + r_\gamma k + \min_{0<l<k} \{\overline{\mathbf{x}}[l] - r_\gamma l\}. & (6.12) \end{aligned}$$

Consider the expression in the brackets of Equation 6.12. Set $r_\gamma = 0.5$ and recall Equation 6.5. The expression depicts a random walk [90], where at each step 0.5 is randomly added or subtracted. Therefore, computing the minimum is equivalent to finding the minimum in a random walk process. Computing the minimum of random walk processes is possible in a probabilistic sense. In the book by Grimmett and Stirzaker [42] it is shown that

$$\mathbf{P}\{\min_{l}\{\overline{\mathbf{x}}[l] - r_\gamma l\} = z\} = 2\mathbf{P}\{\overline{\mathbf{x}}[k-1] - r_\gamma[k-1] \geq z+1\} + \mathbf{P}\{\overline{\mathbf{x}}[k-1] - r_\gamma[k-1] = z\}, \tag{6.13}$$

where

$$\mathbf{P}\{\overline{\mathbf{x}}[k-1] - r_\gamma[k-1] = z\} = \begin{pmatrix} k-1 \\ \frac{1}{2}(k-1+z) \end{pmatrix} p^{\frac{1}{2}(k-1+z)} (1-p)^{\frac{1}{2}(k-1+z)}. \tag{6.14}$$

Connecting the shaping of a Bernoulli process to a random walk is an interesting result. However, integrating this result into the entire minimum computation of Equation 6.10 is tedious as the terms II and III are statistically dependent. Further, analyzing the service curve enforcer is even more tedious, hence we take recourse to a simulative study, which is described in the following section.

6.3 Simulations

6.3.1 Methodology

In this section, the results of a simulation of a bounded queue as given in Figure 6.2 are presented. The steps for the simulation are outlined in Figure 6.5.

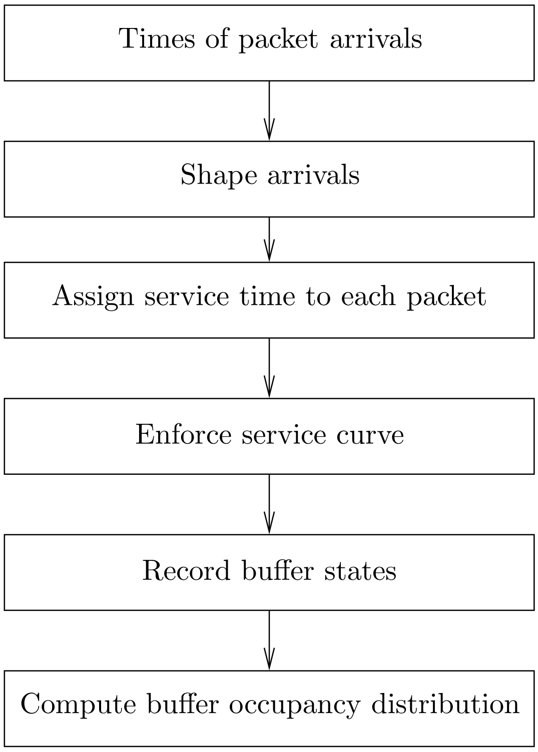

Figure 6.5: Simulation methodology

First arrival times for packets are randomly generated creating a Poisson process. Then the Poisson process is shaped. As shaping curve the token bucket from Equation 2.31 is used. Each packet is assigned a service time. Then the service curve enforcer manipulates the service times. The service curve of choice is the latency rate curve from Definition 2.8. The state of the queue is observed throughout the simulation and finally the buffer state probabilities computed.

We call packets which are delayed by the shaper, shaper manipulated packets. Accordingly, we call packets which are served earlier due to the service curve, server manipulated packets.

Therefore, the parameters of our system model are $(\lambda, \mu, b_\gamma, r_\gamma, l_\phi, r_\phi)$, which denote the arrival rate, service rate, token bucket depth, token bucket rate, latency of the latency rate scheduler and rate of the latency rate scheduler, respectively.

k	$\mathbf{P}_A(k)$	$\mathbf{P}^l_A(k)$	$\mathbf{P}^u_A(k)$	$\mathbf{P}_{0.58}(k)$
0	0.4180	0.4152	0.4207	0.4179
1	0.2831	0.2810	0.2852	0.2433
2	0.1581	0.1561	0.1600	0.1416
3	0.0796	0.0782	0.0809	0.0392
4	0.0380	0.0367	0.0392	0.0480
5	0.0162	0.0152	0.0171	0.0279
6	0.0058	0.0053	0.0064	0.0163
7	0.0013	0.0010	0.0016	0.0095
8	0	0	0	0.0055
9	0	0	0	0.0032

Table 6.1: Bounded Queue vs. M/M/1

Qualitatively, we expect the following behavior in the simulations of the bounded queue. Trivially, the probability of the states higher than the Backlog Bound from Theorem 2.10 will be 0. The probability of state 0 will remain unchanged in the bounded queue. The reason for this is that the shaper and the service curve enforcer are both inactive when the system is empty. There will be a strong increase in probability mass at the state 1, due to the shaper. The shaper causes the inter-arrival times of packets at the queue to be more equally distributed than in a pure exponential distribution. Note that asymptotically, i.e., when $r_\gamma \ll \lambda$, all inter-arrival times are $\frac{1}{r_\gamma}$ after the initial tokens in the bucket have been emptied. Both, the traffic shaper as well as the service curve enforcer, cause the probability mass to shift towards the lower states. Therefore, the higher states of the bounded queue will be less probable than the same states of the M/M/1 queue.

6.3.2 Results

As parameters of the first simulation we use $(\lambda, \mu, b_\gamma, r_\gamma, l_\phi, r_\phi) = (2, 3, 6, 2, 1, 2)$ Using Theorem 2.10 we obtain that the maximum buffer state is 7. There are 5000 arrivals per run and the simulation is repeated 30 times. The confidence intervals are also given. In Table 6.1 the values of this simulation are given in numerical form. The simulation result is depicted in Figure 6.6.

The terms $\mathbf{P}^l_A(k)$ and $\mathbf{P}^u_A(k)$ denote the lower and upper 0.95 confidence interval end-point, respectively. The average number of input and shaper manipulated packets are 4933 and 276.2, respectively. As a reference, the state probabilities of the corresponding M/M/1 Queue, i.e., with $\rho' = \frac{\lambda'}{\mu} = \frac{1.85}{3.18} = 0.58$, are given. Since the difference to the M/M/1/N queue is marginal, we can neglect it. What is striking here is that the probabilities of the high states of the Bounded Queue are lower than of those states in the M/M/1 case. The probability mass of the higher states is neither distributed evenly among the allowed states, nor is it collected in the last allowed state. This result confirms our assumptions, that putting structure in form of input shaping and service curve enforcement makes the low buffer states more likely at the cost of the high ones.

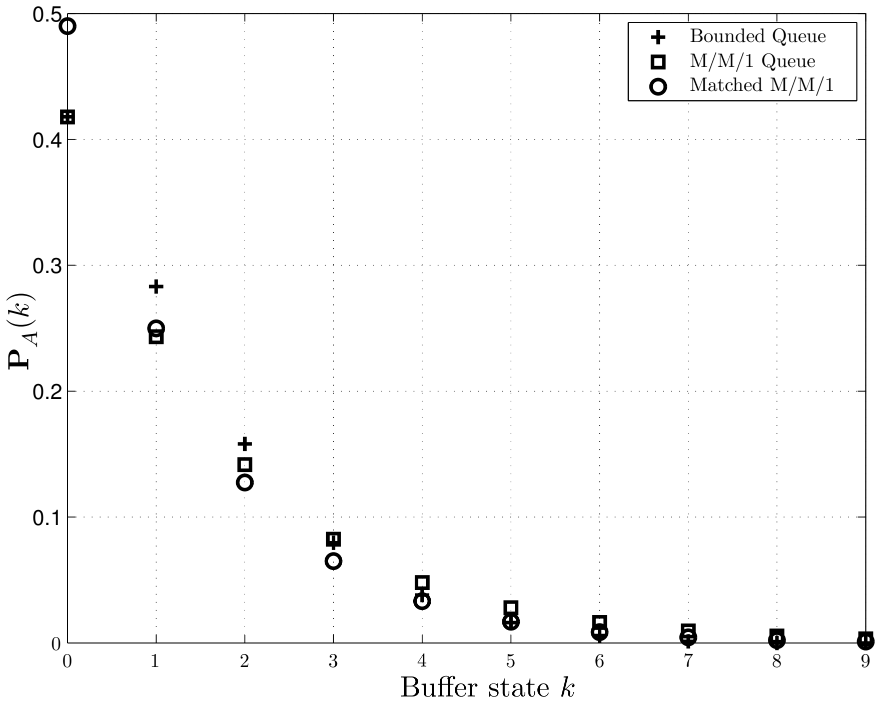

Figure 6.6: Bounded queue vs. M/M/1

The higher buffer states are the relevant ones, as they are responsible for high delays or even loss. The behavior of the queue in the relevant buffer states becomes better. Therefore, we are interested in those states. Considering state 5, we find that the bounded queue has a better behavior than a M/M/1 queue with $\rho^* = 0.51$. This gives us an adjustment factor

$$o = \frac{\rho^*}{\rho'} = 0.88. \tag{6.15}$$

We now analyze several parameter sets in order to get an insight into the influence of the adjustment factor. We hold the parameters $\lambda = \frac{2}{3}$ and $\mu = 1$ and set the token bucket rate equal to the rate of the latency rate service curve $r_\gamma = r_\phi = 1$. In order to compare the buffer occupancy distributions in a fair manner, we ensure that the backlog mound is constant at 7. We therefore set $b_\gamma = 1, 2, \ldots, 7$ and accordingly $l_\phi = 7, 6, \ldots, 1$. The buffer occupancy density functions are shown in Figure 6.7.

