Extended Service Level Management for the Provisioning of Streaming Internet Services

DISSERTATION

DER WIRTSCHAFTSWISSENSCHAFTLICHEN FAKULTÄT DER UNIVERSITÄT ZÜRICH

zur Erlangung der Würde eines Doktors der Informatik

vorgelegt von
HELMUT KNEER
aus
Deutschland

genehmigt auf Antrag von

Prof. Dr. Kurt Bauknecht
Prof. Dr. Burkhard Stiller

Januar 2003

Die Wirtschaftswissenschaftliche Fakultät der Universität Zürich, Lehrbereich Informatik, gestattet hierdurch die Drucklegung der vorliegenden Dissertation, ohne damit zu den darin ausgesprochenen Anschauungen Stellung zu nehmen.

Zürich, den 4. Dezember 2002*

Der Lehrbereichsvorsteher: Prof. Dr. Martin Glinz

**Datum der Promotionsfeier*

Acknowledgments

This thesis was generated during my postgraduate studies at the Department of Information Technology (Institut für Informatik, IFI), University of Zurich. I would like to thank all the people who have contributed to the success of this thesis.

Special thanks to my doctoral advisor Prof. Dr. Kurt Bauknecht for his support and encouragement along the process of writing this thesis and for the opportunity, during my four years of IFI membership, to gain scientific experience within SNF research projects.

Special thanks also to my second advisor and referee Prof. Dr. Burkhard Stiller for his helpful comments on this thesis and his motivating cooperation within SNF research projects and publications.

Furthermore, I would like to thank the following people:

- Dr. Urs Zurfluh and Dr. Harald Häuschen for their supervision within CATI and StreamCom, respectively
- My colleagues and ex-colleagues at the IFI, especially Dr. Henrik Stormer and Dr. Thomas Gaugler for constructive discussions
- Thomas Buchheister, Raymond Marfurt, and Christian Matt who helped to sharpen my ideas with their diploma theses
- Joshua Bongard, Raja Dravid, and Chandana Paul for proof-reading parts of this thesis
- Heinz Erny (Erny Consulting), Helmut Augsburger, and Gerd Tichy (SOG Corporation) (all formerly with SOLCON IT-Management AG) for discussions and the provision of hands-on material
- Walter Vogt (Perseo Consult AG) for the provision of the books 'Nutzen ohne Frust' [VFB00] and 'Service Level Management' [NO97]
- The Swiss National Science Foundation (SNF) and the Canton of Zurich for the financial support, and all my project partners within CATI and StreamCom for fruitful discussions

Nicht zuletzt möchte ich meinen Eltern Theresia und Karl Kneer danken, die immer daran geglaubt haben, dass ich diese Dissertation zu einem erfolgreichen Abschluss bringen werde.

Abstract

The global Internet is widely used to trade products, digital contents, and services. A new form of service over the Internet, also referred to as E-Commerce service, is the delivery of real-time multimedia streams on demand. The service of providing streamed information comprises the reliable transmission of single data packets from the E-Commerce Service Provider (ESP) to the end-customer via several intermediate Internet Service Providers (ISP). Reliable service delivery implies the transmission of data with specifications for the Quality of Service (QoS), hence called service provisioning. A big problem for service provisioning on today's best effort Internet is the lack of traffic prioritization mechanisms to guarantee QoS and the fact that many ISPs are involved in the delivery process.

There are approaches to enable QoS-based end-to-end communication on different levels. On the *communication level*, the problem is approached rather technically by implementing new Internet Protocol Architectures. Generic business models are developed on a *business level*, while Service Level Management (SLM) provides a framework on the *contracting level* to specify, negotiate, stipulate, and monitor quality and quantity of service levels between service provider and service customer.

Since all of these approaches do not enable service provisioning of streaming Internet services independent from each other, we have designed and implemented a holistic approach that connects all three levels of abstraction. On the business level, we have developed an *Integrated Business Model* for streaming Internet services with a focus on charging and accounting for the data transport. Our Integrated Business Model includes a role model, a service model, and a business process model, which combines tasks and roles during different phases of the service delivery. On the contracting level, we have enhanced the concepts of the general SLM to form the *Extended Service Level Management*, which translates business needs on the business level into infrastructure requirements on the communication level. Our Extended Service Level Management is based on the specification and stipulation of Service Level Agreements (SLA) and Operational Level Agreements (OLA) among all the involved business entities, and it provides architectures for negotiating, trading, and monitoring these service contracts.

Zusammenfassung

Das globale Internet dient vielfach dem Handel von Produkten, digitalen Inhalten und Services bzw. Dienstleistungen. Eine neue Form von Service im Internet, auch als E-Commerce Service bezeichnet, stellt die Übertragung von Multimedia Streams in Echtzeit dar. Dieser Service umfasst die zuverlässige Übertragung einzelner Datenpakete vom E-Commerce Service Provider (ESP) zum Endkunden über mehrere, zwischenliegende Internet Service Provider (ISP). Zuverlässige Dienstleistungserbringung impliziert eine Datenübertragung unter Spezifikation der Dienstgüte (Quality of Service, QoS). Ein Problem bei der qualitätsbasierten Dienstleistungserbringung sind fehlende Mechanismen zur Gewährleistung der Dienstgüte und die Tatsache, dass mehrere ISPs miteinbezogen werden.

Es gibt Lösungsansätze auf verschiedenen Levels zur qualitätsbasierten Ende-zu-Ende Kommunikation. Auf dem *Communication Level* wird das Problem technisch angegangen, indem neue Internet Protokoll Architekturen entwickelt werden. Generische Geschäftsmodelle werden auf dem *Business Level* erstellt. Und auf dem *Contracting Level* definiert Service Level Management (SLM) ein Rahmenwerk zum Spezifizieren, Verhandeln, Vereinbaren und Überwachen von Qualität und Quantität des Dienstgütegrades zwischen Service Provider und Kunde.

Da all diese Lösungsansätze nicht unabhängig voneinander zur qualitätsbasierten Dienstleistungserbringung führen, haben wir einen holistischen Ansatz erarbeitet, der alle drei Levels kombiniert. Auf dem Business Level haben wir ein *Integriertes Geschäftsmodell* für streaming Internet Services entwickelt mit Fokus auf Verrechnung des Transportdienstes. Unser Integriertes Geschäftsmodell umfasst ein Rollen-, Service- und Geschäftsprozessmodell, das die Aufgaben und Rollen während der verschiedenen Phasen der Dienstleistungserbringung kombiniert. Auf dem Contracting Level haben wir das allgemeine SLM angepasst, um das *Erweiterte Service Level Management* zu bilden, welches Geschäftsbedürfnisse auf Business Level in Infrastrukturanforderungen auf Communication Level übersetzt. Unser Erweitertes Service Level Management basiert auf der Spezifikation und Vereinbarung von Service Level Agreements (SLA) und Operational Level Agreements (OLA) zwischen allen involvierten Partnern, und es beschreibt Architekturen zum Verhandeln, Handeln und Überwachen dieser Serviceverträge.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Problem and Motivation

Over the past few years, the globalization of market structures and information infrastructures has increased significantly in the fast growing development of E-Commerce. E-Commerce no longer only implies business relationships between enterprises (Business-to-Business), but rather covers supply-and-demand relations between enterprises, households (i.e. residential consumers), and public administration [Gau99]. Communication networks, as the underlying information infrastructure, represent the medium to support global business transactions between enterprises, residential consumers, and public administration.

The most commonly used information infrastructure today is the Internet. It is a global network of networks, which interconnects networks or domains of several Internet Service Providers (ISP). Communication within such a network requires a standardized protocol, the Internet Protocol (IP) [Pos81, DH98], that allows the different domains (also referred to as Autonomous Systems (AS)) to exchange information in the form of single but coherent data packets. In order to achieve global connectivity for every single host on the Internet, every domain needs to route the data within its scope (intra-domain routing) and between other neighboring domains (inter-domain routing).

Unlike connection-oriented telephony, where a fixed physical connection is maintained during the whole duration of a phone call, current IP networks are connectionless and packet-switched. Though based on sophisticated routing protocols (e.g. Shortest Path First (SPF)), an Internet connection is variable in bandwidth with equal forwarding preferences for all the data packets (also referred to as best effort Internet). In addition, the Internet provides a ubiquitous infrastructure that is accessible and economically usable by more and more enterprises, residential consumers, and public administration.

Doing business over the Internet reveals different facets of E-Commerce. The trade with *products* is well established by now and ordering books, CDs, or groceries over the vendor's web page is very common and widely practiced. The publication of *digital content* of newspapers and magazines on the Internet is another form of E-Commerce, where publishing companies sell their digital contents similarly to the traditional paper version. Offering *E-Commerce Services* over the Internet is extensive and comprises manifold offers from plain Internet access to costly application services.

The recent development of E-Commerce services on the Internet shows an upcoming demand for *streaming* real-time multimedia applications in the form of IP-telephony, gaming, video-conferencing, video-, audio-, news-on-demand applications, and live broadcasting of sports events. Besides the time and resources that long and expensive downloads entail, streaming technology offers a cheap solution to broadcast widely for instance live sports events onto residential consumers' computer screens. Streaming means that continuous data is cut into single units (packets) and subsequently sent from a sender to a receiver. The sequence of these single packets is called a stream [Ste99]. Streaming technology makes high demands on streaming systems due to high capacity storage needs and especially on the underlying communication infrastructure due to the transportation of large quantities of data. Real-time services in general and streaming Internet services in particular are very sensitive to delays in the arrival of subsequent data packets. In order to characterize the sensitivity of such a service, performance-related service parameters are required to specify the Quality of Service (QoS).

Since the number of Internet users is increasing steadily as well as the supply and demand of E-Commerce services, data traffic is rising more and more as a logical consequence. Traffic congestion occurs if too many data packets are routed through limited network capacities of single network domains. This means that the current best effort Internet, without existing service prioritization mechanisms, does not allow for guarantees concerning the quality of reliable and fast data transport. This results in poor quality for E-Commerce services in general and streaming Internet services in particular, especially when network congestion occurs.

Thus, the Internet lacks technical mechanisms to implement business needs and allow guaranteed delivery of streaming services. Contractual mechanisms are provided by the general Service Level Management (SLM) but must be extended to be applied to the Internet. As a consequence, this thesis aims at providing a framework to enable the delivery of streaming services over the Internet based on the stipulation of service contracts and their compliance.

1.2 Objectives and Scientific Contributions

The objective target of this thesis is to design and implement an Extended Service Level Management framework that allows service providers to deliver QoS-based streaming Internet services by combining customer-oriented business needs and technological infrastructure capacity.

QoS-based end-to-end communication over the Internet is a difficult task and approaches of researchers exist to solve the problem from different perspectives. One approach aims at investigating the communication infrastructure and trying to find possibilities as to facilitate smooth service delivery. Thus, the problem is approached rather technically on a *communication level* (compare to bottom level in Figure 1.1). Since 1995, the *Internet Engineering Task Force* (IETF) has published several Request for Comments (RFC) to specify the next generation Internet Protocol, IPv6 [Par95, DH98, CD98]. The *Asynchronous Transfer Mode* (ATM) evolved as a networking technology for high-speed packet switching in the nineties [Orl00, Gib01, Ste99]. Besides the development of various protocols to support the transmission of real-time data (e.g. Stream Protocol (ST-II), Real-time Transport Protocol (RTP)), the IETF suggested further extensions to existing Internet architectures to come up with new *Internet Protocol Architectures* [BC94, BBC+98, RVC01].

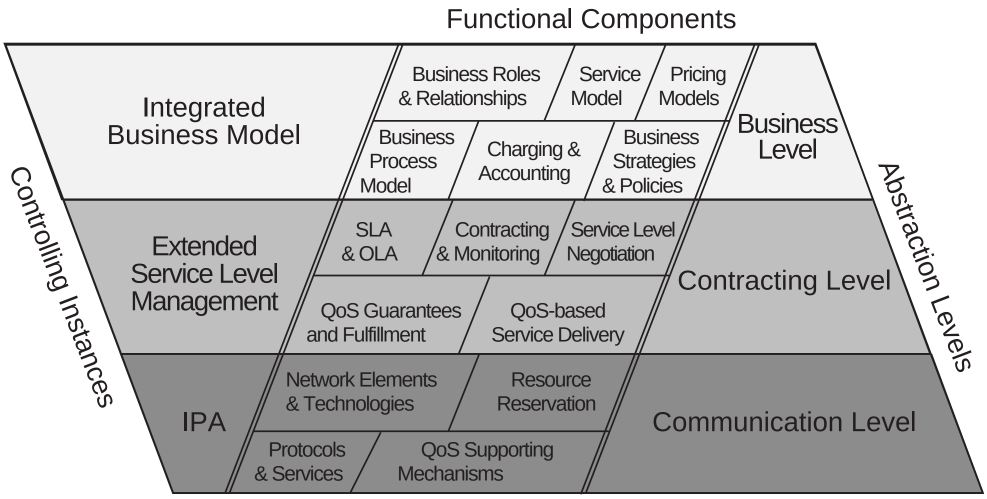

Figure 1.1: The classification model with the controlling instances on different levels of abstraction in conjunction with their functional components.

Another approach to improve QoS-based service delivery over the Internet aims at investigating the Internet users' demands for E-Commerce services that service providers have to meet. Thus, the problem is approached on a *business level* by means of developing business models (compare to top level in Figure 1.1). A business model includes and abstracts knowledge of the functionality of an enterprise: its structures, flows of information and material, products, and external relations [Bai97]. The *Telecommunication Information Networking Architecture Consortium* (TINA-C) has defined a generic business model framework (TINA) for tomorrow's E-Commerce services based on general business roles and their relationships. Other currently existing business models for E-Commerce services on the Internet focus on single-provider services (e.g. application service providing, web hosting), information providing services, or connectivity providing services [And00].

And yet another approach for improving QoS-based service delivery is given by the *Service Management* (SM) as a framework to support, plan, control, and administrate service delivery at customer satisfaction [VFB00, Per00]. A central component within the framework of Service Management is the *Service Level Management* (SLM), which includes the definition, negotiation, monitoring, and management of service contracts between service provider and service customer [NO97]. SLM is based on the existence of service contracts as a means to quantify customer-oriented business needs and maps these onto the existing infrastructure. Thus, the problem of QoS-based service delivery is approached on a *contracting level* by means of SLM or SM, respectively (compare to middle level in Figure 1.1). The *IT Infrastructure Library* (ITIL) is the only widespread and publicly available code of practice for planning, delivering, and supporting IT services[1]. It comprises a layered network of different sets and modules, which describe functions of IT management and their implementation in practice.

However, all these approaches to communication, business, and contracting level do not enable QoS-based service delivery independent from each other. Internet Protocol Architectures lack standards, business models are too generic without a clear focus on a specific E-Commerce scenario, and Service Level Management deals with single business relationships between customer and supplier and is too narrow-gauged to be applied to service delivery on the Internet.

Therefore, to overcome the weaknesses of these single approaches, a holistic approach for the delivery of streaming Internet services is designed and implemented within this thesis, and thus represents its scientific contribution. The holistic approach implies different levels of abstraction, i.e. the business, contracting, and communication level, with a special focus on the *Integrated Business Model* based on a streaming scenario and the *Extended Service Level Management* that

[1]Accessible at URL: http://www.itil.co.uk

considers multiple business relationships to translate business needs into communication infrastructure. Figure 1.1 shows the developed classification model of the holistic approach with the Integrated Business Model, Extended Service Level Management, and Internet Protocol Architectures (IPA) as the controlling instances on the corresponding level of abstraction. Functional components form the building blocks for the controlling instances and are further explained throughout this thesis.

The functional components of the Integrated Business Model are based on a specific application scenario and network model. The components include charging and accounting issues, pricing models, the different occurring business roles and their relationships, the service model with a variety of services that are provided, applied business strategies and policies, and a business process model with four coherent phases and inherent billing and payment schemes.

Functional components of the Extended Service Level Management are based on Service Level Agreements (SLA) and Operational Level Agreements (OLA) as service contracts between service providers and service customers. Furthermore, functional components include service level negotiation, contracting and monitoring to stipulate and check compliance with service contracts, QoS guarantees of service contracts and their fulfillment, and QoS-based service delivery as the essential criterion for a successful Extended Service Level Management.

In order to complete the classification model of Figure 1.1, the functional components of the currently given Internet Protocol Architectures include network elements and technologies, network communication through protocols and services, resource reservation and other QoS supporting mechanisms.

1.3 Structure of Thesis

The thesis is separated into three parts as depicted in Figure 1.2. Part one describes an example of use and introduces QoS. It consists of the following chapter:

- Chapter 2 introduces streaming technology as an example of use for a reliable real-time service. Furthermore, the term ‘QoS’ is defined before certain QoS parameters are specified and classified.

Part two presents and discusses the basics of Internet communication, business models, and Service Level Management. It comprises the following chapters:

- Chapter 3 provides basics in Internet communication and focuses on network technologies, protocols, and services for Internet communication.

- Chapter 4 introduces Internet Protocol Architectures and mechanisms to support QoS-based communication on the Internet.

- Chapter 5 describes business models in general to support business activities on the Internet.
- Chapter 6 introduces Service Level Management as part of the Service Management between customer and provider.

Part three represents the scientific contributions of this thesis, since it develops the proposed solution to enable the delivery of QoS-based streaming Internet services. It includes the following chapters:

- Chapter 7 starts with an Integrated Business Model that we developed for the delivery of streaming Internet services in consideration of charging and accounting aspects.
- Chapter 8 illustrates our designed Extended Service Level Management for managing QoS on the Internet as a refinement of the general Service Level Management as described in chapter 6.
- Chapter 9 describes and discusses our implemented architecture for negotiating and trading service contracts.
- Chapter 10 presents our developed architecture for monitoring compliance with service contracts.
- Chapter 11 provides a practical example to illustrate the negotiation of service contracts and the monitoring of their compliance.

Finally, chapter 12 summarizes the thesis, discusses the scientific contributions, and draws major conclusions. It also gives an outlook on further research topics.

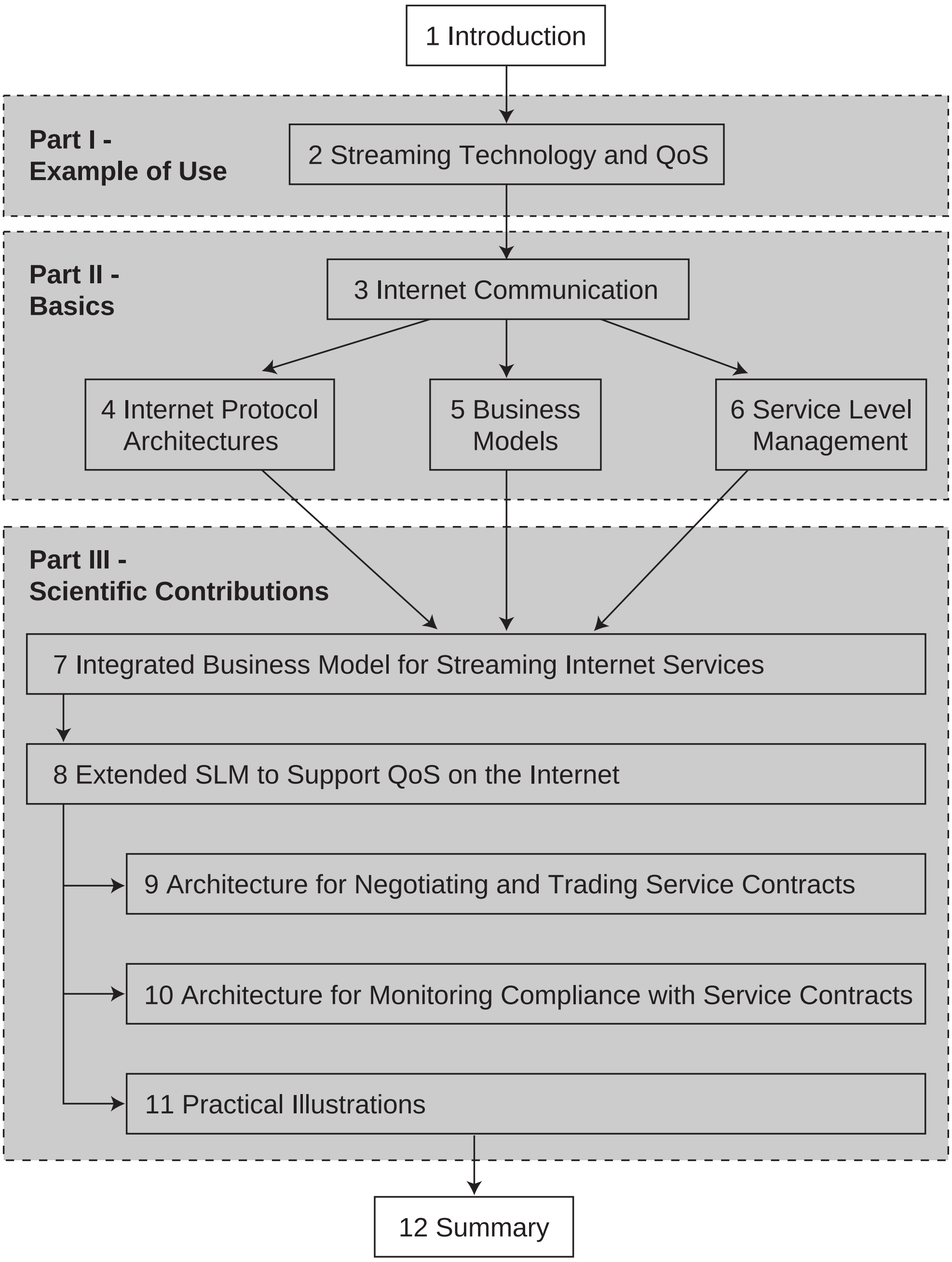

Figure 1.2: The structure of the thesis.

Part I

Example of Use

Chapter 2

Streaming Technology and Quality of Service

Streaming, as one example of a real-time service, is particularly suitable to illustrate requirements of real-time systems and problems associated with QoS-based multimedia services. The chapter starts with an introduction to multimedia and extends, via networked multimedia, to streaming media as one special case of networked multimedia. In addition to the different transmission modes of data streams, the chapter discusses advantages and disadvantages of streaming media and deals briefly with encoding and compression of data streams. Based on these technological basics, the term 'Quality of Service' is defined as well as its meaning in the specific context of QoS-based real-time multimedia systems and services. The chapter closes with the classification and specification of QoS parameters that are important for delivering QoS-based real-time services.

2.1 What is Multimedia?

The term *Multimedia* is hard to define and it is not defined uniformly in literature. If ten experts were asked to give a definition of multimedia the result would be ten different explanations with more or less strict conditions. Literature of the 80s or early 90s have not defined the meaning of multimedia itself but rather use the term in association with *computer animation*, *multimedia-systems*, or *multimedia-databases*.

Analyzing the word multimedia from its origins, we can decompose the word into its two constituents, a prefix 'multi' and a root 'media'. 'Multus' originates from Latin and means 'numerous'. 'Media' is the plural of the Latin word 'medium', which means 'middle' or 'center' and is used in terms of 'mediator' or 'intermediary'. Meyer [Mey75] defines 'medium' generally as a mediative ele-

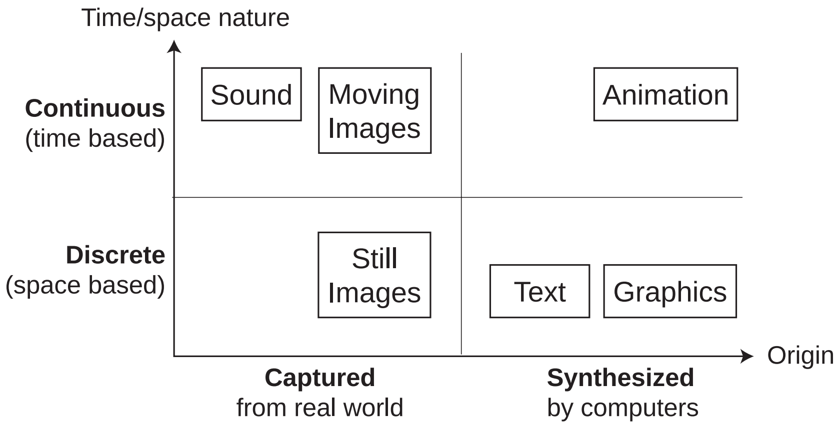

Figure 2.1: Conventional classification of different media types (from [Flu95, page 8]).

ment, especially (if used in the plural) as a means to circulate information by language, gesture, mimic, script, and view (. . .). Applied to the area of information technology, a medium is generally speaking a means or instrument to distribute and represent information. Examples of such means are text, graphics, images, language, sound, and music. Accordingly, water and air represent media for physical or chemical processes.

A further differentiation of the term medium based on various criteria has been performed by [MHE93] where between perception, representation, presentation, storage, transmission, and information exchange medium was distinguished. Media can be classified into *continuous* (time based) and *discrete* (space based) media. Continuous media (sound, moving images, animation) are characterized by a time dependency between subsequent elements of the media as inherent parts of the total information. In contrast, discrete media (still images, text, graphics) are only focused on spatial representation of information where the chronology of its presentation is not important. Depending on the origin, media can further be distinguished whether they are captured from the real world or synthesized by computers. Figure 2.1 shows a conventional classification of media types.

Sloppy definitions consider data already as multimedia data if they include only discrete media (e.g. still images, text). However, a strict definition of multimedia implies a combination of at least one continuous medium (e.g. sound, animation) and a discrete medium that can be used for interactive purposes. Another important issue for classification of multimedia is whether multimedia offers are *offline*, e.g. on a CD-ROM, or available *online* via telecommunication networks. Throughout the discussion about different continuous and discrete media, it is assumed that information exists in form of digital media, not analogous media. Multimedia as used within this thesis is defined as a generic term for all the computer-controlled applications that integrate generation, manipulation, representation, storage and communication of information in form of at least one continuous and one discrete medium.

2.2 Networked Multimedia

Stand-alone multimedia refers to applications that use only local resources to represent the content of multimedia services. The process of generating, manipulating, storing, and representing the multimedia content is performed by a local multimedia-system equipped with memory, CPU, I/O peripherals, and local devices such as CD-ROM, microphones, cameras, etc. The concept of stand-alone multimedia can be deployed very useful in the field of computer-based training where the knowledge in a specific field (e.g. engineering, medicine, biology) is provided on some storage medium (e.g. CD-ROM), which contains information in form of ‘multi-media’. Stand-alone multimedia implies stationary applications that represent digital content without communicating with other computers.

The idea behind *networked multimedia* is that data is stored on one multimedia system and used on another. The communication and the exchange of the data is performed via communication networks (see chapter 3 for details). According to Fluckiger [Flu95], such multimedia systems meet the following four criteria where the fourth criteria is not a mandatory feature for a system to be called multimedia system.

1. Multimedia systems are *computer controlled*.

2. They must *integrate* different kinds of media.

3. The information they handle must be represented *digitally*.

4. The interface to the final user may permit *interactivity*.

Data can be generated, communicated, and represented by multimedia systems in various ways [Neb97]:

- **Offline.** *Offline multimedia* corresponds to the previously explained stand-alone multimedia where multimedia data is stored and represented locally on a computer system or where multimedia data has been stored on a portable storage unit (e.g. CD) and represented exclusively on a local computer.

- **Online.** *Online multimedia* implies the linking of several computer systems via fixed or mobile communication networks. Information has already been generated in advance and is stored on some multimedia system available to be communicated to some remote multimedia system via terrestrial or satellite connections. An example of such an application would be an e-learning scenario where the contents of a virtual class have been made available and accessible for the students of that class.

- **Live.** *Live multimedia* is one form of real-time applications and its data transfer corresponds to a live broadcast on television. The content for live multimedia scenarios can simply not be generated in advance since the content for that multimedia application only exists at the time it is being communicated. Examples of live multimedia include video conferencing or visual telephones. The particularity and difficulty of such live multimedia applications is that they have to meet strict communication requirements in order to be perceived as real-time applications. The maximum time lag for video data from the sender to be displayed at the receiver is said to be 150 ms to be perceived as real-time [Kun98, page 4].

There are various reasons why communication over networks is advantageous or even a precondition for multimedia. It is the purpose of many multimedia applications to offer communication services over geographical distances. Applications of that kind include video conferencing, packet-based audio and video transmission, or multimedia emails. Another issue concerns the timeliness of information that looses its value when out-dated. Networked multimedia guarantees a data exchange based on constantly updated information before being communicated. An example is E-learning as the enhanced and networked form of computer-based training that combines communication, education, information, and training in order to keep peoples knowledge and skills constantly updated. Another aspect that justifies networked multimedia is the implementation of *client/server* models, where resources, such as multimedia storage, are provided on a server that can be accessed remotely from clients. Considering the large amount of required memory for storing multimedia files, the storage facilities of servers and their network file systems can save the clients a lot of money.

2.3 Streaming Media

A special case of networked multimedia is the use of streaming media that have become really popular over the past years. Moreover, the widespread usage of the Internet as the global communication network virtually pushes the idea of distributing digital content via streamed media. The idea behind streaming technologies implies that data is sent from a remote server to be displayed on the client side as soon as it arrives. In order for multimedia systems to communicate (both continuous and discrete) media, the information is first divided into single units (packets) by the sending system (server) and then transmitted to the receiving system (client). The sequence of these single data packets is called a stream [Ste99].

2.3.1 Transmission Modes of Data Streams

According to [Ste99], transmission of data from a sending to a receiving system can be classified into three different transfer modes. The *asynchronous* transfer mode does not include any kind of time restrictions for the transmission of the data packets. Data packets are transmitted as fast as possible. The *synchronous* transfer mode for the transmission of data packets defines a maximum end-to-end delay for every single data packet. Similar to the maximum end-to-end delay of the synchronous transfer mode, the *isochronous* transfer mode defines an additional minimum end-to-end delay in order to delimit delay variations of arriving data packets.

According to the above definitions for the transmission of data it is now possible to characterize different transmission modes for streaming media. Three different kinds of transmission modes of data streams can be distinguished:

1. **Download.** *Downloading* multimedia data from a remote server onto a local client is currently the kind of transmission mode the most often used to access and communicate multimedia data. It corresponds to the previously defined asynchronous transfer mode and it implies that multimedia data (e.g. video files) is accessed and copied as a whole, stored on the local hard disk before it is represented on the local system. This transmission mode requires enough storage capacity on the local system for the downloaded data (e.g. downloading MP3 files).

2. **Progressive Download.** *Progressive download* (or *HTTP streaming* [CC00]) is a refinement of the regular download mode and is similar to the previously defined synchronous transfer mode. Using progressive download for the transmission of data streams, the file is still ‘downloaded’ from a remote server onto the client’s disk but it starts illustrating the content as soon as enough of the stream has arrived. The *time delay* (called *delay offset*) before representation of the stream is useful for the transmission of video data if the required data transmission rates are higher than the actual available network resources. Transmitted data is stored in a temporary storage (called

buffer) to be represented thereafter without interruptions. After the download, the data remains either in the cache memory of the application (e.g. in the browser) or on the local hard disc of the client system.

3. **True Streaming.** In contrast to the previous transmission modes, *true streaming* does not store any data of the stream on the user's disk [CC00]. Like with the isochronous transfer mode where both maximum and minimum end-to-end delays are defined, true streaming may also use a small buffer to smooth out delay variations of single packets. But effectively each packet in the stream is represented as soon as it arrives on the client side. This form of transmission mode is used for live video, video conferencing, or Video-on-Demand (VoD) applications [LL99]. The quality for these applications is limited to the capacity and performance of network resources.

From this point on, terms or expressions that talk about streaming imply the true streaming transmission mode for communication.

2.3.2 Advantages and Disadvantages of Streaming Media

Streaming media is a new and popular form of communicating data but the technology also implies disadvantages that need to be considered. The following discusses advantages and disadvantages of streaming technology:

+ Streaming of multimedia includes central storage of data on a multimedia server at some location available to remote client multimedia systems. Communication networks enable worldwide access to these huge amounts of multimedia data. Changes to out-dated information can easily be made on the server side only, without performing any changes at remote systems. The storage of multimedia data requires big amounts of storage capacity and hardware equipment (e.g. CD drives, hardware decoder) only on the server side which saves remote users money.

+ With streaming media there is no time delay between the reception of information and its local representation. The client multimedia system does not loose time for prior downloads or intermediate storage. Information is available as soon as it arrives.

+ Besides regular TV broadcasts, streaming technology provides means to transmit live events (e.g. an olympic sprint competition, basketball match) onto remote computer screens. No download or any prior activity is possible before the actual live event takes place.

+ The global spread of the Internet and the simple use of multimedia soft- and hardware at relatively modest cost, makes streaming accessible for everyone.

+ One possible and aspiring application for streaming technology is Video-on-Demand, where a video provider company maintains a big database of video files to be streamed via the Internet to paying customers. Since streaming does not store any data on customers' disks, this form of transmission mode is advantageous for the copyright holders of the content of the video files, be it the video provider itself or some other business entity. Copyright protection in this context is currently of big interest for researchers and investigated in [Kne01].

+ Another advantage of streaming media is the fact that it can be interactive. In conjunction with VoD, interaction includes fast-forward or rewind mechanisms, analogous to VCR functionality, or the possibility for viewers to have an influence on the plot or single acts of the movie.

- One major disadvantage of streaming technology is its high demand for communication bandwidth, its dependency on the underlying communication network, and the variations of available networking resources to transmit the single data units from sender to receiver. First of all, digitized multimedia data requires much memory for its storage especially if video data are to be digitized and stored. Then, in order to reduce the high amounts of multimedia data, data gets encoded and compressed before it is communicated over the network. Encoding and compression does reduce the amount of data by a multiple, however, the required network capacity is not necessarily available during the whole transmission period due to network congestion, which makes an uninterrupted presentation of streamed multimedia content impossible for a remote system. Performance of underlying networking infrastructure is a major constituent for the realization of streaming applications and will be studied further in chapters 3 and 4.

2.3.3 Encoding and Compression of Data Streams

As pointed out in the previous section, data streams with multimedia content carry discrete media such as plain text mixed with continuous media such as graphics, audio, or video sequences. Compared to plain text, graphics and especially audio and video are characterized by requiring big amounts of data memory for storage. As a consequence, the processing and communication of these media makes high demands on multimedia systems and the underlying communication network. The transmission of an uncompressed video sequence over the Internet requires so much bandwidth that it would be hard to realize even one single point-to-point stream. Therefore, *encoding* and *compression* algorithms have been developed to enable fast, flexible, and cost-effective solutions to process and distribute multimedia streams.

In order to avoid confusion about the two terms ‘encoding’ and ‘compression’, a few things have to be clarified. Digital encoding comprises the representation of both analog and digital information through a sequence of bits. Usually, encoding schemes try to keep the sequence of bits as short as possible in order to reduce the amount of digital memory to a minimum. Compression, on the other hand, is the process of intentionally reducing the amount of data needed for the storage or transmission of a given piece of information, typically by the use of encoding techniques. The definitions of both terms are different but very often they are mixed up or lead to confusion since both techniques have one primary goal, i.e. data reduction. Throughout this work both terms are used in an equivalent manner provided that they have a common goal.

The primary goal of encoding and compression is to represent digital content (e.g. a video file) with as little data (in bits) as possible without suffering substantial losses in quality. Encoding and compression represents a tightrope walk between reducing memory storage and loosing quality of the content. One can distinguish between *lossy* and *loss-free* encoding schemes where loss-free means that the original data can be restored exactly without any losses. Recovery of lossy encoded data only allows an approximation of the original. The key of data compression is the elimination of redundancy and in case of lossy encoding the elimination of irrelevance. Irrelevance includes information that is imperceptible or just very hard to perceive for humans. Compression rates can be considerably increased by eliminating irrelevance. Further important goals of data encoding and compression comprise according to [LeG91]:

- **Error correction.** The encoding scheme must be robust against occurring errors during transmission over networks, replay, storage, etc. The encoding scheme should allow fast and controlled recovery of these kinds of errors.

- **Scalable encoding and decoding effort.** It should be possible to encode and decode data efficiently with low effort and loss of quality.

- **Diversity of format and resolution.** Different (standardized and other) formats and resolutions should be supported.

- **Cost.** The algorithms should be cost-effective as far as its realizability with hardware is concerned.

Furthermore, encoding schemes of video sequences should include random access to single pictures, fast forwarding/rewinding, audio-visual synchronization and enable backwards replay of single sequences for interactive features [Hei96].

Encoding and Compression of Images

Graphics or still images contain information that can neither be perceived by our eyes nor processed by our brain. Compression algorithms take advantage of these weaknesses in order to eliminate redundancy and irrelevance [SS00, pages 157-199].

Since 1992, *JPEG* is an international standard (defined by the *International Standards Organization* (ISO)) for the compression of still images [ISO93] that was developed by the *Joint Photographics Expert Group*. Through the fast encoding and decoding of single still images, moving images can also be processed which is called *Motion JPEG*. JPEG is a standard for compression of halftone pictures in greyscales or color. It is a hybrid encoding technique that supports several modes for a lossy or loss-free compression. The goal of JPEG is to incorporate a wide palette of different quality levels for the compression specified through parameters. Thus, the user can decide between compression rate and quality of the image. The JPEG standard distinguishes between four different operation modes as described in [Ste99]: *sequential encoding* (lossy), *progressive encoding* (lossy), *loss-free encoding* (reversible), and *hierarchical encoding*.

Practical usage of JPEG compression showed that the standard is suitable for photographs without abrupt transitions in form and color. JPEG is not suitable for unrealistic pictures such as tables or angular computer graphics. The compression rate depends on the complexity of the image and reaches for lossy operation modes compression rates of up to 25:1. In case of loss-free compression, it only reaches rates of about 2:1 [Flu95, page 535].

Encoding and Compression of Audio Data

Audio data is more difficult to compress than still images since humans are much more sensitive to sound. Therefore, compression rates are much lower than the ones for image compression.

Before audio data is compressed it needs to be digitized through quantization of the samples. The *International Telecommunication Union* (ITU) developed a standard (G.711) for uncompressed audio digitization, which defines different quality levels for sampling and quantizing audio signals (e.g. telephone and CD signals). According to G.711, telephone quality is reached if the telephone signal is sampled with 8 kHz with 8 bit code words. This results in a total bit rate of 8000 x 8 = 64 kBit/s for uncompressed telephone data.

The digitization of CD data requires more granular sampling and quantization and also two channels to reach stereo quality. A sampling rate of 44.1 kHz, 16 Bit code words, and two channels result in a total bit rate of 44100 x 16 x 2 = 1.411 MBit/s for uncompressed CD data.

For the compression of audio data with telephone quality, the ITU developed the G.721 standard, which allows a compression rate of 2:1 without audible losses in quality. An increase of the compression rate to 4:1 is given by standard G.728, which allows a bit rate of only 16 kBit/s. Further compression algorithms have been developed but suffer of substantial losses in quality.

For the compression of audio data with CD quality, a working group called *Moving Pictures Expert Group* (MPEG) was formed with the goal to develop standards for recording and transmission of digital audio-visual information. The result was a MPEG family with different single standards (MPEG-1, MPEG-2, MPEG-4, and MPEG-7) for various applications. Part of the MPEG-1 standard is *MPEG-Audio* that is used for compression of audio data. MPEG-Audio consists of three layers of lossy encoding and compression schemes that achieve compression rates of up to 12:1 [Flu95, page 525].

Encoding and Compression of Video Data

Video data is multidimensional. Besides local extension in two dimensions, video data comprises additional time, color, and luminance (lightness and brightness) information. The digitized information is very large and requires lots of memory storage. That is why compression of video data is very important for the success of streamed video data [SS00, pages 323-467].

The spatial extension of a video picture is defined by two dimensions, its height and width. A picture consists of a matrix of pixel values, where a pixel carries the necessary information to represent a picture value. One way to encode pixel values is through an additive mixture of the colors Red, Green, and Blue (RGB format). Since the human visual system is more sensitive to luminance than to color, the color signal may be transmitted or represented with less accuracy. Therefore, the RGB signal is transformed into another set of signals, i.e. one luminance signal Y, and two chrominance signals U and V (defined by the PAL standard, the NTSC standard refers to the luminance as Y and the two chrominance signals as the I and Q components) [Flu95, Kun98, Ste99].

A video sequence consists of several single pictures, the so-called *frames*. The faster the subsequent frames are played successively, the less it is possible for the human visual system to recognize the single pictures. Experiences in movie and TV technology have shown that this is the case with a frequency of 16 Hz. 25 frames per second can even be visualized without flickering [Kun98]. The refresh rate for videos is called *frame rate* and specified in Hertz (Hz) or frames per second (fps). There are different international standardized formats for representing digital video. Given the European PAL norm for TV broadcasting (352 x 288 pixels, 25 fps) an uncompressed digitized video with a 24 bit color encoding scheme would require a bandwidth of 7.6 MByte/s. According to this, a 90 minute movie would require 41 Gigabyte of memory storage. This simple example shows how difficult transmission of uncompressed video streams would be nowadays. The huge amount of data storage and the high bandwidth make high demands on multimedia systems and networks which makes compression even more valuable and necessary.

Compression of video sequences considers two components for reducing or eliminating redundancy or irrelevance. It can be distinguished between *spatial correlation* of single frames and *temporal correlation* of subsequent frames.

- Spatial correlation refers to single frames of a video sequence. The redundancy or irrelevance within a frame is reduced or eliminated. Used techniques are identical to the ones used for JPEG compression.