In Figure 6.8 the values for λ', μ', ρ' and ρ^* are shown. As reference, λ and μ are also shown. It can be seen that when the bucket depth is low, and consequently the latency is high, μ' is close to μ. Similarly, when the latency is low and the bucket depth high, λ' is close to λ.

In Figure 6.9 the adjustment factor is plotted as a function of the bucket depth b_γ, which decides the number of manipulated packets. It can be seen that it is lowest for the endpoints. This implies that tight shaping or tight service curve enforcement has a stronger influence on the adjustment factor than some shaping and some service curve

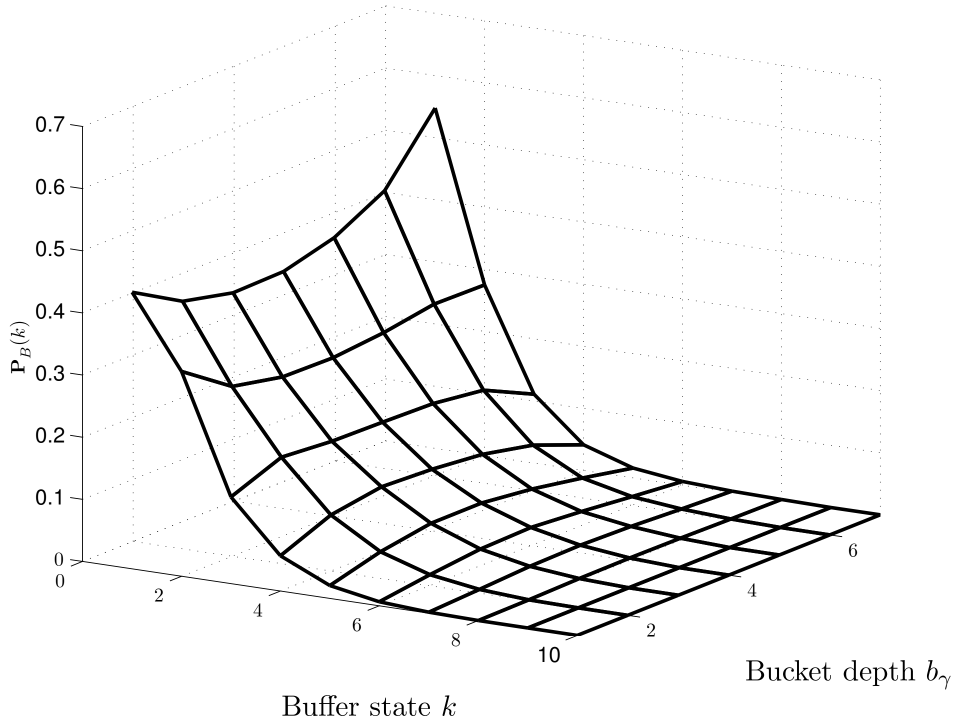

Figure 6.7: Bounded buffer occupancy distribution (I)

enforcement combined. The influence of tight shaping is stronger than that of tight service curve enforcement.

In the following set of simulations we alter only μ. We set it to $\mu = \frac{4}{3}$ and therefore have a utilization of $\rho = 0.5$. The remaining parameters are set similar to the previous simulations. Altering λ is not worth studying further as the ratio $\frac{\lambda}{r_\gamma}$ quickly leads to extreme results, i.e., no packets being touched by the shaper on the one side and, on the other side, losing stability when $\lambda < r_\gamma$. We insist on keeping r_γ constant in order to have a fair comparison. The results are shown in Figures 6.10, 6.11 and 6.12.

The adjustment factors are lower for the less utilized queue when the shaping is tight and as the bucket depth increases, the adjustment factors move closer together. This is counter-intuitive, in the sense that a higher μ causes less instances of the service curve enforcer being active, which implies less structure, which leads to a higher adjustment factor. The reason this logic does not apply, is that tight shaping reduces the number of server manipulated packets, which can be seen to be virtually 0 by comparing μ' to μ. Since we have not altered λ, λ' remains unchanged too. In this scenario, the effect that lower utilized queues decrease faster dominates, which causes the adjustment factor to decrease as well.

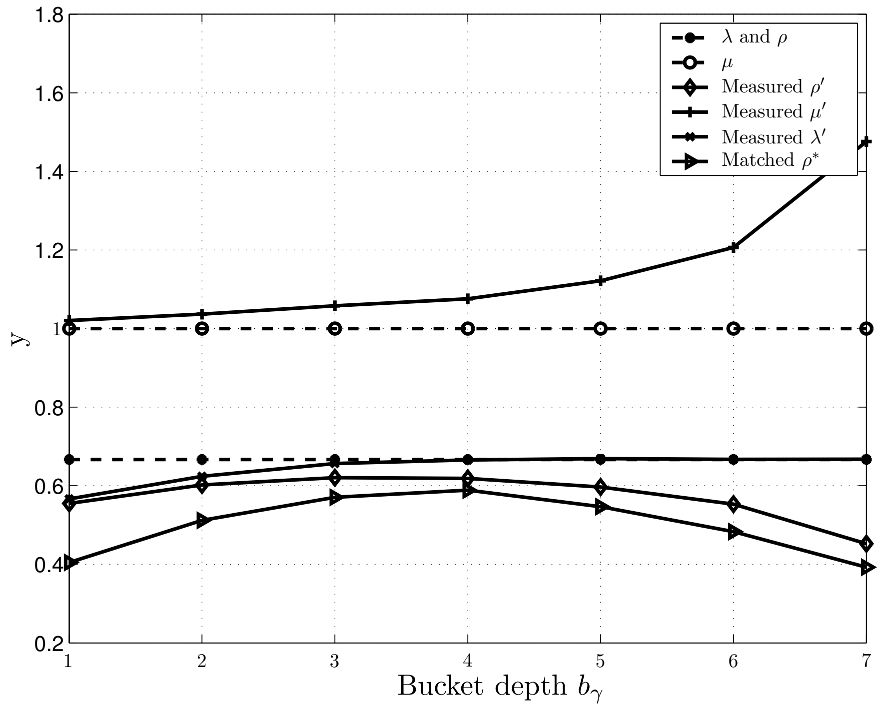

Figure 6.8: Comparison of arrival rates, service rates and utilizations (I)

6.4 Recipe for Application

The results in this chapter provide an alternative or an additional performance analysis to a conventional queuing analysis. The steps to apply them are outlined in Figure 6.13. First the input process has to be determined. If a Poisson process is taken to hold, this only consists of identifying the rate. Then the input process is shaped according to Theorem 2.6. Each packet is assigned a service time, e.g., one which is exponentially distributed. The fourth step is checking whether the service times are in accordance with the underlying service curve, and modifying it if necessary. Finally, the buffer occupancy distribution is computed by simulation.

The difference to the conventional queuing analysis is that steps 2 and 4 are skipped. Then the buffer occupancy can many times be computed analytically, which is less complex than the simulation. Whether the increased complexity of the bounded queuing analysis is worthwhile depends on the application.

6.5 Summary

In this chapter a pioneering effort to bring together network calculus and queuing theory was undertaken. It was shown that the analytical approach is not tractable

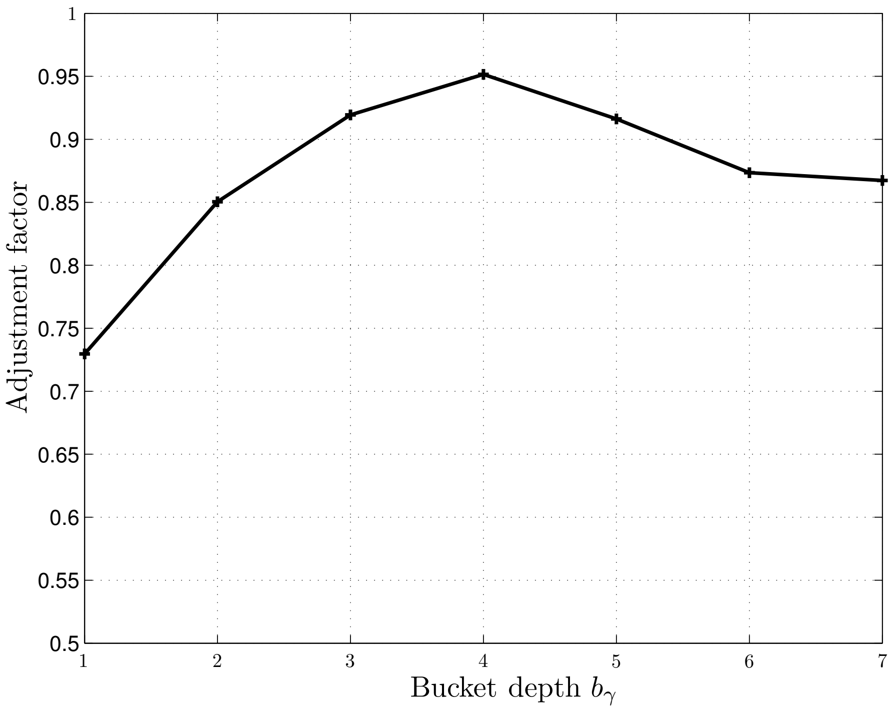

Figure 6.9: Adjustment factors (I)

and therefore not suited for application by practitioners. Simulations of a M/M/1 queue enhanced with a shaper and service curve enforcer were conducted and the relationship of such a queue to a conventional M/M/1 queue pointed out.

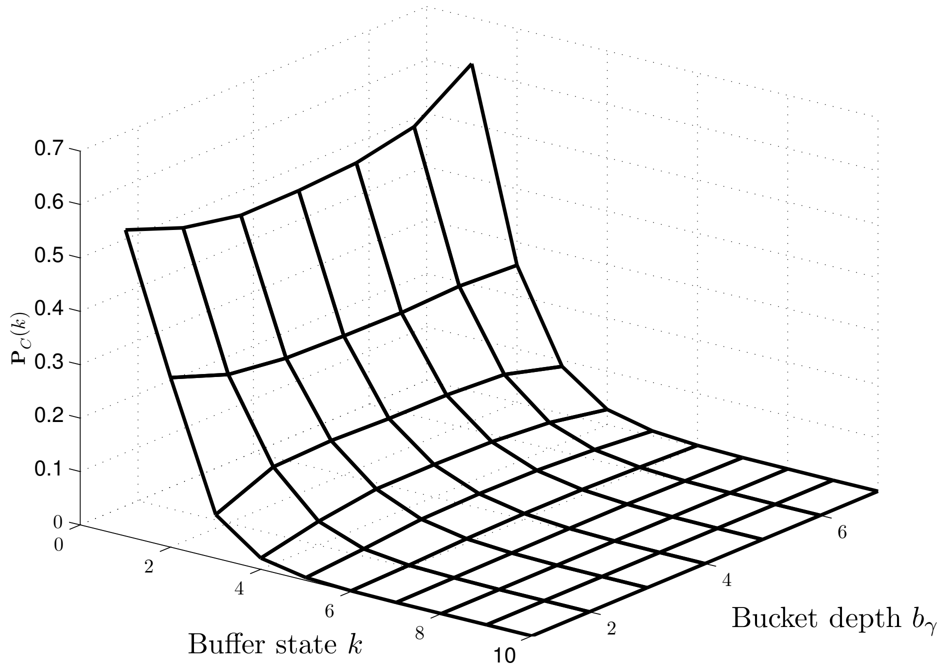

Figure 6.10: Bounded buffer occupancy distribution (II)

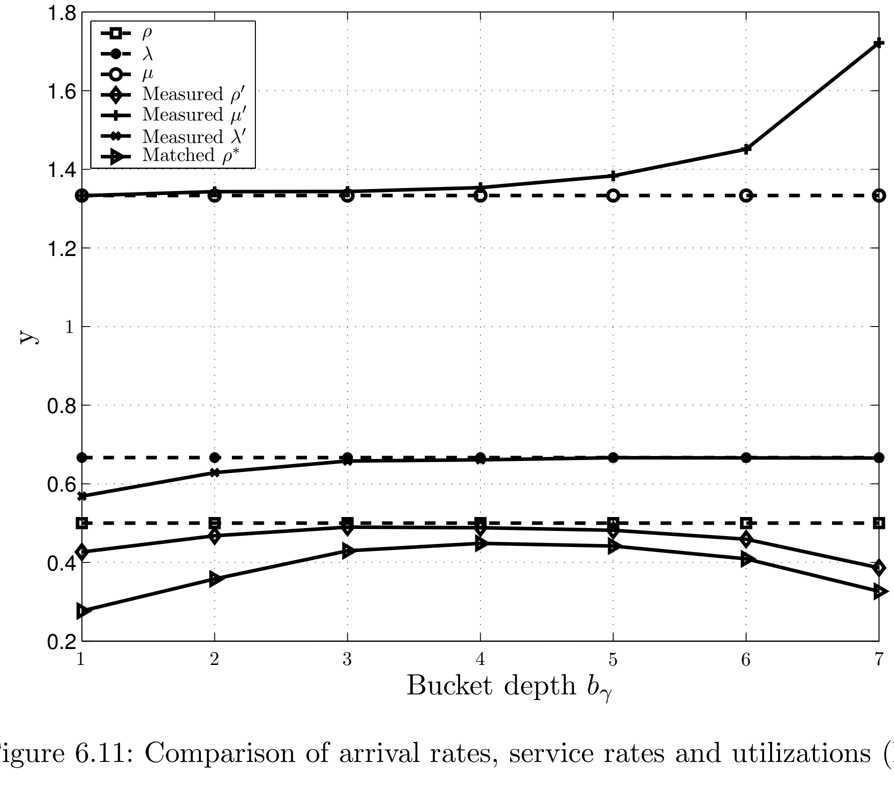

Figure 6.11: Comparison of arrival rates, service rates and utilizations (II)

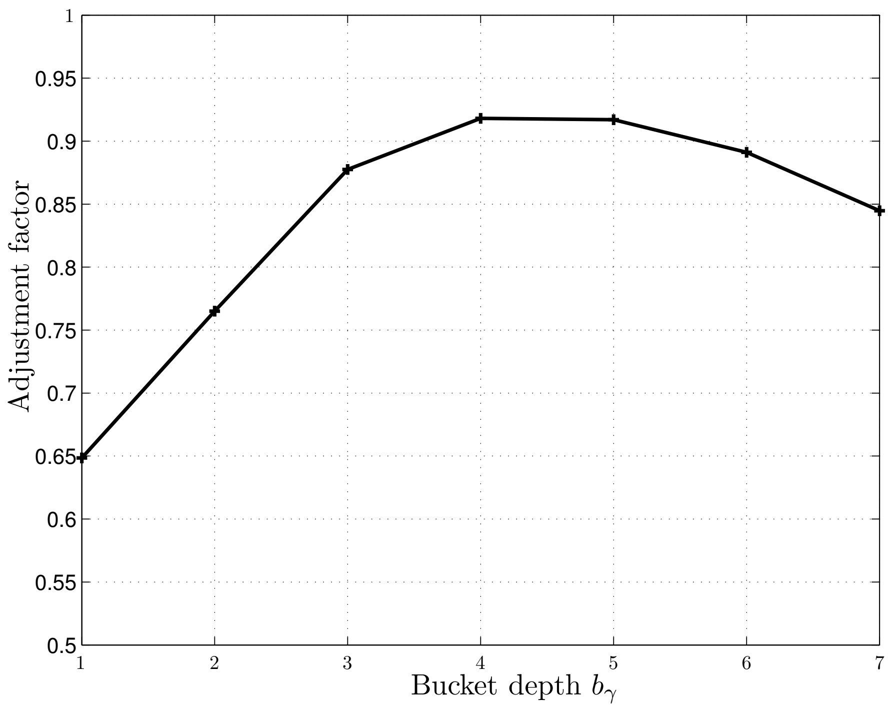

Figure 6.12: Adjustment factors (II)

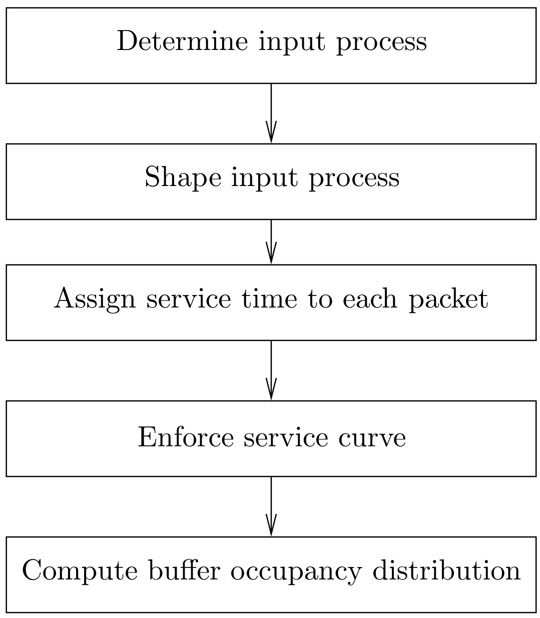

Figure 6.13: Five steps for the bounded queuing analysis

7 Conclusions

In this dissertation elements of a theoretical model for the performance analysis of QoS in networks are presented. The long term vision is to have a model which plays a role similar to the linear, time-invariant system model of the conventional system theory, which is employed in communications or control systems. Ultimately, such a model would support research and development in this field. The basis of this work is network calculus. It is supplemented by queuing theory and linear system theory.