- Temporal correlation refers to differences in subsequent video frames. Existing redundancy is eliminated by encoding only the differences to the previous or subsequent frame.

Different compression techniques can be distinguished whether they focus on eliminating only spatial or temporal correlation or both. One example of a video compression algorithm focusing on spatial correlation only is *Motion-JPEG* (M-JPEG) where every single frame is compressed independently with JPEG compression techniques. The advantage of M-JPEG is that compression is fast and the loss of single video frames does not influence previous or subsequent frames. The previously introduced MPEG standard is an example of a compression technique that eliminates both spatial and temporal redundancy and irrelevance. The compression rate is much higher but single frames are inter-correlated and thus more sensitive to transmission errors. More details to the MPEG standard, see [LeG91, Ste99, Flu95, Kun98, SS00].

The H.320 norm defines another standard specified by the ITU for visual telephony and video conferencing over digital telephone lines. Part of the H.320 family is the standard H.261, which defines a standard for encoding and compression of moving images (e.g. video sequences) [Say00, pages 529-560]. Work and research on video compression algorithms and techniques will further advance in the future but the success of these methods strongly depends on the industries who have to implement hard- and software for different streaming applications.

2.4 Quality of Service

Two basic concepts for the operation of multimedia systems are *Quality of Service (QoS)* and *Resource Management*. In order for multimedia systems to generate, manipulate, store, represent, and communicate continuous multimedia data, its processing and communication must meet certain requirements in order to satisfy the users of multimedia applications. Performance-related service guarantees correspond to the Quality of Service, which implies that necessary operating resources (e.g. system memory, communication bandwidth, processor power) need to be managed by the resource management. The following sections give a definition of Quality of Service with further focusing on QoS-based operation of multimedia systems and their parameters. Resource management concepts are a logical consequence of strict compliance with QoS and its service parameters and will be further discussed in section 3.1.

2.4.1 Definition of Quality of Service

The term *Quality of Service* in connection with multimedia refers to the supply of a multimedia service offered by a service provider to a service customer. A service is a beneficial value for a service customer that is financially compensated towards the service provider. The *International Telecommunication Union - Telecommunication Standardization Sector* (ITU-T) defines a service generally as *"a set of functions offered to a user by an organization"* [ITU94b]. In the framework of streaming multimedia, a service implies that a service provider makes multimedia data available to requesting service customers by help of a communication infrastructure and with financial compensation. The advantages of streaming multimedia had been discussed in section 2.3.2 but one reason for its recently gaining attractiveness is the offer of various applications such as live streaming, Video-on-Demand, etc. However, in order for these streaming multimedia applications to become accepted, all the involved resources in the streaming process have to meet certain requirements. One of the most important requirement is the operation in real-time, which will be discussed in the next section.

The market for multimedia services and their performance strongly depend on the system requirements of service providers and customers, but mostly on the network resources of the communication infrastructure, i.e. the Internet in a general streaming scenario. Due to the heterogeneity of multimedia system and communication networks and the strict service requirements for streaming services, it is absolutely indispensable to parametrize multimedia systems and their service provisioning. The parameters allow a flexible description of a service and a classification into different service classes. Thus, the total of service parameters specify the *Quality of Service*.

According to the ITU-T recommendation [ITU94b], the generic definition of Quality of Service is *“the collective effect of service performance which determine the degree of satisfaction of a user of the service”*. This recommendation defines QoS as a concept how well the service is performed and it implies the evaluation of the service by the user’s perception. It leaves out quantitative and measurable ratings of the performed service. The ITU-T Recommendation X.641, or the ISO/IEC International Standard 13236 respectively [ITU97], defined later in 1997 Quality of Service as *“a set of qualities related to the collective behavior of one or more objects”*, where a managed object is *“a view of a resource within the OSI environment that may be managed through the use of OSI management protocols”*. Similarly, this definition only covers qualities of a service without specifying quantitative measures to classify service performance. Steinmetz[1] characterizes QoS as *“. . . the defined and controllable behavior of a system regarding quantitatively measurable parameters”* [Ste99]. According to Steinmetz, the following definition of Quality of Service will be valid throughout the thesis:

Definition 2.1 (Quality of Service) *Quality of Service (QoS) is defined by a set of parameters that specify the performance of a service in both qualitative and measurable manner.*

2.4.2 Requirements of Real-time Systems and Services

As pointed out in the previous section, the real-time feature of streaming multimedia is quite important for the operation and the acceptance of streaming multimedia services. Real-time systems in general are characterized by severe temporal restrictions of their performance. The DIN norm 44300 [DIN85] defines real-time operation of a system as *‘a mode where system programs are always ready to process data such that processing results are available within a given period of time’*. A real-time system has to fulfill the service requests within the given time limits, hence, a real-time system offering real-time services has to meet certain guaranteed response times. But it is not only speed and efficiency of the responses that specify a real-time system, it is also the fine tuning between different service responses. Regarding streaming multimedia, neither representing single video frames too fast nor too slow is really advantageous for the quality of the video sequence. For the correct replay of a high-quality video stream, a real-time service has to consider both synchronization between single frames and between audio and video data.

[1] Citation is translated from German to English.

A deterministic and predictable behavior of a real-time system requires guarantees for time-critical service performance. The following requirements of a real-time system have to be met among others (see [Ste99, pages 233-234]):

- A fast and predictable processing of time-critical events and service requests which implies short interrupt delays and quick changes of context
- A high degree of schedulability of the resources to determine temporal guarantees for services
- Stability of the system to avoid system and service interruptions in case of overloads
- Ability to run several tasks in parallel (multi-tasking) in order to answer several service requests
- Appropriate mechanisms for inter-process communication (between services) and synchronization (within a service)

Real-time systems are characterized by their correct operation within strict time limits. Their usage is predestinated for important tasks where human life could be dependent on. Typical areas of application include the anti-skid system in vehicles, autopilot systems in aircrafts, or military environments. Recently developed considerations to support medical doctors for remote surgeries emphasize the importance of precisely operating real-time multimedia systems and services.

2.4.3 Characteristics of QoS-based Real-time Multimedia Systems and Services

Streaming audio or video data consists of single, periodically sequent data consisting of samples or frames. The representation of these data units needs to follow certain temporal and synchronistic restrictions to keep the stream as smooth and fluent as possible. However, the streamed information is generated and stored on a sending system, represented on a receiving system, and communicated over a networking system. And if the streaming service wants to guarantee a certain QoS, it requires not only single systems to follow the temporal and synchronistic restrictions but all the involved systems. Thus, the given restrictions need to be followed *end-to-end*.

In order to provide a QoS-based end-to-end streaming service, different requirements have to be met:

- As in traditional real-time systems, multimedia systems depend primarily on *availability*, *temporal reliability*, and *correctness* of the transmitted information. Inaccessible or lost data in a video stream just causes anger with the users but erroneous or delayed data to support a remote medical surgery can harm human life.

- The transmission of time-sensitive audio or video streams is specified by *time limits* and *intervals*. This includes the definition of end-to-end jitter[2], synchronization levels (e.g. between audio and video channel), and a maximum end-to-end delay of language processing applications. While the end-to-end delay and the resulting time offset does not really matter for streaming video applications, it is jitter that mainly influences the quality of the video stream whereas live audio lives on short end-to-end delays.

- A streaming multimedia service requires a *high data throughput* since multimedia data is usually large although compressed in most of the cases. A streaming service requires a high throughput of both processing systems and communication networks.

- In order to avoid *scaling problems*, multimedia systems and their compression algorithms need to perform dynamically and adjust to changing environments. For example, by changing encoding parameters the quality of the compressed information stream needs to be adjusted dynamically to the available bandwidth (*scalable video*).

- The demand for quality and resources of streaming services results in a *cost-based separation of service classes*. Multimedia systems and the services they provide need to hold mechanisms to reserve system and network resources and to make service guarantees according to the price that is paid for. Furthermore, they need to offer fair mechanisms, which allow distinction between time-critical and time-uncritical services.

2.4.4 Specification of QoS Parameters

The specification of services with the aid of quantifiable parameters helps to define and classify the service levels and the QoS. QoS parameters can be represented by a vector of scalar values that is negotiated between the service providing and the utilizing system. However, as emphasized in the previous section, QoS also depends on an intermediate network system that is responsible for the communication of information. Therefore, QoS parameters can be categorized into *system*

[2] Jitter defines the variation of end-to-end delays of single, transmitted data packets.

parameters for sending and receiving systems and *network parameters* for the transmitting communication infrastructure.

The ITU-T recommendation E.800 [ITU94b] provided a QoS framework with performance concepts and influencing factors which contribute collectively to the overall QoS as perceived by the user of a service. However, these performance concepts represent a complicated netting and are lacking quantitative indications to measure QoS. The ITU-T recommendation/ISO standard X.641 [ITU97] has elaborated on the relationships between QoS concepts and developed a complex listing of various QoS characteristics as quantifiable criteria of QoS.

Steinmetz defines QoS parameters in a layered QoS model corresponding to networked multimedia systems [Ste99, pages 237-245]. The model consists of three layers: *application*, *system*, and *devices*. In case of having a human user, the model may also have a *user QoS layer*. These different layers can have sublayers associated with further QoS specifications.

EURESCOM[3] distinguishes QoS parameters within a QoS framework between *primary* and *derived* parameters [Eur99]. *"A primary QoS parameter is determined on the basis of direct observations of 'events' at interaction points"* whereas *"a derived QoS parameter can be determined as a function of other, previously defined QoS parameters"*. For the classification of the primary QoS parameters, they use a 3x3 matrix approach with *access*, *information transfer*, and *disengagement* as three generic functions applied onto three criteria for characterizing *speed*, *accuracy*, and *dependability*.

In order to simplify and structure the diverse classification of different QoS parameters in the literature, our graph in Figure 2.2 helps to characterize QoS parameters.

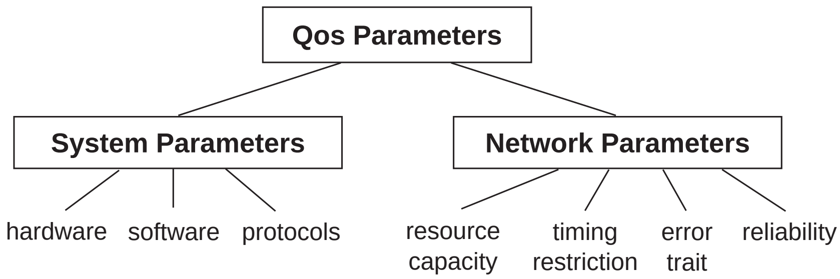

Figure 2.2: Classification of QoS parameters into system and network parameters.

QoS parameters can be classified into *system parameters* for sending (providing) and receiving (utilizing) systems and *network parameters* for the transmitting

[3]The European Institute for Research and Strategic Studies in Telecommunications; accessible at URL: http://www.eurescom.de

communication infrastructure. The system parameters can be further classified into *hardware*, *software*, and *protocols* as the main classes for QoS system parameters. Table 2.1 shows different QoS criteria referring to all three classes of system parameters. They are not further studied since the main focus of QoS parameters is on the network parameters.

hardware	processor power memory operations (on RAM or hard disk) in-/output devices (e.g. cameras, microphones) internal bus system performance
software	operating systems databases application programs (e.g. encoding and compression, grouping of frames and packets) security operations (e.g. encryption)
protocols	stream synchronization correctly ordered packet delivery error recognition/correction packet scheduling

Table 2.1: A selection of QoS criteria concerning hardware, software, and protocols as the three classes of QoS system parameters.

Network parameters can be further classified into *resource capacity*, *timing restriction*, *error trait*, and *reliability* as the four main classes to specify QoS network parameters. Resource capacity specifies the (maximum, minimum, average) *bandwidth* of a digital network link in bits per second (bps) as the number of bits per second that can be transmitted on that link. However, bandwidth specifies a theoretical measure of resource capacities whereas the practical value for the measure very often strongly varies. In order to distinguish theoretical measures from practical values, *throughput* is very often used in literature to emphasize the actual measured performance of the network.

Timing restriction comprises (maximum, minimum, average) *end-to-end delay* (in literature also sometimes called *latency*[4], which corresponds to how long it takes a single bit to propagate from one end of a network to the other. Delay is measured strictly in terms of time (e.g. milliseconds). It is important to know

[4]According to [PD96], latency can be specified by the speed of light propagation delay, the transmission time for a data unit, and the queueing delays inside a network: $Latency = Propagation + Transmit + Queue$, where $Propagation = Distance/SpeedOfLight$ and $Transmit = DataSize/Bandwidth$.

that there are situations in which it is more interesting to know how long it takes to send a bit from one end of a network to the other and back, rather than the one-way delay. This is called the *round-trip time (RTT)* of a network. Related to that is a measure that is often referred to as the *response time* of a system, which measures the elapsed time between the service request and the corresponding response. The response time specifies the behavior of interactive services (e.g. web requests) where the QoS depends on both system and network parameters. Another QoS network parameter in this category is *jitter*, which measures delay variations (also in terms of time, e.g. ms) and which strongly influences the quality of a streamed video as explained earlier in this chapter.

The class error trait contains parameters specifying the faultiness of the network by defining the *packet loss rate* and the *error rate*. Both parameters are usually specified by a percentage throughout a whole service while packet loss rate measures packets that got lost (or arrived too late) and error rate measures the number of bits that were erroneous during a transmission.

Reliability comprises the specification of the network *availability* as a percentage of a day or week. Integrity of transmitted data is specified and measured within the class error characteristics and other security preferences (e.g. confidentiality) are covered by software and end-to-end protocols of sending and receiving systems. Availability denotes the time the network is accessible for transmission services. Table 2.2 summarizes the introduction of the QoS network parameters with the measuring units.

resource capacity	(max., min., avg.) bandwidth (in bits/s)
timing restriction	(max., min., avg.) end-to-end delay (in ms) jitter (in ms)
error trait	packet loss rate (in % of a service) error rate (in % of a service)
reliability	availability (in % of a day/week)

Table 2.2: QoS criteria concerning resource capacity, timing restrictions, error trait, and reliability as the three classes of QoS network parameters.

2.5 Summary

Within the first part, this chapter described streaming technology as an example of use of a real-time service that can be provided over the Internet. Streaming is particularly suitable to show the meaning of Quality of Service (QoS) and illustrate requirements of QoS-based real-time systems. The next part provides basics for the communication of data over the Internet with the help of business models, Internet Protocol Architectures, and Service Level Management.

Part II

Basics

Chapter 3

Internet Communication

Communication over networks allows humans and computers to exchange data, and thus provides the technological basis for any form of electronic business. The chapter starts with an introduction to resource management and then describes general basics in network communication focusing on network technologies, protocols, and routing mechanisms on the Internet.

3.1 Resource Management

The idea behind networked multimedia in general and streaming applications in particular is that content of multimedia data is stored on one multimedia system and viewed or represented on another. The content is transmitted via communication networks from the sending system to the receiving system. That means that there are basically three principal resource components that are required to establish the multimedia communication and which are responsible for the provided QoS of the multimedia service [Bal96]. These include the *multimedia server*, the *multimedia client*, and some *communication system*.

3.1.1 Resource Assets

Resource assets for multimedia systems (both servers and clients) can be subdivided into hardware and software components (compare to section 2.4.4). Hardware assets on the server side consist of high-performance processors, plenty of RAM and hard disc memory, and a fast internal bus system. Hardware assets can further consist of external multimedia equipment such as CD players, digital video cameras, or microphones. Software assets for a multimedia server comprise large databases as the data repositories, as well as reliably running operating systems and application programs such as encoding algorithms or streaming software.

The multimedia client on the customer side represents an interface to a human user and must be equipped with input and output facilities for any kind of interaction with the server, as well as processing and representation capabilities in soft- and hardware. The performance of the client system is one factor among others that influences the overall QoS of the streaming service.

Resource assets for communication systems consist of hardware devices that interconnect communication networks. These devices include repeaters, bridges, routers, gateways, hubs, switches, and, of course, the cable itself. These network assets are of primary interest for this work since they play the major role for the overall QoS of streaming services. Network assets enable a global communication and usage of multimedia content and they implement strategies and policies that decide about how fast data packets of multimedia services are communicated over the networks.

3.1.2 Resource Sharing

A communication network provides connectivity among a set of computers. Sometimes it is enough to limit the connectivity to only a few selected computers (e.g. internal corporate network) while in most of the cases, communication networks are designed to connect to basically any computer in the world. The physical medium that interconnects computers, such as coaxial cable or optical fiber, is called a *link* whereas the connected computers and the previously mentioned hardware devices (e.g. routers) are often referred to as *nodes*. Network nodes are sometimes connected by a direct link, also called a *point-to-point link*, while in other cases, more than two nodes may share a single physical link (often referred to as *multiple-access link*). The question to be answered is how do many nodes share one and the same resource if used for several parallel (multimedia) applications.

The fundamental idea behind network resource sharing is the concept of *multiplexing*, which means that network resources are shared among multiple users. As illustrated in Figure 3.1, three pairs of hosts try to communicate over one physical link where Switch 1 multiplexes the data onto that link and Switch 2 *demultiplexes* back into the corresponding *connections* or *flows*.

There are several different methods for multiplexing multiple flows onto one physical link. One method, which is commonly used for telephone networks, is called *Synchronous Time-Division Multiplexing* (STDM). The idea is to split the time into equal-sized quanta and give equally each pair of communicating hosts the chance to transmit data during that period of time. Another method is the *Frequency-Division Multiplexing* (FDM), which allows several users to transmit data over a link at a different frequency, just as different TV channels are broadcasted over one physical cable. However, these two methods for resource sharing

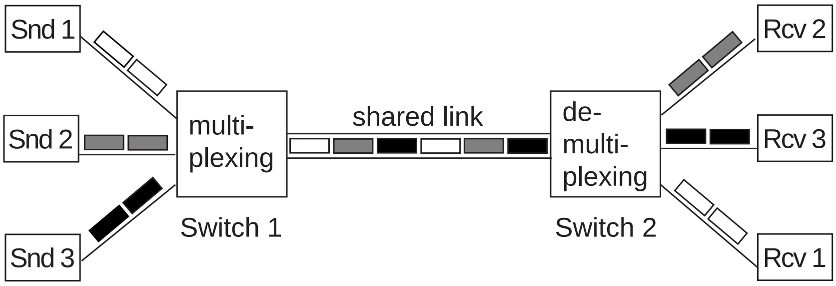

Figure 3.1: Multiplexing three logical connections over one physical shared link (from [PD96, Figures 1.7 and 1.8]).

are not optimal since, firstly, they are limited to a fixed number of flows that must be known ahead of time, and, secondly, they are ineffective due to idle links if one connection does not have any data to send.

The form of multiplexing to overcome these problems is called *statistical multiplexing*. Just like STDM, the physical link is shared over time by use of single time slots but instead of transmitting data strictly during these predetermined time slots, it is rather transmitted on demand. That means if one flow has no data to send its time slot can be used by other flows. This method avoids idle time and allows several flows to share a common resource efficiently. Statistical multiplexing defines an upper bound on the size of a data unit to be transmitted at one time, which explains the term *packet* in connection with data transmission.

Referring again to Figure 3.1, there are packets of three different incoming flows multiplexed onto one single link by a switch. Thus, it is the switch's decision made on a packet-by-packet basis as to which flow's packet to send next. That decision can be made in a number of different ways, e.g. by using a round-robin approach or rather based on a *first-come-first-served* basis. The decision depends on the *scheduling policy* of the network administration, which is described in more detail in section 3.2.4.

It might be possible that a switch receives packets of incoming links faster than the shared link can accommodate. In such a case, the switch is forced to buffer packets in its memory. However, if incoming packets extend the buffer size, some packets will have to be dropped and both switch and shared link are said to be *congested*. *Congestion control* is a complex issue within the field of communication networks and also influenced by the scheduling policy.

3.2 The Global Internet

The previous section introduced some of the most important terms in the field of communication networks and it also described how network resources can be shared by several hosts and connections. However, single networks are built using point-to-point links, shared media, and switches, which all represent and use different technologies. In order for all the different networks to communicate with each other they need to be interconnected with each other.

The term *internetwork* is a combination of single networks to establish some sort of host-to-host packet delivery service. A Local Area Network (LAN), as one form of an internal communication network, can be interconnected with one or several other LANs at different sites by means of a bridge. Extending several internetworks to a global reach results in an interconnection of physical networks or a *network of networks*, which is also referred to as *the Internet*. The nodes that interconnect the networks are called *routers* or *gateways*. A discussion about bridges, switches, and routers is given in [PD96, page 234-237].

3.2.1 Network Technologies

Communication networks play the key role for multimedia applications since they transmit the data from a sending host to a receiving host. A great influence on their performance have the different network technologies and their features. Networks can be subdivided into three categories depending on the regional extent of hosts: *Local Area Networks* (LAN), *Metropolitan Area Networks* (MAN), and *Wide Area Networks* (WAN).

A LAN is characterized through its local extent of only a few kilometers and high data rates of up to 1 GBit/s. The number of connections to a LAN is limited to typically a few hundred hosts. A MAN typically connects several LANs and it can be extended to an urban area with an available bandwidth that is generally higher than in LANs, i.e. more than 100 MBit/s. The number of connected hosts can reach several thousand.

A MAN spans a larger geographic area than a LAN, but a smaller area than a WAN. A WAN typically spans over larger regions, it can even extend to whole countries. WANs allow to interconnect LANs and MANs and hence, they play an important role for the access of distributed multimedia applications. A WAN interconnects organizations, companies, private users, etc. to a large network. Network owners are private or public telecommunication companies that run their networks across countries and that interconnect with other network providers to establish a wide network. Data rates in WANs are usually smaller than in LANs, but protocols and services can be used to improve the reliability of the connection (see section 3.3).

Ethernet

Nowadays, *Ethernet* is the most widely distributed form of a *Local Area Network* (LAN). Either coax cable or fiber is used as transmission medium. *Fast Ethernet* uses coax cable with a data rate of 10 or 100 MBit/s, whereas the high-speed network *Gigabit Ethernet* uses fiber cable and can therefore transmit data up to 1 GBit/s (compare [Tan98, Ste99]).

Ethernet is based on a bus system that connects all the communication systems (e.g. hosts and switches) by means of a transmission path. Ethernet uses *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) where a sending system first checks whether the bus is ready to transmit and then sends the data. If several systems try to send at the same time, a collision is detected, which causes the systems to wait a random period of time before trying again. Ethernet does not guarantee any maximum end-to-end delays which are required for the transmission of continuous media. It does not offer any other service guarantees either, but nevertheless, there is a number of possibilities to still use Ethernet for the transmission of audio and video data (compare [Ste99, page 412]):

- Continuous data can be transmitted over Ethernet without special treatment if network usage is kept at a moderate level. However, the number of occurring errors is not satisfactory.

- Using Ethernet requires a scalable encoding of the content that allows to adapt the transmission rate dynamically. However, occurring errors cannot be excluded either.

- The use of a separate Ethernet LAN dedicated only to the transmission of continuous media is a very costly solution and requires a lot of effort.

- Instead of connecting hosts to a shared Ethernet bus, they could rather be connected directly via a switch (*Switched Ethernet*), that provides every host the full range of available bandwidth. However, a data transfer could still interfere with an audio or video transmission.

Token Ring

Token Ring is another form of a LAN. It offers data rates of 4 or 16 MBit/s and is based on a well-ordered access mechanism. All the hosts are connected by a logical ring that rotates a token, which symbolizes the right to send data. In order to send, a host has to keep the token. Unless reserved by another host (with priorities between 0 and 7), a host can claim the token. If the token is free again, the host with the highest priority gets it next. The system requires mechanisms to implement a fair distribution of the token. The priority scheme and the maximum possible delay within the token ring allows the guaranteed support of continuous media. The following aspects have to be considered for the transmission of audio and video in a token ring (compare [Ste99, page 417]):

- Just as with Ethernet, a scalable encoding scheme of the content improves the dynamic of the transmission.
- Continuous media should always have a higher priority than discrete media. A high network usage makes the usage of waiting queues essential in order to reduce delays.
- Among the continuous media, audio transmission should have a higher priority than video transmission.
- It is possible to run both audio and video transmission on the same priority level and then reserve resources both statically and dynamically.

Fiber Distributed Data Interface

Fiber Distributed Data Interface (FDDI) can be considered as an enhancement of token ring. It consists of two uni-directional fiber rings that transmit data in opposite directions with a total data rate of 100 MBit/s. The transmission distinguishes a *synchronous mode*, that allows the reservation of bandwidth, and an *asynchronous mode*, that is similar to the token ring protocol. Due to the possibility to reserve bandwidth, the synchronous mode is suitable for the transmission of audio and video. However, the reservation process requires a considerable amount of time. Furthermore, FDDI supports *multicasting* (described in section 3.3.3) that allows the transmission of a data stream to several hosts, whereas a synchronization of different data streams is not supported.

Digital Subscriber Line

Digital Subscriber Line (DSL) is a modem technology that converts regular twisted-pair copper telephone lines into high bandwidth connection links to the Internet. Thus, it allows to establish high data rates that are required for the transmission of multimedia data. DSL covers a whole ‘family’ of similar technologies, also referred to as *xDSL*, which include ADSL (*Asymmetric DSL*), SDSL (*Symmetric DSL*), HDSL (*High-bit-rate DSL*), RADSL (*Rate-adaptive DSL*), and VDSL (*Very-high-data-rate DSL*). The currently most common technology is ADSL, which allows more bandwidth *downstream*, i.e. from the local network provider to the customer, than *upstream*, i.e. from the subscriber to the connecting station of the network provider. Technically, ADSL allows a maximum bandwidth of more than 6 MBits/s downstream and 640 kbit/s upstream. However, bandwidth varies since it is limited by the local providers.

3.2.2 Switching and Forwarding

So far, terms like ‘packets’ or ‘networks’ had been introduced but now it is time to explain the idea about how packets are sent across multiple networks. In other words, how is it possible for two hosts to communicate with each other although they are not directly connected with each other. Similarly to telephone networks where calling persons are not connected directly with each other but through a *circuit switch*, internetworks are connected by use of a *packet switch*. A packet switch enables packets to be transmitted from one host to another although there is no direct connection between the two hosts. Thus, a switch is a hardware device with several inputs that get mapped onto several outputs. The process of mapping incoming packets of different flows onto the correct outputs for the corresponding flow is called packet *switching* or *forwarding*. There is a *connectionless* approach (using *datagrams*) or a *connection-oriented* approach (using *virtual circuits*) for the communication of data packets. *Source routing* is another approach to direct the data packets correctly along their way from sender to receiver which is simple but less common than the previous two.

Source Routing

The idea behind *source routing* is to assign a number to each output of a switch and put it in the header of a packet. Since there may be more than one switch between sending and receiving host, the approach requires an ordered list of switch ports to store forwarding information for every switch along the path. One way to handle the header information for source routing is to rotate the list by every switch so that the next switch in the path is always at the front of the list. Another way is to strip off the first element as the switch uses it. Alternatively, an extra pointer field that indicates the current port could be updated by every switch.

The main reason why source routing is not so common is the fact that it scales badly for large networks. It is very hard for a sending host to know the complete path of the data packets in order to construct the correct headers. Furthermore, a path between sending and receiving host can contain a variable number of intermediate switches which means that the header field needs to be of variable size with no upper bound.

Virtual Circuit Switching

Another approach for packet switching is called *virtual circuit switching* which is based on a *connection-oriented* model just like *Public Switched Telephone Networks* (PSTN). However, the virtual circuit switching approach is packet-based and shares common network resources. At first, it requires that a virtual connection is set up from the sending to the receiving host. The way this works is as follows.

Assume, host A wants to send packets to host B with three intermediate switches along the path. Host A sends a connection request message to host B where the message contains a *Virtual Circuit Identifier* (VCI), let's say 5, chosen by host A as a unique identifier between host A and switch 1 for all the subsequent packets belonging to that flow. When switch 1 receives the connection request on let's say port 2, it knows[1] how to get to host B and forwards the request message on to switch 2 through let's say port 1. Since the identifier needs to be unique for every link, switch 1 chooses a new identifier, let's say 11, for the link to switch 2. Consequently, switch 1 makes a new entry in its *virtual circuit table* which contains information about the incoming port, the incoming identifier, the output port, and the outgoing identifier. When switches 2 and 3 receive the connection request message they behave similarly to switch 1 and if all three switches accept the connection, an acceptance notice is sent back to host A and the data transmission can begin. When host A is done sending packets to host B, it tears down the connection by sending a teardown message to switch 1, which is forwarded to switch 2 and 3. That way, unnecessary table entries can be deleted.

Advantageous with this approach in contrast to source routing is that only the connection request contains the full address for host B while each data packet only contains a small identifier. Also, the knowledge of the network topology is shared among the switches, where the knowledge of each switch is limited to making the right forwarding decisions only for its own output ports. Disadvantages imply the waiting time to set up a connection and the fact that a new connection needs to be established if a switch or a link in a connection fails.

It is possible to allocate resources to virtual circuits when the connection is established. *X.25* is another networking technology (similar to *Frame Relay*) that allows to allocate buffers to each virtual circuit (see section 3.3.3). These buffers are managed by flow control protocol to avoid an overflow of the buffers. This basic strategy is also called *hop-by-hop flow control*. This strategy in general permits the implementation of virtual circuits with different QoS levels where network resources are reserved in order to make some kind of performance-related guarantees. More on QoS supporting mechanisms in chapter 4.

---

[1] Switch 1 knows about the network topology through the usage of routing protocols and routing tables, which will be subject of section 3.2.3.

Datagrams

As opposed to the previously described connection-oriented approach of circuit switching, the *datagram* model tries to forward data packets based on a *connectionless* approach. The idea is very simple: instead of setting up a connection in advance, a datagram, as the basic transmission unit in the Internet architecture, is equipped with all the necessary information to be delivered to its destination. That is, every datagram contains the complete destination address. Just like switches keep virtual circuit tables in the connection-oriented model, every switch in the datagram model keeps its own *forwarding* or *routing table* that contains enough information about the network topology. These tables help the switches to decide how to forward a datagram on its way from source to destination based on its destination address.

The advantage of the datagram approach is that there is no round-trip-time delay in waiting for a connection setup. A host can send datagrams as soon as it is ready. However, a host can never be sure whether the network is ready to transmit the datagrams or whether the destination host is even prepared to receive the datagrams. Every datagram is transmitted independently from its predecessors which means that a link or switch failure does not necessarily force the connection to break but instead subsequent datagrams may be switched to an alternate route if the forwarding tables are updated accordingly. But since every datagram carries the full destination address, the header size of a datagram is much bigger than for a data packet in the connection-oriented model.

3.2.3 Internet Routing

After describing the switching and forwarding process in the previous section, this section finally gives information about how either hosts (in the case of source routing) or switches gain their knowledge about the network topology in order to make the right decisions on which outgoing port to chose for a packet or datagram. In the previous two approaches (virtual circuit switching and forwarding datagrams), the switches made use of virtual circuit tables or forwarding tables, respectively. The fundamental purpose of *routing* is to generate for every switch such a table with the correct information about how to reach distant hosts or switches in the network. Notice, that this provides an indication of the difference between forwarding and routing. Forwarding describes the process of looking at a packets destination address and sending it through an outgoing port according to a forwarding table. Routing, on the other hand, describes the process of accumulating and sharing knowledge of the network topology and possible routes between source and destination of a packet or datagram.

Routing can be based on problems of graph theory where the nodes of the graph are either hosts or switches, and the edges are network links. Each edge has an associated cost or metric, which is a measure for the effort to transmit data packets or datagrams across that link. The goal of routing is to find the lowest-cost path between any two nodes, which is a very complex and dynamic process due to the constantly changing network situation (changing number of nodes and the cost of network links). Routing is performed by use of *routing protocols* that are run in a distributed and dynamic manner among switches in order to find the lowest-cost path. Two main classes of routing protocols can be distinguished and are further described in the following: *distance vector routing* and *link state routing*.

Distance Vector Routing

Distance vector routing is based on the assumption that each node knows the cost (or distance) to reach each of its directly connected nodes. A link that is down is assigned an infinite cost. The protocol implies that each of the nodes sends regular messages to its directly connected nodes containing its knowledge about distances to neighboring nodes. If a node A receives a message from node B with a path to a node C that is shorter (less expensive) than the one it currently knows about, node A changes the entry in its *routing table* by updating the length (cost) of the path and the new route to node C to go through node B. Thus, after every node has exchanged a few updates with its directly connected neighbors, all nodes in the network will know the least-cost path to all the other nodes.

The messages are either sent regularly, to let the other nodes know that it is still running, or the messages are *triggered* whenever the network situation changes and causes other nodes to update their entries in the routing table. The failure of a node A can be detected by a neighboring node B either if node B does not receive an acknowledgment on a control packet that it sent out or if node B does not receive the periodic routing updates from node A. If node B detects the failure of node A, it sets the distance to node A to infinity and passes that new information on to its neighboring nodes, so that alternative paths can be found that go around node A. Eventually, several messages need to be exchanged until the protocol converges and the system can become stable again. As far as speed of convergence is concerned, the following link state routing outplays distance vector routing.

Link State Routing

Similar to distance vector routing, *link state routing* also assumes that each node knows the state and the cost of the links to its neighboring nodes. The basic idea of link state routing is similar to the one in distance vector routing: the knowledge of every node about how to reach neighboring nodes is disseminated not only to directly connected nodes but to all the nodes in the network, which allows each node to build a complete map of the network. Thus, link state routing is based on two mechanisms, *reliable flooding* for the dissemination of link-state information among all the nodes and *route calculation* as the subsequent process of construction a network map by every node with routes and distances between any two nodes.

The idea behind reliable flooding is that each node creates a packet, called *link-state packet* (LSP), and sends it out to all its directly connected nodes, with each of these nodes forwarding the same packet to all of its directly connected nodes, and so on. This process continues until the LSP has reached every node in the network. The LSP of a specific node contains the following information:

- the ID of the node that created the LSP
- a list of the neighbors of that node with the cost of each of the links
- a sequence number
- the time to live (TTL) for that packet

The first two items are used for route calculation, the last two are needed to make flooding reliable since there may be multiple, contradictory LSPs of the same node traversing the network. Every time a node generates a new LSP, it increments the sequence number by one. The TTL field is used to delete old LSPs if the value reaches 0.

Route calculation answers the question how every node can build a map of the network topology out of the LSP information and calculate routes and distances between any two nodes. Route calculation is based on a well-known algorithm from graph theory, *Dijkstra's shortest path algorithm*, which is described in detail in [CLR90, page 527-532].

Advantages of link-state routing are the fact that it converges quickly to a stable network situation and that it reacts rapidly to any changes in the network topology without generating too much traffic. However, the stored LSP information of all the nodes in the network can be quite large.

3.2.4 Queuing and Congestion Control

Congestion in communication networks (such as the Internet) occurs if more data packets request to be transported (i.e. routed) than there are available network resources. In such a case, a router or switch has so many packets queued that it runs out of buffer space and has to start dropping packets. Congestion control is a complex task and has been studied ever since the first network was designed. One reason for its complexity is that it implies routers and switches to implement queuing algorithms while it requires congestion control mechanisms to be implemented in the transport protocol running on the end hosts.

FIFO (first-in-first-out), also called *FCFS* (first-come-first-served), is a simple queuing algorithm where the first packet that arrives at a router is the first packet to be transmitted. If a packet arrives at a router and the queue (buffer space) is full, then the router discards that packet regardless of the kind of information the packet carries. Thus, FIFO queuing puts all the responsibility for congestion control to the end points of communication, the end hosts. Variations of the FIFO queuing imply subtle decisions about which packets to transmit and which packets to discard if the queue is full. Another variation on simple FIFO queuing is to implement priority queues where the routers support multiple prioritized FIFO queues and have the sources mark each packet with a priority. However, the problem with priority queuing is how to keep the sources from marking all their as being high priority.

The main problem with FIFO queuing is that it does not distinguish between packets of different traffic flows, which allows a malicious source to flood routers with its data packets and thus capture an arbitrarily large fraction of the network capacity. *Fair queuing* (FQ) overcomes this problem by maintaining separate queues for each flow that is handled by the router. The router then serves these queues in a round-robin manner. When a source sends packets too quickly, then its queue fills up only and leaves the other flows their appropriate share of network capacity. However, FQ does not inform the sources about the state of the router and thus requires the use of an end-to-end congestion control mechanism.

An example of end-to-end congestion control on today's Internet is implemented by the *Transmission Control Protocol* (TCP), which will be introduced in section 3.3.2. The basic idea behind the TCP congestion control is for each source to determine how much capacity is available in the network and then decide upon that information how many packets it can safely have in transit. This algorithm requires the introduction of an ACK as a signal that one packet has left the network and arrived at the receiver. The difficulty with this algorithm is that every source must be able to adjust the number of packets it has in transit to the constantly changing situation of available bandwidth. For more details on queuing mechanisms and congestion control see [PD96, pages 392-434].

3.3 Protocols and Services

The previous sections introduced different networks, their technologies, and their suitability for transmitting multimedia data. This section gives an overview of network and transport protocols and services that enable communication between two connected nodes in a network.

In order to understand the modes of operation of the protocols, it is important to understand the structure of the different protocols. It was the *International Standards Organization* (ISO) that formally defined a way to structure protocols in a protocol graph. The so called *Open Systems Interconnection* (OSI) architecture describes a reference model for partitioning network functionality into seven layers. From bottom to top these are the *physical layer*, *data link layer*, *network layer*, *transport layer*, *session layer*, *presentation layer*, and *application layer* [PD96, Tan98]. The different protocol layers implement the functionality that was assigned to them and provide the necessary prerequisites for the upper layer protocol according to an interface specification.

However, an alternative view of a network architecture is the *Internet Architecture*, which evolved out of experiences with one of the first packet-switched networks, the *ARPANET*, as the originator of the later Internet. Both the Internet and the ARPANET were funded by the *Advanced Research Projects Agency* (ARPA), one of the R&D funding agencies of the US Department of Defense. The Internet architecture has succeeded over the OSI model and consists of only four layers: the *physical layer*, *network layer*, *transport layer*, and *application layer*. The focus of the Internet architecture is definitely on the two middle layers, which run the two main protocols on the network and transport layer, the *Internet Protocol* (IP) and the *Transmission Control Protocol* (TCP) (more details on these two protocol in the following sections 3.3.1 and 3.3.2). That is why the Internet Architecture is also called *TCP/IP Architecture*.

3.3.1 Network Layer

Special requests for the transmission of multimedia data are directed to the network layer, which embraces the supply of high bandwidth, multicast communication, reservation of network resources, and the support of different QoS levels. While the current implementation of the Internet protocol has not implemented any mechanisms to guarantee QoS levels, there are developments and ongoing research in finding mechanisms to support QoS (see chapter 4).

Internet Protocol

The currently mostly used version of the *Internet protocol* (IPv4) only offers a connectionless, *best effort service* of datagrams across the Internet, which means that the delivery of data packets is attempted as its best effort but cannot be guaranteed [Pos81]. This feature is very disadvantageous and does certainly not fulfill the QoS requirements for audio and video transmission. The basic implementation of the protocol is not equipped with mechanisms to reserve network resources and can therefore not ensure any service guarantees.

The IP datagram[2] was introduced in section 3.2.2 as part of a connectionless approach to forward data packets on the Internet. A datagram is the basic transmission unit in the Internet architecture and is equipped with all the necessary information to find its way from sender to receiver. No path setup is required in advance, and if something goes wrong with the packet delivery, i.e. the datagram gets lost, corrupted, misdelivered, or simply fails to reach its destination, it is not the responsibility of the Internet protocol. Higher level protocols, such as TCP, or applications that run above IP need to be aware of all these possible failure modes. This behavior demonstrates the advantages of a modular Internet architecture protocol stack, i.e. every layer has its own task independent of upper or lower layers. The big advantage of IP is that it can basically run over any other networking technology (e.g. Ethernet, FDDI, Token Ring).

When a message is sent from a sender to a receiver, it is first fragmented into several datagrams, then the single datagrams of the message are transmitted over the network before they are finally reassembled to the original message. This process of *fragmentation and reassembly* is important for the forwarding of IP messages across heterogeneous networks that have their own idea of how large a data packet can be. For example, Ethernet can accept packets up to 1500 bytes long, while FDDI packets may be 4500 bytes long. The form of an IP datagram is standardized and consists of a header (of length *HLen*) followed by a number of bytes of data (see Figure 3.2).

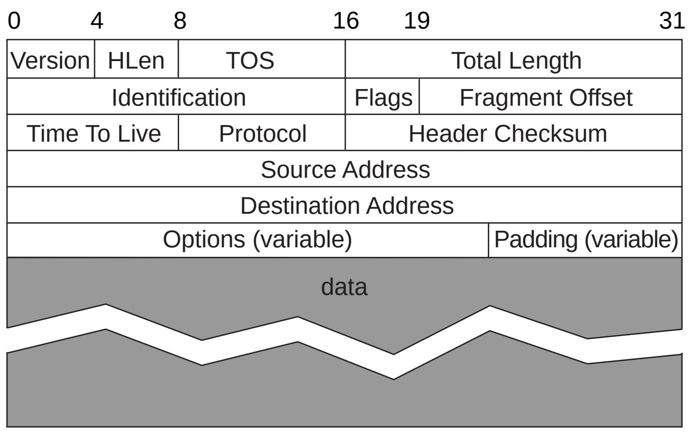

Figure 3.2: IPv4 packet header.