The requirements and building blocks of such a model are presented. Open issues are identified and contributions to resolving them given.

One relevant topic is the application of network calculus for modeling new network paradigms. In Chapter 4 it is shown how a dynamically reconfigurable network can be identified as a network calculus model. This enables its worst case delay analysis and buffer dimensioning to be performed succinctly. Formulating an optimization problem based on the results could serve as a pointer for future work. Beyond that, this modeling technique can be applied to other network architectures.

Another issue is finding a suitable transform for network calculus. With transforms, signals can be represented adequately in order to handle systems with ease. Furthermore, they can reduce the complexity of the computation of the output of the system. A transform for network calculus is derived in Chapter 5. Since it does not directly facilitate the computation of the output, theorems are additionally developed. The insights resulting from the new methods on computing the output are applied to infer optimal node service curves from the optimal network service curve. Furthermore, the implications of deficient nodes are pointed out. The future work for this chapter could be to derive algorithms which apply the results. In addition, such a system could be implemented or simulated.

Queuing theory has been the method of choice in analyzing packet switched networks for many years. In Chapter 6 network calculus and queuing theory are combined to obtain the best of both worlds. This is related to exisiting work on the combination of queuing theory and Petri Nets ("Stochastic Petri Nets"). It is shown how the results of the queuing analysis can be improved by using available additional knowledge. The additional knowledge is brought in by network calculus style traffic shaping and service guarantees. This chapter also contains several pointers for future work. The queuing analysis can be extended from the M/M/1 queue to more sophisticated models as well as queuing networks. In addition, the analytical approach can be pursued and ideally brought together with stochastic network calculus.

On a final note, the field of QoS performance analysis is far from being exhausted. The road to fulfilling the vision of a comprehensive model might have become a bit shorter than before, but, a long way still lies ahead.

Bibliography

[1] Dirk Abendroth and Ulrich Killat. Intelligent Shaping: Well Shaped Throughout the Network? In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), New York, USA*, 2002.

[2] Ralf Ackermann. *Gateways and Components for Supplementary IP Telephony Services in Heterogeneous Environments*. PhD thesis, Technische Universität Darmstadt, July 2003.

[3] Farshid Agharebparast and Victor Leung. Modeling Wireless Link Layer by Network Calculus for Efficient Evaluations of Multimedia Quality of Service. In *Proceeding of IEEE International Conference on Communications (ICC 2005)*, May 2005.

[4] Rajeev Agrawal, Francois Bacelli, and Rajendran Rajan. An algebra for queuing networks with time varying service. Technical Report ECE-98-2, University of Wisconsin - Madison, 1998.

[5] Rajeev Agrawal, Rene Cruz, Clayton Okino, and Rajendran Rajan. Performance Bounds for Flow Control Protocols. *IEEE/ACM Transactions on Networking*, 7(3):310–323, June 1999.

[6] Francois Baccelli, Guy Cohen, Geert-Jan Olsder, and Jean-Pierre Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. John Wiley and Sons, 1992.

[7] Francois Baccelli and Dohy Hong. TCP is Max-Plus Linear - and What it Tells us on its Throughput. In *Proceedings of the ACM SIGCOMM Conference*, pages 219–239, 2000.

[8] Ilia Baldine and George Rouskas. Traffic Adaptive WDM Networks: A Study of Reconfiguration Issues. *IEEE/OSA Journal of Lightwave Technology*, 19(4):433–455, April 2001.

[9] David Black, Stephen Blake, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An Architecture for Differentiated Services. Informational RFC 2475, December 1998.

[10] Stefan Bodamer. *Verfahren zur Relativen Dienstgütedifferenzierung in IP-Netzknoten*. PhD thesis, Universität Stuttgart, 2004. 88. Bericht über vekehrstheoretische Arbeiten (Dissertation).

[11] Robert Boorstyn, Almut Burchard, Joerg Liebeherr, and Chaiwat Oottamakorn. Effective Envelopes: Statistical Bounds on Multiplexed Traffic in Packet Networks. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000), Tel Aviv, Israel*, pages 1223–1232. IEEE, March 2000.

[12] Robert Boorstyn, Almut Burchard, Joerg Liebeherr, and Chaiwat Oottamakorn. Statistical Service Assurances for Traffic Scheduling Algorithm. *IEEE Journal on Selected Areas in Communications. Special Issue on Internet QoS*, 2000.

[13] Jean-Yves Le Boudec and Patrick Thiran. *Network Calculus*. Number 2050 in Lecture Notes in Computer Science. Springer-Verlag, 2001.

[14] Robert Braden, David Clark, and Scott Shenker. Integrated Services in the Internet Architecture: an Overview. Informational RFC 1633, June 1994.

[15] Robert Braden, Lixia Zhang, Steven Berson, Shai Herzog, and Sugih Jamin. Resource Reservation Protocol (RSVP) - Version 1 Functional Specification. Proposed Standard RFC 2205, September 1997.

[16] Lee Breslau and Scott Shenker. Best Effort Versus Reservations: A Simple Comparative Analysis. *ACM SIGCOMM Computer Communication Review*, 28(4):3–16, 1998.

[17] Andrew Brzezinski and Eytan Modiano. Dynamic Reconfiguration and Routing Algorithms for IP-over-WDM Networks with Stochastic Traffic. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, 2005.

[18] Cheng-Shang Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.

[19] Cheng-Shang Chang, Wen-Jyh Chen, and Hsiang-Yi Huang. Birkhoff-von Neumann Input Buffered Crossbar Switches. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, 2000.

[20] Cheng-Shang Chang, Rene Cruz, Jean-Yes Le Boudec, and Patrick Thiran. A Min,+ System Theory for Constrained Traffic Regulation and Dynamic Service Guarantees. *IEEE/ACM Transactions on Networking*, 10(6):805–817, December 2001.

[21] Anna Charny and Jean Yves Le Boudec. Delay Bounds in a Network with Aggregate Scheduling. In *Proceedings of Quality of future Internet Services Workshop (QofIS 2000), Berlin, Germany*, pages 105–116. Springer LNCS, September 2000. ISBN 3-540-41076-7.

[22] Florin Ciucu, Almut Burchard, and Joerg Liebeherr. A Network Service Curve Approach for the Stochastic Analysis of Networks. In *Proceedings of ACM SIGMETRICS Conference*, 2005.

[23] Rene Cruz. A Calculus for Network Delay. I. Network Elements in Isolation. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[24] Rene Cruz. A Calculus for Network Delay. II. Network Analysis. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

[25] Rene Cruz and M. Taneja. An Analysis of Traffic Clipping. In *Proceedings of the 1998 Conference on Information Sciences and Systems, Princeton, NJ*, 1998.

[26] Klaus Dolzer. *Mechanisms for Quality of Service Differentiation in Optical Burst Switched Networks*. PhD thesis, Universität Stuttgart, 2004. 86. Bericht über vekehrstheoretische Arbeiten (Dissertation).

[27] Qing Duan and John N. Daigle. Resource Allocation for Quality of Service Provision in Multistage Buffered Crossbar Switches. *Elsevier Computer Networks Journal*, 46(2):147–168, October 2004.

[28] Markus Fidler. Extending the Network Calculus Pay Bursts Only Once Principle to Aggregate Scheduling. In *Proceedings of the 1st International Workshop on Quality of Service in Multiservice IP Networks (QoS-IP 2003)*, 2003.

[29] Markus Fidler. On the Impacts of Traffic Shaping on End-to-End Delay Bounds in Aggregate Scheduling Networks. In *Proceedings of 4th International Workshop on Quality of Future Internet Services (QoFIS 2003)*, 2003.

[30] Markus Fidler. An End-to-End Probabilistic Network Calculus with Moment Generating Functions. In *Proceedings of the 14th IEEE International Workshop on Quality of Service (IWQoS 2006)*, 2006. To Appear.

[31] Markus Fidler, Oliver Heckmann, and Ralf Steinmetz. Preserving the Independence of Flows in General Topologies using Turn-Prohibition. In *Proceedings of the 13th IEEE International Workshop on Quality of Service (IWQoS 2005), Passau, Germany*, June 2005.

[32] Markus Fidler and Stephan Recker. A Dual Approach to Network Calculus Applying the Legendre Transform. In *Proceedings of the 3rd International Workshop on Quality of Service in Multiservice IP Networks (QoS-IP 2005)*, 2005.

[33] Markus Fidler and Stephan Recker. Transformation-based Network Calculus Applying Convex/Concave Conjugates. In *14. Fachtagung Kommunikation in Verteilten Systemen*, 2005.

[34] Markus Fidler and Stephan Recker. Conjugate Network Calculus: A Dual Approach Applying the Legendre Transform. *Elsevier Computer Networks*, 50(8):1026–1039, June 2006. In press.

[35] Markus Fidler and Jens Schmitt. On the Way to a Distributed Systems Calculus: An End-to-End Network Calculus with Data Scaling. In *Proceedings of ACM SIGMETRICS/Performance 2006 (SIGMETRICS 2006)*, St. Malo, France, June 2006.

[36] Victor Firoiu, Jean-Yves Le Boudec, Don Towsley, and Zhi-Li Zhang. Theories and Models for Internet QoS. *Proceedings of the IEEE - Special Issue on Internet Technology*, August 2002.

[37] Sally Floyd and Eddie Kohler. Internet Research Needs Better Models. In *1st Workshop on Hot Topics in Networks*, October 2002.

[38] International Organization for Standardization. Information Processing System - Open Systems Interconnection. International Standard 8824, December 1987. Obsoleted by ISO/IEC 8824-1:1998.

[39] Manuel Goertz. *Effiziente Echtzeitkommunikationsdienste durch Einbeziehung von Kontexten*. PhD thesis, Technische Universität Darmstadt, July 2005.

[40] Marcel Graf. VBR Video over ATM: Reducing Network Resource Requirements through Endsystem Traffic Shaping. In *Proceedings of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 1997), Kobe, Japan*, pages 48–57. IEEE, 1997.

[41] Marcel Graf. *Traffic Shaping-Verfahren in ATM-Netzen für die Übertragung Komprimierter Videodaten*. PhD thesis, Universität Stuttgart, 2002. 84. Bericht über vekehrstheoretische Arbeiten (Dissertation).

[42] Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford University Press, 2nd edition, 1992.

[43] Xin Guan, Ian Li-Jin Thng, Yuming Jiang, and Hailong Li. Providing Absolute QoS through Virtual Channel Reservation in Optical Burst Switching Networks. *Elsevier Computer Communications Journal*, 28(9):967–986, June 2005.

[44] Simon Haykin. *Neural Networks - A Comprehensive Foundation*. Prentice Hall, 1998.

[45] Oliver Heckmann. *A System-oriented Approach to Efficiency and Quality of Service for Internet Service Providers*. PhD thesis, Technische Universität Darmstadt, December 2004.

[46] Oliver Heckmann, Krishna Pandit, Jens Schmitt, and Ralf Steinmetz. KOM ScenGen - The Swiss Army Knife For Simulation And Emulation Experiments. In *Proceedings of the International Workshop on Multimedia Interactive Protocols and Systems (MIPS 2003)*, pages 91–106, November 2003.

[47] Oliver Heckmann and Jens Schmitt. Best Effort Versus Reservations Revisited. In *Proceedings of the 13th IEEE International Workshop on Quality of Service (IWQoS 2005), Passau, Germany*, June 2005.

[48] Juha Heinanen, Fred Baker, Walter Weiss, and John Wroclawski. Assured Forwarding PHB Group. Proposed Standard RFC 2597, June 1999.

[49] Jean-Baptiste Hiriart-Urruty and Claude Lemarechal. *Fundamentals of Convex Analysis*. Springer-Verlag, 2001.

[50] Paul Hurley, Mourad Kara, Jean-Yves Le Boudec, and Patrick Thiran. ABE: Providing a Low-Delay Service within Best Effort. *IEEE Network Magazine*, 15(3), May 2001.

[51] Van Jacobson, Kathleen Nichols, and Kedar Poduri. An Expedited Forwarding PHB. Proposed Standard RFC 2598, June 1999.

[52] Holger Jaekel. *Ein Beitrag zur Algebraischen Analyse Verteilter und Ereignisdiskreter Systeme*. Logos-Verlag, 2003. ISBN 3-8325-0238-6.

[53] Raj Jain. *The Art of Computer Systems Analysis. Techniques for Experimental Design, Measurement, Simulation and Modeling*. Wiley-Interscience, May 1991.

[54] Yuming Jiang. Analysis of Stochastic Service Guarantees in Communication Networks: A Basic Calculus. Technical report, Centre for Quantifiable Quality of Service in Communication Systems (Q2S), Norwegian University of Science and Technology, 2005.

[55] Yuming Jiang and Peder Emstad. Analysis of Stochastic Service Guarantees in Communication Networks: A Server Model. In *Proceedings of the 13th IEEE International Workshop on Quality of Service (IWQoS 2005), Passau, Germany*, pages 233–245, 2005.

[56] Yuming Jiang and Peder Emstad. Analysis of Stochastic Service Guarantees in Communication Networks: A Traffic Model. In *Proceedings of the 19th International Teletraffic Congress (ITC 2005)*, 2005.

[57] Martin Karsten. *QoS Signalling and Charging in a Multi-service Internet using RSVP*. PhD thesis, Technische Universität Darmstadt, July 2000.

[58] Martin Karsten and Jens Schmitt. Admission Control based on Packet Marking and Feedback Signalling - Mechanisms, Implementation and Experiments. Technical Report TR-KOM-2002-03, Technische Universität Darmstadt, May 2002.

[59] Martin Karsten and Jens Schmitt. Integrated Load Estimation based on Binary Packet Marking. In *Proceedings of 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005), Nice, France*, May 2005.

[60] Martin Karsten, Jens Schmitt, and Ralf Steinmetz. Implementation and Evaluation of the KOM RSVP Engine. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), Anchorage, USA*, pages 1290–1299. IEEE, April 2001. ISBN 0-7803-7016-3.

[61] Hwangnam Kim and Jennifer Hou. Network Calculus Based Simulation of TCP Congestion Control: Theorems, Implementation, and Evaluation. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, 2004.

[62] Leonard Kleinrock. Information Flow in Large Communication Nets. PhD thesis proposal, Massachusetts Institute of Technology, 1961.

[63] Leonard Kleinrock. *Queuing Systems, Volume 1: Theory.* John Wiley and Sons, 1975.

[64] Leonard Kleinrock. *Queuing Systems, Volume 2: Computer Applications.* John Wiley and Sons, 1976.

[65] Edward Knightly and Ness Shroff. Admission Control for Statistical QoS: Theory and Practice. *IEEE Network Magazine*, 13:20–29, March/April 1999.

[66] Anurag Kumar, D. Manjunath, and Joy Kuri. *Communication Networking: An Analytical Approach.* Morgan Kaufmann Publishers, 2004. ISBN 0-12-428751-4.

[67] Karl Küpfmüller. *Systemtheorie der Elektrischen Nachrichtenübertragung.* Hirzel Verlag, 1949.

[68] James F. Kurose and Keith Ross. *Computer Networking: A Top-Down Approach Featuring the Internet.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[69] Averill Law and David Kelton. *Simulation Modeling and Analysis.* McGraw Hill, New York, 1991.

[70] Will Leland, Murad Taqqu, Walter Willinger, and Daniel Wilson. On the Self-Similar Nature of Ethernet Traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.

[71] Chengzhi Li, Almut Burchard, and Joerg Liebeherr. A Network Calculus with Effective Bandwidth. Technical Report CS-2003-20, University of Virginia, Department of Computer Science, 2003.

[72] Joerg Liebeherr, Almut Burchard, and Stephen Patek. A Calculus for End-to-End Statistical Service Guarantees. Technical Report CS-2001-19, University of Virginia, Department of Computer Science, 2001.

[73] Joerg Liebeherr, Stephen D. Patek, and Erhan Yilmaz. Trade-Offs in Designing Networks with End-to-End Statistical QoS Guarantees. In *Proceedings of the 8th IEEE International Workshop on Quality of Service (IWQoS 2000)*, pages 221–230, 2000.

[74] Yong Liu, Chen-Khong Tham, and Yuming Jiang. A Stochastic Network Calculus. Technical report, National University of Singapore, 2004.

[75] Martin Lorang. *Verbindungsorientierte und Verbindungslose Mechanismen eines Skalierbaren Verkehrsmanagements für IP-Netze.* PhD thesis, Universität Stuttgart, 2005. 90. Bericht über vekehrstheoretische Arbeiten (Dissertation).

[76] Robert M. Metcalfe and David R. Boggs. Ethernet: Distributed Packet Switching for Local Computer Networks. *Communications of the ACM*, 19(7):395 – 404, July 1976.

[77] Alan Oppenheim, Alan Willsky, and Syed Nawab. *Signals and Systems.* Prentice Hall, 2nd edition, 1996. ISBN 0-13814-757-4.

[78] Krishna Pandit, Claus Kirchner, Jens Schmitt, and Ralf Steinmetz. Optimization of the Min-plus Convolution Computation under Network Calculus Constraints. Technical Report TR-KOM-2004-04, Technische Universität Darmstadt, June 2004.