If no options are set, the header is usually 5 words (20 bytes) long, while the size of the data varies but can at most be 65535 bytes long (including the header)

[2]Note, that 'IP datagram' is used equivalently with 'IP packet' as the basic transmission unit for Internet communication.

that is specified by the 16 bits in the *Total Length* field. The 8-bit *TOS* (Type of Service) field has originally been reserved to specify different service levels for certain applications (e.g. real-time streaming applications) but has never been used widely enough to be really useful. For a detailed description of the header information see [PD96, Tan98, Pos81].

Next Generation Internet Protocol

The wide usage of IPv4 on the Internet, the increasing need for QoS-based multimedia applications, the lack of any security features in IPv4, and the growing demand for a bigger address space has finally led the *Internet Engineering Task Force* (IETF) in the early 90s to post a 'Call for Proposals' for an Internet protocol of the *next generation* (IPnG). From 1995 on, different Request for Comments (RFC) had been published to specify the next generation Internet protocol, IPv6 [Par95, DH98, CD98].

The primary motivation to develop IPv6 was the increasing shortage of available IP addresses. That is why the length IPv6 addresses were quadruplicated from 32 bits in IPv4 to 128 bits in IPv6. The whole header in IPv6 is only twice as big as the existing one but much simpler since some of the IPv4 header fields were deleted or designated to optional fields which is very advantageous for the network overload and the process of header information. Thus, the IPv6 header is always 40 bytes long (compared to 20 bytes in IPv4 in the absence of options) and options are treated as *extension headers* that appear in specific order after the regular header information (see Figure 3.3). For a detailed description of the IPv6 header, see [PD96, Tan98, DH98].

Unlike IPv4, the next generation Internet protocol took into account the concept of *flows* as a sequence of data packets that belong to one group of packets exchanged between a sender and a receiver (e.g. a multimedia application flow). The IPv6 header contains a 20-bit field for the *Flow Label* that a router for example can use to forward the packets accordingly. Another field in the IPv6 header is the 8-bit *Traffic Class* field, that is used by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets. This feature could be used for real-time multimedia streams that require special treatment and fast forwarding as opposed to regular data traffic with low priority.

As far as security is concerned, IPv6 supports functionalities to guarantee authenticity of the communicating entities, data integrity, and confidentiality regarding the content of the exchanged data. These features had to be implemented on an application level in IPv4, but has now become redundant in IPv6.

An important issue in the discussion about the replacement of the current Internet protocol by IPv6 is the question how the transition is going to be performed.

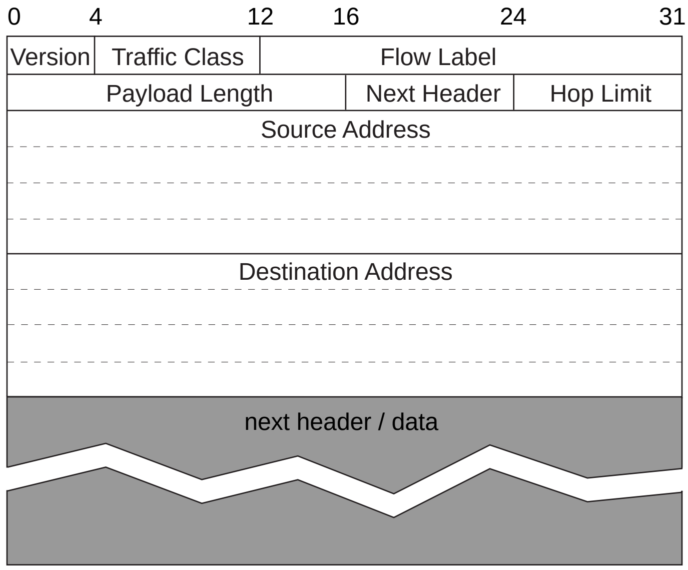

Figure 3.3: IPv6 packet header.

The Internet is far too big and decentralized to have a ‘flag day’, a specific day where every host and router in the network is upgraded from IPv4 to IPv6. Therefore, smooth transitions are rather discussed where either IPv6 nodes run both IPv6 and IPv4 and use the *Version* field to decide which protocol to run (*dual-stack operation*) or where IPv4 packets are encapsulated into IPv6 packets and vice versa depending on the running protocol (*tunneling*). A further discussion on improvements, advantages, and the migration from IPv4 to IPv6 are discussed in [PD96, Ste99].

3.3.2 Transport Layer

Transport protocols are responsible for the addressing of communication end-points, connection setup and teardown, the fragmentation and reassembly of data packets, the flow control, and the error detection and correction of transmitted data. Some of these tasks play a major role for the reliable transmission of real-time streams, which are addressed in the following.

Transmission Control Protocol

The *Transmission Control Protocol* (TCP) provides a reliable and connection-oriented transmission service in duplex mode. The implemented flow control mechanisms, such as the *sliding window algorithm*[3], guarantee the arrival of all the sent data packets in the correct order. This is very advantageous for reliable applications without real-time restrictions, like a simple file transfer, but for real-time services, the maximum end-to-end delay is often exceeded.

Furthermore, the implemented *slow start algorithm*[4], that reacts upon high network traffic, is not suited for the transmission of real-time data, since it reduces the data rate if network congestion occurs. Hence, TCP was developed as a protocol for the reliable data transport of non-real-time applications but is not applicable for real-time applications.

User Datagram Protocol

The *User Datagram Protocol* (UDP) is a simple extension of the Internet protocol IP, which provides a connectionless and unreliable transport service for data packets. Besides the multiplexing of datagrams and the creation and control of checksums, UDP does not offer any other services. Higher level protocols that use UDP have to provide control mechanisms like flow control in case of packet loss or the fragmentation and reassembly on their own. Nevertheless, UDP is used by many multimedia applications since it is simple and provides certain real-time transport features. UDP is widely used and constantly subject to enhancements and optimization.

Xpress Transport Protocol

The *Xpress Transport Protocol* (XTP) was developed for high-speed networks with low error rates [SDW92]. XTP is flexible, adaptive, supports multicasting, and offers flow control mechanisms. Originally, XTP was developed for the use of a number of applications, e.g. real-time control systems, remote procedure calls in distributed operating systems and databases, or the transport of high amounts of data. XTP offers different strategies for error control, such that acknowledgments are required only for selective packets or for lost packets (negative acknowledgments), or when selective packets are retransmitted.

[3] The sliding window algorithm allows the sender to transmit multiple packets (up to the size of the window) before receiving an acknowledgment. As soon as acknowledgments for the first packets in the window arrive, the window 'slides' and enables more packets to be sent [PD96].

[4] The slow start algorithm is a congestion-avoidance algorithm for TCP that attempts to slow down outgoing traffic in case of high traffic. For every acknowledgment that is returned, two additional packets are sent [PD96].

3.3.3 Other Protocols and Services

Asynchronous Transfer Mode

The *Asynchronous Transfer Mode* (ATM) is both a communication service and a networking or switching technology with minimal network functionality. ATM treats packets as *cells* with a fixed total size of 53 bytes, consisting of 5 bytes for the header information and 48 bytes user data (see Figure 3.4). There is no error handling and flow control mechanisms with ATM which requires higher layer end-to-end protocols to deal with reliability issues. The data rate for a transmission in *duplex mode*[5] is typically around 155 MBit/s or 622 MBit/s, but can also reach up to 2.4 GBit/s and it is best suited for fiber cable.

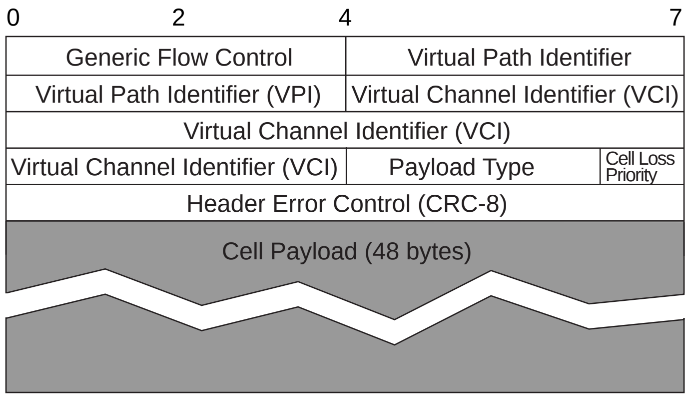

Figure 3.4: ATM cell format.

ATM is a connection-oriented, packet-switched network that uses *virtual circuits* (by means of a *Virtual Path Identifier* (VPI) and a *Virtual Channel Identifier* (VCI)) for cell switching. ATM provides a transport service for data packets with much higher rates than other packet-switched networks such as Ethernet or FDDI. ATM requires a connection setup, called *signaling*, before cells can be exchanged. This allows ATM to reserve and allocate network resources at switches along the virtual circuit, which in turn enables the specification and guarantee of different QoS levels. If resources are not available the connection cannot be setup. This feature was one of the main reasons for the ITU-T to use ATM as the transmission mode for *Broadband Integrated Services Digital Networks* (B-ISDN) as the future high-speed WAN technology (for details in ATM and ISDN, see [Orl00, pages 93-238] and [Gib01, pages 129-177]).

[5] Full duplex mode refers to transmission of data in two directions simultaneously, e.g. by sending and receiving data on different links.

Besides traffic parameters that characterize a specific connection, ATM also defines five different service categories out of which some qualify for the transmission of audio or video streams (see Figure 3.5). The service categories differ from each other through their QoS parameters and it can be distinguished between real-time and non-real-time services and whether the services are rather simple or complex (compare to [Ste99, pages 460-464]):

1. **Constant Bit Rate (CBR).** This category was specified for real-time applications that require a constant bit rate and that do not tolerate service shortcomings. The service implies that a certain static bandwidth is reserved in advance for the whole duration of the connection. This service category is ideal for (compressed) audio or video transmission with evenly distributed data traffic.

2. **Unspecified Bit Rate (UBR).** UBR was specified for applications without real-time restrictions where no limits for delay or jitter have to be met. No promises can be made concerning the data rates or the fairness of the connection. UBR is principally used for traditional data transfer such as in file transfers or emails and it reflects the service model of the current best-effort Internet.

3. **Real-time Variable Bit Rate (rt-VBR).** The service category rt-VBR was specified for real-time applications without a constant data rate. The advantage over CBR is the fact that network resources are used much more efficiently and with a lower loss tolerance. This service class is ideal for the synchronous transmission of compressed real-time audio or video, such as video conferences or VoD.

4. **Non-real-time Variable Bit Rate (nrt-VBR).** The nrt-VBR service was specified to support applications without real-time requirements that are characterized by variable sending of data packets. It also improves the parameters 'delay' or 'packet loss' since bandwidth is reserved for that service. Preferred applications in that service category could be multimedia emails, cached audio or video transmissions, or transaction systems.

5. **Available Bit Rate (ABR).** The ABR service was developed for non-real-time applications that are not subject to any delay or jitter restrictions. Instead, ABR offers a flow control mechanism to minimize packet loss and to maximize fairness among all operating ABR connections. The service class can be used for LAN emulations and for current Internet services.

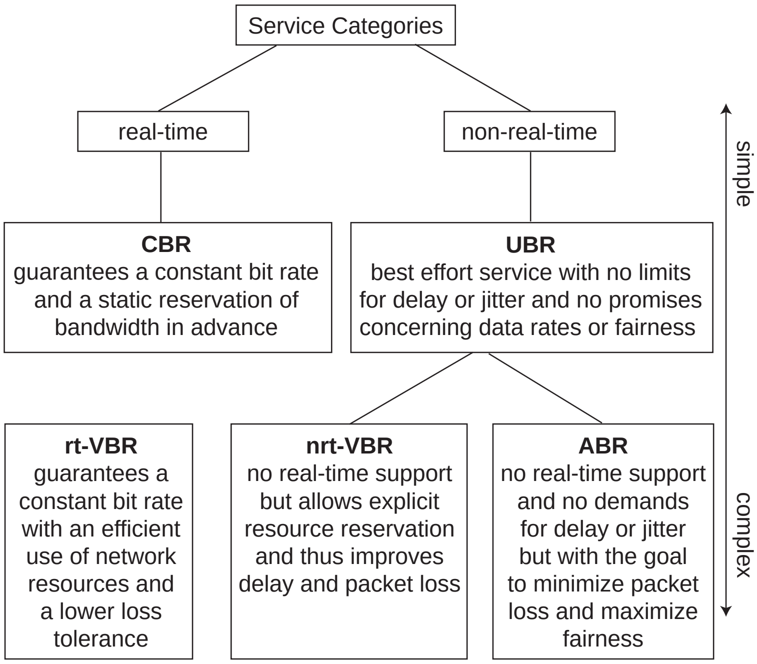

Figure 3.5: Relationship between the different ATM service categories (from [Ste99, Figure 14-16].

Frame Relay and X.25

Frame Relay (FR) is a high-performance WAN protocol that operates at the physical and data link layers of the OSI reference model. Originally, as initial proposals for standardization were presented in 1984, it was designed to be used across ISDN networks only but nowadays, it serves as a protocol for a variety of network interfaces. Frame Relay was intended to solve the demand for high bandwidth networking. It is a packet-switched technology, which relies on reliable underlying data links and high performance processors. Frame Relay is often described as a streamlined version of *X.25*, that offers more of the robust capabilities such as windowing and retransmission of lost packets. The X.25 standard was initiated by common telephone companies in the 1970s when unreliable packet-switched telephone networks served as the platform for X.25 WANs.

Since X.25 also provides services at the third layer (the network layer) of the OSI reference model, it enables Frame Relay to offer higher performance and greater transmission efficiency, which makes Frame Relay more suitable for current WAN technology, such as LAN interconnection. Both protocols provide a connection-oriented data link layer communication with *virtual circuits* as logical connections created between two hosts across a FR/X.25 network. Two types of virtual circuits are distinguished, *switched virtual circuits* (SVC), which are temporary connections used for sporadic data transfers, and *permanent virtual circuits* (PVC), which are permanent connections used for frequent and consistent data transfer. More information on frame formats and implementation in [DFL+98].

IP Multicast Transmission

After the expression 'multicasting' was mentioned earlier in section 3.2.1, it is finally described shortly. Multicast is a network protocol that allows the sending of a datagram to more than one receiver. The datagram is sent only once by the sender and, where necessary, switched and/or duplicated by the multicast-enabled routers. *Multicast groups* form the basis for any multicast communication. Therefore, a multicast extension to the Internet Protocol was proposed in RFC 966 in 1985 [DC85], as a standardized interface for the usage of IP multicast. The RFC defined the *Internet Group Management Protocol* (IGMP) that supports the implementation of joining or sending to multicast groups.

IP Multicast transmission is realized by the assignment of special (Class-D) addresses. Data packets with that address are passed on to all the members of that multicast group, and thus, a parallel transmission to several receivers is possible. Multicast on the Internet is also called *IP Multicast* and is based on the UDP protocol which means that it does not provide any service guarantees for delivered packets. Multicast transmission on the Internet requires special precautions. The transport of multicast packets is realized by a special network, called the *Multicast Backbone* (MBone)[6]. Several multicast routers form a virtual network based on the existing Internet topology. Multicast packets are *tunneled* between multicast routers so that they look like regular IP packets for intermediate non-multicast routers. More information on multicast routing in [DC90, DEF+94, Tan98].

[6]Accessible at URL: http://www.mbone.com

Chapter 4

Internet Protocol Architectures to Support Quality of Service

This chapter describes, on the lowest 'communication level' (compare to bottom level in Figure 1.1), mechanisms and approaches to support QoS-based Internet communication using new *Internet Protocol Architectures*. This chapter first describes general mechanisms and protocols to reserve network resources, and thereafter introduces and discusses different Internet Protocol Architectures that allow the transport of real-time data through the Internet.

4.1 Mechanisms to Support Quality of Service

The development of the Internet from the actual precursor, the military ARPANET, resulted in a global infrastructure that attached great importance to simple but reliable communication connections. Ever since the beginning of the widespread usage of the Internet, treatment of time-critical data packets across networks has never been an issue. But nowadays, through an increasing need and demand for high-bandwidth Internet connections (e.g. through the use of multimedia and other time-critical applications), a best effort service for the transport of data packets across network connections is not sufficient anymore and other service models and Internet protocol architectures are required to meet the high QoS demands.

The goal of new Internet protocol architectures must be to replace or at least enhance the well-established best effort service of today's Internet protocol (IPv4) that predominates in routers and network equipment. A simple *First-In-First-Out* (FIFO) strategy is not sufficient in routers if data packets of real-time applications have a greater urge to be switched faster through a network than others.

Besides the briefly mentioned switching strategies in network routers (in section 3.2.4), it is the reservation of network resources that experts attach great importance to. Based on the idea to reserve resources, the following introduces different mechanisms (protocols and architectures) to support QoS requirements.

4.1.1 Resource Reservation

Basically, the reservation of resources can be considered as a task of the network management as an intermediary entity between sender and receiver. Resource reservation could be performed manually or by an automatic process that requires information about sender, receiver, and the specified QoS. However, this approach is unusual and rarely used in practice.

A more common approach is the resource reservation initiated by the end systems that transmit control information along the path on a *hop-by-hop* basis (from router to router). In that case, a router decides whether it can or want to meet a certain QoS specification and then forwards the message on to the next router. Resource reservations can be subdivided into *sender-oriented* and *receiver-oriented* approaches depending on whether the sender or the receiver initiated the resource reservation.

Furthermore, resource reservation can be distinguished between a *hard state* and *soft state* approach. Hard state means that the reservation of resources is maintained until an explicit tear down message deallocates the reserved resources for a connection. The soft state approach requires periodic refresh messages in order to keep up the reservation of resources. The soft state approach is more flexible and allows a fast release of resources in case of connection interrupts but is more expensive to implement.

4.1.2 Stream Protocol

The *STream Protocol* in its version 2 (ST-II) is a network layer protocol with the ability to replace IPv4 since it can handle real-time data accordingly. ST-II offers a connection-oriented transport service with QoS guarantees based on sender-oriented reservation of resources. ST-II consists of the *STream Control Message Protocol* (SCMP), that realizes the exchange of protocol messages, and of the ST-protocol itself for the reliable transport of data [Ste99, pages 487-488].

Resources are reserved by means of a SCMP message that allows to set parameters for the *flow specification* such as minimum packet size, data rate, end-to-end delay, jitter, etc. The flow specification is forwarded from router to router along the path and if the requirements can be met a connection will be established otherwise refused. The protocol uses the hard state approach for the reservation of resources.

ST-II was developed for the transmission of multimedia data, and therefore equipped with the ability to reserve network resources between sending and receiving system. But in case of a multicast scenario with several receivers, resource reservation is performed badly with high redundancies and overloads since multiple reservations cannot be combined.

4.1.3 Real-time Transport Protocol

The *Real-time Transport Protocol* (RTP) is an end-to-end protocol that was designed to support transmission of real-time data. The goal was to develop a consistent and flexible protocol that performs well in tasks like packaging of data, proper labeling of data encoding, adding of time information, and other control tasks. RTP was not meant to replace an existing transport protocol, yet it is independent of any transport protocol and can operate with protocols such as ST-II or UDP/IP as well as in ATM networks.

Its main scope of application is the support for multimedia teleconferencing with several participants. It therefore supports both unicast and multicast communication. It also supports individual requirements of heterogeneous participants and communication paths by usage of a *translator* and *mixer*. They are used to change the characteristics of the transmitted data by scaling, changing data formats, or by mixing several data streams to an aggregated stream.

RTP is a soft-state protocol that uses the *Real-time Control Protocol* (RTCP) to exchange control information in regular time intervals to keep up the soft state in routers ([Ste99, pages 495-500], [CHW99, pages 123-128]).

4.2 An Architecture for Integrated Services

As a consequence of constant demands for provisioning QoS-based Internet services, the IETF formed a working group to discuss extensions to existing Internet architectures and protocols to provide integrated services, i.e. to support real-time as well as non-real-time IP services. The working group came up with a Request for Comments in 1994, and it contained *an Architecture for Integrated Services in the Internet* (IntServ) [BC94].

IntServ defines a networking framework, which supports unidirectional end-to-end flows as a distinguishable stream of related datagrams that results from a single user activity and requires the same QoS. In principle, the IntServ framework consists of four basic components or mechanisms: *packet classifier*, *admission control*, *resource reservation*, and *packet scheduler*. Figure 4.1 illustrates how the components are arranged in an IntServ router. Notice that other secondary control units (e.g. policy control or routing) are summarized by the component *other control units*.

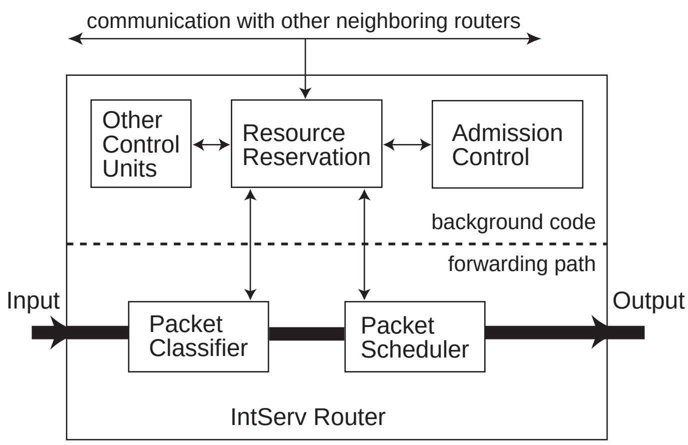

Figure 4.1: IntServ framework showing the different components in a router (from [BC94]).

With a regular best-effort switching service for Internet traffic, datagrams are equipped with the destination address and then simply sent out on the network hoping they would reach their destination. The IntServ framework foresees two service classes to distinguish between real-time streams and traditional data streams: *Guaranteed Service* [SPG97] and *Controlled Load Service* [Wro97]. While the guaranteed service class guarantees a specified maximum delay for its data packets, the controlled load service rather predicts packet delays and is therefore also referred to as *predictive service*. In order to determine the membership to either service class, packets must be *classified* and mapped into the corresponding service class. This classifying process is based on a set of information that is also called *flowspec*. A flowspec is a list of parameters used by an application to specify the desired QoS. This flowspec is carried by the *resource reservation protocol*, passed to admission control, and finally used for packet scheduling.

When the flowspec arrives at a router or host to request resources for a certain service, it is up to *admission control* that decides whether a new flow can be granted the requested resources without interfering with earlier guarantees. Admission control implements a decision algorithm that determines whether a service can be provided or not. If for example 10 users ask for a service that requires 2 MBit/s of link capacity, and they all share a link of 10 MBit/s, admission control will have to accept some of the requests and reject others.

The central idea of the IntServ framework is based on the fact that resources need to be reserved if a real-time service wants to be provided. Therefore, *resource reservation* is the central component to allow QoS-based service delivery. As part of the IntServ framework, the *Resource ReSerVation Protocol* (RSVP) was developed as a *signaling protocol* to exchange service requests, transport flowspecs from router to router, and to carry admission control decisions.

When flows have been described by the flowspec and admission control decisions have been made, network routers need to forward packets of the different flows accordingly in order to meet the QoS requirements. *Packet scheduling* is the mechanism that manages the way packets are queued and scheduled for transmission in routers and switches.

4.2.1 Resource Reservation Protocol

As mentioned in the previous section, the *Resource ReSerVation Protocol* (RSVP) was developed as a signaling protocol to establish a connection between a sender and a receiver based upon IP as the network layer protocol. According to [BC94], RSVP was designed to meet the following requirements:

- The protocol should be designed for a multicast environment.
- It should accommodate heterogeneous requirements concerning different service needs of different receivers.
- It should allow a flexible control over the way reservations are made along branches of a multicast tree.
- It should have elementary functionality for adding or deleting a sender or receiver to an existing set.
- It should be robust and scalable for the usage with large multicast groups.
- Finally, it should allow resource reservations in advance.

Reservations of resources are receiver-oriented in RSVP and performed exclusively unidirectionally. In order to inform the receiver(s) about the trait of the data stream, and thus to give them indications about reservation requirements, the sender sends a *PATH message* to all the potential receivers. A receiver, that is interested in the described data stream, generates a *RESV message* and sends it back to the sender along the same path that was used to transmit the PATH message. Each router on the path looks at the reservation request and tries to allocate the necessary resources to satisfy the request. If the reservation can be made, the RESV message is passed on to the next router until it reaches the sender, otherwise an error message is returned.

RSVP messages are transmitted as IP datagrams and generate a soft state at the routers along the path between sender and receiver. If the reservation request is not renewed after a certain period of time, the reservations are deallocated from the routers. As opposed to ST-II, reservations with RSVP can be combined and merged (for multicast scenarios) if the same data stream shares common resources on the way to several receivers.

Reservation in advance is a very helpful functionality that allows to reserve network resources for applications (e.g. teleconferences, VoD applications) whose point of time for the beginning process is known some time in advance. It separates the reservation process from the actual provision of the resources. *Resource Reservation in Advance* (ReRA) is such a concept that was developed by Wolf and Steinmetz [WS97].

4.3 An Architecture for Differentiated Services

Advantageous features of the IntServ and RSVP approach encompass a receiver-oriented QoS specification, the support of multicast traffic and its merging for reservations, and the soft state approach for maintaining context data of a flow. However, the support on a per-flow basis is assumed to show scalability problems with respect to a large number of flows and states to be kept in routers. The per-flow granularity imposes overhead which may not be necessary for a certain number of situations.

Due to the assumed scalability and overhead problems in case of many single flows, the IETF formed another working group to develop *an architecture for differentiated services* (DiffServ) in 1998 [BBC^{+}98]. Instead of treating a single flow as the entity of interest, the Differentiated Services Internet handles traffic based on the notion of aggregated flows and fixed numbers of service levels (or classes) in terms of service profiles. This approach is supported by a *domain-concept*, where a group of routers implements a similar number of service classes (e.g. *Expedited Forwarding* (EF) [JNP99] and *Assured Forwarding* (AF) [HBWW99]) and the appropriate set of policies. This DiffServ domain is defined by a fixed boundary, consisting of *ingress* and *egress routers* (so called *edge routers*). However, traffic traversing such a DiffServ domain is required to be marked.

This marking is performed on a per IP-packet basis at the ingress routers using the DiffServ (DS) field [NBBB98] as a replacement header field for the existing IPv4 *TOS* octet [Pos81] or the IPv6 *Traffic Class* octet [DH98] (compare with IPv4 and IPv6 headers in section 3.3.1). This DS field allows the definition of *per-hop behaviors* (PHB), which determines the service level and the forwarding behavior applied to a packet. Once the DS Code Point (DSCP), as the significant 6-bit part of the 8-bit DS field, has been set, the packet 'tunnels' through the DiffServ domain being treated equally in every interior router. Similarly, as with the IntServ approach, an aggregated DiffServ flow is considered to be unidirectional.

Since a certain service level may be required for an aggregation of flows over a longer period of time, a longer termed service contract is required between the different DiffServ domains that are maintained by service providers. The contracts are set up between interconnecting service providers to describe and maintain the desired service level for the aggregated flows. Considerations to combine RSVP and DiffServ have been made and are still ongoing.

4.4 Multiprotocol Label Switching Architecture

MultiProtocol Label Switching (MPLS) is called 'multiprotocol' because its techniques are applicable to any network layer protocol (layer 3 of the OSI reference model). However, its use is focused on IP as the most widely used network layer protocol. A router that supports MPLS is known as a *Label Switching Router* (LSR).

The basic concept of MPLS is to use labels as short, fixed length, locally significant identifiers (typically 20 bits long) that determine the *Forwarding Equivalence Class* (FEC). A FEC describes a class of IP packets that are forwarded by all routers in the network in the same manner, i.e. over the same path, with the same forwarding treatment, etc. The class of service can be fully or partially inferred from the label, which represents the combination of a FEC and a service class.

One of the fundamental features in MPLS is that no further network header analysis must be performed by subsequent routers, once a packet is assigned to a FEC, all the forwarding is controlled by the label. The label is used as an index into a table that specifies the next hop, and also a new label. That means that labels are local to a link between two LSRs (and possibly several intermediate routers or switches). That implies that the intermediate routers can switch the packets based on the label (as part of layer 2) without considering the network layer anymore.

The old label is replaced with the new label before the packet is forwarded to its next hop (*label swapping*) [RVC01]. A label distribution protocol is required where different LSRs inform each other about their label bindings. The base protocol developed for that purpose is the *Label Distribution Protocol* (LDP) as specified in [ADF$^+$01].

Within the DiffServ approach data packets are marked when entering a DiffServ domain to indicate each packet's service class and forwarding behavior. The MPLS approach counts on a network of sophisticated LSRs that are able to read header information and assign data packets to specific paths like virtual circuits in a switched network.

Current efforts of the MPLS working group (also formed by the IETF) are focusing on the standardization of a base technology for using label switching and the implementation of label-switched paths over various link-level technologies, such as Frame Relay, ATM, or LAN technologies (e.g. Ethernet, Token Ring). The efforts include the development of procedures and protocols for the distribution of labels between routers, encapsulations, and the support for multicast communication. Furthermore, there are several considerations to use MPLS for Differentiated Services [FWD+01].

Chapter 5

Business Models

This chapter introduces, on the top 'business level' (compare to top level in Figure 1.1), the basics of business models, which include economic aspects of E-Commerce. This chapter starts with an overview of existing business models, continues with economic aspects of business models, finally leading to terminologies and problems with charging and accounting of Internet services.

5.1 Existing Business Models

The use and invention of new Internet protocol architectures (as introduced in chapter 4) make the evolution of new business models be only a logical consequence. It is the challenge for most companies to take advantage of these new technologies and turn them into profit-making solutions and businesses. The definition of new business models is required to formulate the way of doing business. However, there is no uniform definition of the term 'business model' although widely used in scientific publications. According to Bailer [Bai97], a business model includes and abstracts the relevantly formulated knowledge of the functionality of the enterprise, its structures, flows of information and material, products, and external relations. A business model is a map of the professional language as it is used in the business world.

According to Birkhofer [Bir01], business models can be divided into two general groups based on their level of application:

- **Enterprise level.** The focus of this group of business models is on organizational, structural, and cultural issues within an enterprise, e.g. management systems, motivation and guidance of employees. The business model is a substitute for business strategies, and examples of such business models were developed before the times of the Internet. Hence, they do not quite apply anymore in the area of E-Commerce.

- **Business level.** The focus of this group of business models is on new technologies and their possibilities to improve the core business of an enterprise, the financial returns, the competitive situation, and the customer relationship. Business models of this group translate E-Commerce strategies into operative business processes and they determine sources of income.

A simple transformation of an existing conventional business model (e.g. of the first group) into an E-Commerce environment has only small chances of success. In order to define an E-Commerce business model, three characteristic elements must be considered (according to Birkhofer [Bir01]):

1. Definition of the customer's benefit
2. Definition of the business activities within the business process
3. Definition of the sources of revenues

According to Timmers [Tim98], E-commerce over the Internet may be either complementary to traditional business or represent a whole new line of different businesses that require emerging business models. His classification of existing business models was performed into a grid along two dimensions. The first dimension gives the *degree of innovation*, the second dimension is the extent of *functional integration*, ranging from single function business models, (e.g. e-shops that only provide marketing function over the Internet), to fully integrated functionality such as value chain integration. Unlike many authors of current literature, he gives a definition of what is meant by the term 'business model'. According to him, three aspects of a business model need to be considered:

1. An architecture for the product, service and information flows, including a description of the various business actors and their roles
2. A description of the potential benefits for the various business actors
3. A description of the sources of revenues

According to Timmers, a business model in itself does not yet provide understanding of the business mission of the roles that contribute to the realization of the project. It requires a marketing strategy of the companies who take a role in such an E-commerce scenario. This implies precise statements to the companies' economical benefits and revenues and their motivation to participate in such a scenario.

The concept of roles and relationships is essential for defining business models. In a business and technical environment, roles have the advantage to logically separate the business model and define what tasks and resources are used by them and what responsibilities the different roles take. Lawrence [Law93] defines a role based on the kind of job it executes within an organization. A role can therefore inherit purely technical functions but also administrative ones. The concept of roles is also applied for the security management of business applications where different users of application are assigned to specific roles with limited functionality and access or execution rights.

Transferring this approach onto the business case of streaming Internet services, the roles and their functionality are represented by different business entities that are inherent in an application scenario as will be illustrated in section 7.1. The business model behind this scenario focuses on the cooperation and the relationships between the different roles to facilitate or even enable the provision of a video stream from the video provider to the end-customer. According to Kalakota and Whinston [KW96], a business model defines structures and processes of the business environment, both internal and external to business entities.

Another approach towards new business models for tomorrow's E-commerce services has been developed by the *Telecommunication Information Networking Architecture Consortium* (TINA-C[1]) that was founded by 40 telecommunications operators, telecommunications equipment and computer manufacturers in 1993 with the goal to improve the way services are designed and offered to the customers. TINA defines a generic business model framework to specify reference points and propagate requirements on a TINA system. A reference point specifies conformance requirements for TINA and provides an interface for different players in the industry in order to interoperate with each other [YTD$^+$97]. TINA logically separates the high level applications and the physical infrastructure from the need to directly communicate with each other. It defines a few general business roles and their relationships and mainly focuses on the reference points within TINA. The business model framework lacks a clear focus on a specific E-commerce scenario (e.g. streaming VoD service), charging and accounting issues, and the cooperation of several business roles in order to provide QoS-based end-to-end communication.

Currently, there exist different business models for service providers on the Internet that offer services to Internet users focusing on single-provider services. Service providers make money from subscription charges, advertising, E-commerce, and technical support [OS00]. Very common these days is the business model of *Application Service Providers* (ASP) that own applications (complex data analysis tools or sophisticated business software) and provide it online to a wide range of different enterprises (e.g. SAP[2]). Other business models are applied

[1] Accessible at URL: http://www.tinac.com
[2] Accessible at URL: http://www.sap.com

by telecommunication operators that offer access to the Internet for both residential users and companies as well as web hosting (e.g. sunrise[3]). Mobile telephony operators rather focus on packet-switched technology for mobile services. Other business models on the Internet offer information services, email gateways, hard disc storage, security services (e.g. Virtual Private Networks (VPN), firewalls), web-site design, server hosting, leased Internet access lines, ISDN, and many more [And00].

The growing deployment of E-Commerce applications and the convergence of data and telecommunication make the introduction of proper business models and well-defined business processes indispensable, especially for multi-provider environments. Applying business models in the technical area of telecommunication requires a set of mechanisms being available (e.g. charging and accounting functionality), especially for QoS-based end-to-end communication. An Integrated Business Model will be introduced in chapter 7 where charging and accounting functionality is applied for the provisioning of streaming Internet services.

5.2 Economical Aspects of E-Commerce

The Internet was introduced in 1969 when the *Advanced Research Projects Agency* (ARPA), a department of the American Defense Ministry, developed a network (the ARPAnet) to connect the high-level military network with university computers. The TCP/IP protocol (see section 3.3) was developed as the standard communication protocol for the ARPAnet.

In the mid 80s, the *National Science Foundation* (NSF) created the NSFNET to interconnect their super-computers and to offer other services to their users. The NSFNET adopted the TCP/IP protocol and provided the first high-speed *backbone*[4] for the Internet. After the American government allowed the Internet to be used for commercial purposes, *Mosaic*[5] was the first readily-available graphical Web browser, followed by *Netscape Navigator*[6], that enabled a global exchange of business data over the Internet, and thus, formed an optimal E-Commerce platform.

[3] Accessible at URL: http://www.sunrise.ch

[4] A backbone network has a very high bandwidth made up of a fiber-optic network, often capable of sending several hundred MBit/s. Backbone networks interconnect other networks even internationally and they do not have direct users.

[5] Mosaic was the first web browser developed by the *National Center for Supercomputing Applications* (NCSA) (accessible at URL: http://www.ncsa.edu) after Tim Berners-Lee, while employed by CERN (Conseil Européen pour la Recherche Nucléaire, accessible at URL: http://www.cern.ch), conceived the idea of hypertext as a way to link and access information on the web.

[6] Accessible at URL: http://www.netscape.com

In 1995, the NSFNET backbone was taken out of service after the NSF grant was finished. The annual expenses for the operation of the NSFNET in 1993 was estimated to be around 12 Million US dollars. Besides the expenses for the Network Operations Center to monitor traffic flows and solve network problems and the subsidies for regional networks, it was another considerable amount the NSF had to raise for the 20 Million Internet users. In the end, the expenses were simply too high for the NSF and already then it was clear that charging and accounting of Internet services would be of major interest for the global Internet services market.

The number of Internet users had been growing tremendously during the past years as well as the number of interconnected networks and the variety of applications running across the Internet. Nowadays, the Internet is accessible even for residential users, and new E-Commerce applications and services offer a wide spectrum of possibilities to do business electronically. Due to available bandwidth and powerful hardware, the idea arose to use the Internet infrastructure for an integration of data and telecommunication to entertain customers with multimedia data and services like video streams or teleconferencing.

The growing number of Internet users, applications and services but also the integration of data and telecommunication causes Internet traffic to grow accordingly, which results in too much traffic congestion on the Internet. Traffic congestion makes reliable transmission of delay-sensitive data difficult or even impossible. High-quality Internet applications or services are very susceptible to delayed delivery of data packets, since their quality strongly depends on reliable network parameters such as throughput, bandwidth, and latency.

Unfortunately, packet forwarding in today's Internet works on a best-effort basis, which results in poor quality for multiplexed audio or video streams, especially when network congestion occurs. One solution to overcome this problem is the introduction of a multi-service-level Internet, which offers different QoS levels for the data transport through the Internet. This means that Internet traffic is classified and the customers charged accordingly. Charging and accounting of Internet services affects the Internet traffic in such a way that the classified traffic of the different applications is treated fairly according to their service level and price [SBGP99]. This means that the network providers have to establish a basis for a multi-service-level Internet in which different service levels are offered and the desired QoS can be guaranteed. This would help both the network providers to cover their operating costs by collecting usage fees from their customers and the customers themselves to receive the desired QoS that they chose and paid for. IP telephony, as an example for such a multi-level Internet application, was investigated in [SFJ+99].

As a prerequisite for the implementation and realization of charging and accounting mechanisms, there has to be a business model that defines processes and phases from the initiation to the final payment of the service delivery. Our developed Integrated Business Model in chapter 7 describes on the *business level* what business entities there are, what roles and functions they have, and what business relationships are essential for performing electronic business for QoS-based end-to-end communication.

5.3 Charging and Accounting

Charging and accounting can be applied to three different domains, depending on the matter that is considered. Firstly, charging and accounting can be *product-based*, where the value of the physical product determines the price that is paid. This kind of charging is very common these days (e.g. Amazon.com[7]) and can be applied to all kinds of physical goods such as CDs, books, software, etc. Charging for physical goods within E-Commerce scenarios is well understood and will not be studied any further here.

Secondly, charging and accounting on the Internet can also be *content-based*, where the value or the content of information determines the charges. The content is independent from the number of data packets necessary to transmit the information but instead represents a non-tangible good. Examples for charging digital content include video clips, online news and magazines, or the content that is transmitted in large-sized data backups.

Thirdly, charging and accounting on the Internet can be *transport-based*, where the price for an Internet service is aligned with the way data packets are transmitted through the Internet. It is the transport service the ISPs provide for the forwarding of the data packets through their network that turns the balance for the price. While email requires a transport service that is time-independent and tolerant to network congestion, a VoD stream on the other hand is very delay-sensitive and packet forwarding needs to be within a certain time limit. On the other hand, large-sized data backups depend on a transport service that can be variable in time and delay but that must be accurate since only one bit error can make the backup unusable. Factors that are taken into consideration for a transport-based charging and accounting include the transmission period, the ordered/perceived QoS, used bandwidth (throughput), delay times, loss rates, etc.

5.3.1 Terminology

According to Stiller et al. [SFPW98], the terminology for the transport-based charging and accounting of Internet services embraces definitions for the terms *accounting*, *charging*, *pricing*, and *billing*. These terms describe different func-

[7]Accessible at URL: http://www.amazon.com

tionalities in the whole charging and accounting process but they are all related to each other and some provide input for others (see Figure 5.1).

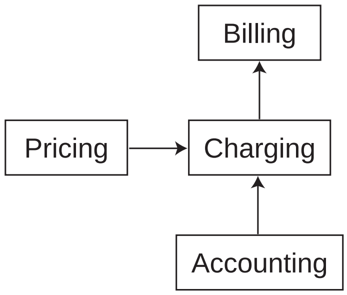

Figure 5.1: The charging model describes the correlation between the different terms and their functionalities (from [SFP98]).

Definition 5.1 (Accounting) *Accounting is defined as collecting user information for the utilization of a certain service or for the usage of network resources. This information includes technical parameters such as throughput for the service, service duration, delays, loss rates, etc. Accounting determines a function that transforms network resources into technical values that are stored in so-called* ***accounting records****. This record depends on both the underlying network infrastructure (e.g. ATM, Frame Relay) and the provided service.*

Definition 5.2 (Charging) *Charging is defined as the price calculation for an accounting record based on a pricing scheme for that service. Thus, charging determines a function that transforms technical values, stored in accounting records, into monetary values that are stored in so-called* ***charging records****.*

Definition 5.3 (Pricing) *Pricing is defined as the process of setting a price for a service. The price determines how much a user for example has to pay for transmitting one Megabyte of data in the afternoon over a distance of ten hops. This price depends on the business structure of the ISP or the current market situation.*

Definition 5.4 (Billing) *Billing is defined as the process of collecting user-specific charging records, adding them up, and generating a bill for that specific user. A bill contains the user information within a time period (e.g. one month). The bill summarizes the full cost a service user has to pay.*

5.3.2 Telephone Service

The traditional telephone service is one of the oldest examples of a telecommunication service that has considerable experience with accounting, charging, pricing and billing. The *Public Switched Telephone Network* (PSTN) provides a variety of telephone networks and services worldwide and is therefore comparable to the current Internet infrastructure. Telecommunication services such as telephony, fax, or email form the basis for E-Commerce concerning both the network infrastructure and the balancing of the charges.

Most of the network hardware of today's Internet consists of switches, routers, and connecting cables. Regional networks and backbones still support leased phone lines for interconnecting networks but the ratio of fiber-optic cable is steadily increasing. Transfer rates in the late 80s were at 1.5 MBit/s (so called *T-1 connection*), at 45 MBit/s (*T-3 connection*) in the early 90s, while currently 155 MBit/s, 622 MBit/s, and even up to several GBit/s is possible. However, data transport on the Internet is packet-switched while the telephone service is circuit-switched. One reason why packet-switched Internet technologies caught on in the Internet lies in the calculation of profitability for the purchasing and maintenance expenses for the hardware. Circuit switched networks maintain many physical wires in order to keep the delay time for the routing as short as possible. Such networks are beneficial if wires are relatively cheap compared to switches or routers. Since the 70s, switches and computers had been relatively cheaper than wires, which implicates that the expensive wires are shared among several connections. This again leads to an overload in the network nodes (switches, routers), which causes congestion and longer delay times.

The telephone service is based on an international standard with a fixed bandwidth of 64 kbit/s for digital telephony or 3.1 kHz for an analog service and a maximum delay time between the calling partners. That means that no bandwidth has to be reserved explicitly. A fixed bandwidth is reserved and fully dedicated to the telephone call even if nobody talks and the line is idle. On the other hand, the telephone call will be charged even if no packets are crossing the line.

The charges for a regular telephone call within PSTN is normally based on time ticks, where a time tick depends on the duration of the phone call, the distance to the calling partner, and the daytime when the phone call is made. This three-dimensional pricing scheme is applied by many of the national telecommunication providers (abbr.: *telco*) such as Deutsche Telekom, Swisscom, or British Telecom. Charging and accounting of mobile telecommunication services is a bit more complicated through the usage of foreign and possibly changing national networks (*roaming*). The transparency of the different tariffs and the cost for the phone call gets lost. Anonymous usage of mobile phones poses another challenge for correct charging and accounting but also to the security of users and data.

5.3.3 Charging and Accounting of Internet Services

Accounting on the Internet is not that simple due to packet-switched Internet technology. Information is fragmented into single datagrams (the average size is 200 Bytes) by the sender before transmitted over the network and then reassembled by the receiver of the information. All the datagrams of an application or a flow are independent since they carry enough information to find their way from sender to receiver (compare to section 3.3). While a one minute telephone call of a connection-oriented network requires one accounting entry, a T-3 link in a packet-switched network would require to register about 1700 single datagrams in a database at the same time.

In order to equip the datagrams with corresponding accounting and charging information, an immense overhead would be required to be attached to the datagrams, which, in return, would cause more traffic and more congestion. Accounting would therefore be more practical and less expensive if aggregates of data flows are considered just as done in the DiffServ Architecture (see section 4.3).

Another problem with accounting on the Internet is given by the fact that not single hosts, registered by their IP addresses, are to be charged for a service but the users of the hosts. This would require and expensive and complex authorization and accounting software installed on every single host in order to identify the responsible users. Problems are also given by aggressive flows that compromise the fairness for the Internet traffic through unfair methods (e.g. several parallel connections between client and server, persistent and quick executions of certain transactions). Furthermore, the combined payment of both sender and receiver of a data flow imposes problems for the accounting on the Internet while the telephone service offers the possibility for the callee to pay for the call.

There are many open questions for solving accounting problems of Internet services but from an economic point of view, why should Internet services, especially the reliable transport of datagrams, not be paid for? Why should one pay for regular telephone services while the usage of Internet resources is for free or at least based on a flat fee, independent of the kind of services or applications used and independent of the amount of datagrams sent or received? The charges for a product or a service correspond to the equivalent monetary value. Charging for Internet traffic supersedes an assignment of priorities for the datagrams and it avoids that traffic of a high-level user does not interfere with the traffic of a user that pays less. Thus, charging for the transmission of packets is a means to control Internet traffic.

The importance of the transport service determines the price an Internet user is willing to pay. That way, a high-level user will not get restricted or hindered by a user that pays less. However, this approach requires that Internet traffic will be quantified into service classes with an according price.

MacKie-Mason and Varian [MMV95] introduced a usage-based pricing scheme based on a *smart market* whereupon Internet traffic is only charged in case the Internet is congested. The basic idea behind this smart market is that a user can furnish the datagrams of an application or service with a bid that determines how much a user would pay for a congestion-free forwarding of its data packets. The actual price a user has to pay is the one with the lowest bid for the packet that is just above the tolerated limit of forwarded packets. A problem with this approach is the fact not all packets arrive at the same time and, thus, the market situation is changing constantly. Once the market price is determined, the network situation might have changed in the meantime.

5.3.4 Pricing Models for Telecommunication Services

A pricing model for transport-based telecommunication services was developed by Stiller et al. [SFPW98] that includes three basic components. Firstly, there is an *access fee* dependent on the capacity of the link. Secondly, there is a *connection fee* for the reservation of the network resources or the setup of the connection. And thirdly, transport-based pricing includes a fee for the *usage of network resources* based on time, volume, QoS, distance, bandwidth (throughput), or a combination of all of them. Additionally, a content fee can be charged separately from the transport service (see Figure 5.2).

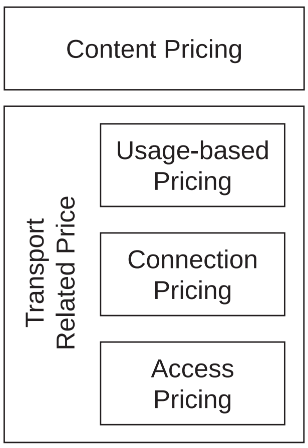

Figure 5.2: The pricing model shows different basic components of transport-based pricing (from [SFPW98]).

Apart from the three basic components, pricing could be performed as a combination of all three. Stiller et al. [SFPW98] classified pricing of telecommunication services into *flat fee* (currently applied by local ISPs for residential Internet access), *usage-based*, *reservation-based* (usually a one-time fee for reserving network resources), *volume-based* (depending on the number of data packets), *service class-based* (if different service classes are offered for an application (e.g. ATM, IntServ, DiffServ)), *bandwidth-based* (according to the provided bandwidth for the connection), *spending cap* (setting an upper limit for the price), *distance-based* (depending on the number of ISP networks or hop counts), *network-based* (traffic is preferred that stays within an ISP's network), and *connecting time-based* (as for telephone services).

Originally, the Internet was subsidized from the NSF and first used for military purposes and then for education, research and development. Pricing models for the Internet are therefore based on the fact that there is currently only one service class available that provides a transport service based on best effort. The lack of accounting mechanisms has led to simple pricing models in reality. Most ISPs charge a monthly flat fee or a connecting time-based fee depending on the time users are online. However, this is different in the USA, since local calls are for free. For many small and local ISPs, a simple monthly flat fee is not sufficient for their economic survival. More detailed pricing models for the Internet show problems since multiple ISPs are involved in a service provisioning scenario and prices can be in the range of only a few cents (*micro charges*), which makes charging and accounting very costly. Therefore, appropriate business models for charging and accounting are an emerging need for all participating business entities that allow a fair joint service provisioning for both service providers (ESPs and ISPs) and service customers.

Chapter 6

Service Level Management

This chapter discusses, on the middle 'contracting level' (compare to middle level in Figure 1.1), the basics and limitations of Service Level Management (SLM) as part of IT Service Management. This chapter is based on the documentation of the IT Infrastructure Library (ITIL), which is the only, publicly available standard to define a framework for IT Service Management consisting of modules and processes. The ITIL standard provides basics for the Service Level Management between a single service provider and a customer but it is insufficient to be applied for QoS-based service delivery in a multi-provider environment. Therefore, chapter 8 will introduce the Extended Service Level Management, as one part of the scientific contributions of this thesis, for managing QoS on the Internet. Starting with IT Service Management in general, this chapter introduces the ITIL standard and focuses on Service Level Management as the central process for realization and control of QoS-based service delivery between a service provider and its customer.

6.1 Service Management

As illustrated in chapter 4, Internet Protocol Architectures are a necessary prerequisite on the 'communication level' to make QoS-based service delivery even possible. But just the management of network resources, systems, and single network components is not sufficient for QoS-based service delivery to a customer. Neither is the plain existence of general business models (as described in chapter 5) satisfactory to solve the problem of delivering reliable end-to-end services across the Internet. *Service Management* is another approach for improving QoS-based service delivery by connecting the communication level and its technical control of infrastructure with the business level and its processes between service provider and customer.

The lack of a connecting level between the business and the communication level was realized and picked out as the central theme in many publications (see for example [You99, HS98, SMJ00]) and as a consequence of it, ‘service management’ was identified as the integral and extensive part of service delivery. The goal of IT service management was clearly defined as providing customer-oriented and optimized services. Due to missing market forces, service management strives for similar competitive mechanisms in order to fulfill business needs in a goal-oriented manner. Developing competitive forces among service providers with a clear definition of service goals and prices, service delivery becomes comparable for service customers and as a consequence more reliable.

According to Young [You99], a service provider has four reasons to establish service management:

1. Satisfying customers and providers.
2. Managing expectations.
3. Containing costs.
4. Justifying information system’s existence.

For Tritt [Tri00], customer satisfaction is the center of service delivery. For him, customer satisfaction is the result of a comparison between customer expectations of a service and its actual received performance. Low customer satisfaction results in a loss of customers, market shares, sales, and profit. Only a minority of customers complains about bad service quality, the majority changes the service provider quietly. Besides, it is five times as expensive to acquire new customers than retaining old customers. High customer satisfaction on the other hand increases market shares, sales, and profit and leads to a higher *Return On Investment* (ROI). Furthermore, satisfied customers also access other services and represent good references for interested customers.

According to Hofmann and Schmitt [HS98], problems with today’s service delivery are based on a lack of customer-orientation, low level of transparency of the service performance, sub-optimal resource usage, and the focus of technical needs instead of business needs. IT service management overcomes these problems by clearly defining service levels, matching customer needs and resource requirements, and using resources efficiently through control of network components. This leads to a higher customer-satisfaction, a better service quality, and higher productivity of service providers.

A business process incorporates several services, which, in turn, require control of several network components for its performance. Therefore, ‘service’ is the central component to connect the ‘business level’ with the ‘communication level' as illustrated in Figure 6.1. This means that service management combines *Business Process Management* (BPM)[1] and Network Management.

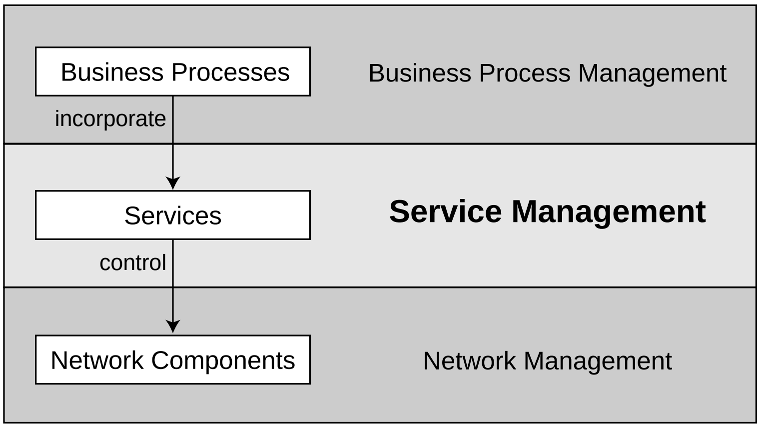

Figure 6.1: Placement of the service management between business process management and network management (adapted from [Aut00, Figure 1.6]).

Service in this context is used similarly to chapter 1 when it was introduced in form of an E-Commerce service. A service is an operating business activity of a service provider with beneficial value to a service customer and financial compensation. A service is part of a business process and requires support of resources in form of underlying network components as illustrated in Figure 6.1 (see also definitions in section 6.1.1).

Depending on the service customer, services can be delivered to an internal business unit or the service customer can be an external customer. Therefore, services can be distinguished between *internal services* and *external services*[2]. According to Piede, "*internal services are most simply defined as the services a firm requires to operate but for which there is no direct tie to revenue*" [Pie96]. Internal service customers are the employees of the different business units of a company. Internal services are designed to support internal business processes and increase their overall productivity and quality and add value to their organization

[1]Business Process Management (BPM) is a widespread and extensive research area on its own and is not further discussed herein. However, for clarification of the term, it can shortly be said that BPM comprises mechanisms to represent, organize, monitor, and control business activities and operations, information, and knowledge [vDO00].

[2]The distinction into internal and external services applies to a service provider's view. From a service customer point of view, having a service delivered by an external third party is considered as *outsourcing*. "*Outsourcing is the movement of an internal service outside the firm and the establishment of a business relationship with another company to provide the service*" [Pie96]. More on outsourcing in [Buc99, Ear96].

[Mur96]. A typical representative of an internal service provider is the business sector *IT operations* that operates the IT infrastructure of a company[3]. However, the focus of this thesis will be on external service delivery from a service provider to a third party service customer with financial compensation for the provided service.

As illustrated in Figure 6.1, service management is the interface between network management and business processes management. The central position of IT service management effects a subordination of single network components under the services layer and puts services in charge of controlling network components. One advantage is that complexity is reduced by managing tens of services as the matter of interest for the service customer instead of controlling thousands of single components. Another advantage of service management is that error messages and problem notifications are gathered and represented on this service management level.

Within the context of delivering QoS-based end-to-end services on the Internet, IT service management is limited to administrative domains of single ISPs. Since service delivery on the Internet depends on the cooperation of several independent business entities, ordinary service management between a single service provider and its customer, as described within this chapter, is not sufficient and must therefore be extended to become a key factor for reliable and satisfying service performance. Therefore, chapter 8 introduces the Extended Service Level Management as part of the scientific contributions of this thesis to enable QoS-based service delivery on the Internet.

6.1.1 Definitions

This subsection gives important definitions of terms that are used throughout this chapter. The definitions are derived from previous discussions and summarized as follows. Section 6.3 will add more specific definitions to the area of Service Level Management.

The term ‘service’ is of central importance to the thesis in general and to the field of IT service management in particular. It has been used already in practical context in chapter 1, when it was introduced in form of an E-Commerce service. However, there are different meanings of the term ‘service’, depending on the context it is used in (e.g. application service, communication service, network service, protocol service). In the context of communication protocols, a service comprises the capabilities at an interface that one protocol layer provides to an

[3]IT operations is the sector of a company responsible for the operation of IT services and network infrastructure by combining hardware and software having regard to the software life cycle [Krc00]. For tasks of IT operations see [Krc00, Hei02].

upper protocol layer [ITU94a, Sti95].

Narrow-band ISDN [ITU88] distinguishes between *bearer services*, *teleservices*, and *supplementary services*. Bearer services provide information exchange between two network nodes, teleservices support communication capabilities between end-systems, and supplementary services provide added value for teleservices [Bru00]. In the Internet context, a service usually refers to an end-to-end service from a service provider to a service customer where it is distinguished between higher level application services (e.g. email, web requests, streaming applications) and lower level network services (e.g. network configuration, control of network components).

The ITU-T defines a service generally as *"a set of functions offered to a user by an organization"* [ITU94b]. Similarly, the following definition will define service as used within this thesis.

Definition 6.1 (Service) *A service is an operating business activity of a service provider with beneficial value to a service customer and financial compensation.*

According to the IT Infrastructure Library[4] (ITIL), which will be explained in section 6.2, a service is one or more IT systems which enable a business process[5]. Consequently, ITIL defined service management as *"the management of services to meet customer's requirements"*.

Service management has also been defined by the ITU-T within the standardized framework of the *Telecommunications Management Network* (TMN) [ITU92]. The definition says that *"service management is concerned with, and responsible for, the contractual aspects of services that are being provided to customers or available to potential new customers.*

The service management layer has principally the following four roles:

1. *Customer facing and interfacing with other administrations. Customer facing provides the basic point of contact with customers for all service transactions including provision/cessation of service, accounts, quality of service, fault reporting, etc.*

2. *Interaction with service providers.*

3. *Maintaining statistical data (e.g. QoS).*

4. *Interaction between services.*

[4] Accessible at URL: http://www.itil.co.uk

[5] Definition is accessible at URL: http://www.itil.co.uk/online_ordering/itil_glossary.htm

The service management layer is responsible for all negotiations and resulting contractual agreements between a (potential) customer and the service(s) offered to this customer."

The definition is very customer-oriented and covers aspects of customer relationships and interaction based on contracts. A similar definition is given by Hofmann and Schmitt [HS98] where service management consists of the four components *Customer Care Management*, *Systems Management*, *Change Management*, and *Knowledge Management*. Customer Care Management negotiates service contracts with customers, Systems Management assures control over processes and network elements, Change Management realizes necessary adaptations for changing business and infrastructure needs, and Knowledge Management provides a knowledge base to solve customer problems as fast as possible.

The following definition as used within this thesis is a more general definition of service management while more details will be included in the definition of Service Level Management (see section 6.3).

Definition 6.2 (Service Management) *Service management describes a framework of single processes to support, plan, control, and administrate service delivery at customer satisfaction.*

This definition is very general and comprises many aspects of service delivery and support. However, service management consists of a netting of management tasks (according to ITIL) in which *Service Level Management* plays a central role. SLM is an integral part of the whole service management and is therefore further investigated in section 6.3. Furthermore, it must be said that literature does not clearly distinguish between service management and Service Level Management and both terms are used in a similar context. A more precise definition of SLM is given in section 6.3.

Previous chapters have talked about service delivery and service provisioning within the same context of providing a service to a service customer. While Brunner focuses on network capabilities when talking about service provisioning (see definition in [Bru00, page 29]), it is rather the QoS constraints that make the difference within this thesis. Therefore, the following two definitions are made accordingly.

Definition 6.3 (Service Delivery) *Service delivery describes the activity of a service provider to provide a service to a customer.*

Definition 6.4 (Service Provisioning) *Service provisioning describes service delivery under compliance with QoS requirements. This might include the reservation of network resources.*

6.2 ITIL Standard

The IT Infrastructure Library (ITIL) is a consistent and comprehensive documentation of best practice for IT service management. ITIL is an extensive framework that describes the overall IT service management based on modules and sets. The ethos behind the development of ITIL is the recognition that organizations in general and service providers in particular are becoming increasingly dependent on IT and network infrastructure in order to satisfy their corporate aims and meet the business needs of their customers. This growing dependency leads to a growing requirement for high quality IT services (e.g. streaming Internet services).

ITIL was developed by fellow employees of IT operations, suppliers, consulting experts, instructors, and national organizations[6]. The goal of ITIL is to significantly increase efficiency and effectiveness of service provisioning by using IT infrastructure in compliance with a high degree of standardization.

ITIL is the only widespread and publicly available code of practice for planning, delivering, and supporting IT services. The broad adoption of the ITIL guidance in practice within organizations and service providers and its references in numerous scientific publications have made ITIL the standard within the context of IT service management. ITIL is supported by a number of tools and there exist training programs offering formal qualifications and certificates.

ITIL comprises a layered network of different sets and modules, which describe functions of IT management and their realization in practice. The modules work autonomously but some are linked and their results influence each other. The network consists of three layers, the *strategic layer*, *tactical layer*, and *operative layer* as illustrated in Figure 6.2. The ITIL standard defines a detailed process and structural model, especially for the last two layers. An overview of the tasks of the respective layers is given in the following, according to [VFB00, Per00].

6.2.1 Strategic Layer

The *strategic layer* of the ITIL concept handles the management of IT services. Tasks of the different modules within the scope of the strategic layer essentially comprise planning, controlling, and marketing of IT services. These tasks include the planning of IT infrastructure architectures, support of software and hardware lifecycles, quality management of IT services, and maintenance of customer and supplier business relationships.

[6] ITIL is a registered trademark of the Office of Government Commerce (OGC) (accessible at URL: http://www.ogc.gov.uk), which works closely with the British Standards Institution (BSI) (accessible at URL: http://www.bsi-global.com) and the IT Service Management Forum (itSMF) (accessible at URL: http://www.itsmf.com) in setting an international standard for IT service management.

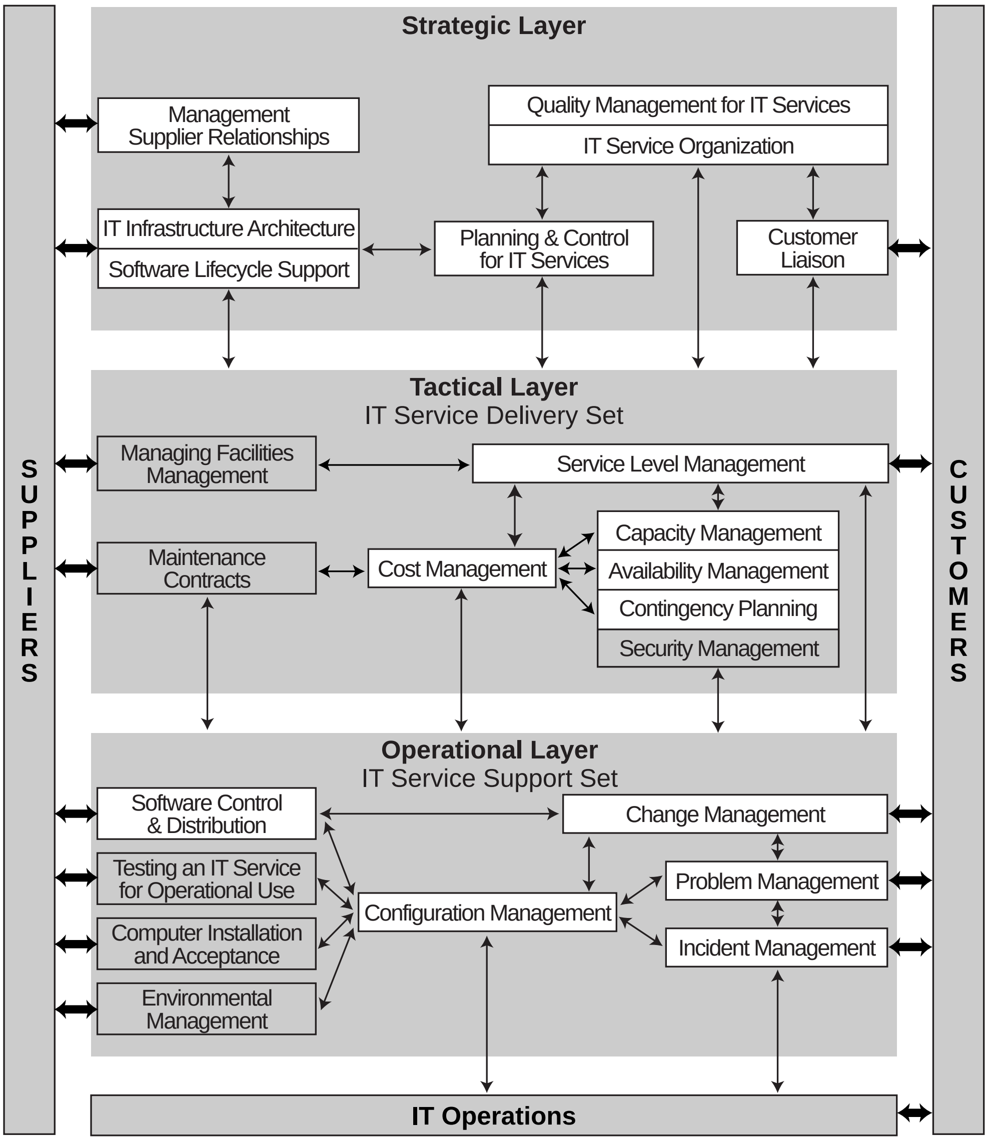

Figure 6.2: The layered concept of ITIL (from [VFB00, page 17]).

6.2.2 Tactical Layer

Within the framework of ITIL, the *tactical layer* is called ‘IT Service Delivery Set’ and comprises mainly planning and controlling processes of IT services. These tactical processes realize a detailing of prior service agreements with the customer and implement the necessary steps for a customer-oriented service provisioning. The following five processes are part of the IT service delivery set.

Service Level Management

Service Level Management is a central process within IT service management and it serves as a controlling process for the service quality in an environment of changing demands and requirements. Main tasks of the SLM comprise the definition and negotiation of service agreements between service providers and customers as well as their documentation, administration, monitoring, and management having regard to cost, service quality, and resources. SLM according to the ITIL standard is described in more detail in section 6.3.

Cost Management

As illustrated in chapter 5, charging and accounting is an integral part of a business process for delivering QoS-based Internet services. Besides charging and accounting tasks, the cost management process embraces costing, cost allocation, financial budgeting, controlling, and statistical evaluation of business ratios as its core tasks within the IT service delivery set.

Capacity Management

The primary goal of the capacity management process is to reliably fulfill customer demands with the aid of IT resources regarding transaction control, throughput, and response time of the service. This requires to calculate, plan, reserve, allocate, and monitor IT resources in time and at low cost. A well-working capacity management avoids an overload of IT resources and therefore shortcomings or failures of the service provisioning process. Together with the availability management and contingency planning process, capacity management is responsible for the usage of IT resources to realize QoS-based service provisioning.

Availability Management

Availability means that a service can be used by a customer whenever he/she desires. High availability requires low error rate. If a problem occurs, it must be solved as quickly as possible. Availability management makes sure that IT resources and services provide highest benefit to the customer.

Contingency Planning

Contingency planning includes measures to restore the IT service provisioning process in case of a total collapse of systems or environment. A contingency plan contains clear statements how and when the corresponding measures come into operation. Contingency planning improves the availability of services, reduces the risk of a collapse by preventive steps, and provides measures to restore IT resources and services in case of a collapse. Figure 6.3 shows the ITIL Service Delivery Set with correlations of the five previously described processes.

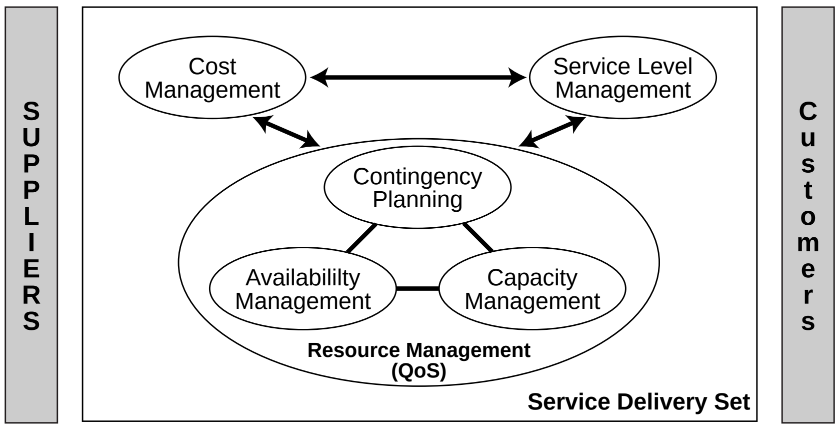

Figure 6.3: The correlation of modules within the ITIL Service Delivery Set (from [VFB00, page 18]).

6.2.3 Operative Layer

The *operative layer* within the ITIL framework supports the delivery of IT services and is therefore called ‘IT Service Support Set’. The modules within this set take tasks to support IT services on an operational level during service operation. The way how processes are implemented in order to support the IT service delivery set is described by the following five modules.

Change Management

Change management provides measures to securely check, control, and monitor changes of operational infrastructure. Change requests usually result from solving occurred problems, installation and upgrading of system components, or adaptation of system software to new or changed technological or business conditions. The task to make changes without aftereffects and wrong decisions is very critical for the performance of IT services.

Configuration Management

Since service delivery is dependent on IT resources and their operation, it is important to know these resources in detail including their features and correlation with other components. The configuration management takes care of an efficient and effective administration and control of used IT assets. By storing current and exact information about available configuration units in a database (e.g. version, status, owner), the configuration management is indispensable for successful IT service management.

Incident Management

The incident management represents a reactive process within customer-oriented service delivery. The process implements a transformation of customer requests and needs, and reports about service failure or shortcomings into documented support requests. Its main goal is to facilitate the recovery of service delivery as fast as possible. Incident management is often implemented in form of a help desk as the interface between IT Operations and customer.

Problem Management

The problem management is responsible for trouble shooting. In case the problems could not be solved by the incident management, it is the task of the problem management to find and implement a solution in order to quickly reestablish service delivery under normal conditions. By means of preventive measures, problem management makes sure that occurring problems have no or only little influence on the service delivery process.

Software Control and Distribution

Software control and distribution (also called *release management*) assures successful planning and realization of hardware and software installations and their maintenance. A special focus is on the protection of systems and services by using formal methods and checks and by avoiding incompatibilities in cooperating hardware and software. Figure 6.4 shows the ITIL Service Support Set with the five interacting processes as previously described.

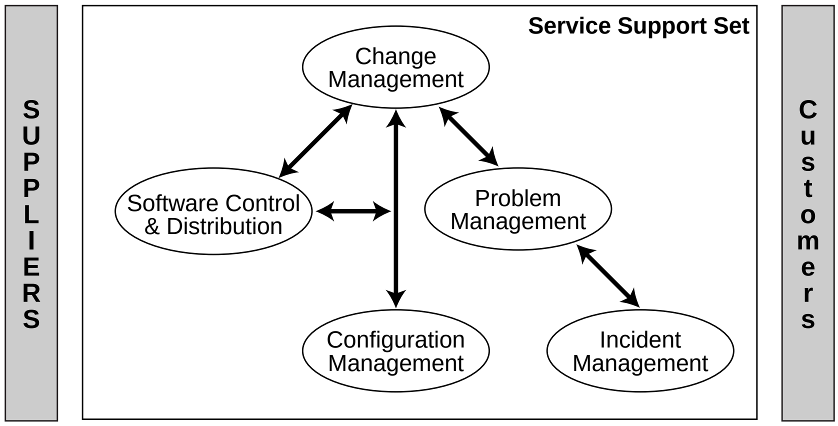

Figure 6.4: The correlation of modules within the ITIL Service Support Set (from [VFB00, page 18]).

6.3 Service Level Management

As introduced in the previous section through the ITIL concept, Service Level Management plays a central role for the service provisioning. The importance of SLM emerges of the fact that both technical components and business cases are necessary prerequisites for service delivery but yet not sufficient to satisfy customer needs and problems [Buc00].

According to Berry et al. [BZP90], "*customers are the sole judge of service quality. Customers assess service by comparing the service they receive (perceptions) with the service they desire (expectations). A company can achieve a strong reputation for quality service only when it consistently meets customer service expectations.*" Through numerous qualitative studies, the authors generated a set of five dimensions which have been consistently ranked by customers to be most important for service quality, regardless of service industry. These dimensions are defined as follows:

1. **Tangibles.** The appearance of physical facilities, equipment, personnel, and communication materials.

2. **Reliability.** The ability to perform the promised service reliably and accurately.

3. **Responsiveness.** The willingness to help customers and to provide prompt service.

4. **Assurance.** Knowledge and courtesy of employees and their ability to convey trust and confidence.

5. **Empathy.** The provision of caring, individualized attention the firm provides its customers.

A negative customer perception arises from insufficient cultural anchorage and enforcement of customer- and service-oriented service delivery on the part of the service provider. So far, service provisioning was based on a best effort service with partly existing but meaningless service guarantees. The service concept and the attitude towards reliable service delivery is very often vague and untrustworthy, which results in bad reputation for QoS-based service delivery and low motivation to provide or make use of service guarantees. According to Berry et al. [BZP90], a common failing in service organizations is the management's failure to properly define and reinforce the role of service for service provisioning. The result is *service role ambiguity*, the concept of service is vague and non-credible. Potential causes of service role ambiguity are manifold and include the following [BZP90, page 30]:

- *No service standards* exist, which results in a decreasing credibility about the importance of service.
- *Too many service standards* exist, which are sometimes even contradictory, and thus, avoid to focus on the most essential service priorities.
- Only *general service standards* exist, which only provide basic understanding and guidance on controlling and measuring service performance.
- *Poorly communicated service standards* make existing standards a guessing game for both service providers and their customers.
- *Service standards unconnected to performance measurement, appraisal, and reward systems* provide no motivation and urge for compliance or improvement of the service performance.

For a service customer to evaluate service quality, it is the deviation between expected and perceived service quality that is decisive. The guidance of customer expectations is therefore a central task of Service Level Management. In order to avoid service ambiguities, the following measures can be taken (compare with [Kot00, page 439] or [BP97]):

- **Coordination of service customer expectations and service provider perception.** The service provider should always be aware of customer needs and their service expectations since they provide a frame of reference for their assessment of the service.

- **Specification of service quality.** In order to avoid misunderstandings and disappointments about the service performance on both service provider and customer side, it is inevitable to specify service quality before the service is delivered.

- **Guarantee for service delivery.** Specifying service quality levels is of no use and not authentic if no service guarantees are offered to the customer. This includes measurement of the service performance and punishment in case of shortcomings or service failure.

- **Transparent communication of service quality.** In order to avoid unrealistic and unrealizable customer expectations concerning service quality, transparent communication between service provider and customer must be established.

- **Capture customers' words.** Since service customers judge service quality, it is important for the improvement of the service quality to take customers' words seriously, solve their problems immediately, and capture their words to improve service quality.

Service Level Management provides means for both service provider and service customer to negotiate, explicitly formulate, and adhere to agreements concerning service quality. Furthermore, these agreements are reached in advance and they include steps that are taken in case of service shortcomings or service failure. Service Level Management allows service providers to deliver customer-oriented, clearly defined, and fairly priced services based on necessary service requirements. Therefore, quintessences of Service Level Management cover the following aspects:

- **Contractual Agreements.** Contractual agreements are required to stipulate service conditions and parameters. Service provider and customer jointly define the content of the service and its quality by means of specifying service parameters. This offers service providers the possibility to effectively provide the service quality that is desired by the customer. On the other hand, it is a means for service customers to express their expectations of that service and thus avoid speculations about the service quality. Through mutually stipulated contractual agreements, it can be clearly distinguished between mandatory service requirements and ones that are nice to have.

- **QoS Guarantees and Fulfillment.** Contractual agreements support the service provider to be taken seriously and trustfully in providing QoS guarantees. Such a contract is a symbol of trust and confidence in the service providers QoS guarantees. In order to determine the degree of performance, and hence, the fulfillment of the service contract, it is inevitable to stipulate appropriate, comprehensible, and most important, measurable service metrics. Furthermore, the service contract needs to contain regulations about steps that are taken in case of service failure or service shortcomings. This supports closure of the credibility gap between service provider and customer for any QoS guarantees.

- **Contracting and Service Level Negotiation.** Before a service is performed, service levels need to be negotiated between service provider and customer and finally stipulated in a service contract. This requires a contracting process beforehand whereupon the service customer requests the desired service quality in form of service requirements and whereupon the service provider acknowledges the request or rejects it. In case of an acknowledgment, the service contract is stipulated between service provider and customer, in case of a rejection, negotiation can be restarted with another service provider or with different service requirements.

- **Monitoring.** A core issue of successful Service Level Management is the monitoring of the service performance in order to decide about compliance or non-compliance with the service contract. This implies a mechanism of transparent and periodic reporting in order to keep the service customer informed about service performance and eventual problems. Monitoring presumes the definition of measurable service metrics within the service contract that had been stipulated within the contracting phase. Due to pre-defined service metrics, the service provider knows the relevant service criteria that decide about success or failure of the service delivery. It therefore helps the service provider to focus on a systematic management of network resources in order to be compliant with the service contract. Again, transparent and detailed reporting increases trust with the customer and credibility for QoS guarantees.

- **Pricing.** The fact that network resources may be reserved in order to provide a QoS-based service implicates an adequate financial compensation for the service provider. The price for the service delivery corresponds to the service level that is provided to the customer. This is beneficial for both service provider and customer since pricing is based on operating expenses, which again determine the service performance.

- **Motivation and Attitude.** Due to clearly defined service levels and QoS specifications, service delivery is based on pre-defined service quality. This will lower unrealistic customer expectations and increase service provider performance and thus minimize discrepancies. This will improve motivation for the service provider to satisfy customer needs and it will change the attitude of service customers to take service guarantees seriously and trustfully.

- **Simplicity.** The effort for the introduction and maintenance of the Service Level Management should not exceed its benefit. Therefore, one key success factor for a successful implementation of a SLM concept lies in a highly automated and simple administration of contractual agreements. Keep it as short and simple as possible! There are supporting tools on the market that provide technical support for successful implementation of the Service Level Management (see section 6.3.3).

Figure 6.5 illustrates the quintessences of Service Level Management and places SLM in the center of business activities between service provider and service customer. The surrounding components of SLM represent the previously described quintessences that are part of the whole SLM concept connecting service provider and service customer. The two bold right arrows in Figure 6.5 symbolize that the service provider offers a service with a certain quality based on resources at a certain cost whereas the service customer receives that same service at a certain price.

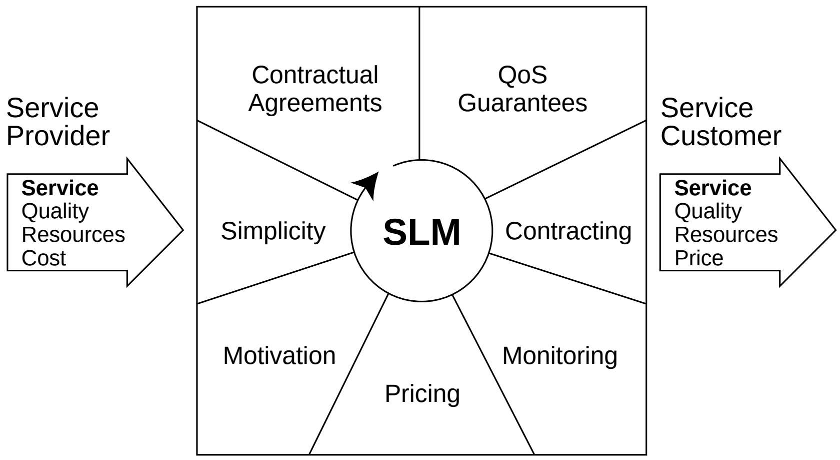

Figure 6.5: SLM concept connecting service provider and service customer.

6.3.1 Definitions

Before the Service Level Management concept can be further illustrated on the basis of a Service Level Management process, it is required to introduce and define a few central terms that will be used throughout this chapter. The scope of the definitions is limited exclusively to the context of Service Level Management. The definitions are based on the ITIL standard[7], as well as [VFB00, Hei02, NO97].

Based on previous definitions of service and service management (see section 6.1.1) and the introduction of the ITIL standard (see section 6.2), it can be proceeded to focus on the central term of Service Level Management. For Niessen et al. [NO97, page 9], *"Service Level Management is the process of managing a delivered IT service, in terms of quality, quantity, and cost."* As defined in the ITIL standard, they also see Service Level Management as the key role within the IT service management process, while emphasizing its connecting functionality between service provider and service customer (compare to the ITIL concept of Figure 6.2). For them, *"the service level management process can be seen as the 'bridge' between customer and supplier, which:*

- *integrates the disparate elements that make up service provision,*
- *packages them into an easily used and understood service, and*
- *expresses that service in terms the customer can understand, i.e. in business terms."*

Within the ITIL standard itself, Service Level Management is defined in a similar manner, as *"the process of defining, agreeing, documenting and managing the levels of customer IT service, that are required and cost justified."*

Both definitions leave out the fact that the management of service levels is based on contractual agreements between the business entities. Vogt et al. [VFB00] stress the fact that written service contracts are required to make IT service delivery be based on. They comprehend the meaning of Service Level Management as the process, which regulates quality and quantity of an offered IT service based on a written agreement or contract between service provider and customer. Service Level Management is seen as the interface between the operational resources and the formulated customer needs. It provides techniques in order to coordinate, specify, agree, monitor, and improve the required quality of the delivered service. According to these previous definitions, Service Level Management will be defined as follows:

Definition 6.5 (Service Level Management) *Service Level Management (SLM) describes the process of specifying, negotiating, agreeing, and monitoring quality and quantity of service levels in form of service parameters based on a fair price and contractual agreements between service provider and customer.*

[7]Definitions are accessible at URL: http://www.itil.co.uk/online_ordering/itil_glossary.htm

This definition contains the notion of *service level*, which is also part of the term 'Service Level Management' itself. According to the definition of ITIL, a service level is *"the expression of an aspect of a service in definitive and quantifiable terms*." This definition points out the fact that a service must be quantifiable in order to be managed and delivered definitively. This leads to the following definition:

Definition 6.6 (Service Level) *A service level describes a quantifiable degree of service performance based on the specification of service parameters (QoS) including the price.*

The Service Level Management process includes the stipulation of the service level in a written contractual agreement as previously mentioned. Such an agreement is called a *Service Level Agreement* that documents agreed service levels for an IT service (compare with ITIL, [NO97], and [VFB00]).