[79] Krishna Pandit, Jens Schmitt, Martin Karsten, and Ralf Steinmetz. Bottleneck Estimation for Load Control Gateways. In *Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2003)*, pages 279–286, October 2003. ISBN 0-7695-2039-1.

[80] Krishna Pandit, Jens Schmitt, Claus Kircher, and Ralf Steinmetz. A Transform for Network Calculus and its Application to Multimedia Networking. In *Proceedings of the SPIE Conference on Multimedia Computing and Networking (MMCN 2006)*, January 2006.

[81] Krishna Pandit, Jens Schmitt, Claus Kirchner, and Ralf Steinmetz. Optimal Allocation of Service Curves by Exploiting Properties of the Min-plus Convolution. Technical Report TR-KOM-2004-08, Technische Universiät Darmstadt, November 2004.

[82] Krishna Pandit, Jens Schmitt, and Ralf Steinmetz. Aggregation of Heterogeneous Real-Time Flows with Statistical Guarantees. In *Procceedings of 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2002), San Diego, USA*, pages 57–64. SCS, July 2002. ISBN 1-56555-252-0.

[83] Krishna Pandit, Jens Schmitt, and Ralf Steinmetz. Network Calculus Meets Queueing Theory - A Simulation Based Approach to Bounded Queues. In *Proceedings of the 12th IEEE International Workshop on Quality of Service (IWQoS 2004)*, pages 114–118, June 2004.

[84] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, Inc., 4th edition, 2002.

[85] Abhay K. Parekh and Robert G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, 1993.

[86] Larry L. Peterson and Bruce S. Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[87] William Press, Saul Teukolsky, William Vellerling, and Brian Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.

[88] Thomas Robertazzi. *Computer Networks and Systems: Queueing Theory and Performance Evaluation*. Springer, 3rd edition, 2000.

[89] Utz Roedig. *Firewall-Architekturen für Multimedia-Applikationen*. PhD thesis, Technische Universität Darmstadt, November 2002.

[90] Joseph Rudnick and George Gaspari. *Elements of the Random Walk : An Introduction for Advanced Students and Researchers.* Cambridge University Press, 2004.

[91] Henry Sariowan, Rene Cruz, and George Polyzos. SCED: A Generalized Scheduling Policy for Guaranteed Quality of Service. *IEEE/ACM Transactions on Networking*, 7(5):669–684, October 1999.

[92] Jens Schmitt. *Heterogeneous Network Quality of Service Systems.* Kluwer Academic Publishers, June 2001. ISBN 0-793-7410-X.

[93] Jens Schmitt. On the Allocation of Network Service Curves for Bandwidth/Delay-Decoupled Scheduling Disciplines. In *Proceedings of the IEEE Global Telecommunications Conference 2002 (GLOBECOM 2002), Taipei, Taiwan.* IEEE, November 2002.

[94] Jens Schmitt. On Average and Worst Case Behaviour in Non-Preemptive Priority Queueing. In *Proceedings of the 2003 Internation Symposium on Performance Evaluation of Computer and Telecommunication Systems*, pages 197–204, 2003.

[95] Jens Schmitt, Paul Hurley, Matthias Hollick, and Ralf Steinmetz. Per-flow Guarantees under Class-Based Priority Queueing. In *Proceedings of IEEE Global Telecommunications Conference 2003 (GLOBECOM 2003), San Francisco, CA, USA*, December 2003.

[96] Jens Schmitt, Martin Karsten, and Ralf Steinmetz. On the Aggregation of Guaranteed Service Flows. *Computer Communications (Special Issue on QoS-Sensitive Network Applications and Systems)*, 24(1):2–18, January 2001. ISSN 0140-3664.

[97] Jens Schmitt and Utz Roedig. Sensor Network Calculus - A Framework for Worst Case Analysis. In *Proceedings of IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS 2005), Marina del Rey, USA*, pages 141–154. Springer, LNCS 3560, June 2005. ISBN 3-540-26422-1.

[98] Jens Schmitt and Utz Roedig. Worst Case Dimensioning of Wireless Sensor Networks under Uncertain Topologies. In *Proceedings of 3rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt 2005), Workshop on Resource Allocation in Wireless Networks, Riva del Garda, Italy.* IEEE, April 2005. ISBN 0-9767294-0-7.

[99] Markus Schumacher. *Engineering with Security Pattern.* PhD thesis, Technische Universität Darmstadt, May 2003.

[100] Scott Shenker, Craig Partridge, and Roch Guerin. Specification of Guaranteed Quality of Service. Informational RFC 2202, September 1997.

[101] David Starobinski, Mark Karpovsky, and Lev Zakrevski. Application of Network Calculus to General Topologies Using Turn-prohibition. *IEEE/ACM Transactions on Networking*, 11(3):411–421, June 2003.

[102] David Starobinski and Moshe Sidi. Stochastically Bounded Burstiness for Communication Networks. *IEEE Transactions on Information Theory*, 46(1):206–212, Jan 2000.

[103] Ralf Steinmetz and Klara Nahrstedt. *Multimedia Fundamentals Volume 1*. Prentice Hall, January 2002.

[104] Ralf Steinmetz and Klara Nahrstedt. *Multimedia Applications*. Springer, April 2004.

[105] Ion Stoica and Hui Zhang. Providing Guaranteed Services Without Per Flow Management. *ACM Computer Communication Review*, 29(4):81–94, September 1999. Proceedings of SIGCOMM 1999 Conference.

[106] Ion Stoica, Hui Zhang, and T. S. Eugene Ng. Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Service. *ACM Computer Communication Review*, 27(4), October 1997. Proceedings of the ACM SIGCOMM Conference.

[107] Andrew Tanenbaum. *Computer Networks*. Prentice Hall, 4th edition, August 2002.

[108] Lothar Thiele, Samarjit Chakraborty, Matthias Gries, Alexander Maxiaguine, and Jonas Greutert. Embedded Software in Network Processors - Models and Algorithms. In *Proceedings of the First Workshop on Embedded Software (EMSOFT 2001)*, Lecture Notes in Computer Science, pages 416–434. Springer-Verlag, October 2001.

[109] Jonathan Turner. New Directions in Communications (or Which Way to the Information Age?). *IEEE Communications Magazine*, 24(10):8–15, October 1986.

[110] Milan Vojnovic and Jean-Yves Le Boudec. Bounds for Independent Regulated Inputs Multiplexed in a Service Curve Network Element. *IEEE Transactions on Communications*, 51(5):735–740, May 2003.

[111] John Wroclawski. Specification of the Controlled-Load Network Element Service. Informational RFC 2211, September 1997.

[112] Opher Yaron and Moshe Sidi. Performance and Stability of Communication Networks via Robust Exponential Bounds. *IEEE/ACM Transactions on Networking*, 1(1):372–385, June 1993.

[113] Michael Zink. *Scalable Video-on-Demand Systems*. John Wiley and Sons, October 2005.

A Rebalance Theorem

In the following an interesting result is presented, which was found during the work on Chapter 4 but is off-topic from the core of this thesis. What we can say about the choice of dominating matrix and decomposition, is that it is favourable to have as few diagonal entries $\boldsymbol{\alpha}_{i,i}[k] > 0$ as possible. A diagonal entry $\boldsymbol{\alpha}_{i,i}[k] > 0$ indicates that a topology must be switched that leaves nodes idle. If instead the topologies involving all nodes are used, the overall throughput of the network increases. In the next section we devise a theorem and some remarks regarding diagonal entries, with which performance can be increased and flexibility added.

The following theorem shows it is possible to always obtain a dominating arrival matrix with at most one diagonal entry $\boldsymbol{\alpha}_{i,i}[k] > 0$. As the only relevant parameter of the arrival curves $\boldsymbol{\alpha}_{i,j}[k]$ for stability is the sustained rate $r^s_{i,j}$, we again consider rate limited flows. The term $r^*_{i,j}$ denotes the sustained rate of the dominating arrival matrix.