Definition 6.7 (Service Level Agreement) *A Service Level Agreement (SLA) is a written contract between a service provider and its customer that specifies and documents the service levels in terms of content, quantity, and quality of service (QoS) including the price.*

As mentioned in section 6.1, it can be distinguished between internal and external services. In case of internal services, there are forms of contracts that help IT operations operate services. An internal agreement that guarantees service support for IT operations is referred to as *Operational Level Agreement* [NO97, page 76]. A similar contract with an external supplier is referred to as *underpinning contract* [NO97, page 76].

In case of external service delivery on the Internet, as it is the focus of this thesis, the terminology and scope of these definitions has been widened in order to be applied in this context. Service provisioning on the Internet is dependent on the cooperation of several service providers, which represent 'external' service providers. The outcome of this is the following definition in contrast to Service Level Agreement.

Definition 6.8 (Operational Level Agreement) *An Operational Level Agreement (OLA) is a written underpinning contract between two service providers that stipulates the supply of additional services to operate and fulfill the Service Level Agreement. In that case, one service provider becomes a customer of the other.*

The DiffServ Architecture (as described in section 4.3) uses SLAs to describe a "*service contract between a customer and a service provider that specifies the* *forwarding service a customer should receive"* [BBC+98]. In particular, the detailed technical parameter specification is considered as being part of a second container, which, in case of DiffServ, is termed *Service Level Specification* (SLS). It mainly includes those parameter specifications, which cover values and traffic data within a dedicated DiffServ domain. The *Traffic Conditioning Specification* (TCS) forms an inherent part of a SLS and defines the detailed classifier rules and traffic profile values [Gro02]. Therefore, a SLS can be considered as a DiffServ OLA.

As a starting point for SLA negotiations, Service Level Requirements are specified by the service customer as formulated demands on the service level [VFB00, page 201]. Furthermore, these Service Level Requirements need to be translated into Operational Level Requirements as equivalent demands on supporting external service provider. Accordingly, the following two definitions are made:

Definition 6.9 (Service Level Requirements) *Service Level Requirements (SLR) form the starting basis in the process of service level negotiation where the service customer makes demands on the service level regarding content, quantity, and quality of service (QoS).*

Definition 6.10 (Operational Level Requirements) *Operational Level Requirements (OLR) represent a means between two service providers to describe the technical demands for supporting a service as specified within the SLR, i.e. regarding content, quantity, and quality of service (QoS).*

In order to specify SLR/OLR and subsequently monitor compliance with the agreed SLA/OLA, service metrics and indicators are required as measurable quantities that provide a basis for evaluation of the service quality.

Definition 6.11 (Metric) *A metric is a quantifiable measure used to calculate the performance or efficiency of a process, system, or network component.*

Definition 6.12 (Service Indicators) *As opposed to direct measures for characteristics, Service Indicators (SI) are a set of pre-defined metrics used to determine both the required and the actual provided service level (performance). It thus provides an indirect measure of quality for the provided service.*

Definition 6.13 (Service Quality) *The service quality determines the degree of accordance between the measured service level (performance) and the stipulated service level. It thus represents the difference between actual value and target value of the service level.*

6.3.2 Service Level Management Process

As defined previously, Service Level Management describes the process of specifying, negotiating, agreeing, and monitoring quality and quantity of service levels based on a price and contractual agreements between service provider and customer. Similar to managing any other project, the approach to introduce and implement the Service Level Management process (SLM process) includes activities from defining the process over to executing and controlling it. This also requires to make both service providers and customers be aware of their IT infrastructure and service requirements and be confident of the service levels that had been specified and that will be provided accordingly, independent of the number of service providers that contribute to provision the service.

According to the ITIL standard, the whole SLM process can be characterized by the four components 'goal', 'results', 'scope', and 'activities' (see [NO97, pages 17-30]). Service Level Management is an integral part of IT service management. The *goal* of the SLM process is to deliver a cost effective service whose quality and quantity is specified beforehand and whose actual performance at least meets or better exceeds customer expectations.

In order to pursue the goals, it is absolutely necessary to specify and quantify the targeted *results* of the SLM process. Typical quantified results are the following:

- All the offered services are catalogued.
- All the catalogued services are quantified by metrics that are understandable by both service provider and customer.
- Both internal and external target values for the service levels are defined and agreed.
- Agreed service levels are achieved.

The *scope* of the SLM process concerns the management of IT services between:

- the customer organization and the service provider
- the service provider and its external suppliers
- the service provider and its internal departments

Activities comprise actions that are performed within the SLM process depending on the stage the process is in. According to Niessen et al. [NO97], the whole SLM process can be subdivided into three phases, a *definition phase, execution* *phase*, and a *controlling phase*. The phases describe a quality cycle for continuous improvement of the SLM process in order to pursue the goal with the specified results within the respective scope. Figure 6.6 shows the phases of the cycle and activities that describe the SLM process.

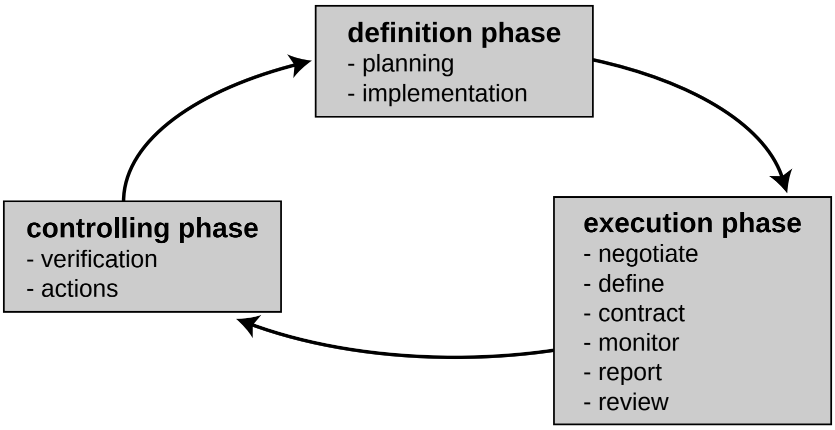

Figure 6.6: Three phases and activities that describe the SLM process (adapted from [NO97, page 20]).

In the definition phase, the SLM process must be planned by defining roles, tasks, and responsibilities, creating awareness, quantifying service levels, human and network resources, identifying risks, structuring SLAs and OLAs, creating project plans and time tables, etc. The planning activities result in an implementation plan which includes procedures for realizing targeted goals. In order to achieve the implementation in a timely and effective manner, it is necessary to treat the introduction of SLM as a project.

Actions performed within the execution phase comprise the negotiation of targets and service levels between service provider and customer and the definition of Service Level Requirements. The negotiation results in a service contract in form of a SLA or OLA respectively. Monitoring of service performance can only be performed properly if service metrics were clearly defined and measured from a customer's perspective. Service reports must be generated regularly to show the measured service achievements against the agreed service targets. Service levels must be reviewed on a regular basis in order to confirm conformance with the service contract or to spot service shortcomings or breaches.

The final controlling phase takes actions either in a *proactive* or *reactive* way. Proactive control anticipates future problems and deficiencies, while reactive control takes measures upon problems and shortcomings signaled from the help desk or the monitoring activities. Steps of the controlling phase have effects on the definition phase in order to constantly improve the quality and customer satisfaction of the whole SLM process.

The SLM process illustrates a procedure to introduce SLM as a concept within a service providing organization in order to establish cost-effective and quantity-based service delivery that meets or exceeds customer's expectation. However, circumstances are different for the provisioning of real-time services over the Internet, which requires a wider scope and adjusted activities for the implementation of the SLM concept. As a solution to this, an Extended Service Level Management with a refinement of the SLM process will be introduced in chapter 8.

6.3.3 Supporting Tools for Service Level Management

Successful Service Level Management requires supporting tools for monitoring service levels and for determining compliance with the service contracts. Supporting tools also generate regular reports for the communication between service provider and customer.

Monitoring and measurement of service performance is an essential process in many respects. First of all, it is a matter of whether a service provider keeps its promises regarding the guaranteed service levels or not. Then, in order to compare services of different service providers it is of fundamental importance to measure service performance and compare prices. After all, services and processes can be controlled only if they are measured. Finally, reporting measured values that are above a predefined threshold help to improve the business relationship between service provider and customer due to increasing mutual trust. The most important metric is the customer's experience!

Supporting tools for the SLM help service providers to keep track of the status of network components, processes, and services. These tools measure stability, efficiency, performance, availability, and/or responsiveness of applications and services. The range of currently available products is endless and out of scope, but nevertheless, a few basic features of the tools and their underlying concepts shall be introduced in the following. Well-designed Service Level Management tools should offer the following key characteristics:

- **Detecting.** Detect network congestion or services that are not operational, experience slow response times, or do not meet service level objectives.

- **Comprehensive.** Monitor critical business applications on heterogeneous systems (e.g. legacy mainframes, client/server systems), websites, e-commerce transactions, etc.

- **End-to-end.** In order to monitor customer perception, monitoring should be performed from the service providing system all the way to the customer host. The metric that truly matters is the customer experience.

- **Real-time.** Monitoring in real-time enables a rapid detection and early diagnosis of problems as they occur.

- **Proactive.** A consistent sampling of availability and response time allows to recognize and resolve problems before they impact the service quality. Some tools provide audible alarms to warn of eventual service breaches or SLA/OLA violations.

- **Realistic.** Monitoring the provided service from the perspective of a service customer is far more realistic than measuring single system or network components and processes. They all contribute to the overall service quality but what counts is a realistic view of the provided service.

- **Non-intrusive.** Monitoring should be performed completely transparent without impacting activities of service customers or modifying applications in any way.

- **Visible.** Some monitoring tools provide the feature to display a real-time picture of exactly how business processes, network components, and services are performing.

Among the currently available SLM supporting tools, there are four different concepts that can be differentiated according to [Ser98, Com99, But99, Ruh01]. However, commercial tools can hardly be applied in heterogeneous environments and most of them only measure application response times. Furthermore, the tools were designed for supporting the internal SLM process, i.e. monitoring business processes within an enterprise. These concepts are not implemented as such for the provisioning of services over the Internet due to the number of entities involved in the whole SLM process.

- **Application Instrumentation.** By instrumenting the source code in applications, a service provider can determine the exact start and end of business transactions by measuring the retention period of data packets in network and system components (also referred to as *Application Response Monitoring* (ARM)). An evaluation of these measurement data allows precise conclusions of the total round-trip response time of the application. However, this concept is highly intrusive, costly, and difficult because it requires modifications to the application source code as well as maintenance of these modifications. In the meantime, current developments also allow external ARM.

- **Network X-Ray.** This classical analyzer approach (also referred to as network *sniffer*) is based on evaluating probes spread out at strategic locations across the network to interpret data streams on a protocol level. It reads packet headers and calculates response times as seen from the probe point but does not capture response time from the end-user perspective. Therefore, this concept mainly serves to make network traffic transparent but not for an end-to-end service monitoring.

- **Client Capture.** Client capture is a very promising concept since every desktop is equipped with an intelligent agent that monitors real transactions and then captures the response time of applications and services. This concept is application independent and complements network and systems management solutions. On the other hand, it generates high amounts of extra traffic and causes legal problems (especially in Germany) concerning data protection due to installation of monitoring modules in customer desktops.

- **Capture/Playback.** Another promising concept is that of Capture/Playback (also referred to as *Client Simulation*), where application response times are measured from a dedicated measuring station. This measuring station simulates user keystrokes in order to measure response times of applications and services. This approach does not measure the actual end-customer's response time experience but potential application and service performance of a simulated transaction. It approximates end-customer experience automatically, can be implemented quickly, and is very flexible.

6.4 Summary

Within part two (i.e. chapters 3 to 6), basics were provided that help to understand how data is communicated and services are delivered over the Internet. Furthermore, this part introduced and discussed mechanisms and approaches on different levels of abstraction to support service delivery over the Internet. However, all these approaches do not enable QoS-based service delivery independent from each other. Internet Protocol Architectures lack standards, business models are too generic, and Service Level Management is too limited to be applied to service delivery on the Internet. Therefore, the following part three designs and implements our holistic approach consisting of an Integrated Business Model and an Extended Service Level Management to enable the provisioning of streaming Internet services.

Part III

Scientific Contributions

Chapter 7

Integrated Business Model for Streaming Internet Services

This chapter describes, on the top 'business level' (compare to top level in Figure 1.1), an Integrated Business Model for streaming Internet services. The Integrated Business Model represents one part of the scientific contributions of this thesis since it is developed for a multi-provider E-Commerce scenario with a focus on charging and accounting of the data transport. Starting from an E-Commerce scenario of a streaming video transmission within a general network model, the chapter introduces our developed role model for that scenario with different roles and their business relationships. Furthermore, we designed a service model that illustrates the single services within such a scenario in conjunction with the roles that perform the services. Finally, we developed a business process model, which demonstrates the tasks that the roles perform within different phases in order to deliver a QoS-based streaming Internet service.

7.1 E-Commerce Scenario and Network Model

As seen in chapter 5, there are current approaches to define generic business models for the E-Commerce but they are lacking a clear focus on a specific scenario with a well-defined business process and charging and accounting aspects. Other more specific business models only focus on the business relationship between service provider and service customer. However, service provisioning over the Internet involves a number of different and independent service providers that need to cooperate in order to deliver an E-Commerce service to the customer, especially for QoS-based end-to-end communication.

The Integrated Business Model includes the course of business events and reflects the processes within an economic system. The economic system introduced in this section describes an scenario consisting of a business relationship between a customer and a merchant doing business over the Internet. The classic E-Commerce scenario is the one where the customer orders products from the merchant by means of Internet communication. The products offered by the merchants include *physical goods* that could also be purchased in stores, e.g. books, cars, CDs.

Nowadays, use of *digital content* and *Internet services* is getting more and more popular with the Internet users. Digital contents offered by the merchants comprise digital information as non-tangible goods in form of bits and bytes. Digital contents include for example the information of an online book or a (news) magazine, digitized songs of a music CD in form of mp3 files or digitized moving images of a video file. The number of Internet services offered these days by the merchants is huge and comprises different categories. Of special interest for the performance of the service delivery are streaming real-time services such as Video-on-Demand applications or video conferences.

From a business point of view, customer and merchant on the Internet form a simple business relationship. However, since customer and merchant communicate over the Internet the customer can be seen as an *end-customer*, to stress the fact that it is an end point of a communication connection, and the merchant that offers goods in form of products, digital contents, and services can generally be modeled by the role of an *E-Commerce Service Provider* (ESP) (see Figure 7.1). The advantage of such an E-Commerce business relationship is the fact that information can be made available to all the end-customers in the world by means of a web browser and the communication between business entities is quick and simple (e.g. email).

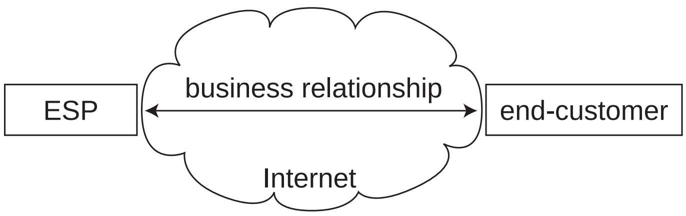

Figure 7.1: An E-Commerce business relationship between an end-customer and an ESP over the Internet.

The basic idea of an E-Commerce business relationship between an end-customer and an ESP is relatively simple and well understood if products are purchased with the help of Internet communication. Such a scenario includes the process of exchanging information to keep the end-customer informed about the offered products of the ESP. And in case of a purchasing interest, some minor issues (e.g. shipping, payment) are negotiated between the two before the product is finally purchased and money is exchanged between the two entities.

However, problems for an economic system arise with the introduction of performance and QoS requirements for the delivery of Internet services and digital content. As illustrated in chapter 2, the transmission of real-time data such as audio or video streams makes high demands on the communication of the single packets from sender to receiver. The delivery of real-time streaming information under compliance with QoS requirements (also referred to as *service provisioning*) necessitates of both ESP and end-customer to make sure to be able to provide and receive the service according to the QoS specifications. But since the Internet is used as the underlying networking infrastructure, several other business entities are also involved in providing services between end-customers and ESPs.

The Internet is composed of several independent and inter-connected networks (or domains) whereas every single domain is administrated and operated by an *Internet Service Provider* (ISP). Since the networks of ISPs form the path between an end-customer and an ESP, ISPs are also referred to as *intermediaries*. The difficulty with QoS-based end-to-end communication is the transport of single data packets across the Internet under strict QoS requirements (e.g. delay, jitter, etc.). Therefore, the crucial part of the communication is taken by the ISPs that bear the brunt of the work.

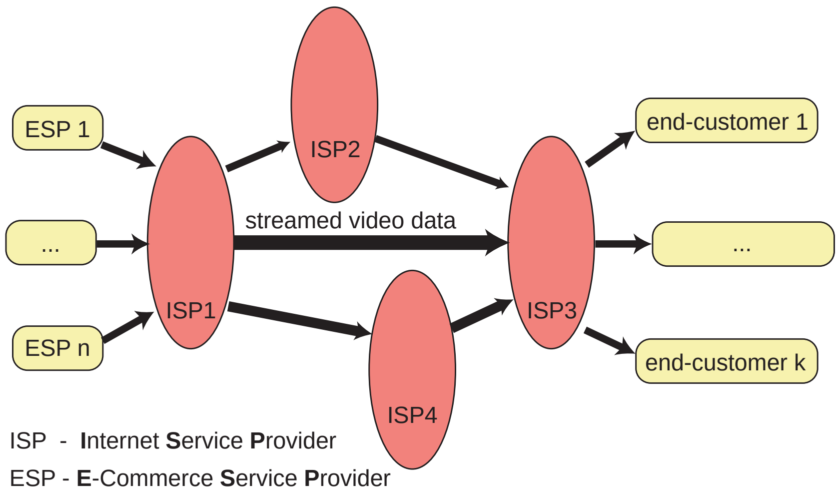

Figure 7.2: A general network model of a streaming E-Commerce scenario (adapted from [KSHS02]).

The network model in Figure 7.2 is an extension of Figure 7.1 and it shows a general network model of an E-Commerce scenario where a streaming service (streamed video data) is delivered between an ESP and an end-customer across several ISPs. The model interconnects different ISPs, ESPs, and various end-customers. Abstracted from any business roles, that will be described in the next section, an E-Commerce scenario can be generally defined as follows:

Definition 7.1 (E-Commerce scenario) *An E-Commerce scenario is an economic system of interconnected and cooperative service providers with the goal to provide services to service customers via the Internet.*

From a *business level* point of view, it is the end-customer that communicates only with the ESP in order to negotiate service conditions and QoS parameters. But from a *communication level* point of view, there are network connections that need to be established between two neighboring entities in order to make a communication path from end-customer to ESP. It is the cooperation of many entities, the widespread business environment, and the competition among business entities that make the whole Integrated Business Model a challenge for charging and accounting on the Internet.

7.2 Business Roles and Relationships

As known from the Public Switched Telephone Network (PSTN), both service provisioning and the subsequent charging and accounting tasks are implemented in a distributed manner, such that accounting tasks are performed at certain locations in the PSTN, charges are calculated, and billing information is collected and sent to customers. Based on a variety of pricing models for these telephony services (including dynamic, static, content-based, or usage-based pricing), similar economic considerations arise for today's Internet.

Within the Internet, a quite similar distributed situation can be envisioned, but with the major difference that communication is connectionless and data is transmitted in datagrams. Furthermore, services offered for the Internet have a much broader spectrum of availability than pure telephony service and the partners are manifold in nature and existence. An Internet service is a beneficial value for a customer that is financially compensated towards the service provider. However, as illustrated in the scenario of Figure 7.2, the current network structure is such that there are many service customers who receive services and many service providers who offer services. Therefore, this different situation requires a clear identification of relevant roles in an open and global Internet with a dedicated focus on provisioning transport services.

7.2.1 Role Model

Our developed role model includes different correlated roles that are characterized by a certain functionality and scope of duties. In business and technical environments, roles have the advantage to refine the business model and specify what tasks and resources are used by them and what responsibilities the different roles take. L.G. Lawrence [Law93] defines a role as "*. . . mainly a definition of a job at the lowest level of granularity used in the organization. According to the business concerned it may be in a position description, establishment position statement or other term which addresses what must be done by the occupant of that job regardless of who that person is*". A role can inherit purely technical as well as administrative functions. Furthermore, the concept of roles is advantageous for the security, quality, and resource management. Securing data and managing resources can be based on role-access relationships where certain business entities are assigned to roles with restricted functionality.

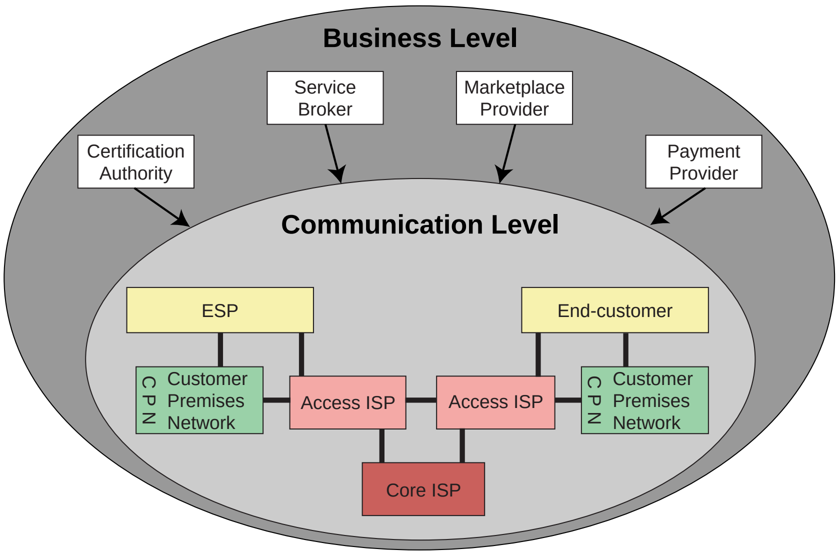

Figure 7.3: The role model shows different business roles both on a business and communication level.

Figure 7.3 further breaks down the multi-provider scenario of Figure 7.2 and shows our role model with a set of different business roles that occur both on the *business* and *communication level*. The communication level is characterized by the physical arrangement of the different roles and their relationships in the service delivery process as illustrated in Figure 7.4. It shows the interconnection of the business entities and the physical communication path from ESP to the end-customer along several intermediate ISPs (compare to bottom level in Figure 1.1).

The business level shows additional roles that have no influence on the technical process of service provisioning but on administrative issues concerning the business process (compare to the top level in Figure 1.1).

A role implies a set of functions and responsibilities that allows any business entity that holds that role to provide the right services accordingly. An instance of a role will interact and communicate with another instance of a role in order to provide a service. Additionally, one business entity can fulfill the tasks of several roles and thus adopt these roles. One example is Akamai[1] that takes the role of both ISP and ESP by providing communication networks and content.

The focus on the communication level is on the roles of the *ESP*, *ISP*, *end-customer*, and *Customer Premises Network* (CPN) knowing that these roles are not limited on the interface but on the quality of the service they provide. On the business level, there are the additional roles of a *Certification Authority* (CA) and a *Payment Provider* as instances of a *Trusted Third Party* (TTP) in order to fulfill security and payment services. Furthermore, there is a *Service Broker* and a *Marketplace Provider* that are responsible for the trading environment. There might be more roles inherent in an E-Commerce scenario [And00] but the focus will be on these roles and their relationships.

E-Commerce Service Provider

The role of an *E-Commerce Service Provider* (ESP) symbolizes a merchant within the business relationship with an end-customer. The ESP offers goods online via the World Wide Web (WWW) to every Internet user in the world and represents the seller of goods. ‘Goods’ is considered as a general term for any kind of product, content or service that is offered or sold electronically on the Internet. Therefore, the role of an E-Commerce Service Provider can be defined as follows:

Definition 7.2 (E-Commerce Service Provider) *An E-Commerce Service Provider (ESP) is a business role within an E-Commerce scenario that offers products, digital contents, and services to end-customers via the Internet.*

Special occurrences of an ESP are represented by *retailers*, *Application Service Providers* (ASP), or *Content Service Providers* (CSP). An Internet retailer buys products or contents from the holder and resells them to end-customers (e.g. Amazon[2]) or a retailer buys the license to sell a service or to show and distribute contents as performed on TV (e.g. live coverage of sports events). An ASP offers the usage of software applications it owns. The application software provider maintains and manages the software for its customers. Examples

[1] Accessible at URL: http://www.akamai.com

[2] Accessible at URL: http://www.amazon.com

of such application services are web hosting or application servers (e.g. SAP[3]). The CSP collects, organizes, and communicates information in certain specialized areas. While there are CSPs that distribute news and real-world information (e.g. CNN[4]), there are other CSPs that focus on entertainment such as audio or video (e.g. streaming media offered by Akamai).

The revenues for an ESP result from the price that ESPs charge their customers for accessing the content or receiving the service (e.g. Playboy[5] charges 36 Euro a year for the membership that allows free access to the full web content). The current trend among more and more CSPs is to charge for the web content, which was so far accessible for free on the Internet. Eventually, the content is charged depending on the provided QoS that the end-customer had chosen before the service is performed.

End-customer

The role of an end-customer symbolizes a customer at the end point of a communication connection with the ESP. The end-customer can be a residential user or an enterprise. The end-customer can get online information about the offered goods within the ESP's product or service catalog and may choose electronically by selecting certain items. The end-customer can compare prices of different ESPs in order to find the optimal product or service concerning quality and price. The end-customer purchases products, contents, or services from the ESP and represents the buyer of goods. The end-customer does not resell goods that it consumes. If possible, the end-customer chooses the QoS parameters (e.g. bandwidth of a stream) before a service is performed, which might have an influence on the price.

Definition 7.3 (End-customer) *An End-customer is a business role within an E-Commerce scenario that buys products, digital contents, or services from the ESP via the Internet.*

Internet Service Provider

The role of an *Internet Service Provider* (ISP) is divided into the roles of an *Access ISP* and *Core ISP* according to their scope of network provider duties and the range of hardware and software equipment. Access ISPs support local access to internal or external networks and provide Internet connection between end-customer, be it directly or through a *Customer Premises Network* (CPN) (see following section). An Access ISP covers the 'last mile' between the end-customer

[3]Accessible at URL: http://www.sap.com
[4]Accessible at URL: http://www.cnn.com
[5]Accessible at URL: http://www.playboy.de

and the Core ISP, utilizing copper lines, fiber line, or radio technologies (e.g. satellite or mobile communication). Examples of Access ISPs are dial-in modem providers (e.g. AOL[6]) or local telephone companies (e.g. sunrise[7]) that provide their telephone lines for Internet access. Furthermore, there are mobile and wireless Access ISPs that allow connection to the Internet (e.g. D2-Vodafone[8]). Access ISPs charge their end-customers (or CPNs) for the time being connected to their network and charges might be demanded from connected ESPs that provide services or contents to their end-customers. Thus, the Access ISPs might claim a fair share from the ESPs' business profit.

Many Core ISPs increase the reach of Access ISPs to a global extent and form the backbone of the Internet. They handle the data transport interconnecting multiple Access ISPs. There might be none or more than one instance of a Core ISP involved in a communication connection between an end-customer and an ESP. It depends on the connectivity of the ISPs and the geographical distance of end-customer and ESP. In case there is only one instance of an ISP involved between end-customer and ESP, it takes the role of both Access ISP and Core ISP. Thus, the role of Access ISP and Core ISP may be physically inherent in one business entity. Examples of Core ISPs are Bluewin[9] in Switzerland or MCI Worldcom[10] for the United States. Core ISPs maintain *peering agreements* with other Core ISPs which allow them to transport data packets for each other for no cost. However, Core ISPs may get usage-based revenues from connected Access ISPs, which represent a share for the connected end-customers and ESPs exchanging data through the Core ISPs' networks. The revenues of the Core ISPs depend on the size of the network, its location within the global network, and the amount of data it carries from other Core ISPs or for them.

Definition 7.4 (Internet Service Provider) *An Internet Service Provider (ISP) is a business role within an E-Commerce scenario that provides data transport services (Core ISP) and eventually Internet access (Access ISP) for end-customers.*

Customer Premises Network

It is also possible that end-customers are affiliated to a *Customer Premises Network* (CPN), e.g. a LAN of an enterprise or university that groups end-customers and establishes the connection to the Access ISP. A CPN may offer additional private applications or extra conditions for network services to their end-customers.

[6] Accessible at URL: http://www.aol.com
[7] Accessible at URL: http://www.sunrise.ch
[8] Accessible at URL: http://www.d2privat.de
[9] Accessible at URL: http://www.bluewin.ch
[10] Accessible at URL: http://www.worldcom.com

A CPN represents a group of users in terms of a common policy and conceals individual end-customers from the Access ISP. An end-customer can be connected to a CPN or directly to the Access ISP if the end-customer is a residential user.

Definition 7.5 (Customer Premises Network) *A Customer Premises Network (CPN) within an E-Commerce scenario is a facility of accumulated end-customers with a common policy and connection to an Access ISP.*

Service Broker

Traditionally, a broker is an individual or a company that acts as an agent on behalf of a customer to execute buy and sell orders submitted by another individual or company. The broker charges its customer a commission for its services.

On the Internet services market, a *service broker* acts as negotiator on behalf of both service provider and service customer, similar to a traditional broker. A service broker observes the marketplace and tries to match offered products, contents, or services of service providers with customer requests in an optimal and efficient way. Through its general conspectus and presence on the marketplace, the service broker can save its customers (service providers and service customers) precious money and time if it finds the perfect match for supply and demand.

Definition 7.6 (Service Broker) *A Service Broker (SB) is a business role within an E-Commerce scenario that acts as a negotiator between service provider and service customer to match supply and demand.*

Marketplace Provider

A *marketplace provider* makes an environment available to service providers and service customers for trading products, contents, and services. By bringing together large numbers of buyers and sellers, marketplaces offer service providers to meet new customers and they help service customers to expand choices for services and compare performance and price. Furthermore, electronic marketplaces can reduce transaction cost for all the business entities through automated transactions and also increase transparency and quality of offered products, contents, or services. A traditional example for trading products is ebay[11] that is well-known today and widely used.

Definition 7.7 (Marketplace Provider) *A Marketplace Provider is a business role within an E-Commerce scenario that provides an environment (i.e. a marketplace) for service providers and service customers to meet and trade products, digital contents, and services.*

[11] Accessible at URL: http://www.ebay.com

Certification Authority

A *Certification Authority* (CA) issues digital certificates for the use within an E-Commerce scenario. A digital certificate represents one possibility to guarantee the identity of a business entity since it links the public key with certain information concerning the identity of the owner of the certificate (e.g. name, address). In order for any business entity to accept such a digital certificate, it must be signed by an entity that is trustful. The CA represents such a Trusted Third Party (TTP) that testifies with its digital signature that the information on the certificate is true. The CA and the use of digital certificates is helpful for any non-trusted business relationship between entities where secure and important business data is exchanged. From a security point of view, such a certificate helps to guarantee *confidentiality*, *integrity*, *authenticity*, and *non-repudiation* of exchanged business data. More information on security in [Sch95, Kne01].

Definition 7.8 (Certification Authority) *A Certification Authority (CA) is a trusted business role within an E-Commerce scenario that provides security services for other business roles by issuing digital certificates.*

Payment Provider

Whenever products, contents, or services are financially balanced other than with cash, a *payment provider* is required to perform the settlement. A payment provider provides the payment system and the necessary infrastructure to enable money transfers between service providers and customers. An example of a payment provider is a bank or a credit card company that charges the account of the buyer and credits the amount onto the seller's account. Similar to the Certification Authority, the payment provider represents a TTP and any kind of problems and fraud would harm its reputation. The payment provider grants credits and performs payment orders in favor of its customers. Thus, the payment provider takes the guarantee and the risk for its customers financial solvency. Furthermore, the payment provider makes sure that every service provider receives the money for the provided services. However, billing and payment in multi-providers scenarios is difficult and will be further investigated as part of the business process in section 7.4.

Definition 7.9 (Payment Provider) *A Payment Provider is a trusted business role within an E-Commerce scenario that provides financial services by transferring money between service providers and service customers.*

7.2.2 Relationship Model

The relationship model in Figure 7.4 shows business relationships between entities on the communication level. The business roles on the business level provide services to all the entities on the communication level and are therefore not further included in the Figure. The arrows represent the direction of the service delivery and point from the service provider to the service customer. ESP $\rightarrow$ end-customer means that the ESP provides a service to the end-customer. The figure points out that the ISPs provide services to each other while the ESP for example only provides services (in form of products, contents, and services) to the end-customer. The end-customer only consumes services.

The variety of different kinds of services offered by the service providers is further explained by the service model in the next section. It is important to note that individual business relationships among business entities on the communication level are not trusted in equal measure. Usually, business relationships are not trusted (except for relationships with TTPs), and therefore, the exchange of data requires security measures depending on the level of trust and security risks [KZM99].

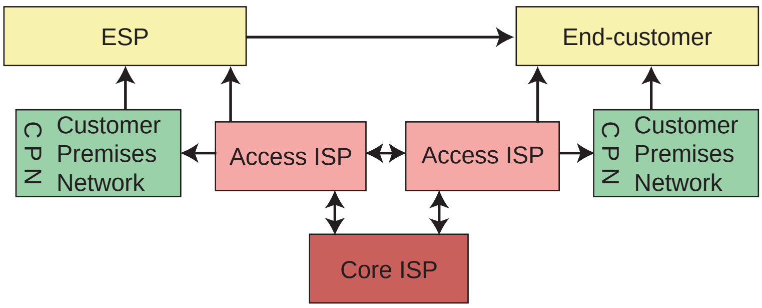

Figure 7.4: The relationship model shows business relationships between the different business entities on the communication level.

7.3 Service Model

Economic incentives for providing high-quality end-to-end communication imply the development and usage of E-Commerce services that contribute an added value to the service provisioning over IP infrastructure. Interacting and cooperating business roles are required to perform the service delivery based on predefined policies.

The previous section defined different roles, described the different relationships between them, and pointed out their functionality within the business scenario of Figure 7.2. At this stage different kinds of services are discussed, including facets that these roles can provide and add to the whole process of service provisioning.

7.3.1 Services

Figure 7.5 shows our developed service model (see also Kneer et al. [KZS01]) with the previously defined roles of the communication level, the end-customer, ESP, and ISP in form of ovals. The model abstracts from the technical solution where the end-customer is connected to an ESP via the IP infrastructure of an Access ISP. Instead, the service model is based on the business relationship between end-customer and ESP including different services that are provided on top of the IP infrastructure of the ISP. The end-customer uses the E-Commerce Service, offered by the ESP, as an interface for further value-added service provisioning. This E-Commerce service comprises the offer of products, contents, and services that are offered online by the ESP.

Figure 7.5 also implies the possibility for end-customers to directly use the IP infrastructure offered by ISPs in order to communicate over the TCP/IP-based infrastructure. This might be useful to up or download data onto or from a file server with no extra services and without time restrictions for the data transport.

Ordering an E-Commerce service from the ESP gives the end-customer a wide range of accompanying value-added services. Particularly emphasized in Figure 7.5 are the *access*, *reservation*, and *transport service* but the module 'Other Services' combines other services (e.g. security services, synchronization services, charging services) that could find their spot within the service model. These modules of different single services are offered by both ISP and ESP and can be recombined and added to increase the value of the delivered E-Commerce service. The following explains the purpose and functioning of access, reservation, and transport service in more detail.

Access Service

Internet Access is offered to end-customers to connect to the Internet via an ISP platform. Service functionality comprises among others access control, charging and billing, logging in and out. An Access ISP holds the position of the closest ISP for both end-customer and ESP, which is a special role since access conditions are individually and contractually fixed. Access conditions determine the kind of connection such as its bandwidth, availability, or security measures, but also pricing models for the Internet access. Since these conditions are fixed on a contractual base, they have to be kept otherwise reimbursements have to be settled to compensate the failure. Within Public Switched Telephone Networks, the

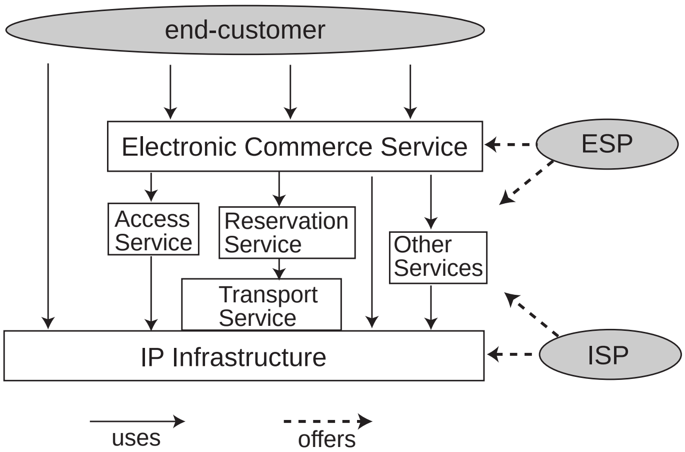

Figure 7.5: The service model including roles and services (from [KZS01]).

phone access service is charged with a monthly flat fee depending on the access technology (e.g. analog or ISDN) rather than on usage.

Recently, similar models are applied on the Internet and *Service Level Agreements* (SLA) have been introduced to stipulate the business conditions between customer and ISP[12]. The content of a SLA always depends on the service negotiated between the business entities. It usually contains customer and provider identification, the general service description, and additional service parameters such as service monitoring and control information, pricing-, charging-, and business-related information as well as legal aspects[13]. More about the content of Service Level Agreements and their practical usage in the following chapters 8 to 11.

Unlike PSTN access, Internet access implies global connectivity for the price of a local phone call to the Access ISP. This is an important difference for both residential users and also enterprises with a private network infrastructure. However, today's packet-switched Internet does not provide any QoS-based transport services as opposed to the circuit-switched PSTN with a direct connection between two end-customers and a constant transfer rate.

Transport Service

Data *transport service* provided by the role of the Internet Service Providers (both Access and Core ISPs) is of fundamental importance for the quality of the delivered communication services. Data needs to be carried reliably from one end

[12] UUNET: *Service Level Agreements*; accessible at URL: http://www2.uu.net/customer/sla

[13] Bluewin: *Leistungsbeschreibung Bluewin Internet Access Service*; accessible at URL: http://www2.bluewin.ch/services/terms/servicerange_hw_d.html

point to another. Packet forwarding on todays Internet works on a best-effort basis, which means that single packets are transmitted independently from each other from sender to receiver as fast as possible. However, transmitting video or audio frames with good quality requires a type of network connection with a certain guaranteed bandwidth or maximum delay. A video or audio stream is delay-sensitive which requires timeliness of packet delivery and allows only small variations in the transfer rate due to buffering mechanisms. The service quality of the stream depends on the delay between every transmitted frame where the frame size varies depending on the amount of changes in the moving picture. Thus, the transfer rate varies from frame to frame. A video conference on the other hand is rate-sensitive and needs to constantly transmit frames of the same size. Providing a high-quality video conference stream requires a guaranteed bandwidth with a certain constant data transport rate [DFL+98].

IP infrastructure running the TCP/IP protocol suite provides basic transport features (e.g. global connectivity or reliable byte stream channel) for best-effort traffic. However, high-quality real-time multimedia traffic requires reliable QoS packet delivery. In order for ISPs to provide reliable transport services according to the required QoS of the E-Commerce service, it is necessary, firstly, to classify traffic into different service classes and, secondly, to treat data packets differently depending on the service class and the according price. Distinction and prioritizing of special traffic could be one solution to overcome the problem of network congestion, since it treats data packets fairly according to the service level and price [SBGP99]. Delay- or rate-sensitive applications or high-quality E-Commerce services receive a better transport service and thus are more expensive than regular best-effort services. Consequently, the quality of an E-Commerce service depends on many factors, but mainly on the quality of the underlying transport service.

Reservation Service

The reservation service is provided by both Access ISP and Core ISP in order to reserve network resources for the data transport service but also by the ESP to reserve an E-Commerce service. The reservation service negotiates service conditions for the E-Commerce service and it stipulates network parameters for the transport service (including Internet access) between end-customer and ESP. Before the E-Commerce service is performed via the Internet, network resources have to be reserved within the infrastructure of one or several ISPs. The Access ISP as the closest ISP for the end-customer plays a special role in the reservation process as far as the negotiation of the transport service and the Internet access is concerned. Depending on the requirements for the E-Commerce service, the end-customer chooses the desired QoS for the transport service by selecting the appropriate QoS parameters (e.g. bandwidth, latency, loss rates), which of course affect the price. The reservation of a service is usually launched by the end-customer but can also be launched by the ESP depending on their business relationship (e.g. cost sharing for the transport service between ESP and end-customer).

One can distinguish three different kinds of models for the reservation of a service:

1. An immediate short-term reservation for one period of time (default is for example 30 seconds) where the end-customer reserves a service immediately for a period no longer than the default.

2. An immediate long-term reservation for a longer period of time where the end-customer reserves a service immediately for a period that is longer than the default.

3. A reservation in advance where the end-customer reserves a service that will be delivered in the future.

The connection between the end-customer and the ESP can be established across a network of several ISPs, where every connection (or flow) is assigned a unique identifier to distinguish different flows within the network of an ISP. As described in section 4.2, IntServ defines a networking framework, which supports unidirectional end-to-end flows [Par92]. These flows may request a certain QoS and may use a Controlled Load Service or a Guaranteed Service. Within the IntServ framework, the Resource Reservation Protocol (RSVP) is used as a signaling protocol to reserve network resources for single flows of applications. RSVP can be used to establish a network connection on a dedicated path between the end-customer and the ESP. It can also be used to negotiate service parameters and a price for the transport service. RSVP extensions were created within CATI[14] that carry accounting and charging relevant data on a per-flow basis [FSVP98].

Instead of treating a single flow as the entity of interest, the Differentiated Services Internet (DiffServ) as a different networking framework was recently developed by the IETF that handles Internet traffic based on the notion of aggregated flows and fixed numbers of service levels in terms of service profiles (see section 4.3). This approach solves the problems of scalability and overhead as it occurs with single flows but traffic traversing DiffServ domains need to be marked which is also complex and time-consuming.

Figure 7.6 gives a final overview of the previously defined roles and services and it shows the affiliation of certain roles to both service provider and service customer. The left column shows the service providers offering services to the

[14] Accessible at URL: http://www.tik.ee.ethz.ch/ cati

service customers in the top row. The ESP, for example, offers E-Commerce services including reservation to the end-customer. The Access ISP (AISP) gives Internet access to end-customer and ESP, while reservation and transport services are also offered to other ISPs to establish global connectivity. Core ISPs (CISP) do not negotiate directly with end-customers or ESPs, hence the fields are marked with a cross.

Service Provider \ Service Customer	end-customer	ESP	AISP	CISP
ESP	E-Commerce Reservation			
AISP	Internet Access Transport Reservation	Internet Access Transport Reservation	Transport Reservation	Transport Reservation
CISP			Transport Reservation	Transport Reservation

Figure 7.6: Overview of services offered from service providers to their service customers.

7.3.2 Strategies and Policies

As described implicitly by the previous sections, strategies and policies inherently form a major constituent for roles to enable a structured interaction. Note that *strategy* at this point defines the set of business rules for establishing, controlling, and operating service provisioning. This terminology applied here is addressed to the higher level, termed *business level*. The technical management of communication networks, in particular with respect to policies, determines all details on a given communication level to allow for the automated decision on, e.g. access rights, security levels, performance issues, or service quality. Therefore, a *policy* is applied within the *communication level* and it serves as a means to technically support business management strategies and business cases. Traditionally, policies have been intended to be used for network management purposes only [SL00].

Interaction between different ISPs is based on a federal approach, which means that the business entities are widely self-dependent and responsible for their service delivery. Within the approach, the Access ISP has a special role as the point of contact for the end-customer with the responsibility to technically set up an E-Commerce service. This implies a fair business relationship among service providers since reliability of their own service delivery determines the trust in a business entity. A model to describe these interactions for a given service will be defined in section 7.4 as a business process. The selection of such a business process in a given market situation determines the strategy of a particular enterprise, an Access ISP for instance.

To be able to implement such a strategy on the Internet, a set of communication means is essential, basically driven by the Internet and its protocols as well as existing management means. SLAs form such crucial technical means to implement a business process depending on an ISP strategy. They are applied between business entities in general, ISPs as well as end-customers, to negotiate business conditions on a contractual basis. They define among others a service control policy that regulates objectives and responsibilities for the service delivery under a given business process.

Different coherent phases of such a business process have to be run through in order to fulfill business conditions at every stage. Considering the economic dimensions of strategies and policies, the approach to charge for communication services is an essential constituent for a business process. Therefore, the charging of services (as described in the following business process) needs to be supported by appropriate technology [SBGP99, SGRF01].

In addition, the performance in economic terms of the service delivery between service providers shall be based on a 'win-win' strategy, which implies service delivery and financial compensation until the service has finished or failed. This implies fair business relationships among providers, since no financial losses result from service failures. The effects of such a strategy, backed by SLAs, policies, and network management functionality, will be illustrated at the protocol interface of the Internet.

It is represented by a particular business process (section 7.4), which is applied specifically onto different Internet technologies to argue that the strategy as well as instances of policy sets are independent on the underlying technology. However, it has to be noted that an instance of such a policy set will deal with technology details, such as protocol IDs, flows, code points, or QoS. Above all, the performance of the service delivery as mentioned is critical for the satisfaction of service customers.

As known, traffic handling on todays best-effort Internet is difficult and results often in poor quality service delivery. One way to solve this problem is to introduce pricing models for packet-based Internet as an economic-driven policy for traffic control on the Internet [RLS99]. This includes a variety of service priority models, auction mechanisms, or pricing models based on particular Internet technology, such as the RSVP protocol or Differentiated Services Code Points. Another way to solve this problem will be based on purely circuit-switched approaches with guaranteed resource reservations for all traffic entering the network.

Due to the fact that the commercialization of services and service provisioning has reached a major status in todays communications business, the former approach seems likely to be implemented. Therefore, charging and accounting terminology was introduced in chapter 5 and will be applied within the business process in the following section 7.4.

7.4 Business Process Model

Starting from the E-Commerce scenario and network model of section 7.1, this chapter so far presented our designed role and service model as part of the Integrated Business Model. This section introduces the business process that we have developed based on the E-Commerce scenario of Figure 7.2, where an end-customer requests an E-Commerce Service from an ESP. This includes the demand for any of the previously mentioned services. The flow of this business process is restricted to only the operative sequence of steps and phases without specifying any strategies for negotiating service parameters or taking market situations into consideration. The business process is part of the Integrated Business Model [KZDS00] and it shows the different coherent phases and their correlation with the previously identified roles and services (see Figures 7.8 and 7.9). Furthermore, these phases will be mapped onto Internet protocol architectures of best-effort Internet, IntServ, and DiffServ resulting in approaches to solve tasks of these phases.

The business process contains four different but coherent phases, the *information phase*, *reservation phase*, *service phase*, and *clearing phase*. The initial information phase is run through only once in order to introduce the business partners to each other on a contractual base for further cooperation and service provisioning. Within the reservation phase, the end-customer negotiates service parameters and reserves the desired service before the actual service is performed in the service phase. The final clearing phase serves to charge, bill, and pay for delivered services considering different payment schemes and mechanisms. After making payments either during the service or the clearing phase, the end-customer can jump back to the reservation phase and reserve a new service again. Figure 7.7 shows graphically the coherence of the different phases that form the business process.

The business process is modeled by means of a flow chart, where the clouds show sub-phases of the four coherent phases and the rhombuses symbolize 'forks in the road' that decide which way the process leads. Attached to the clouds are the roles in the gray boxes that represent the involved actors of that sub-phase (see Figures 7.8 and 7.9).

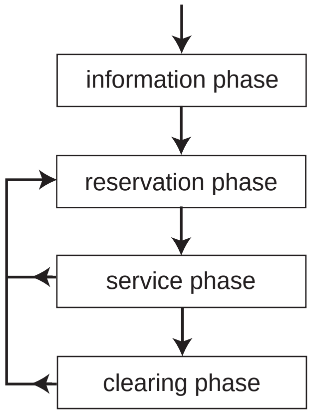

Figure 7.7: Business process model with a sequence of coherent phases (adapted from [KZDS00]).

7.4.1 Information Phase

The *information phase* serves service providers and service customers to get to know each other and to establish a business relationship based on a *skeleton agreement* as a kind of contract. In many cases, the service provider wants to know some information (e.g. name, address, email) about the customer before any services are provided. This information helps the service provider to protect itself against malicious customers, but also to distinguish between 'good' (loyal) customers and 'bad' (untrustworthy) ones. The good customers might benefit from special offers, prices, services, etc. The information phase is run through in the very beginning before service provider and service customer know each other. It does not have to result in a skeleton agreement, it can simply be run through if the end-customer just needs some information. The end-customer can remain anonymous to the service provider.

Initially, the end-customer logs in with a service provider, be it an ESP or ISP, to request a service. Login with an ESP means that the end-customer is interested in getting some information about the ESP and if satisfied, the end-customer will eventually request a product, content, or service from the ESP. A login with an ISP occurs, if the end-customer is interested in the ISP's access service if he/she wants to get access to the Internet or change his/her Access ISP. Thus, a login with an ISP serves to technically set up Internet access on the *communication level*.

If the end-customer is registered with the service provider, meaning a *contract exists* already, the information phase is skipped since it is run through only once. The *general information phase* is optional and provides general information about the service provider and the services it offers. It contains information such as a service overview, pricing information, business conditions, operational status, etc. If the end-customer wants to remain anonymous and use the service with *pre-paid payments*, the *skeleton agreement phase* can be skipped and no registration and identification with the service provider is necessary. One possibility for the service provider to protect itself against malicious end-customers is to charge for the service in advance in order to avoid at least financial damage. This can be done by requesting pre-paid payments from the end-customer. Of course, at this point no pre-paid payments are made yet but it implies either that the end-customer wants to remain anonymous and not reveal personal information or he/she is not interested in requesting a service from the service provider.

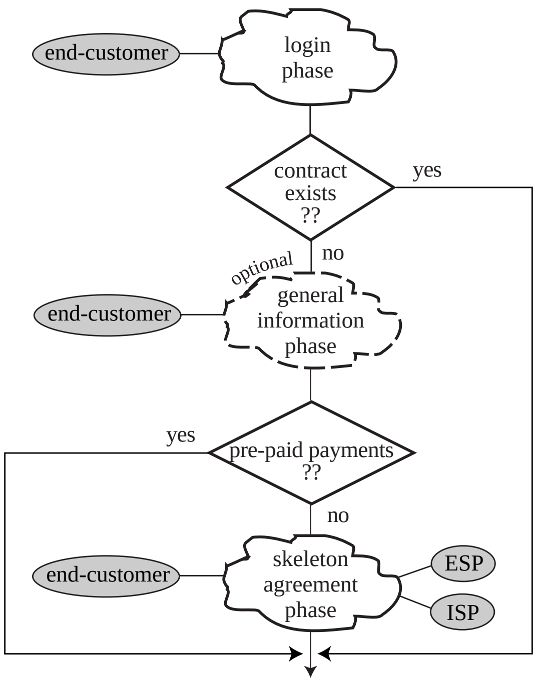

Figure 7.8: Information phase with different sub-phases preceding the phases of Figure 7.9 (from [KZS01]).

In the final *skeleton agreement phase*, both service provider and customer are involved in finalizing the skeleton agreement. General business conditions are stipulated (and signed) in such a skeleton agreement which means for the service customer that he/she agrees to the general terms and business conditions and that he/she is a registered customer to have access to certain products, contents, or services in the future. For the service provider this means that it gained another registered customer that might request some of its products, contents, or services in the future. Eventually, at this point, the customer might be charged for the registration and money will be exchanged.

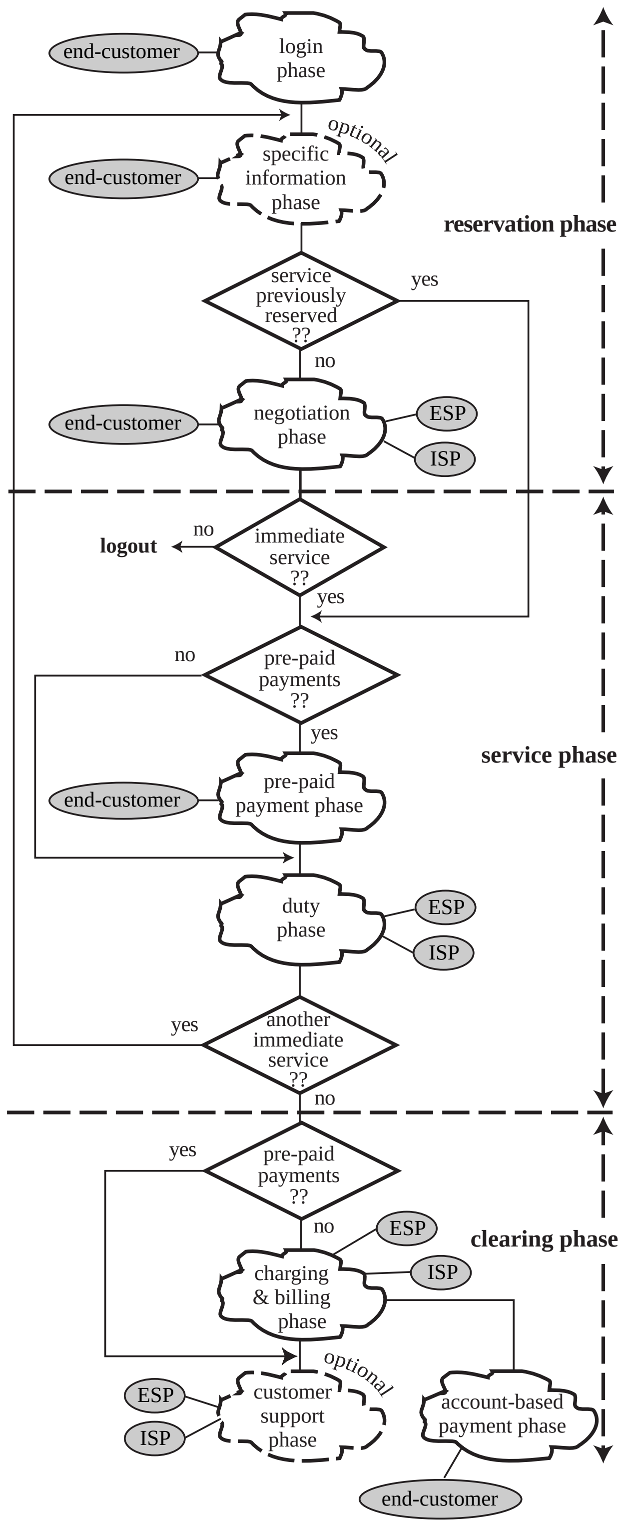

Figure 7.9: Entire business process with reservation, service, and clearing phase (from [KZS01]).

7.4.2 Reservation Phase

The *reservation phase* is run through either immediately following the information phase (Figure 7.8) or at any other time. It is the starting phase if an end-customer is interested in requesting a service. At this point, the flowchart in Figure 7.9 describes the business process of an end-customer requesting an E-Commerce service (e.g. video stream) from an ESP that also requires transport and reservation service by intermediate ISPs. Within the reservation phase, different service parameters are negotiated.

During the *login phase*, the end-customer logs in with the ESP to reserve a service. In an optional *specific information phase* the end-customer gets customer-specific information about the short-term market situation, e.g. price tender, network traffic, or comparison with other service providers. If a *service* had been *previously reserved* and the end-customer wants to receive the service immediately, the *negotiation phase* can be skipped. Otherwise, if no service had been previously reserved, the service parameters have to be negotiated among the business entities within the negotiation phase.

Within the negotiation phase, the end-customer negotiates service parameters with both ESP and all the intermediate ISPs. Service levels are defined and a certain QoS is promised by the ISPs. Considering the exemplary E-Commerce scenario in Figure 7.2, there might be several ISPs involved in the service provisioning process, each providing a transport service for the E-Commerce service between ESP and end-customer. This negotiation process is crucial for the service provisioning and for the performance of the service since the received QoS depends on the cooperation of all the ISPs and the compliance to their promises.

The negotiation process is performed in four steps and its sequence across two ISPs is illustrated in Figure 7.10. In the first step, the end-customer sends out a signal to ISP1 (the end-customer's Access ISP) to indicate a *reservation request*. This request contains information about the kind of reservation (e.g. service description, starting time, duration), and the desired QoS for the service. The request is sent across other ISPs (in Figure 7.10 this is ISP2) to the ESP, which replies with a *reservation offer*, if it accepts the end-customer's request. If one ISP or the ESP does not agree upon the reservation request, it may reject the request and send the denial back to the end-customer (dashed arrow in Figure 7.10). In that case, the negotiation starts anew.

In the second step of the negotiation process, the ESP replies to the reservation request with a reservation offer informing the end-customer on acceptance or denial of the reservation request. The reservation offer is sent the same way back to the end-customer, where the intermediate ISPs reserve the requested resources for that particular request or they add pricing information in case the reservation request was only a price request [SFJ+99]. If one entity does not accept the reservation offer, the end-customer must re-send a reservation request with different parameters.

The third and fourth step of the negotiation process (i.e. *reservation adoption* and *reservation confirmation*) are optional and can be used to add sender or receiver provided electronic payments if for example reservation fees are required or payments have to be made in advance [FSVP98]. This four-step negotiation phase has to be run through for every newly requested service or if the end-customer wants to extend the currently performed service.

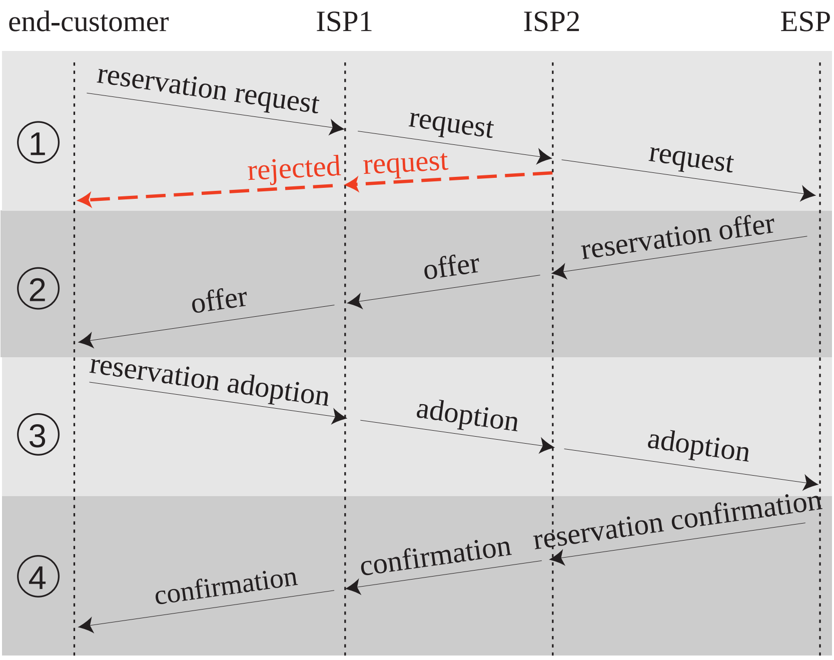

Figure 7.10: Different steps are executed within the negotiation phase (from [KZDS00]).

As mentioned in section 7.3.1, services can be reserved immediately for a short term, immediately for a longer term, or in advance. It is important to note that the negotiation phase for the Integrated Services Internet (IntServ) runs differently from the Differentiated Services Internet (DiffServ). IntServ supports the negotiation of service parameters based on single and eventually short-termed (e.g. 30 seconds) end-to-end flows whereas DiffServ traffic is based on the notion of aggregated flows with fixed numbers of service levels, which requires long-termed service negotiation (e.g. for the whole service duration).

While no reservation or negotiation is possible with the best effort Internet, IntServ and DiffServ is more suitable for possible approaches. The check whether a service was previously reserved could be performed by checking for existing or reserved flows within an ISP domain (IntServ) or by checking the DiffServ Code Point (DSCP) usage of a corresponding microflow at the border router of a DiffServ domain (DiffServ). A microflow is a single instance of an application-to-application flow of packets identified by source and destination address, protocol id, and source port. RSVP messages between end-customer and ESP could be used with IntServ for the negotiation process while negotiation within DiffServ is rather handled by *Bandwidth Brokers* (BB) of different DiffServ domains [KB00] (see also Figure 7.15).

7.4.3 Service Phase

If the business process has finished the reservation phase with a successful stipulation of service conditions, the process leads into the *service phase* (see Figure 7.9). Within the service phase, the actual performance of the requested service takes place. If the reserved service will be delivered in the future, and thus, no *immediate service* is required, the end-customer logs out. Of course, the end-customer can log out at any point during the whole business process. If an immediate service is desired by a registered end-customer the actual service is performed within the *duty phase* by both ESP and the intermediate ISPs. However, if the end-customer decided in the information phase not to register with the ESP/ISPs and instead to remain anonymous, *pre-paid payments* have to be made within the pre-paid payment phase before the service is performed.

After the service is delivered, the business process loops back into the reservation phase if *another immediate service* is requested by the end-customer. This loop can be short-termed (e.g. 30 seconds) or longer-termed (e.g. the whole service duration) depending on whether the reservation had been an *immediate short-term* or an *immediate long-term reservation*. In case of IntServ, the loop between reservation and service phase can be very short-termed if new RSVP messages are sent every 30 seconds (default). The loop is generally longer-termed in case of DiffServ.

A best effort Internet does not require payments for transport services since no resources are reserved and IP packet forwarding is performed without guarantees and priorities. IntServ allows IP packets to be marked with corresponding flow labels while DiffServ marks IP packets at the ingress of a DiffServ domain with the right DSCP that identifies the PHB within the domain. Pre-paid payments could be included within IP packets for a certain period of time (e.g. 10 minutes) with both IntServ and DiffServ. Another immediate service is possible anytime with no guarantees with the current Internet, while IntServ requires new RSVP messages (e.g. every 30 seconds) and DiffServ a similar DSCP marking scheme for packets of subsequent microflows (see also Figure 7.15).

Charging System

Charging and accounting for registered end-customers is performed within the duty phase of the service phase by both ESP and all the ISPs. The duty phase implements the process of transmitting data packets from ESP to end-customer across several ISPs. If content-based or transport-based charging is performed by ESP or ISPs, IP data packets with a certain flow label (in case of IntServ) or with a certain DSCP mark (in case of DiffServ) are accounted and charged accordingly.

A commercialized Internet requires a variety of communication services (as illustrated in section 7.3.1), which encompass technical ones as well as economic ones. While the functionality of technical services includes IP packet delivery, reliability and error control, or security features, economic services focus on feedback information from service usage to optimize the revenue of E-Commerce services. Based on the scenario of Figure 7.2, a charging approach within the service phase of the business process requires a system interface for a commercial information transfer between ESPs and ISPs in order to optimize business strategies.

Optimizing service provisioning requires accounting of services on a timely basis. Traditionally, this accounting has been performed on longer time-scales, nowadays a short-term is envisioned. This leads to the necessity to map these technically accounted information onto a financial dimension in a similar time-scale. Therefore, a charge calculation function will become the major tool for optimizing business strategies and revenues, since soft real-time charging provides important feedback for the service providers involved in the service provisioning process. This may allow for ESPs to select an optimal provider for IP services at a given time. Potentially, an in-operation communication could be switched dynamically, however, technical restrictions have to be considered closely, before practical solutions will be available.

A similar concept is applicable to all service provider roles as defined in section 7.2, where cash flow and further economic factors are to be optimized. As depicted in Figure 7.11, the *charging system* represents that component in the described service model of Figure 7.5, which performs mapping tasks from technically accounted parameters of an IP infrastructure and its usage by end-customers onto financial values required to provide feedback signals to both ESPs and ISPs.

However, technical parameters need to be completed by QoS information directed to the ESP in case a content sensitive data transport service has failed, e.g. a file backup. However, QoS information directed to the ESP is required to cancel the E-Commerce service, and the charges for it, if content sensitive data has been damaged. Detailed functionality of the charging system is discussed in the following section. The feedback functionality of the charging system represents a financial service itself provided by an ISP. Therefore, it needs to be charged as a value-added service offer to potential ESPs as well as end-customers.

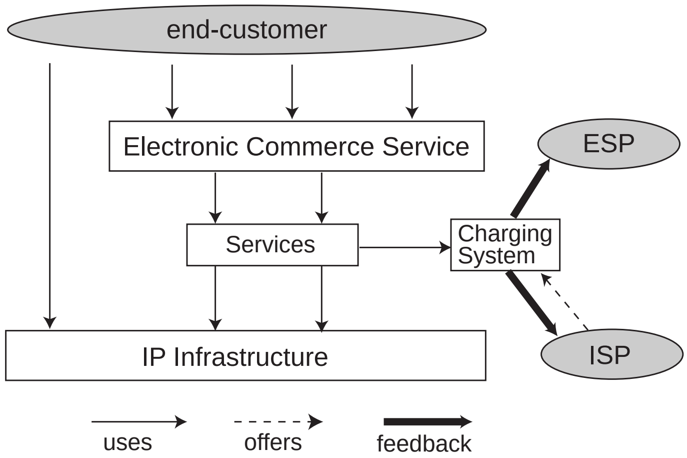

Figure 7.11: The service model of Figure 7.5 extended by the feedback mechanism of the charging system (from [KZS01]).

Charging Communications

In general, charging communications requires a clear differentiation of tasks and functions to be implemented for the network. Within the Internet, an initial step to organize those tasks has been performed [SGRF01] and they offer the necessary functionality to perform economically as well as technically driven control and optimization functions. Therefore, basic components and tasks are distinguished as follows.

Every service provider maintains a network consisting of routers and network links between them, metering functions, mediation systems, accounting systems, and a billing system. *Metering* functions can be implemented in independent devices (*passive metering*) or integrated into routers (*active metering*). In either case, they generate usage data. Due to the fact that the amount of usage data is usually too large to be sensibly processed, firstly, it must go through *mediation* where it is transformed into a form suitable for further processing. The resulting accounting information (*accounting records*) is gathered and accumulated in *accounting* systems. These accounting systems, in turn, forward all accumulated and perhaps abstracted accounting information through a *charge calculation* function towards the *billing* system interface.

This raw usage data is not yet associated with a specific user or customer. Therefore, it is necessary to identify a customer responsible for every sequence of data units sent. This is performed within *mediation*, since the amount of data leaving the mediation components can be much smaller due to the aggregation of usage data. The *charge calculation*, which receives pricing information from the *pricing* component, translates the accounting information into charging records, hence, it maps the resource-oriented information from the accounting systems into monetary values.

Within the charge calculation, discounting strategies, marketing-driven pricing schemes, or simply fixed prices can be applied. Pricing and services-related information is specified in a tariff. Finally, the *billing* system uses these values to prepare the bills to be sent to customers. The settlement with the customer billing and the subsequent payment can be embraced and modeled in the clearing phase of the business process, which basically compensates effort and expenses.[15] These charging and accounting tasks are conceptually coordinated within an overall high-level architecture and need to inter-operate as depicted in Figure 7.12.

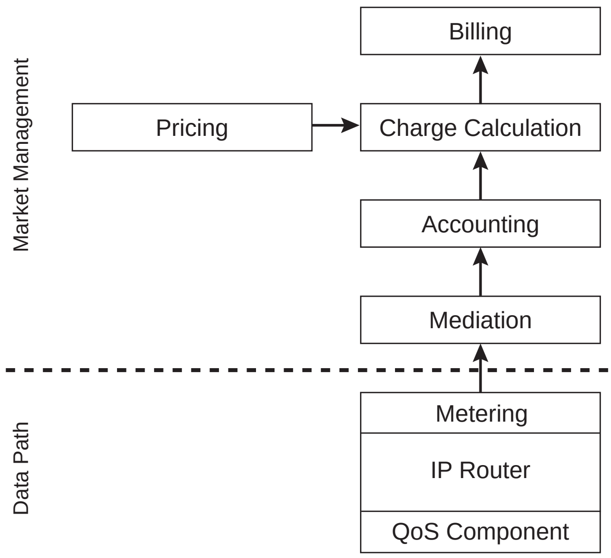

Figure 7.12: Conceptual view of charging and accounting tasks performed during the service phase (from [KZS01]).

7.4.4 Clearing Phase

The final *clearing phase* includes charging, billing, and payment for the delivered service with account-based post-paid payments if the end-customer is registered. If the end-customer is anonymous to the service providers and *pre-paid payments* had been performed in the service phase, the *charging and billing phase* is skipped and no payments are made at this point. The charging and billing phase comprises

[15] As denoted in section 7.4.1, clearing can also be done in advance, especially if the service customer is not known to the service provider.

the accumulation of user specific information (i.e. charging records) on the part of the ESP and the intermediate ISPs in order to calculate the overall price for the delivered service [KZM00].

Additionally, a bill is sent to the end-customer either solely for that service or listed with other services within a certain period of time. Depending on the billing scheme and service the bill is sent directly to the end-customer or via Access ISP or clearing center. Post-paid payments are made within the *account-based payment phase* for example with credit card, debit card, or regular money transfer. A final, optional *customer support phase* concludes the business process where the service providers have the chance to get feedback from the end-customer concerning the performance and the experienced service quality.

While a best effort Internet does not foresee any mechanisms to charge end-customers accordingly, IntServ might be suitable to charge for single flows with the same flow label. Similarly, DiffServ could count IP packets of a microflow as part of an aggregated flow with the same DSCP. This allows both IntServ and DiffServ architecture to add identified charges to a customer bill according to the clearing phase of the business process (see also Figure 7.15).

Billing

Billing can be considered as a service itself, which could be outsourced from the service providers' core businesses. However, billing implies the process of collecting all relevant information from the charge calculation function and accumulating it onto the customers bill (see Figure 7.12). The bill summarizes all charges to a final amount that the customer needs to pay. The bill is sent to the customer immediately after the service delivery or on a regular base, e.g. monthly or weekly.

As shown in our service model (see Figure 7.5), services are delivered from service providers to service customers and billed accordingly. The ESP bills the end-customer for the ordered product, the delivered service (including reservation), or the provided digital content. The ESP presents a bill, e.g. for watching a video clip, ordering a book, or reading the latest news online. It is up to the ESPs marketing strategy to determine prices and special offers for its customers, e.g. watch one video clip and get a second one free.

Access and Core ISPs bill among others for Internet Access,[16] transport, and reservation services, and especially charges for the transport service depend on the applied pricing model, including QoS value specifications. The price for these provided services may consist fully or partially of the following three components:

[16]In case of a direct dial-in connection between the ESP and the end-customer, no Internet Access service is required and the transport service is charged as a direct call from the end-customer to the ESP.

1. An access charge for the network access
2. A one-time connection charge for the connection setup
3. A usage charge based on time, volume, or QoS for the provided service

Accordingly, the charge covers used network resources and the maintenance of the network infrastructure. Usage-based pricing schemes are applied to charge the customer for the actual amount of used network resources [RLS99, SFPW98].

Nowadays, Access ISPs charge a monthly flat fee for residential Internet access, which may even be of no charge,[17] as performed on the Swiss market, and eventually include a one-time charge for the physical installation of required hardware equipment. The charge for the transport service for residential end-customers is billed according to the charge for a local phone connection to the Access ISP. It follows a three-dimensional pricing scheme, depending on (1) the time of a day, (2) the distance, and (3) the duration of a call [SFPW98]. It could even be possible that the transport charge is shared proportionately between the ESP and the end-customer, similar to collect calls where the callee pays. Again, it depends on the marketing strategy of the ESP to take over the transport charge for an ordered E-Commerce service, and thus, make offers more attractive to end-customers.

Within the Integrated Business Model, final prices for E-Commerce, Internet access, and transport service include an optional charge for their reservation. Figure 7.13 illustrates the coherence between the different kinds of services we identified and the billing for their delivery. Horizontally listed, there is the E-Commerce service on top of the Internet access service and the underlying transport service. Reservations for all these services can be made independently from each other. Billing can be performed separately for the horizontally listed services with or without charges for the reservation service.

With classical billing, the customer receives a monthly bill with a list of charges for the claimed goods. This scheme can be applied in a single-provider environment with a classic business relationship between the merchant and the customer. Different billing schemes have to be applied in a multi-provider environment (as shown in Figure 7.2) where the service performance depends on the cooperation of multiple parties.

Edge billing is a scheme applicable for billing transport and reservation services along a path of ISPs. Charges from every ISP are added to one final bill. The end-customer or the ESP (or both) pays the total amount to the Access ISP, which deducts its share and passes the rest on to the next ISP along the path. The next

[17]Note, that this type of zero charge Internet access offer is concatenated with the requirement of being a pre-selected PSTN customer of the PSTN provider, which offers Internet access.

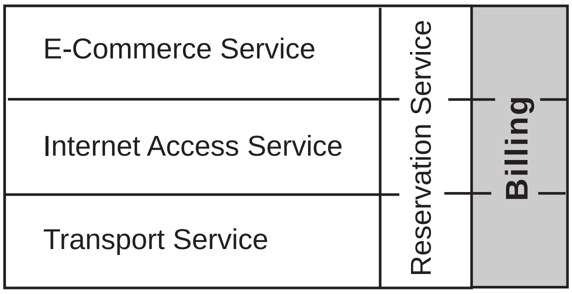

Figure 7.13: The different kinds of services and their billing (from [KZS01]).

ISP also deducts only its share and forwards the rest. This process continues along the paths until the end of network connection. Certainly, bilateral agreements between neighboring ISPs would perform better in terms of effort, but asymmetrical traffic flows and quite drastically changing traffic volumes over time require a flexible solution to short-term service setups. A *centralized billing* model consists of several bills that are passed directly from the ISPs to the end-customer, which results in a complex situation for the end-customer. A *hybrid billing* model is a combination of both edge billing and centralized billing. Other billing schemes comprise clearing centers to perform the clearing between involved parties. See [Kai98] for more details.

Payment

Billing in classical terms requires the exchange of personal data (e.g. name and address) between service customer and service provider. Sometimes it might be desired for the end-customer to remain anonymous to the service provider and still use any of its services. A good example is calling from a public phone, where the caller is unknown to any service provider along the way to the callee. Being anonymous to the service providers affects the payment process such that services have to be paid in advance.

The payment process within our process model follows the billing process and it defines a well-defined scheme how money is exchanged between service customer and service provider. A customer usually pays in advance (called *pre-paid*) if anonymous, and afterwards (called *post-paid*) if known to the service provider. Using a public phone deducts every phone tick in advance off a pre-paid phone card, whereas calling from a home phone registers the phone ticks onto the customer's monthly bill. Figure 7.14 shows our model for classifying payment mechanisms into two important dimensions depending on the possibility, whether the end-customer is anonymous to the service provider or registered.

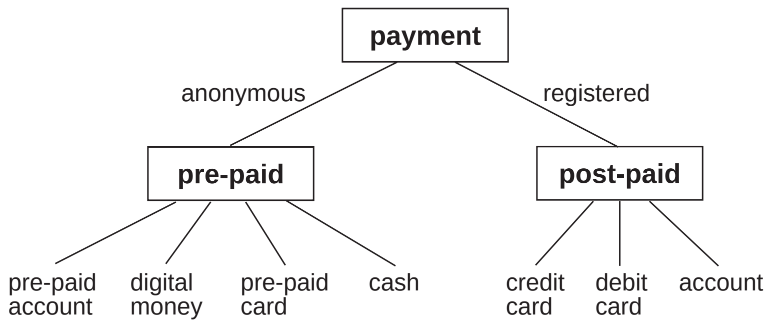

Figure 7.14: Classification of different payment mechanisms (from [KZS01]).

The tree structure of our model shows peculiarities for both *anonymous pre-paid* and *registered post-paid payments*. *Credit* and *debit cards* as well as regular *accounts* are used for conventional payments where the customer has a registered bank account with some credit institution. *Cash* and *pre-paid cards* (e.g. telephone cards) are anonymous means of payment well known today. *Digital money* is just a logical representation of monetary units and is also used anonymously, whereas *pre-paid accounts* are used with a pseudonym where the service availability depends on the current account balance.

Payments can be distinguished, whether they are performed *on-line* or *off-line*. On-line means that an authenticated connection to the payment provider (usually a bank or a credit institution) is set up during the payment process as opposed to off-line payments. On-line payments require the payer to authenticate. Authenticity can be achieved by proofing *possession* (e.g. passport, smart card), *knowledge* (e.g. password), *existence* of a unique feature of identification (e.g. biometrical password), or a combination of them.

It is also important to distinguish *micro-* and *macro-payments*[18] since the value of the bill has a strong influence on the payment mechanism. Values in the range of a few dollars and less are considered to be micro-payments, large amounts above a few dollars are called macro-payments. Not every payment mechanism can handle micro-payments due to high administration cost compared to small recovery cost from the bill [MMW96]. Further details on payment systems and their organizational design and implementation can be found in [Mer96, ZMH98].

Figure 7.15 finally illustrates what had been discussed throughout this section 7.4. Within the table, we compare a best effort Internet with the IntServ and DiffServ framework and how they could technically support tasks within some

[18]Within the SmartPay project, payment mechanisms were developed for very small amounts of correlated payments (MicPay) and uncorrelated payments of larger value (MacPay). Accessible at URL: http://cui.unige.ch/OSG/projects/SmartPay/SmartPay.html

exemplary sub-phases and decision points. It illustrates how best effort does not foresee any resource reservation or service differentiation mechanisms as opposed to IntServ and DiffServ.

		best effort	**IntServ**	**DiffServ**
reservation phase	service previously reserved	not possible	check for existing or reserved flow	check DSCP usage of microflows at border router
	negotiation phase	no negotiation	RSVP messages btw. ESP & end-customer	res. requests btw. bandwidth brokers
service phase	pre-paid payment phase	no payment since no resources are reserved	pre-paid payments within IP packets for a certain period of time (e.g., 10 minutes)	
	duty phase	IP packet routing with no priorities	IP packets with flow labeling	IP packets with DSCP marked for PHB
	another immediate service	possible anytime with no guarantee	new RSVP message (default 30 seconds)	similar DSCP marking scheme for packets of a microflow
clearing phase	charging and billing phase	no charges	charges for single flows with same flow label	charges for counted IP packets per DSCP
	account-based payment phase	no payment	adding identified charges to a customer bill	

Figure 7.15: Comparison of technical approaches for solving different tasks of the business process (from [KZS01]).

7.5 Summary

This chapter described, on the top 'business level', our developed Integrated Business Model for streaming Internet services as one part of the scientific contributions of this thesis. Integral parts of the Integrated Business Model were the role and service model to illustrate tasks and responsibilities, and the business process model with different phases for delivering QoS-based streaming Internet services.

The following chapter 8 illustrates, on the middle 'contracting level', the Extended SLM as an enhancement of the general SLM approach to support QoS on the Internet based on the stipulation of service contracts between the business entities. In analogy to the Integrated Business Model, the Extended SLM includes an extended SLM process whose phases are run through independently and in parallel to the phases of the business process. Based on the Extended SLM, chapters 9 and 10 present our developed architectures for negotiating, trading, and monitoring service contracts as an implementation of the different phases of the extended SLM process.

Chapter 8

Extended Service Level Management to Support QoS on the Internet

This chapter describes, on the middle 'contracting level' (compare to middle level in Figure 1.1), the Extended Service Level Management to support QoS on the Internet. The Extended Service Level Management represents the fundamental part of the scientific contributions of this thesis, since it extends the general SLM approach and adjusts it to the multi-provider environment given on today's Internet. As part of the Extended Service Level Management, we have refined an extended SLM process consisting of different phases for specifying, negotiating, agreeing, and monitoring service contracts. Finally, we propose a structure for the content of service contracts that form the basis for the implementation of the Extended Service Level Management. Our implementation of different phases of the extended SLM process will be introduced in the subsequent chapters 9 and 10.

8.1 Refinement of the SLM Process

The SLM process, as introduced in chapter 6, described goal, results, scope, and activities for implementing Service Level Management for service delivery between service provider and customer. The *goal* of the SLM process is to deliver a cost effective service whose quality and quantity is specified beforehand and whose actual performance at least meets or better exceeds customer expectations. In order to pursue the goals, it is absolutely necessary to specify and quantify the targeted *results* of the SLM process, e.g. the definition of specific metrics that allow to measure whether agreed service levels are achieved. The *scope* of the SLM process concerns the management of IT services between customer, service provider, internal departments and external suppliers. *Activities* comprise actions that are performed within definition, execution, and controlling phase.

While goal and results remain identical, it is the scope and its activities that need to be extended and refined in order to form an extended SLM process for QoS-based Internet service provisioning. Service contracts in form of SLAs and OLAs provide the basis for implementing the extended SLM process (see section 8.2).

The *scope* of the extended SLM process for the provisioning of real-time services over the Internet is wider than that for the delivery of internal services (as described in [NO97]). The scope focuses on the external service delivery between service provider and customer and the support by external service providers. Therefore, the scope comprises the management of services between the following business entities:

- **Service customer and service provider.** Default services and service levels are defined in a *Service Catalogue*, while specific services and service levels are negotiated between the two entities based on Service Level Requirements (SLR). The result of the negotiation is a Service Level Agreement that stipulates service levels in terms of content, quantity, and quality of service (QoS).

- **Service provider and external service provider.** On the Internet, negotiated and agreed Service Level Agreements need support from external service providers in order to deliver the service to the customer. This requires the stipulation of Operational Level Agreements (OLA) in order to guarantee and fulfill the Service Level Agreement.

An outline of the scope of the extended SLM process and its basic components is illustrated in Figure 8.1.

Activities within the extended SLM process can be refined and further subdivided into other sub-phases. Of special interest is the execution phase that can be subdivided into a *negotiation phase*, *delivery phase*, and *monitoring phase*. The negotiation phase includes the definition and negotiation of SLR and the contracting of SLAs and OLAs, the delivery phase provides service provisioning according to agreed service levels in addition to regular reporting and reviewing, and finally, the monitoring phase supervises service levels according to pre-defined metrics. The sequence of sub-phases is discontinuous through reciprocal interaction between the delivery and monitoring phase. Monitoring takes place during service provisioning and can therefore influence the delivery process proactively.