Theorem A.1 (Rebalance Theorem) *For any $n \times n$ dominating arrival matrix it is possible to rebalance it to a dominating arrival matrix with at most one diagonal entry greater than zero.*

Proof Assume a dominating matrix

$$\underline{\boldsymbol{\alpha}}^*[k] = \begin{pmatrix} r^*_{1,1}k & \dots & r^*_{1,n}k \\ \vdots & \ddots & \vdots \\ r^*_{n,1}k & \dots & r^*_{n,n}k \end{pmatrix} \tag{A.1}$$

Without loss of generality, we assume that

$$r^*_{1,1} \le r^*_{2,2} \le r^*_{3,3} \le ... \le r^*_{n-1,n-1} \le r^*_{n,n}. \tag{A.2}$$

With the following algorithm a dominating matrix, with only one non-zero diagonal entry.

$$\underline{\boldsymbol{\alpha}}^+[k] = \begin{pmatrix} 0 & r^+_{1,2}k & \dots & r^+_{1,n-1}k & r^+_{1,n}k \\ r^+_{2,1}k & 0 & \dots & r^+_{2,n-1}k & r^+_{2,n}k \\ \vdots & \vdots & 0 & \vdots & \vdots \\ r^+_{n-1,1}k & r^+_{n-1,2}k & \dots & 0 & r^+_{n-1,n}k \\ r^+_{n,1}k & r^+_{n,2}k & \dots & r^+_{n,n-1}k & r^+_{n,n}k \end{pmatrix} \tag{A.3}$$

Step 1: Set $r^+_{1,1}$ to 0.
Step 2: Set $r^+_{1,2} = r^*_{1,2} + r^*_{1,1}$ and $r^+_{2,1} = r^*_{2,1} + r^*_{1,1}$
Step 3: Set $r^+_{2,2} = r^*_{2,2} - r^*_{1,1}$. Using Inequality A.2 we find that $r^+_{1,1} \leq r^+_{2,2} \leq r^+_{3,3}$. The above three steps can be iteratively repeated until all $r^+_{1,1} \dots r^+_{n-1,n-1} = 0$. If Ineq. A.2 does not hold, the iteration is done sorting the $r^*_{i,i}$ from the smallest to largest. Since in the original arrival matrix $\underline{\boldsymbol{\alpha}}[k]$ by definition all $r^s_{i,i} = 0$, we have

$$\underline{\boldsymbol{\alpha}}^+[k] \geq \underline{\boldsymbol{\alpha}}[k]. \tag{A.4}$$

■

The proof relies on decreasing a pair of diagonal entries while increasing the entries which affect the rows and columns of both. Systematically finding such pairs yields the following results.

If Ineq. A.2 does not hold, the iteration is done sorting the $r^*_{i,i}$ from the smallest to largest.

The first remark states the condition under which it is possible to set all diagonal entries to 0.

Remark A.2 (Complete rebalance) *It is possible to set all diagonal entries $r^*_{i,i} = 0$, if they can be divided into two groups whose sums are equal.*

In some cases it might be beneficial to have the choice which incomplete topology should be used. The following remark states the condition when it is possible to choose which $r^*_{i,i} = 0$.

Remark A.3 (Choice of incomplete topology) *Choosing which one diagonal entry $r^*_{i,i} > 0$, and therefore which incomplete topology is to be used, is possible if*

$$max_i\{r^*_{i,i}\} \leq \sum_j r^*_{j,j} - max_i\{r^*_{i,i}\}. \tag{A.5}$$

The last remark shows how the fraction of the incomplete topology can be reduced.

Remark A.4 (Reducing the weight) *The overall lowest weight of the incomplete topology can be achieved, by starting the iteration with the two largest $r^*_{i,i}$, and conducting it from then on always with the two largest left.*

List of Tables

4.1 Topologies 55

6.1 Bounded Queue vs. M/M/1 92

List of Figures

2.1 Network . . . 8
2.2 Token bucket . . . 22
2.3 Rate function . . . 23
2.4 Cumulated rate function . . . 23
2.5 Arrival curve with violation . . . 24
2.6 Arrival curve . . . 24
2.7 Latency rate service curve . . . 26
2.8 Backlog and delay bound . . . 28
2.9 Output bound . . . 29

3.1 Relations of model types from Law and Kelton [69] . . . 31
3.2 Sum of transformed service curves in the Legendre domain . . . 38
3.3 Output bound in the Legendre domain . . . 38
3.4 Delay and backlog bound in the Legendre domain . . . 39

4.1 Ring topologies . . . 43
4.2 Topology with one idle node . . . 45
4.3 Layout L_1 . . . 49
4.4 Layout L_2 . . . 49
4.5 Service curve of L_1 . . . 50
4.6 Service curve of L_2 . . . 51
4.7 Service curves of L_1 and L_2 . . . 52
4.8 Service curves of L_1 and L_2 . . . 53
4.9 Buffer requirement for $\boldsymbol{\alpha}_1[k]$ under L_1 and L_2 . . . 54
4.10 Worst case delay bound for different burst sizes . . . 56
4.11 Four steps for the analysis of network calculus systems . . . 57
4.12 Six steps for determining service curves of recofigurable networks . . . 58

5.1 Point cloud operators . . . 62
5.2 Point cloud with redundant area . . . 63
5.3 Continuous $\mathfrak{P}$-transform . . . 64
5.4 Function with one convex inflection point . . . 66
5.5 $\mathfrak{P}$-transform of L2R functions . . . 67
5.6 Min-plus convolution of L2R functions . . . 67
5.7 Proof of Theorem 5.3 . . . 68
5.8 Function with arbitrary convex inflection points . . . 73
5.9 Min-plus convolution of arbitrary functions . . . 73
5.10 Proof of Theorem 5.6 . . . 74

5.11 Optimal network service curve 75
5.12 Local reallocation 78
5.13 Compensated latency with deficient peak rate 80
5.14 Compensated latency with deficient peak rate serving time 81
5.15 Three steps for computing the min-plus convolution 81
5.16 Four steps for admission control 82
5.17 Four steps for service curve allocation 82

6.1 Queuing system 86
6.2 Shaped queue 86
6.3 Methodology of the brute force analysis 88
6.4 Brute force analysis (Equation 6.3) 89
6.5 Simulation methodology 91
6.6 Bounded queue vs. M/M/1 93
6.7 Bounded buffer occupancy distribution (I) 94
6.8 Comparison of arrival rates, service rates and utilizations (I) 95
6.9 Adjustment factors (I) 96
6.10 Bounded buffer occupancy distribution (II) 97
6.11 Comparison of arrival rates, service rates and utilizations (II) 97
6.12 Adjustment factors (II) 98
6.13 Five steps for the bounded queuing analysis 98

List of Acronyms

ABE Alternative Best Effort

ATM Asynchronous Transfer Mode

CDM Code Division Multiplexing

EDF Earliest Deadline First

DiffServ Differentiated Services

FFT Fast Fourier Transform

FDM Frequency Division Multiplexing

FIFO First In First Out

FTP File Transfer Protocol

GPS Generalized Processor Sharing

IETF Internet Engineering Task Force

IntServ Integrated Services

IP Internet Protocol

ISO International Standards Organisation

ISP Internet Service Provider

LIFO Last In First Out

MAC Medium Access Control

MWM Maximum Weight Matching

OSI Open Systems Interconnection

PDB Per Domain Behavior

PHB Per Hop Behavior

POTS Plain Old Telephone System

RSVP Resource ReSerVation Protocol

QoS Quality of Service

SDM Space Division Multiplexing

TDM Time Division Multiplexing

TSpec Traffic Specification

TCP Transmission Control Protocol

VoIP Voice over IP

WDM Wavelength Division Multiplexing

WWW World Wide Web

List of Symbols

$*$	Convolution operator
$\star$	Min-plus convolution operator
$\hat{\star}$	Min-plus deconvolution operator
$\oplus$	Addition in the $\mathfrak{P}$-domain
$\otimes$	Multiplication in the $\mathfrak{P}$-domain
$\inf\{\}$	Infimum operator
$\max\{\}$	Maximum operator
$\min\{\}$	Minimum operator
$\sup\{\}$	Supremum operator
$\boldsymbol{\alpha}(t)$	Arrival curve
$\underline{\boldsymbol{\alpha}}[k]$	Arrival matrix
$\underline{\boldsymbol{\alpha}}^*[k]$	Dominating arrival matrix
$\underline{\boldsymbol{\alpha}}^+[k]$	Rebalanced arrival matrix
b_γ	Burst parameter of token bucket
$b^{\zeta,1}$	Maximum packet size parameter of TSpec
$b^{\zeta,2}$	Burst parameter of TSpec
β	Data served in peak rate
$\boldsymbol{\beta}(t)$	Service curve
$\boldsymbol{\beta_L}$	Service curve of layouts for virtual output queues
$c_{i,j}$	Entry in configuration matrix
$\underline{C}$	Configuration matrix
$\underline{C}_I[k]$	Instantaneous configuration matrix
$\mathfrak{C}\{\mathbf{f}(t)\}$	Fenchel transform of $\mathbf{f}(t)$
$\hat{\gamma}$	Horizontal shift operator
$\boldsymbol{\gamma}(t)$	Token bucket arrival curve
$\boldsymbol{\Gamma}(\tau)$	Legendre transformed token bucken arrival curve
D	Tuning latency
$\mathbf{d}(t)$	Impulse function
$\mathbf{d}[k]$	Impulse sequence
$\mathcal{D}\{\mathbf{f}(t)\}$	Legendre transform of $\mathbf{f}(t)$
δ	Maximum delay
δ_L	Average delay of a packet in a queue
$\hat{\delta}$	Vertical shift operator