Applying the extended SLM process to the E-Commerce scenario in section 7.1 where an end-customer receives a streaming service over the Internet, the extended SLM process can be run in accordance with the business process of that

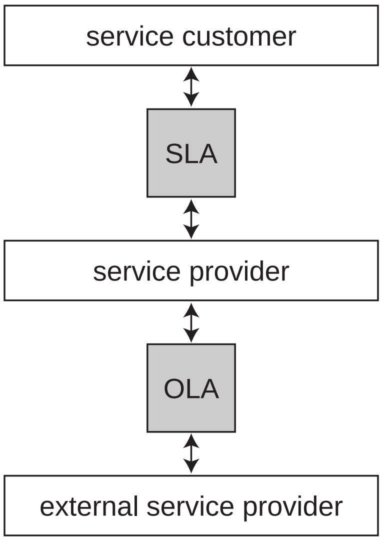

Figure 8.1: The scope of the extended SLM process with the business entities and the contractual agreements.

scenario (compare to section 7.4). The different phases of the extended SLM process can be run through independently and in parallel to the phases of the business process up to the points where activities of the extended SLM process represent the actions required in the business process. More precisely, this means that service level definition and negotiation based on SLR takes place within the reservation phase of the business process although part of the extended SLM process. In other words, the result of the negotiation phase, i.e. the contractual agreements in form of SLA and OLA, is adopted as a result of the reservation phase.

The service phase of the business process contains activities performed within the delivery and monitoring phase of the extended SLM process (e.g. service provisioning, reporting, reviewing, monitoring). While the business process executes the clearing phase, the extended SLM process takes actions to control and improve the quality of the whole extended SLM process by looping back to the definition phase. Similarly, while customers establish business connections with service providers in the information phase of the business process, the extended SLM process is defined, planned, and implemented.

Figure 8.2 outlines the mapping of business process and extended SLM process and graphically illustrates the different phases of the extended SLM process and its correlation with the phases of the business process as already depicted in Figure 7.7. As mentioned earlier, the three sub-phases of the execution phase are of special interest for this thesis and are therefore further described in the following.

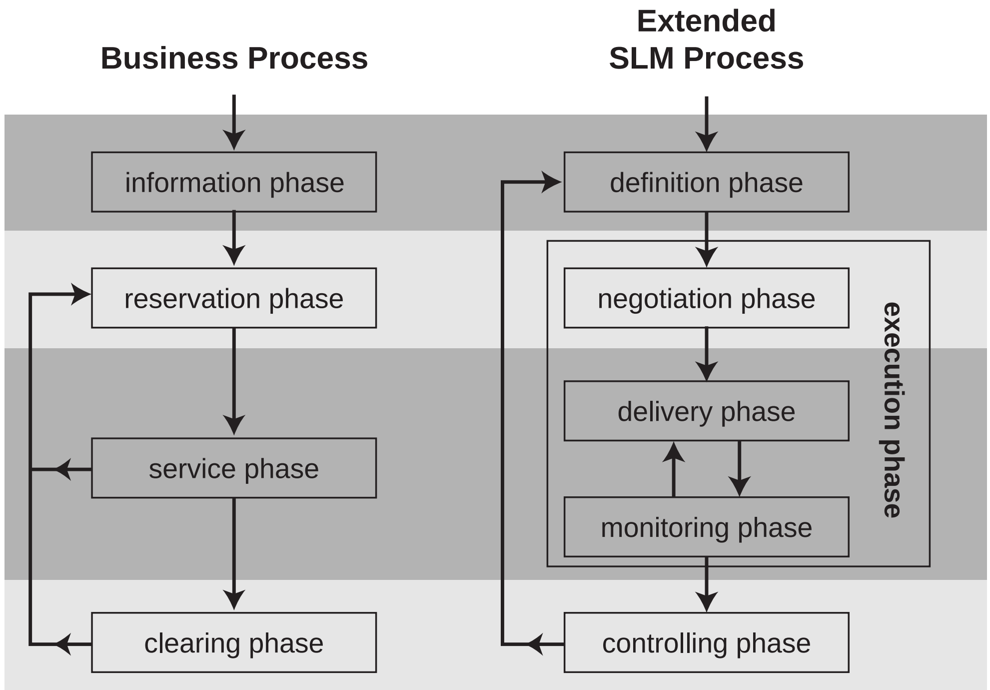

Figure 8.2: Correlation of phases of the extended SLM process with phases of the business process.

8.1.1 Negotiation Phase

Within the negotiation phase service levels are negotiated between service provider and customer. The service customer makes demands on the service provider in form of Service Level Requirements. Negotiation based on quantified service levels in a SLR allows a precise definition of customer demands based on specified metrics. The SLR also requires the specification of a price that aligns with a default value predetermined by the service provider. Furthermore, the SLR defines external targets for the service provider that must be verified first, and if confirmed with an SLA, comply with the contract. It is the service provider's task to translate the SLR into internal target values for required network infrastructure in order to reserve sufficient resources for the service provisioning. According to the ITIL standard, a *Service Quality Plan* is such a written plan with specifications of internal targets designed to guarantee the agreed service levels. For further details on the structure and functionality of a service quality plan see [NO97, pages 61-69].

After the SLR has been translated into an internal service specification in form of an OLR, the service provider can request supportive service from external service providers. It is important to note the difference between external and internal service specification. While external service specification in form of SLR is business and customer-oriented, internal service specification is technology-oriented with technical target values defined in OLR. This means for the service provider to map the requested external Service Level Requirements, that describe the service as a whole, onto atomic internal services and processes of system and network components.

The result of negotiations with a willing external service provider to provide supportive service is stipulated in a binding Operational Level Agreement. The binding agreement between service provider and service customer is represented by a Service Level Agreement as the result of successful negotiation of service levels. In case the negotiation process does not lead to a successful result in form of a SLA/OLA, the service customer can either reformulate his/her SLR or choose a different service provider. Figure 8.3 shows the sequence of a successful negotiation process focused on the performed action, the participants, and the result.

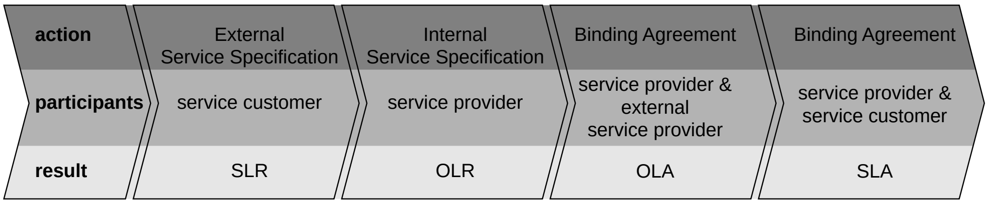

Figure 8.3: Different steps of a successful negotiation process.

Negotiation of service levels and the stipulation of service contracts in form of SLAs and OLAs reveals customer needs for certain services and products. This allows service providers to thoroughly plan service provisioning and reserve necessary resources as required. However, such a negotiation process requires of both service provider and customer to accept compromises. A higher service level automatically means more effort within the SLM process and will therefore be charged more. This also means for customers that services are charged fairly and transparently based on effectively required resources. This again motivates customers to select their required service level more carefully and consciously.

8.1.2 Delivery Phase

After successful conclusion of the negotiation phase, the SLM process leads into the delivery phase where the service provider must provision the agreed service according to the specified service levels stipulated in the SLA. In preparation of the service delivery, the service provider must execute and support the right processes in order to fulfill the required service quality.

In order to maintain a good customer relationship, it is important to keep the customer up-to-date as far as service performance and its compliance with the SLA is concerned. Therefore, regular reporting provides the necessary information for the service customer about positive or adverse variance from the target values.

Furthermore, service contracts and performance have to be reviewed regularly. Due to changing business and system circumstances, processes and systems have to be reviewed in order to fulfill SLAs and OLAs. In case of problems with the compliance of service levels, service contracts might be renegotiated with the service customer. This means to change and adapt service parameters accordingly.

8.1.3 Monitoring Phase

Monitoring compliance with the service levels and the contracts plays a key role within the SLM process. It also contributes to customer satisfaction and develops trust towards the service provider if service levels are monitored continuously. This process makes the specification of metrics even more significant. In case of violations of contractual agreements concerning the Quality of Service, immediate measures have to be taken to overcome the problem. Monitoring should be performed from a customer perspective, i.e. service levels ought to be monitored as close to the customer as possible in order to measure the service the customer actually perceives. In doing so, it is important to have customer-focused quality targets (service indicators) in form of quantifiable service metrics, since only measurable services can be monitored and controlled.

Judging the perceived service quality can be very different from monitoring values of single atomic system or network components. According to the international standard of the International Standards Organization (ISO), quality can be characterized as the collectivity of features of a unit concerning its ability to fulfill predefined requirements [ISO00]. A unit can be a product, a process, a system, an organization, or just a service. Quality in the sense of ISO is goal-oriented and therefore means completion of goals and customer requests. Since this thought is not yet established in the heads of service providers and customers, it is important to run awareness campaigns, enlighten customers, and teach employees already within the definition phase of the SLM process.

As illustrated in Figure 8.2, there is reciprocal interaction between the delivery phase and the monitoring phase. This is conditioned by the fact that results of the monitoring process are linked closely to the executing processes in the delivery phase. Furthermore, these results have to be communicated to the customer in regular intervals predefined within the SLA. A clear and consistent reporting by the service provider reflects the results of the provided service quality. It also contributes to an increase of trust between service provider and customer and by showing the results and observing the development of the service performance, the service provider becomes aware of eventually upcoming problems with the service delivery. An improvement of the perceived service quality can also be stimulated by regularly communicating the service performance and thus raise realistic estimations of the effectively perceived service quality.

As mentioned earlier, monitoring can be distinguished between proactive and reactive monitoring, which is another reason for the reciprocal interaction between delivery and monitoring phase. Proactive monitoring allows to take immediate measures concerning processes, systems, or network components within the delivery phase if service shortcomings are discovered in the monitoring phase (signaled by the arrow to the delivery phase in Figure 8.2). Reactive monitoring takes actions concerning the definition of the SLM process after service failures or shortcomings have been discovered and verified in the controlling phase (signaled by the arrow from controlling to definition phase in Figure 8.2). In the latter case, the service provider has to solve the problem as fast as possible to reestablish agreed service conditions.

As emphasized earlier, the specification of appropriate metrics within the negotiation phase is important for the success of the monitoring process and the satisfaction of the service customer. Within the scope of goal-oriented measurement in software engineering, Basili and Rombach [BR88] developed the *Goal/Question/Metric (GQM) Paradigm*, which can also be applied herein. The GQM paradigm describes a mechanism to formalize the characterization, planning, construction, analysis, learning, and feedback tasks. *"It represents a systematic approach for setting project goals (tailored to the specific needs of an organization) and defining them in an operational and tractable way"* [BR88, page 761]. Systematic appliance of this paradigm for the specification of service levels intends to specify the QoS based on as many service metrics as possible. A good metric is characterized by **SMART** criteria, i.e. **S**imple, **M**easurable, **A**chievable, **R**elevant, and **T**imebound/**T**argeted. Furthermore, there are other important properties a metric should include, such as validity, significance, sharpness, rateability, availability, or stability/reproducibility [Gli02].

Examples of typical and often specified *customer-oriented* metrics to determine and monitor service quality is given in the following (compare to Hofmann and Dülfer [HD99]):

- response time for transactions (in milliseconds)
- rate of the end-to-end communication (in milliseconds)
- duration of application, system, or infrastructure failure (in minutes)
- failure rate of applications, systems, or infrastructure (in % of the operation time)

- reliability (in number of faults per defined time period)
- existence and efficiency of contingency plans
- existence and efficiency of notification channels

Many of the specified service goals in a SLA cannot be determined by just one metric but require the definition of several partial metrics. For example, the availability of an application depends on the application process itself, the status of the host that runs the application, and finally, the capability of the network infrastructure. This example demonstrates how important it is to translate external service goals into internal service targets (Service Quality Plan) and, if required, negotiate them with external service providers based on Operational Level Requirements.

Examples of typical and often specified *technology-oriented* metrics to determine and monitor service quality is given in the following (compare to Niessen et al. [NO97, page 64]):

- CPU power (in Hertz)
- usage of CPU (in hours)
- required storage (in MByte or GByte)
- workload (in MIPS)
- network response times (in milliseconds)
- maximum number of users per shared resource
- number of tapes or cartridges

These technology-oriented metrics are not sufficient enough to determine the external provided service quality but they are a means for the service provider to evaluate its internal performance of single processes and system or network components. These technology-oriented metrics are constituents of the whole provided external service and the performance of all these single metrics contribute to the agreed service quality. If the performance of only one single component collapses, it can decrease the overall external service quality.

8.2 Content of Service Contracts

Since the demand for high-bandwidth multimedia services (e.g. streaming Internet services) is increasing steadily, the expectations for the service quality is growing in just the same measure. This means that service provisioning on today's Internet is very challenging for service providers in order to meet all the QoS requirements. These QoS requirements for services will drive service customers more and more to negotiate service parameters with their service providers.

SLAs and their underpinning OLAs are a means of establishing business relationships between business entities and serve to enable the cooperation and responsibilities between the entities according to predefined service parameters. SLAs and OLAs are negotiated and specified by both service provider and service customer before the service goes into operation. It defines the service conditions and the QoS level for the performance of the delivered service.

The concept of SLAs is fairly well understood and widely used for the Service Level Management of internal services [NO97]. A SLA represents an integral part in today's ISP industry to describe scope and guarantees of the service [Eng99]. As described in section 8.1, the scope of the SLM process for the provisioning of real-time services over the Internet comprises the management of SLAs between service customer and service provider, and the management of OLAs between service provider and external service provider (see Figure 8.1).

For the implementation of SLAs/OLAs, several aspects concerning the content, quantity, and the quality of service have to be precisely specified. For practical cases, UUNET[1] offers worldwide SLAs. For the Swiss market the SLA covers the following service parameters: guarantees on 100% network and service availability, network latency, and outage reporting, or they may show dedicated *Virtual Private Network* (VPN) security characteristics [UUN02]. According to Hofmann and Schmitt [HS98] and Intelligent Communication Software[2] (ICS) [ICS], SLAs/OLAs comprise different aspects that we divided into four sections as illustrated in Figure 8.4.

For the implementation and usage within the Extended Service Level Management, we require no fundamental difference between the structure of SLAs and OLAs. However, since SLAs are stipulated between service providers and end-customers, the language used for describing the contents of these four areas should be less technical. On the other hand, the language to define OLAs can be more detailed and specific, and contents should be precisely defined by numerical targets.

[1]UUNET (accessible at URL: http://www1.worldcom.com/uunet/), founded in 1987 and acquired by WorldCom (accessible at URL: http://www.worldcom.com) in 1996, is well known today for its comprehensive range of Internet communications solutions for business customers.

[2]Accessible at URL: http://www.ics.de

General Contractual Conditions		
Functionality and Performance	Monitoring and Reporting	Customer Support

Figure 8.4: Proposed structure of the content of SLA and OLA, which is divided into four different sections (from [KHB02]).

8.2.1 General Contractual Conditions

General contractual conditions describe a framework with general regulations for a business relationship between two entities. The following lists general conditions that need to be defined as integral parts of the contract.

- **Identification.** The first and most fundamental information a service contract should contain is both service provider and customer identification. This includes name or organization, address or business location, scope of business, email, phone, etc.

- **Contract Duration.** Every SLA/OLA should have a limit for the validity of the contract. The duration of the contract restricts the flexibility of the service and formally states the temporal availability of the service.

- **Contractual Determinations.** It has to be specified under which conditions a change of the contract can be required by one of the two contractual partners. This protects business entities from unexpected service termination due to changing environmental situations.

- **Charging and Pricing.** The delivered service generates costs that are paid by the service customer. The charging mode (i.e. flat fee, volume based, etc.) and specifications concerning the price (e.g. total price, maximum/minimum price, price per volume, etc.) must be stipulated in the SLA/OLA.

- **Payment.** The kind of payment for the service delivery needs to be stipulated. This includes different payment mechanisms (i.e. credit/debit card, bank transfer, immediate pre-paid payments, etc.), charge sharing (sender/receiver pays according to a pre-defined ratio).

- **Jurisdiction.** Each contractual partner has to sign (digitally) the SLA/OLA and certify legally the obligation and liability for the service usage or delivery. The (digital) signature is a means to avoid repudiation of liability.

8.2.2 Functionality and Performance

Service *functionality and performance* contains detailed specification of the mode of service operation as well as definitions of QoS parameters (compare to section 2.4).

- **Service Functionality.** The service has to be described entirely and precisely in detail as far as its content is concerned (QoS parameters are described in the following categories). The textual description of the service contains the service requirements to be provided by the service provider. The more accurately and transparently the service functionality is described, the clearer the expectations of the service customer and the less the disappointments during or after service operation.

- **Performance.** Service performance is determined by specifying QoS parameters. Within a SLA, this is done in customer-oriented terms (e.g. response times of transactions, service availability), whereas an OLA rather contains a technology-oriented description of service parameters[3] (e.g. bandwidth, delays, loss rates). It is the service provider to translate customer-oriented specifications (as given in a SLR) into technology-oriented ones (as given in a OLR). That means, a certain response time of an application determines how much bandwidth and network infrastructure the service provider has to supply in order to fulfill the customer's request.

- **Availability.** The SLA/OLA contains the average availability of the service for the customer. Availability defines the maximum number and duration of a possible service outage. The higher the availability of the service the higher the price for the service customer.

- **Reliability.** This aspect affects the physical quality of the transmission. The reliability is specified by a certain error rate (e.g. bit errors per hour of service operation).

- **Security.** Security aspects include confidentiality, integrity, and authenticity of data packets. Security mechanisms (i.e. cryptographic methods, passwords, public/private keys, watermarks, etc.) help to enforce secure service provisioning.

[3] As described in section 2.4, QoS parameters can be divided into system parameters and network parameters. See also Figure 2.2.

8.2.3 Monitoring and Reporting

The provisioning of the service requires *monitoring and reporting* to inform the service customer about compliance with the service contract. The section of monitoring and reporting defines customer-focused metrics and periods of notification to determine the actual delivered service quality and compliance with the SLA/OLA.

- **Service Metrics/Indicators.** SLA/OLA metrics/indicators have to be defined as well as its periodicity to measure the service quality. This helps for the transparency of service provisioning and consequently, the trustworthiness of the service provider and the business relationship with its service customer. Similar to the specification of the service performance, SLAs contain customer-oriented metrics/indicators whereas OLAs contain the corresponding technology-oriented metrics/indicators.

- **Notification.** The periodicity and detail of the reporting of monitoring results has to be stipulated between service provider and customer before the service delivery (e.g. hourly, weekly, in case of failure or shortcomings).

8.2.4 Customer Support

Customer support contains regulations concerning the support throughout the process of service provisioning. It covers actions to be undertaken by the service provider if service levels are not met as specified in the service contract.

- **Accountability.** This aspect controls the liability of the service delivery and determines the consequences for the service provider in case of service failure, service shortcomings, or even if a service delivery has been overdone. Service failure for example could result in reimbursements of service fees.

- **Recovery.** Service recovery comprises the period of time that has elapsed before a failed/interrupted service has been brought back into operation. This includes existence and capability of contingency plans.

- **Reconciliation.** This aspect includes regulations to allow reconciliation in case of disputes.

- **User Support.** A service provider has to offer information and recommendations for the service customer in case of support queries or service failure. A Help Desk for example could provide such a functionality for the service customer.

Chapter 9

Architecture for Negotiating and Trading Service Contracts

This chapter introduces an architecture that we developed for negotiating and trading service contracts between different business entities. Our implementation of the architecture was performed according to the ‘negotiation phase’ of the Extended Service Level Management presented in chapter 8. Based on a simplified scenario of Figure 7.2, we designed and specified a contracting protocol that allows negotiation and stipulation of service contracts between the involved business entities. We developed the Video-on-Demand Contracting Architecture to serve as a technological framework for the realization and implementation of the contracting protocol. Finally, we introduced a service broker that offers the possibility of trading service contracts in a marketplace.

9.1 Introduction

Section 8.1 introduced goals, results, scopes, and activities as the four components that characterize the whole extended SLM process. While goals and results are of strategic importance for the competitive position of service providers, it is the scope and mainly the activities that contribute on an operational level to a successful Service Level Management process. Activities in the extended SLM process comprise actions that are performed between service provider, service customer, and one or several external service providers that are required to support QoS-based service provisioning.

Other international bodies (e.g. ITU [ITU96], ISO [ITU97], EURESCOM [Eur99]) have contributed to develop frameworks for QoS-based service provisioning. QoS management in a multi-provider environment was also addressed in [JGE99] where a set of user-provider agreements was introduced to define QoS parameters. Existent frameworks provide concepts and terminology but practical usability in application scenarios is missing as well as a detailed description of the service level negotiation and the contracting process with respect to technical implementation.

Therefore, as described in section 8.1, we refined the execution phase of the SLM process and subdivided it into a negotiation phase, delivery phase, and monitoring phase (see Figure 8.2). The negotiation phase provides the fundament of a successful service delivery since it contains the negotiation of service levels and the stipulation of service contracts in form of a SLA between service provider and service customer and one or several OLAs between service provider and external service providers.

However, the negotiation of service levels for the provisioning of streaming services over the Internet is rather difficult since service delivery depends on the cooperation of several independent business entities as described in section 7.1. A general network model of an application scenario was shown in Figure 7.2 where a streaming service was delivered between an ESP and an end-customer across several ISPs. Successful Service Level Management depends on the reliable service delivery of every single service provider according to pre-defined service levels and the stipulation of SLAs and OLAs. This means that all the contributing business entities along the path from ESP to end-customer must be involved into the process of service level negotiation and contracting. The following describes our developed contracting protocol, which pursues this goal within the scope of the extended SLM process by specifying activities between the entities in order to achieve positive results.

9.2 Contracting Protocol

The approach to combine business entities and make them cooperate is of decentralized nature[1]. Several bilateral contracts in form of a SLA and OLAs are required between the entities to enable combined real-time service delivery. Therefore a contracting protocol was developed to show different steps of the negotiation and contracting process. The single steps of the contracting protocol are shown in Figure 9.1 and described below. Compared to the rather complex application scenario of Figure 7.2, the following simplified scenario illustrates a contracting process between one Content Service Provider (CSP), one end-customer, and three intermediate ISPs.

The scenario implies that the end-customer initiates the communication connection with the CSP by requesting a streaming video service from the CSP. One

[1] A centralized approach of the contracting process is described in [KH01].

could imagine that the CSP provides web access with a list of videos and eventually some additional information such as a list of actors, short abstract, preview, etc. The end-customer would choose a video with a starting time at a certain date, the desired QoS, and other parameters concerning monitoring, reporting, and customer support. For the understanding and confidence of the end-customer, QoS parameters are specified in user-friendly terms abstracted from any technical notation. Depending on the selected service parameters the price might change accordingly.

As soon as the CSP receives the request from the end-customer, it is its responsibility to set up the connection to the end-customer via intermediary ISPs. Therefore, a chain reaction is started where the CSP searches for a path to the end-customer that contains ISPs in line that are able to provide the requested service with the specified QoS parameters. At that point, parameters are transferred into technical specifications. As soon as a path is found (this can include one or many ISPs, in Figure 9.1 there are three) from CSP to end-customer, the request of the end-customer can be acknowledged. The following maps this abstract view of the scenario onto a contracting process based on bilateral service contracts between the business entities.

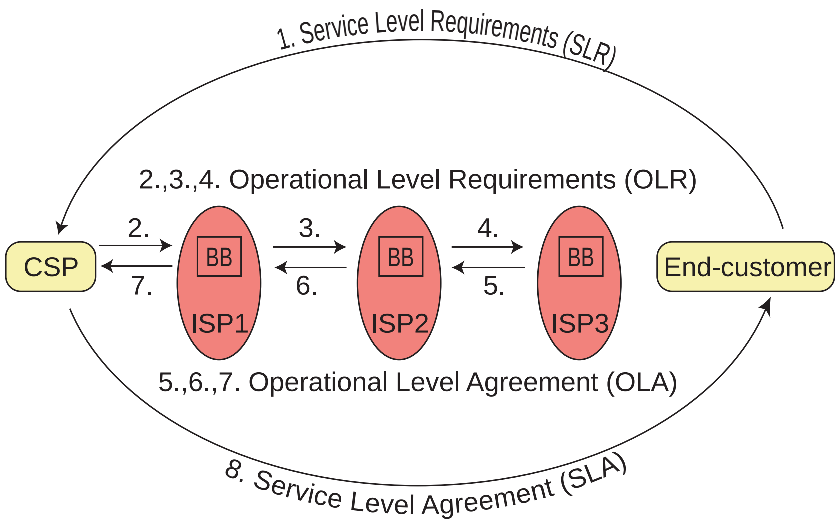

Figure 9.1: Contracting protocol showing a sequence of steps to result in bilateral OLAs and one SLA (from [KHB02]).

In a first step, the receiver of the service (i.e. the end-customer receiving the video stream) defines the Service Level Requirements in informal language and non-technical terms, which include specifications concerning QoS (e.g. colors =

256, frame rate = 16 frames/s, frame size = 352 * 288 pixel, reliability = very good (stall $<$ 1%)), monitoring, reporting, and customer support (as denoted in section 8.2). The CSP who is the end-customer's responsible point of contact cannot guarantee the defined SLR at this point yet. After verifying its own resources required for the service delivery, the CSP needs to check (step 2) with one of its neighboring ISPs (in Figure 9.1 this is ISP1) whether it can provide the required resources to fulfill the SLR. At this point, the SLR need to be transformed into Operational Level Requirements, a technical description of the SLR that can be directly interpreted by the succeeding ISPs (e.g. bandwidth = 1.3 Mbit/s, end-to-end delay = 0.5 s, jitter = 100 ms, packet loss $<$ 1%, error rate $< 10^{-11}$ (fiber)). After verifying its own resources with the help of a network manager (in Figure 9.1 this role is taken by a Bandwidth Broker (BB)), ISP1 creates a new OLR to request the service with ISP2. This process of single and independent service requests continues until, finally, an ISP can deliver the service to the end-customer (in Figure 9.1 this is ISP3 as the end-customer's Access ISP). If an ISP does not agree with the required service conditions, it rejects the OLR and either an alternative route with another connecting ISP can be found or the SLR needs to be redefined by the end-customer to fit the service conditions of all the ISPs.

If ISP3 can guarantee the service delivery according to the defined conditions, it signs the received OLR from ISP2, and thus, makes an Operational Level Agreement out of it (step 5 in Figure 9.1). The cascading process of signing OLRs goes back to the CSP with several OLAs between the neighboring ISPs as a result. Consequently, the CSP can now sign a SLA that guarantees the end-customer the service under the previously defined conditions of the SLR. SLR and OLR are requests that are either rejected or accepted and signed by the ISPs in form of SLAs and OLAs. Several bilateral contracts were settled between all the involved business entities to enable a combined Video-on-Demand service between end-customer and CSP. Therefore, the contracting protocol was developed to show the feasibility of combined service provisioning.

Notice that no OLA is required between ISP3 and the end-customer, since a business relationship between them must exist already on a lower level, the communication level, in form of a simple network access service that implies some availability and bandwidth guarantees. SLAs/OLAs are set up on a higher level, the contracting level, to focus on reliable service provisioning and guaranteed transmission of the content of the service (different levels of abstraction in conjunction with their functional components were illustrated in Figure 1.1).

9.3 Conceptual Design of the Contracting Protocol

Figure 9.2 shows the sequence of the single messages that are exchanged between the business entities according to the contracting protocol as described in the previous section. It shows a cascading process of exchange messages starting from the top with a SLR from the end-customer to the CSP. The SLR is followed by three consecutive OLRs from CSP over ISP1 and ISP2 to ISP3 before they are accepted and transformed into OLAs. The final SLA results as a consequence of a successful contracting process. In case one ISP rejects an OLR, a new ISP has to be identified in order to close the gap and form a continuous path from CSP to end-customer. The following section 9.4 describes an implementation of the contracting protocol, which enables an ISP to divide the total service time into 'time slices' and redirect a data stream over different paths accordingly in order to fulfill the service.

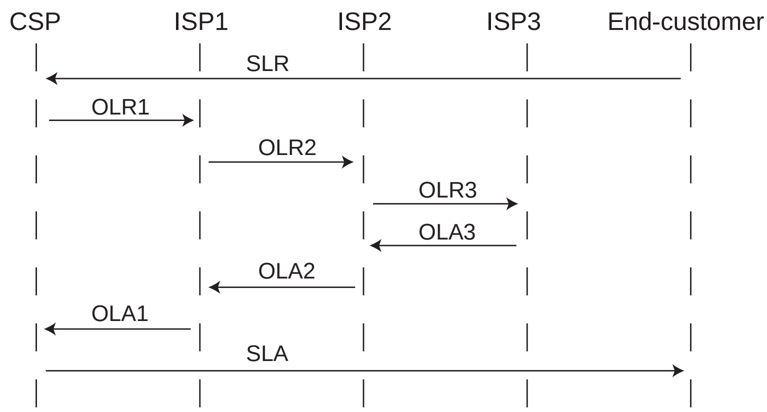

Figure 9.2: Contracting protocol exchange messages between the business entities (from [KHB02]).

According to the contracting protocol, there are the following five steps that are required for every ISP after receiving an OLR from a neighboring ISP.

1. Query to the ISP's resource database for any available network resources according to the requirements of the received OLR. The resource database contains information about the availability of software and hardware resources. If services or network devices are down, it will be stored in that database. In addition, the reservation of resources in advance will be available with the help of network management tools.

2. If resources are available as requested in the OLR, the ISP stores the received OLR in a database with a corresponding identification number. The identification number serves to respond properly (i.e. with an agreement or rejection) to the request as soon as the availability of network resources of neighboring ISPs is clarified.

3. The received OLR is transformed into a new OLR that will be sent to the neighboring ISP. Since the whole contracting process of the contracting protocol consists of several bilateral contracts, every business entity needs to sign two contracts for the service delivery to the end-customer. One contract with the preceding ISP, the other one with the succeeding ISP. Within one contract the ISP is a service customer, within the other contract the ISP takes the role of a service provider. Both contracts carry different identification numbers.

4. The new OLR is also stored in a database to keep reference for the expected reply. If the new OLR is rejected, it will be deleted from the database. In case of a positive answer, the corresponding OLA is stored instead.

5. The new OLR is sent to a succeeding, neighboring ISP to wait for a reply. The reply is either a confirmation in form of an OLA or a rejection. Consequently, the result of the reply needs to be communicated to the preceding ISP.

9.4 Implementing the Contracting Protocol

Implementing the contracting protocol means to find an algorithm that transforms the negotiation process between the business entities into a sequence of OLAs between the ISPs and a SLA between CSP and end-customer. The difficulty within the contracting process is that there are ISPs that cannot fulfill the requested OLR, and thus, reply with a rejection. It is therefore required to find a different route on the way to the end-customer. Another requirement for the protocol concerns the execution time of the contracting protocol, which should perform in real-time since the end-customer expects an immediate response for his/her request. Finally, the resulting service contracts (if concluded) represent legal documents whose authenticity and integrity can be assured by using digital signatures (for security aspects and solutions see [Kne01]).

For the implementation of the contracting protocol, we made the following assumptions:

- The structure of the ISP networks or domains is organized so that there is at least one route from a sender to a receiver. In addition, there are ISPs that have multiple neighboring ISPs, which represent alternative routes to the end-customer in case an ISP rejects an OLR. The predetermined network structure of ISPs for the implementation looked like the one in Figure 9.3 where each ISP is represented by a box with an incoming and an outgoing connection to neighboring entities. To avoid communication cycles between ISPs, packet forwarding is based on information of underlying routing protocols.

- The current situation on the Internet is such that ISPs are not equal in terms of network reach and traffic volume. Tier-1 ISPs are those that have access to the global Internet routing table and do not buy network capacity from other providers. Tier-2 and tier-3 ISPs are regional providers with a smaller national presence and no national backbone that may lease parts of their network from tier-1 ISPs. For routing data through networks, there are peering agreements between tier-1 ISPs at no charge due to an equal amount of exchanged traffic, and there are commercial agreements between tier-1 ISPs and smaller ISPs for routing their traffic [All01]. The implementation of the contracting protocol did not consider any restrictions concerning hierarchical real-world peering conditions among ISPs on the Internet.

- The existence of a well-designed and working network and resource management within networks of ISPs is necessary to provide reservation of network resources with a certain QoS. A broker system is required as a central instance within the network of an ISP that receives and negotiates customer requests and treats them accordingly [KB00]. The implementation did not consider such an instance for managing resources but instead the availability of required network resources was simulated by a variable to express the average network utilization.

- Infrastructure and network equipment allow data streams to be treated according to their service levels. IPv6 and ATM networks for example provide mechanisms to handle different service classes (see section 3.3).

- There is legal coverage and international unanimity for the enforcement of contractual agreements in form of SLAs and OLAs between international service providers.

9.4.1 Simple Protocol

Our first approach for implementing the contracting protocol aimed at specifying and implementing the *Simple Protocol*. The Simple Protocol served as a basis

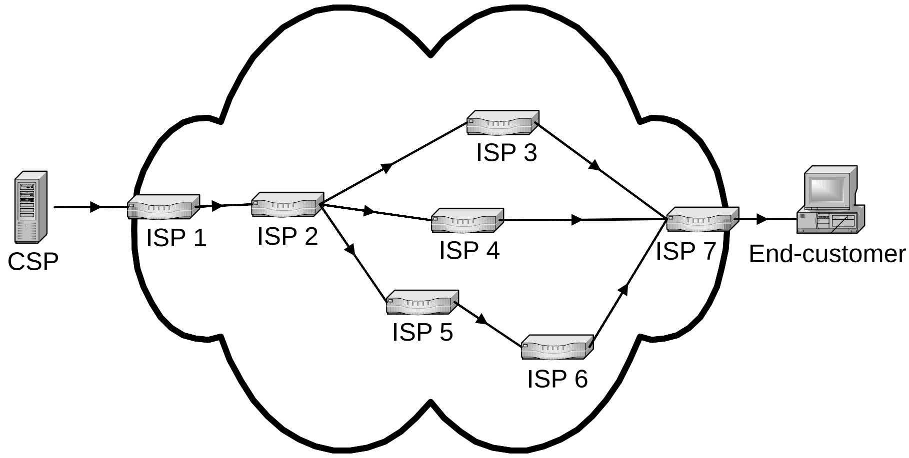

Figure 9.3: Network structure of the implemented application scenario (from [KM02]).

for further investigation, and thus, only implemented parts of the specification requirements as mentioned previously.

The Simple Protocol implements a depth first search algorithm [CLR90] on existing ISPs of the network structure. A path is explored deeper, whenever possible. Starting from the root (i.e. CSP), a route along several ISPs to the end-customer is searched by sending OLRs from ISP to neighboring ISP. As soon as an ISP rejects the OLR, the search 'backtracks' and a different route needs to be found via a different ISP. The main steps of the algorithm are illustrated in Figure 9.4 and explained hereafter. The complexity of the search algorithm is significant since the network structure between neighboring ISPs is manageable and ISP-internal network structures and routing policies are not considered for the performance of the Simple Protocol. For simulation purposes a network structure with seven ISPs was chosen as illustrated in Figure 9.3.

The flow chart in Figure 9.4 shows the steps of the Simple Protocol from a CSP point of view. It starts with the end-customer ordering a video through a web interface (step 1). The implementation is based on registered end-customers that disclose name, address, kind of payment, and other personal data before they receive username and password. The login as a registered customer reveals a list of available movies plus additional information about the length of the movie, actors, description of contents, etc. The customer selects a movie, the date and time (for possible in-advance reservations), and some QoS parameters. In order to gain trust and acceptance from the end-customer, it is very important not to describe QoS parameters in technical terms but informally with natural language.

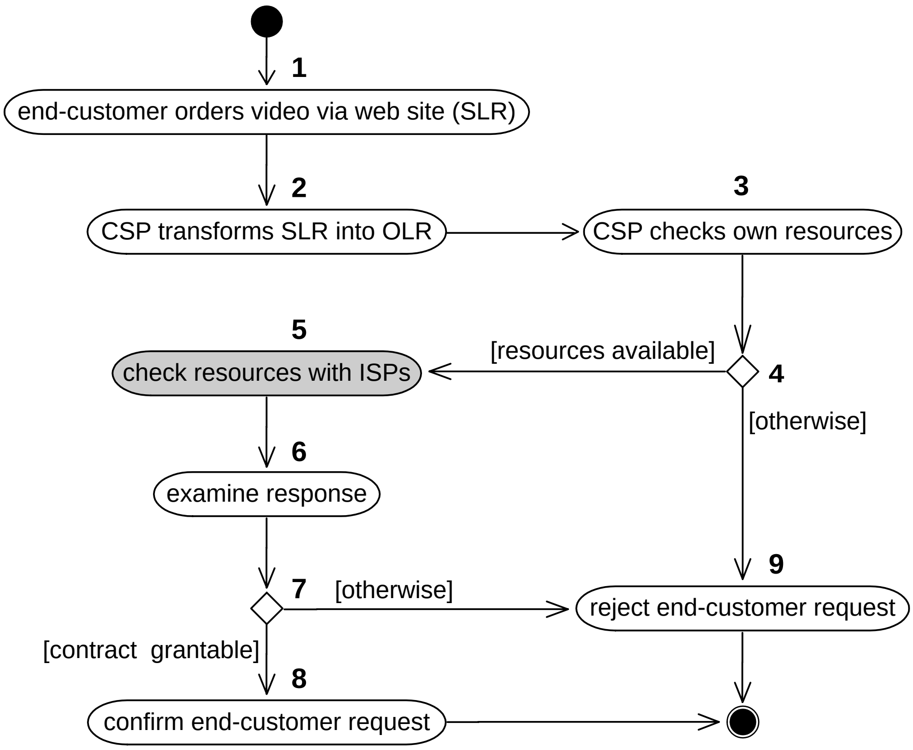

Figure 9.4: Flow chart shows consecutive steps of our implementation of the contracting protocol (from [KM02]).

The CSP transforms the selected informal quality parameters of the SLR into a technically formulated OLR (step 2) before it checks its own resources available for the service delivery (step 3). In case no resources are available to fulfill the SLR the end-customer's request gets rejected, otherwise the OLR is forwarded to ISPs in order to check resources and find a path to the end-customer (step 4).

Step 5 in Figure 9.4 implements the actual search algorithm for a path to the end-customer and contains recursive checks and requests with neighboring ISPs. An OLR is sent to the neighboring ISP who likewise checks its available resources for that service. If network resources are available a new OLR is sent to the neighboring ISP who again checks its resources[2] before forwarding a new OLR. This process continues until the last ISP in line agrees to the OLR, signs it, and sends it back to the preceding ISP who in turn does the same thing with its received OLR.

[2]As mentioned earlier, the implementation did not consider real network management data but instead used random numbers to indicate the availability of network resources in order to create realistic scenarios.

When the CSP receives the answer of its neighboring ISP it examines the response (step 6) and confirms the end-customer request with a SLA (step 8) if and only if the previous SLR is grantable and all the ISPs along the path agreed to the service conditions. Otherwise the SLR is rejected (step 9). It is not very user-friendly to simply reject the end-customer request as being done in step 9. In a real-world business application, the CSP might for example submit a new offer or at least inform the end-customer about the current status of available network resources and give him/her the chance to change QoS parameters in order to align with the realities.

Problems occur when OLRs are rejected. In order to explain the strategy of the Simple Protocol a look at the network structure of Figure 9.3 might be helpful. ISP3, ISP4, and ISP5 are all neighboring ISPs of ISP2 who first sends an OLR to ISP3. In case of rejection, an OLR is sent to ISP4, and if also rejected an OLR is sent to ISP5. If none of the neighboring ISPs accepts the OLR, ISP2 also needs to reject its OLR from ISP1, and the SLR of the end-customer needs to be rejected as well.

The limitation of this approach is obvious and tempting for improvements. Within the Simple Protocol, ISPs only have the option to either accept or reject an OLR depending on the availability of network resources for the whole duration of the service. It does not consider the possibility of ISPs to have resources available for only part of the requested service period. Therefore, the following section illustrates another approach to overcome the rigidity of the Simple Protocol.

9.4.2 Time Slice Protocol

The *Time Slice Protocol*, we developed as a second approach, is an extension of the Simple Protocol with more flexibility to find a path of ISPs from CSP to end-customer. The extension comprises the functionality of ISPs to accept, reject, and also modify an OLR. An OLR can be modified if an ISP can fulfill the requested service for only part of the whole duration. Consequently, the requested service can still be fulfilled within the Time Slice Protocol by splitting the communication path between different ISPs within corresponding 'time slices', whereas the service request had to be canceled in that case within the Simple Protocol. This approach makes sense in cases when data traffic is high within ISP networks and network resources are used to their maximum capacity which makes it difficult to find service coverage for the entire duration of the service.

The flexibility of the Time Slice Protocol increases the complexity for the implementation. First of all, an ISP that gets a number of modified OLRs back from its neighboring ISPs needs to check whether the service duration can be covered by combining time slices with corresponding ISPs. If this is the case, then, secondly, the ISP needs to decide about how the different time slices can be recombined to cover the whole service duration. And thirdly, since the time slices of the modified OLRs most likely overlap, a strategy needs to be found to eliminate this overlapping. Figure 9.5 shows a possible timed segmentation of the path with different routes during appropriate time slices.