ϵ	Zero element in the min-plus dioid
e	One element in the min-plus dioid
f	Frame size
$\mathbf{f}(t)$	General function
$\mathbf{f}[k]$	General sequence
$\mathbf{F}(\tau)$	General transform of $\mathbf{f}(t)$
$\mathbf{f}_c(t)$	Closed convex function
$\mathcal{F}$	Space of wide-sense increasing functions passing through origin
$\mathcal{F}_d$	Space of wide-sense increasing sequences passing through origin
$\mathfrak{F}\{\mathbf{f}(t)\}$	Fourier transform of $\mathbf{f}(t)$
$\boldsymbol{\phi}(t)$	Latency rate service curve
$\boldsymbol{\Phi}(\tau)$	Legendre transformed latency rate service curve
$\mathbf{g}(t), \mathbf{g}[k]$	General function and general sequence
$\mathcal{G}$	Space of wide-sense increasing functions
$\mathcal{G}_d$	Space of wide-sense increasing sequences
$\mathfrak{G}\{\mathbf{f}(t)\}$	$\mathfrak{G}$-transform of $\mathbf{f}(t)$
$\mathbf{h}(t), \mathbf{h}[k]$	Impulse response of a system
i	Index variable
$\underline{I}[k]$	Instantaneous arrival matrix
j	Index variable
k	Index variable
l	Index variable
L	Layout
l_ϕ	Latency parameter in latency rate service curve
$l^{\omega,1}$	Latency parameter of L2R service curve
$l^{\omega,2}$	Inflection point of L2R service curve
l_d	Duration of deficient peak rate
l_t	Deficient duration of peak rate
l_x	Time to compensate deficient duration of peak rate
l_c	Compensated latency
λ	Rate of Poisson arrivals at a queue
λ'	Modified rate of Poisson arrivals at a queue
λ_L	Arrival rate for using Little's law
m	Index variable
μ	Poisson server rate of a queue
μ'	Modified Poisson server rate of a queue

μ^{obs}	Observed server rate
n	Index variable
o	Adjustment factor of matched queue
$\boldsymbol{\omega}(t)$	L2R service curve
$\mathbf{P}(\text{event})$	Probability of an event
$\mathfrak{P}\{\mathbf{f}(t)\}$	$\mathfrak{P}$-transform of $\mathbf{f}(t)$
q	Variable throughput level
Q	Point in the $\mathfrak{P}$-domain
r_γ	Rate parameter of token bucket
$r^{\zeta,1}$	Peak rate parameter of TSpec
$r^{\zeta,2}$	Sustained rate parameter of TSpec
l_ϕ	Rate parameter in latency rate service curve
$r^{\omega,1}$	Peak rate parameter of L2R service curve
$r^{\omega,2}$	Sustained rate parameter of L2R service curve
r^s	Sustained rate
$r_i^{\mathbf{f}}$	The ith slope of a function $\mathbf{f}$
r_d	Deficient peak rate
r^*	Dominating sustained rate
r^+	Rebalanced sustained rate
$\mathbf{r}(t)$	Rate function
$\mathbb{R}$	Set of real numbers
ρ	Utilization of a queue
ρ'	Modified utilization of a queue
ρ^*	Matched utilization
$\mathbf{s}$	Coefficient of point cloud
$\mathbf{S}\{\}$	Mapping operator
σ_A	Maximum of rows and columns
$\boldsymbol{\sigma}(t)$	Greedy shaper
t	Argument of functions
t^ω	Peak rate duration
$T_i^{\mathbf{f}}$	Time of the ith inflection point of a function $\mathbf{f}$
$\hat{T}_i^{\mathbf{f}}$	Time of a convex inflection point
T^{obs}	Duration of observation
τ	Argument of functions
u	Variable in convex combination
$\mathbf{u}(t)$	Step function
$\mathbf{u}[k]$	Step sequence

υ	Maximum backlog
$\boldsymbol{\upsilon}(t)$	Backlog
V	Virtual output queue
w_i^f	Frame with size f of topology i
$\overline{\mathbf{w}}[k]$	Process with Bernoulli arrivals
x	Weight of topologies
$\mathbf{x}(t), \mathbf{x}[k]$	Input of a system
$\overline{\mathbf{x}}[k]$	Input process of a queue
$\underline{\mathbf{X}}[k]$	Occupancy of all queues at time instant k
$\mathbf{X}_{i,j}[k]$	Occupancy of the queue from node i to node j at time instant k
$\mathbf{y}(t), \mathbf{y}[k]$	Output of a system
$\overline{\mathbf{y}}[k]$	Output process of a queue
$\boldsymbol{\psi}(t)$	Output bound
z	Threshold of random walk
$\mathbf{z}(t)$	Function in convex combination
$\mathbb{Z}$	Set of integers
$\mathfrak{Z}\{\mathbf{x}[k]\}$	Z-transform of $\mathbf{x}[k]$

Curriculum Vitae

Personal

Name:	Chennagiri Krishna Subramanya Pandit
Date and Place of Birth:	March 14th, 1978 in Berlin, FR Germany
Nationality:	German

Career

2001 – 2006	Researcher and PhD candidate at Multimedia Communications Lab (KOM), Dept. of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Darmstadt, Germany
1997 – 2001	Student of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Darmstadt, Germany Degree: Diplom-Ingenieur (Dipl.-Ing.)
1988 – 1997	Albert Schweitzer Gymnasium, Kaiserslautern, Germany
1984 – 1988	Grundschule Pestalozzischule, Kaiserslautern, Germany

Professional Activities

Member of the Technical Program Committee of the ACM Multimedia Conference 2005
Reviewer for IEEE Communications Letters
IEEE student member

Publications

Conference Papers

Krishna Pandit, Jens Schmitt, Claus Kirchner and Ralf Steinmetz. A Transform for Network Calculus and its Application to Multimedia Networking. In *Proceedings of the SPIE Conference on Multimedia Computing and Networking (MMCN 2006)*, January 2006.

Krishna Pandit, Jens Schmitt, and Ralf Steinmetz. Network Calculus Meets Queueing Theory - A Simulation Based Approach to Bounded Queues. In *Proceedings of the 12th IEEE International Workshop on Quality of Service (IWQoS 2004), Montreal, Canada*, pages 114–118, June 2004.

Krishna Pandit, Jens Schmitt, Martin Karsten, and Ralf Steinmetz. Bottleneck Estimation for Load Control Gateways. In *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 279–286, October 2003. ISBN 0-7695-2039-1.

Krishna Pandit, Jens Schmitt, and Ralf Steinmetz. Aggregation of Heterogeneous Real-Time Flows with Statistical Guarantees. In *Procceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2002), San Diego, USA*, pages 57–64. SCS, July 2002. ISBN 1-56555-252-0.

Oliver Heckmann, Krishna Pandit, Jens Schmitt, and Ralf Steinmetz. KOM ScenGen - The Swiss Army Knife For Simulation And Emulation Experiments. In *Proceedings of the International Workshop on Multimedia Interactive Protocols and Systems (MIPS 2003)*, pages 91–106, November 2003. Received the best paper award.

Sumeet Sandhu, Krishna Pandit, and Arogyaswami Paulraj. On Non-Linear Space-Time Block Codes. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002), Orlando, USA*, volume 3, pages 2417 – 2420, May 2002.

Technical Reports

Krishna Pandit, Jens Schmitt, Claus Kirchner, and Ralf Steinmetz. Optimal Allocation of Service Curves by Exploiting Properties of the Min-plus Convolution. Technical Report TR-KOM-2004-08, Technische Universiät Darmstadt, November 2004.

Krishna Pandit, Claus Kirchner, Jens Schmitt, and Ralf Steinmetz. Optimization of the Min-plus Convolution Computation under Network Calculus Constraints. Technical Report TR-KOM-2004-04, Technische Universität Darmstadt, June 2004.

Supervised Theses

Julian Eckert. Worst Case Delay Analysis of Switching Schemes in Reconfigurable Networks by using Network Calculus. Studienarbeit, Technische Universität Darmstadt, January 2005.

Ian Hubbertz. Design and Implementation of a Framework for Quality of Service Experiments using a Testbed and Simulator. Studienarbeit, Technische Universität Darmstadt, July 2004.

Claus Kirchner. Optimization of the Min-plus Convolution Computation under Network Calculus Constraints. Studienarbeit, Technische Universität Darmstadt, February 2004.

Matthias Priebe. TSpec Bounded M/M/1 Queue - Design, Analysis and Visualization. Studienarbeit, Technische Universität Darmstadt, August 2003.

Thomas Pfeiffer. Implementierung eines Systems zur Unterstützung von Differentiated Services im Internet. Diplomarbeit, Technische Universität Darmstadt, November 2002.

Julian Eckert. Compression Methods For Transfering Radar Data over the Internet. Diplomarbeit, Technische Universität Darmstadt, February 2006. Co-advised with Dr. Michael Zink.

Matthias Priebe. Analysis and Evaluation of Machine Learning Algorithms for Context-Aware Call Processing in Voice-over-IP Networks. Diplomarbeit, Technische Universität Darmstadt, July 2005. Co-advised with Johannes Schmitt.