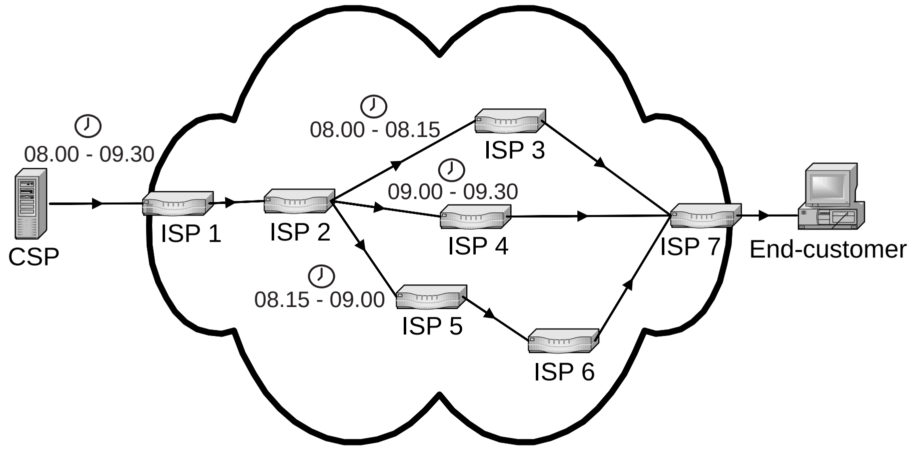

Figure 9.5: Mode of operation for the Time Slice Protocol showing different routes during appropriate time slices (from [KM02]).

The general process of the Time Slice Protocol is similar to the one of the Simple Protocol as illustrated by the 9 steps in Figure 9.4. However, step 5 (“check resources with ISPs”) is enhanced and contains recursive checks and requests to neighboring ISPs in order to find a path to the end-customer. The Time Slice Protocol allows a neighboring ISP to respond with a modified OLR, which contains an adjusted time period during which the service could be fulfilled with the requested QoS parameters. The modified OLRs are no binding counter offers but simply represent the situation of available network resources for the requested service period.

If an OLR is not accepted by a neighboring ISP and instead, a modified OLR is returned with an adjusted time period, the modified OLR is stored and another neighboring ISP is requested for service delivery. At this point, two strategies are possible. The *pessimistic strategy* assumes that it is hard to find a neighboring ISP to cover the whole service duration, and therefore, changes the service duration of every OLR depending on the times not covered by other neighboring ISPs yet. The *optimistic strategy* assumes to find a neighboring ISP for service delivery during the whole duration, and therefore, collects the received modified OLRs without processing them and instead requesting other neighboring ISPs for service delivery during the whole duration. In practice, the choice of the strategy depends on the load capacity of the network. In case of a high workload, the pessimistic strategy might be more suitable, for the implementation the optimistic strategy was used.

If an ISP could not find a neighboring ISP to cover the whole operation of the requested service, a subprocess for analyzing the modified OLRs is executed. Analyzing all the modified OLRs leads to a selection of certain modified OLRs that cover the whole service duration. If not enough ‘time slices’ could be found to cover the whole period, the end-customer request needs to be canceled. To eliminate overlapping, new OLRs with adjusted time periods are resent to the corresponding ISPs, which thereafter start a new search process for a path to the end-customer. As mentioned earlier, modified OLRs are not binding, which implicates that in the meantime network resources of neighboring ISPs might not be available anymore due to the constantly changing network situation. Therefore, a *revocation clause* must be introduced.

Assuming that ISP2 in Figure 9.5 collects the modified OLRs from its neighboring ISP3, ISP4, and ISP5, eliminates overlapping and resends new OLRs to them. ISP3 and ISP4 start searching a path along subsequent ISPs to the end-customer and reply with an OLA for the specified time. If ISP5 in the meantime cannot find any neighboring ISPs to fulfill the service with the specified parameters (as yet indicated by the modified OLR), it replies to ISP2 with a rejection of the OLR. In that case, the end-customer request will be canceled and the OLAs with ISP3 and ISP4 need to be withdrawn. Just for this case, a revocation clause foresees the possibility for the sender of an OLR (i.e. the service consuming entity) to cancel an already received and signed OLA within a certain time limit. Likewise, the service providing entity will be given the right to withdraw from an OLA within a certain time limit.

In case of revocation, alternative implementations could be designed to reanalyze the list of modified OLRs and recombine them in order to find a different segmentation of the path.

9.4.3 Load Slice Protocol

Within the Time Slice Protocol single OLRs were fulfilled by several ISPs during different time slices. A similar approach would be to split bandwidth into single ‘load slices’ and distribute it to several ISPs accordingly. Therefore, we designed the *Load Slice Protocol* to allow ISPs to reply to an OLR with both adjusted times and bandwidth (as one parameter for expressing network resources).

Starting from the network structure of Figure 9.3 again, the total load of required bandwidth could for example be carried by ISP3, ISP4, and ISP5. During the total period of time all three ISPs would have their slices available for the service delivery of the end-customer request. The practical usability of the Load Slice Protocol requires the technical ability to separate data packets of the real-time stream and distribute them to the different ISP connections accordingly. Virtual channels within ATM networks provide a mechanism to support connections with different QoS prerequisites (compare to section 3.3.3).

The Load Slice Protocol has not been implemented explicitly, but problems are foreseeable on a technical level rather than on a conceptual level. Furthermore, just as time and bandwidth can be sliced and distributed to different ISPs, other parameters (e.g. loss rate, availability, delay) could eventually be sliced and distributed as well. Yet another approach may be to divide the time or load slices into several fixed fragments in advance and have the ISPs mark the possibilities of QoS guarantees. Further investigations in this area are required and performed in the future.

9.5 Video-on-Demand Contracting Architecture

The previous section described the contracting process with two implementations of the contracting protocol (i.e. the Simple Protocol and the Time Slice Protocol). In order to embed the implementation, we developed the Video-on-Demand Contracting Architecture (VODCA) with the roles and components as a technological framework for the realization of the previously described contracting protocols (see also [Mar01]).

However, operation of both Simple Protocol and Time Slice Protocol within VODCA was simulated on a single computer according to the network structure of Figure 9.3. This means that CSP was simulated by running a *manager application* to administrate customer and movie data and to control communication with end-customer, database, and other ISPs. Each ISP was simulated on the same computer by an *ISP application* to communicate with other ISPs and the database (see Figure 9.6).

The end-customer interacts with the CSP over a *web interface*. The end-customer registers and specifies the parameters for the SLR through an online form on the web. The SLR is passed from the web interface to the manager application, which transforms the SLR into an OLR. The manager application then starts the protocol (Simple Protocol or Time Slice Protocol) to communicate with its neighboring ISPs. Both manager application and ISP application implement the different steps of the contracting protocol. OLRs and OLAs are exchanged among ISPs and also between CSP and neighboring ISP. If an OLR between ISPs results in an OLA, it is stored bilaterally in both ISPs' and CSP's databases. The SLA is stored only in CSP's database.

Figure 9.7 shows a screenshot of a graphical component of the ISP application simulating ISP3 according to the network structure of Figure 9.3. It contains an incoming contract request (OLR) of neighboring ISP2 with date, time and quality

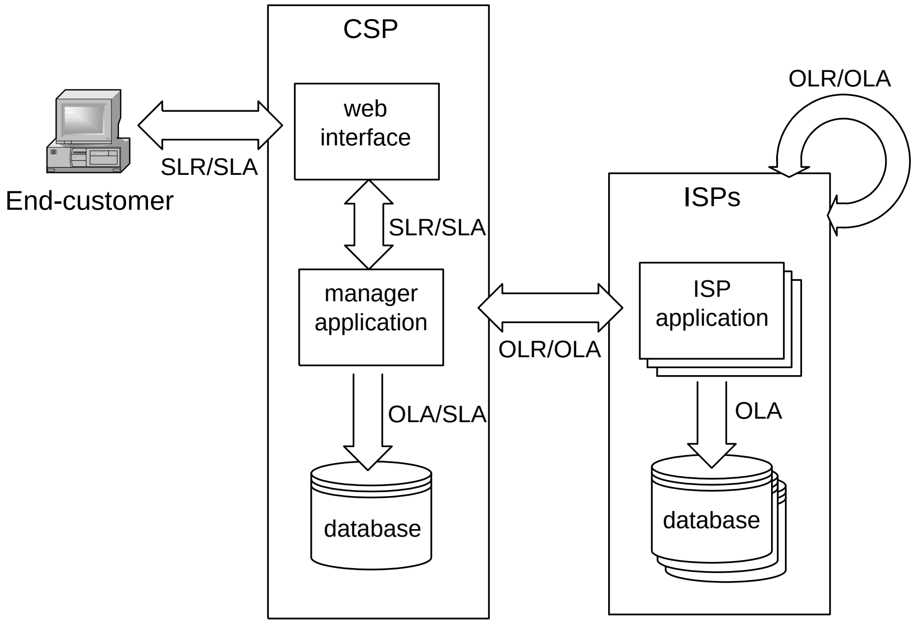

Figure 9.6: The roles and technical components of the Video-on-Demand Contracting Architecture (VODCA) (from [KM02]).

parameters. After checking its own resources, ISP3 sends the modified OLR back to ISP2 with adjusted times for the service delivery. Implementing the contracting protocol within VODCA required a number of technologies that will be introduced briefly in the following subsections (details in [Mar01]).

9.5.1 Java

Both manager application and ISP application were implemented in Java[3]. The object-oriented concept of Java emphasizes objects with their fields, methods, and their interfaces to communicate with other objects. The definition of interfaces and the concept of inheritance were advantageous for a modular design of the application. Interfaces allow the exchange of objects and groups of objects (modules) without negative impacts on the functionality of other objects, e.g. GUI modules or database access can be exchanged on-the-fly. Next, inheritance is used for implementing the functionality of both Simple and Time Slice Protocol. A method `processContract()` for handling the contracting process is declared in a super-class `Protocol` while the functionality is implemented in the according

[3]The Java Language Specification (Second Edition), 2000; accessible at URL: http://java.sun.com/docs/books/jls/second_edition/html/jTOC.doc.html

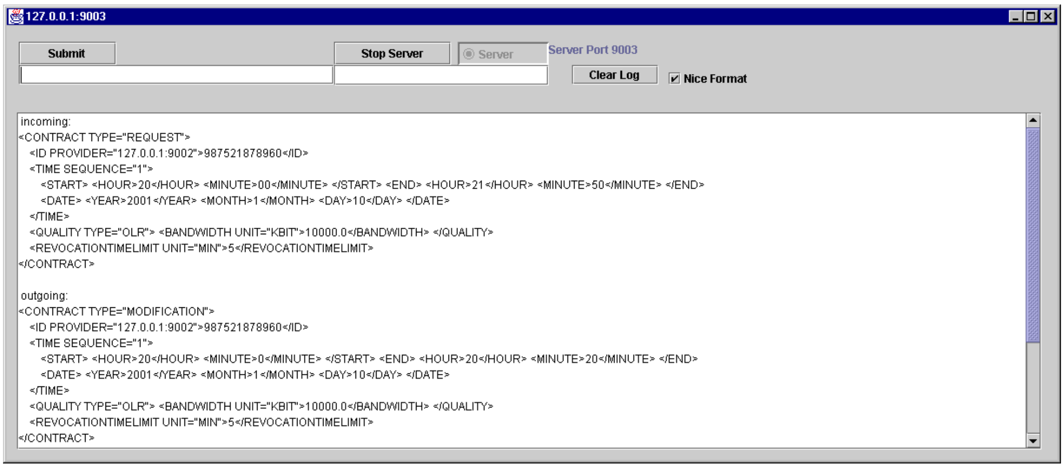

Figure 9.7: Screenshot of an ISP application showing incoming and outgoing contract types (from [KM02]).

sub-class. Thus, the classes `SimpleProtocol` and `TimeSliceProtocol` extend `Protocol` while implementing the process for the Simple Protocol or the Time Slice Protocol, respectively. According to this, variations of the protocol can easily be implemented and used thereafter.

Java code is *platform-independent* since source code is not compiled in platform specific machine code but in platform independent byte code that needs a *Java Virtual Machine* (JVM) and standard runtime libraries to be interpreted[4]. Platform independence of Java code is a desirable feature of VODCA since the application is used by different ISPs with different devices and operating systems. Communication between the distributed instances of ISPs takes place over the Internet. Java provides class libraries to facilitate TCP/IP communication.

Multithreading is another Java feature implemented in the VODCA architecture. A thread is a so-called lightweight process [Tan01]. If several threads run within one process it is called multi threaded. Every instance of a manager or ISP application consists of several threads in order to process several requests. Implementing threads is convenient in Java by inheriting from the class `Thread` as part of the standard library. However, Java consigns thread handling to the operating system, which results in strong distinctions between the operating systems.

---

[4]The Java Virtual Machine Specification (Second Edition), 1999; accessible at URL: http://java.sun.com/docs/books/vmspec/2nd-edition/html/VMSpecTOC.doc.html

9.5.2 Servlets

The web interface of VODCA processes user interactions (e.g. registration, login, service specification, ordering), which requires a dynamic structure of the HTML page implemented with *Java servlets* [DC99]. The servlet technology is based on Java and it makes use of Java classes just as manager and ISP applications do. As opposed to the *Common Gateway Interface* (CGI) [Gun96], servlets have a better performance since dynamic page setup is processed by threads instead of processes. Besides, the fact that servlets are based on Java makes their usage also platform-independent.

9.5.3 XML

The *eXtensible Markup Language*[5] (XML) was used within VODCA to specify the contents of the SLR/SLA and OLRs/OLAs. XML, a subset of the *Standard Generalized Markup Language* (SGML), qualifies to structure a document (i.e. a service contract) hierarchically with markups in order to be processed automatically. Labeling different elements of a document by XML tags enables applications to exchange and process documents without human interaction. The contracts that are exchanged within the applications are valid XML documents in compliance with an existing *Document Type Definition* (DTD) (see valid XML document in Figure 9.7 based on an existing DTD). XML and Java form a profitable combination since the use of both XML and Java is targeted on interoperability and platform-independence.

A XML parser is needed to analyze an XML document according to its DTD in order to get the necessary information out of it. There are basically two kinds of XML parsers, SAX[6] and DOM[7] parsers. SAX parsers implement the *Simple API for XML* (SAX) and they are event-based, which means that they report parsing events (such as the start and end of an XML tag) directly to the calling application through callbacks. In contrast to that, DOM parsers follow the *Document Object Model* (DOM), a standard defined by the *World Wide Web Consortium*[8] (W3C), which describes an interface for the access of HTML and XML documents. A DOM parser builds an internal object tree that can be parsed and manipulated. However, it is expensive to construct and traverse in-memory parse trees for parsing large XML documents, since a lot of memory is required. Nevertheless, a

[5] eXtensible Markup Language (XML) 1.0 (Second Edition), 2000;
accessible at URL: http://www.w3.org/TR/REC-xml

[6] SAX 2.0: The Simple API for XML, 2000;
accessible at URL: http://www.megginson.com/SAX/index.html

[7] Document Object Model (DOM) Level 2 Core Specification, Version 1.0, 2000;
accessible at URL: http://www.w3.org/TR/DOM-Level-2-Core/Overview.html

[8] Accessible at URL: http://www.w3.org

DOM parser was used to implement VODCA since XML documents representing contracts are not too big, and thus, a DOM tree can be built easily. Furthermore, DOM parsers have the ability to edit XML documents, which is advantageous to generate modified OLRs.

9.5.4 TCP/IP

The implementation of VODCA uses the TCP/IP Protocol (see section 3.3) for the communication between the different instances of manager and ISP application. Every instance of manager or ISP application runs a server process that listens for incoming requests on a certain port. The client process contacts the server process via IP address and port number. TCP establishes a connection-oriented communication, which means that a connection is set up, data packets are sent during arbitrary time before the connection is closed down again.

9.5.5 Database Management System

VODCA describes an architecture based on the exchange of XML documents that represent SLR/SLA and OLR/OLA, respectively. Therefore, Tamino[9] as a *DataBase Management System* (DBMS) is used that saves XML objects without conversion into a different data format and without projection onto a relational data structure. This accelerates the database access since data storage and retrieval of data is performed without transformation. Database access was realized through a Java API that provided functionality to store, retrieve, and manipulate XML documents easily. Database operations can be executed using the *XML Query Language*[10] (XQL) as a standard also defined by the W3C.

9.6 Contracting Protocol with a Service Broker

Both contracting protocol and Video-on-Demand Contracting Architecture illustrated how SLAs/OLAs can be defined and negotiated between end-customer, ISPs, and CSP. We assumed that the end-customer establishes direct contact with the CSP at a definite time when he/she requires a service. However, there are some disadvantages and problems associated with it:

- Negotiation needs to take place for every individual business relationship, which can lead to an overload at the CSP's or ISP's servers. To some ex-

[9] Accessible at URL: http://www.softwareag.com/tamino/
[10] Accessible at URL: http://www.w3.org/XML/Query

tent this could be reduced by mirroring services and contents at different locations with different service providers (e.g. Akamai[11]).

- In a distributed environment (as illustrated in Figure 7.2), it is difficult to utilize available resources in a balanced manner. It would be desirable to have a well-balanced network utilization even at marginal times of a day. This could be encouraged, for example, by use of differential pricing models [RLS99], but this is not always flexible enough and nontransparent for the service customer.

- Service providers would prefer to administrate just a few SLAs/OLAs for services with large quantities of required network resources, since economic effort and risk were smaller.

- A reservation of a service in advance (i.e. a service will be provided at a future date) requires a well-working network resource management system. It is desirable for end-customers to be able to make reservations at any time, particularly for special events that are known in sufficient time to obtain the required information at the right time.

- Every SLA/OLA is unique. Therefore, it cannot be re-sold easily if it is not required anymore.

- A service customer can never be certain that it will be possible to obtain a SLA/OLA at any given time for a requested service.

As an approach to solve the previously mentioned disadvantages and problems of the contracting process, we introduced the role of a *Service Broker* (SB) to act as a *negotiator*. To illustrate the concept of such a service broker and to keep the scenario as simple as possible, the service broker operates only between end-customer and CSP. However, service brokers as negotiators between ISPs are very well imaginable.

The service broker represents a point of contact for end-customers, if they request and negotiate a service with certain QoS requirements for the service performance. Furthermore, the service broker collects all the offers from CSPs and tries to match them in an optimal way. In case of similar service requests concerning price or QoS parameters (e.g. several end-customers are interested in a real-time video stream of an Olympic competition), the service broker could collect these requests according to their local distances and gather them in a compound SLA/OLA with the end-customers and the CSP. The service broker could even request resources from a CSP for a service that it is going to sell to end-customers

[11] Accessible at URL: http://www.akamai.com

in the future (e.g. selling the rights to broadcast future sports events). Service requests by end-customers could eventually be renegotiated and resubmitted to the end-customer in case of changes in price or QoS. To summarize, the role and functionality of a service broker ensures efficient utilization of available network resources in a distributed environment.

Figure 9.8 shows the position and behavior of the service broker in analogy to the contracting protocol of Figure 9.1. The service broker could be an independent business entity but in this case, it is ISP3 that takes the role of the service broker, which is very well possible in a real-world scenario. The service broker takes a SLR from an end-customer, translates it into an OLR to transfer it to the CSP. The following steps (steps 2-7) are equal to the ones in Figure 9.1, with the particularity that there exists an OLA between service broker and CSP. Finally, service broker and end-customer settle the SLA with the required service conditions. Notice again that no OLA is required between ISP2 and the service broker, since physical interconnection between the two entities must already exist on a communication level and also, ISP2 is the last ISP in line to agree to the OLR originally requested by the service broker.

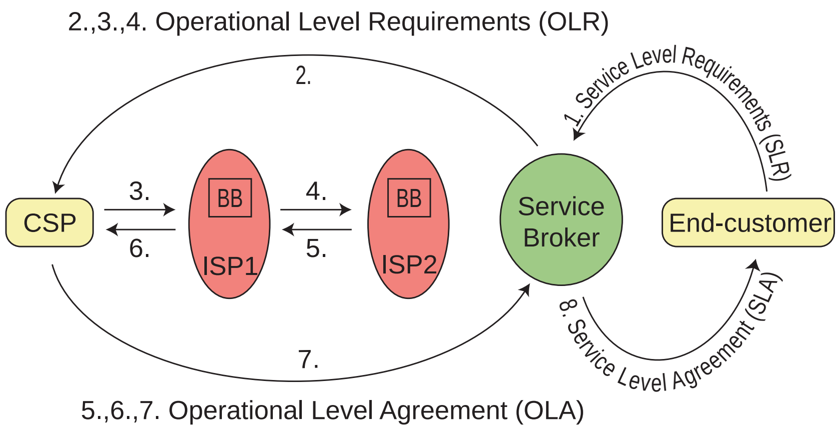

Figure 9.8: The Service Broker as negotiator between CSP and end-customer (from [KHB02]).

In order to support the role of a service broker and the exchange of SLAs between service broker and end-customer (or eventually between other brokers), the contracting protocol as illustrated in section 9.2 requires further considerations:

- The SLA is turned into a negotiable digital good that can be renegotiated at will right up to the date of usage or after its validity has expired.
- A SLA can be formulated and issued at any time.

- When a service comes into operation, a SLA must be shown to proof correctness of service conditions. This way, the SLA serves to authorize the usage of specific resources.
- A platform is established where SLAs can be negotiated. Such a platform could be managed by service brokers.

Three conclusions can be drawn from this in order to make the service broker work precisely and efficiently and particularly satisfying for both service provider and service customer:

1. A marketplace must be established where SLAs can be traded.
2. A certificate must proof existence and ownership of an SLA as prerequisites for the SLA to be traded correctly.
3. A transfer mechanism must enable the transfer of an SLA between seller and buyer as simply and securely as possible.

9.7 Trading of Service Level Agreements

9.7.1 The Marketplace

For trading SLAs, we designed a marketplace with several participants as modeled in Figure 9.9. It should be noted that the model focuses on the trading of SLAs assuming that corresponding OLA trading between CSPs and service brokers takes place in a similar and parallel manner .

- **CSP.** The business entity that provides the content of the service. The CSP is responsible to the service broker for making sure that the service is available as described in the SLA, that it can be clearly identified, and that it meets all legal requirements. However, it is the service broker that holds responsibility to the end-customer for the reliable service provisioning.
- **Owner/Seller.** The business entity that owns the SLA and holds the rights to trade it. Owner/seller of a SLA can be a service broker after reserving a service from a CSP (which was stipulated in form of an OLA). Owner/seller of a SLA can also be an end-customer trying to re-sell a previously purchased SLA.
- **Buyer.** The business entity that purchases a SLA and that will be the future owner and/or re-seller. Both end-customer and service broker could be possible buyers.

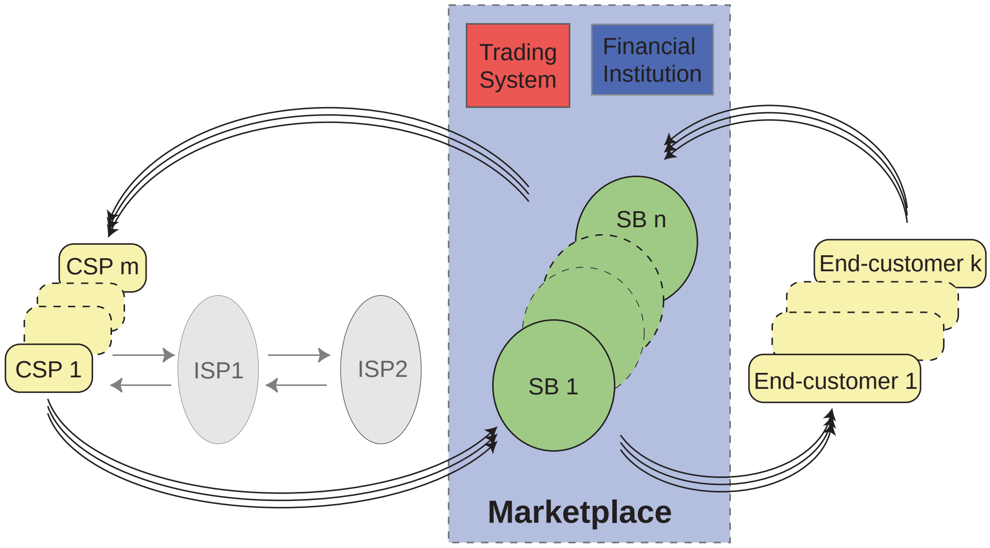

Figure 9.9: Participants in the marketplace for trading SLAs (from [KHB02]).

- **Trading System.** An institution that is responsible for the settlement of the transaction. It is required that all marketplace participants trust the trading system in order to execute SLA trading activities. There might be one or more trading systems present in a marketplace (compared to several *Certification Authorities* (CA) within *Public Key Infrastructures* (PKI)).

- **Financial Institution.** An institution that is responsible for processing payment transactions. Also several financial institutions might be present in a marketplace (compared to several credit institutions for purchasing ordinary products).

9.7.2 The SLA Certificate

The tradable SLA is essentially a certificate, which contains a description of the content of the SLA, the issuer, the lawful owner of the SLA, and the trading system. Thus, the tradable SLA is an XML document named `SLACertificate` and divided up into the four sections `SLA`, `TSIssuer`, `Owner`, and `TS` as illustrated in Figure 9.10.

- **SLADescription.** This section describes the content of the SLA according to section 8.2.

- **TSIssuer.** This section contains information about the issuer of the SLA, which is the trading system that signs the SLA at first. The issuer is always

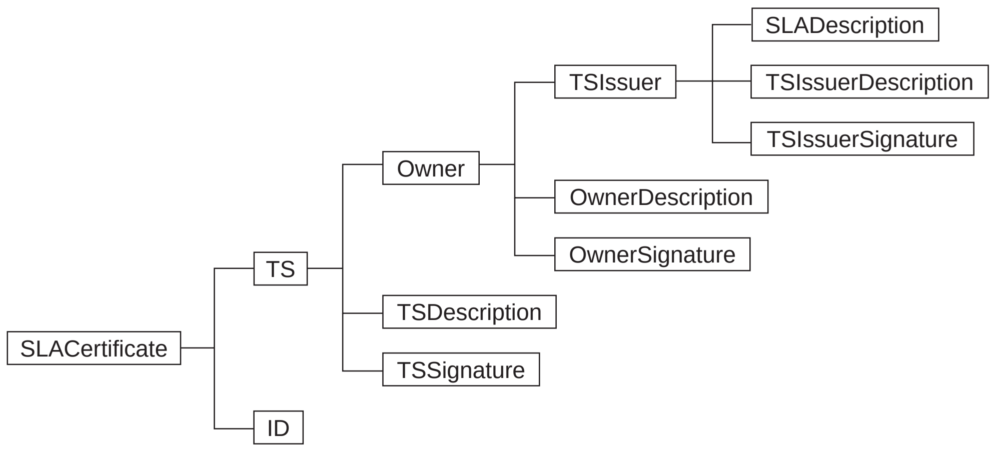

Figure 9.10: XML structure of the SLA certificate (from [KHB02]).

a trading system since it is a trusted entity to all marketplace participants. The issuer digitally signs its identifying information and the description of the SLA. This helps to identify modifications and prevent from fraud and unauthorized changes of the content. No information about the content of the SLA can be altered without the issuing trading system. However, the transfer of ownership can be executed by any other trading system.

- **Owner.** This section contains information about the owner of the SLA. The owner digitally signs his/her identifying information and the TSIssuer section. This digital signature verifies the lawful ownership and further on no changes can be made to the TSIssuer and SLADescription section without the owner's permission.

- **TS.** This section keeps information about the trading system, which has altered the SLA certificate last in order to sign its identifying information and the owner section. This trading system signs the whole SLA certificate at the end, which makes any kind of modifications impossible without any trading system.

9.7.3 Transfer of Ownership

The main task of a SLA trading system, as a trusted entity, is to ensure the secure transfer of SLA ownership together with the settlement of the corresponding payments. A sequence of eight different steps is required to execute a transfer of ownership from a seller of a SLA to a buyer. Figure 9.11 illustrates the sequence of steps with the service broker as seller and the end-customer as buyer of a SLA.

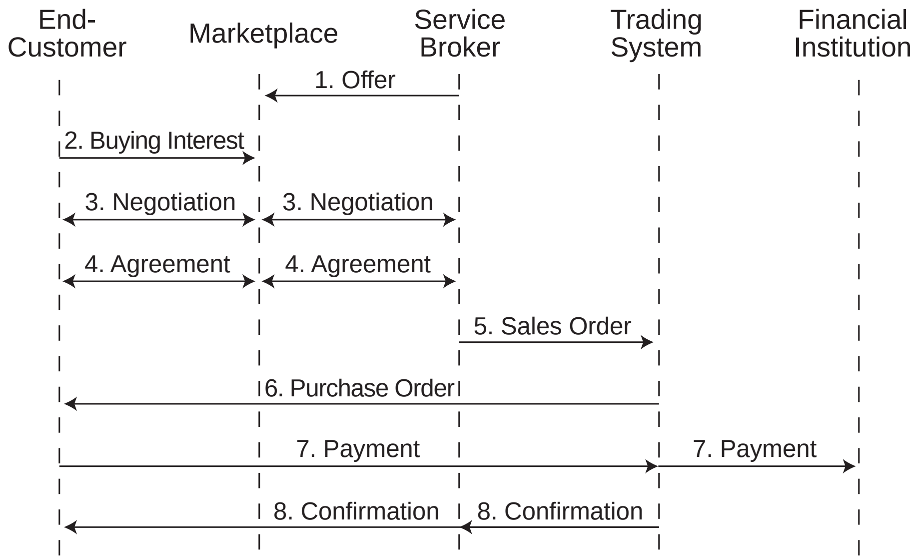

Figure 9.11: Transfer of ownership for a SLA certificate (from [KHB02]).

1. The seller provides his/her SLA certificate to one or several marketplaces where it is to be offered for sale. These marketplaces are not only for trading SLAs, other goods can be traded, too. It is important that existing marketplaces (e.g. auctions, spot markets) are used for trading and negotiating. Only for the final transfer of rights of a SLA an additional mechanism is used.

2. An interested service customer and potential buyer proceeds to the marketplace and selects a SLA. With the help of the trading system, the interested service customer can verify the validity of the SLA certificate, the seller's ownership, and what kind of SLA it is (e.g. QoS parameters, monitoring conditions, etc.).

3. The potential buyer and seller negotiate the price for the SLA certificate. No additional protocols are used for the negotiation process. Negotiation is supported by standard mechanisms of the marketplace where SLAs are offered.

4. The buyer and the seller agree on a price for the SLA.

5. The seller signs the sales order as a means for confirming that he/she wants to sell the SLA for the negotiated price. The sales order is now forwarded to the trading system.

6. The trading system sends the TSIssuer section of the SLA certificate to the buyer, who subsequently signs the owner section accordingly.

7. Payment is executed from the buyer via trading system on to the financial institution.

8. After the financial institution confirms a successful payment transaction, the trading system finally signs the SLA certificate and sends it to the buyer. A confirmation statement goes to both seller and buyer about a successfully completed transfer of ownership.

As soon as the trading system receives a sales order (step 5), the status of the SLA certificate is set to ‘being processed’. This ensures that the SLA cannot be sold again during the transaction process. While the certificate holds this status, both seller and buyer are able to revoke their interest to trade. However, once payment has been initiated (step 7) the status of the certificate is changed to ‘being paid’ and it is no longer possible for seller and buyer to revoke sales and purchase order. If the payment transaction is not successfully terminated the system reverts to the original status, i.e. the original owner remains owner of the SLA certificate.

The sequence of the steps to transfer the ownership of a SLA certificate emphasizes the important role of the trading system during the whole process. It is a specific entity, which is trusted by all the participants of the marketplace. Such entities already exist for other purpose like PKI (e.g. VeriSign[12], GlobalSign[13]). Among other things, the trading system performs the following tasks:

- Transfer of ownership of SLA certificates.
- Processing payments.
- Checking integrity, signatures, authenticity, and validity of SLA certificates.
- Blocking or rejecting SLA certificates.
- Storing of all the legally valid documents.

[12] Accessible at URL: http://www.verisign.com
[13] Accessible at URL: http://www.globalsign.net

Chapter 10

Architecture for Monitoring Compliance with Service Contracts

This chapter introduces an architecture that we developed for monitoring compliance with service contracts. Our implementation of the architecture was performed according to the 'monitoring phase' of the Extended Service Level Management presented in chapter 8. We used mobile agents to monitor service performance within single ISP network domains. Our development and implementation of the agent-based monitoring framework includes the setup of mobile agents, which are responsible for monitoring compliance with the QoS stipulated in the service contracts.

10.1 Mobile Agents for Inter-Domain Monitoring

As illustrated in the previous chapter 9, the setup of bilateral contracts in form of SLA/OLAs coordinates the cooperation of several business entities to provide a real-time streaming service to the end-customer. It ties every business entity to an official document that service customers can rely on if problems occur. Especially the service broker tries to represent a trustworthy entity to its end-customers. In order to maintain trustability and legitimacy of SLAs/OLAs, the existence of a monitoring system to observe compliance with the contracts is absolutely necessary. Therefore, we developed an agent-based monitoring framework to realize inter-domain monitoring across several autonomous domains of ISPs.

Monitoring of SLAs/OLAs is indispensable for an E-Commerce scenario where the end-customer pays money for guaranteed service provisioning and where the process of service delivery is dependent on several independent service providers (CSP and ISPs). The question to be discussed is why to use mobile agents for performing monitoring tasks [G01].

The term ‘agent’ is used by a number of different research communities, for instance within the fields of artificial intelligence (*intelligent agents*) [MJ99] or software engineering and distributed systems (*software agents*) [WJ99]. The *Object Management Group* (OMG) defines a software agent as *“a computer program that acts autonomously on behalf of a person or organization”* [IIFC97]. Additionally, agents are often characterized by the following properties:

- proactive (support of the user’s work)
- adaptive (learning the user’s preferences or the ability to work on different platforms)
- autonomous (not communicating with its creator)
- intelligent (making ‘intelligent’ decisions [Fer99])
- mobile (can actively migrate within networks to different systems and move directly to the local resources, like databases or application servers)

Most of the agents in our framework are mobile (see following section 10.2). Therefore, service providing systems need to install a so-called ‘agent-place’ to create, delete and execute agents. Then, agents can migrate from system to system and perform their tasks locally.

Lange and Oshima [LO99] give seven reasons why using mobile agents in general, which can be favorably applied for the given example of a network model as illustrated in Figure 7.2. For example, there is reduced network load if agents can process and evaluate monitoring data locally instead of being transmitted over the network. The use of mobile agents can also overcome network latency, especially in case of real-time applications like streaming Internet services where quick actions are required in case of service failure or shortcomings. Furthermore, mobile agents can adapt dynamically to changing environments and execute asynchronously and autonomously since they do not require a continuously open connection to a fixed network. And finally, mobile agents can encapsulate protocols by wrapping monitoring information and communicating it based on proprietary protocols.

Lange and Oshima have also defined some applications that benefit from using mobile agents. Included is ‘monitoring and notification’, where the main advantages are the asynchronous nature of the monitoring agents as well as the ability to monitor any given information source without being dependent on the system the information originates from. Providing an agent-place within the service providers’ networks allows the agents to have access to monitoring data in a controlled manner.

10.2 Agent-based Monitoring Framework

Within our developed agent-based monitoring framework, the setup of the agents with the different service providers and the service broker follows a four-step agent setup protocol, as illustrated in Figure 10.1. The protocol is independent from the contracting protocol where the SLA/OLAs are negotiated. The negotiation of service contracts can be performed for an immediate service or service reservations can be made in advance where the end-customer reserves a service that will be delivered in the future (e.g. watching a video stream of an Olympic hurdle race at some point in the future). If the service is provided immediately, the agent setup protocol is executed following the successful termination of the contracting protocol. In case the service is delivered in the future, the agent setup is triggered accordingly by the service broker at a reasonable time before the service.

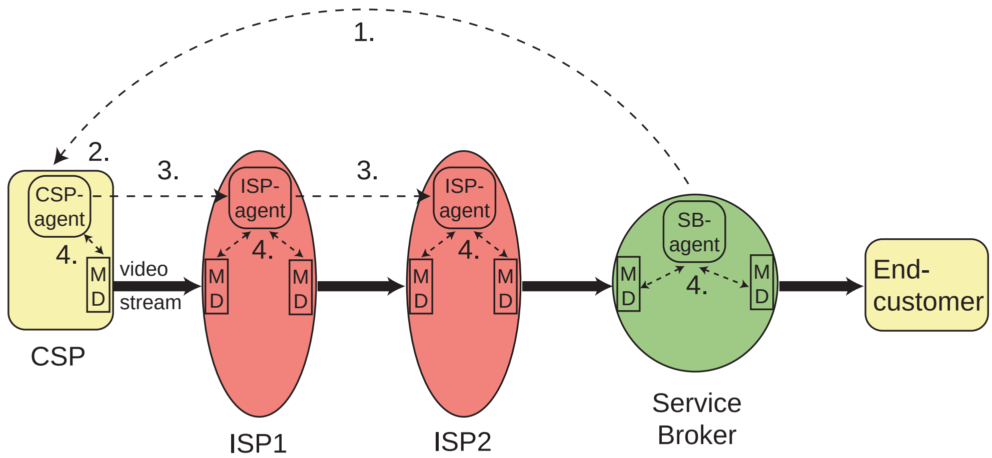

Figure 10.1: The setup of agents within the network of service providers (from [KSHS02]).

Within the monitoring framework, three different kinds of monitoring agents are distinguished at different locations. There is a *SB-agent*, a *CSP-agent*, and possibly several *ISP-agents* with the following functionalities:

SB-agent

- Monitoring of network communication and traffic parameters by collecting monitoring data delivered from the monitoring devices according to the specified filter
- Supervision of overall service provisioning

- Notification and taking steps in case of service failure or shortcomings
- Resides at the service broker

CSP-agent

- Monitoring of network communication and traffic parameters by collecting monitoring data delivered from the monitoring devices according to the specified filter
- Eventually, monitoring of system components and hardware devices that are responsible for transmitting the content (e.g. CPU, operating system, memory, etc.)
- Eventually, real-time evaluation of collected data
- Resides at the CSP

ISP-agent

- Monitoring of network communication and traffic parameters by collecting monitoring data delivered from the monitoring devices according to the specified filter
- Eventually, real-time evaluation of collected data
- Resides at the ISP

As illustrated in Figure 10.1, the sequence of steps to setup and place the different agents within the monitoring framework can be executed as follows (see also Kneer et al. [KSHS02]):

1. In time, before the service is to be monitored, the service broker triggers the setup of the agents by creating the SB-agent. The SB-agent creates a CSP-agent that migrates to the CSP's agent platform.

2. The CSP-agent creates an ISP-agent. The CSP-agent provides the ISP-agent with the necessary information about location of ISP1 in order for the ISP-agent to migrate there.

3. The created ISP-agent migrates to ISP1. Having arrived there at ISP1's agent platform, the ISP-agent generates a copy of itself (also referred to as *clone*), which becomes the future ISP-agent of ISP2. Similarly to step 2, the clone receives the necessary information about ISP2 and migrates there. This procedure of 'copying and migration' repeats until all the ISPs

along the communication path that contribute to the service provisioning process have received their ISP-agents. Information about the neighboring ISPs had been exchanged and stipulated by the bandwidth brokers (BB) in the previously executed contracting protocol.

4. The final step is to connect the ISP-agents as well as CSP- and SB-agent with the local *monitoring devices* (MD) through services and interfaces provided by the respective agent platform. Such monitoring devices could be agents as described in [GB00] or commercial monitoring tools provided for example by NeTraMet[1], Netflow[2], or IUM[3]. As illustrated in Figure 10.1, it is assumes that monitoring devices consist of two devices that monitor both incoming and outgoing traffic at the ingress and egress routers of the ISP network. This has the advantage that all incoming traffic from a particular neighboring ISP can be monitored at the ingress router and compared with the outgoing traffic through the egress router. However, another option could be to place the monitoring devices at different and distributed locations within the network where traffic is reduced and monitoring can be performed more specifically.

After the agents have migrated to the different platforms and the connection is setup to the local monitoring devices, monitoring can be performed during service provisioning according to the QoS specification as stipulated in the SLA/OLAs. After the service has been finished and no monitoring is required anymore, the agents will be deleted.

The CSP-agent and the ISP-agents are local agents that are autonomous but work on behalf of the service broker. That means there is a fixed correlation between the broker agent and CSP/ISP-agents. However, other service brokers also have their broker-agents, which also generate their own CSP/ISP-agents that migrate to CSP/ISP networks accordingly. The monitoring devices are installed locally within the networks and systems of SB/CSP/ISPs and their task is it to provide the SB/CSP/ISP-agents with the relevant monitoring data after successful connection. That means the local monitoring devices provide several local agents with monitoring data. Monitoring devices scan the IP traffic by simply filtering out the information that is relevant to the corresponding agent. The filter is communicated to the monitoring devices during the connection process with the SB/CSP/ISP-agent. Filter functions are provided through specifications on the part of the local agent platform (see section 10.4).

[1] Network Traffic Meter (NeTraMet); accessible at URL: http://www2.auckland.ac.nz/net/Accounting/ntm.Release.note.html

[2] Cisco IOS NetFlow; accessible at URL: http://www.cisco.com/warp/public/732/Tech/netflow/

[3] HP OpenView Internet Usage Manager (IUM); accessible at URL: http://www.hp.com/communications/usage/ium/index.html

The service broker as an intermediary between CSP and end-customer might cause a bundling of several single requests of end-customers to one compound request to the CSP. In such a case, one pair of monitoring devices does not have to monitor several single services but just the one bundled service. Still, the service specifications for the bundle of single services have to be correct and monitored accordingly. After receiving the bundle of data, it is the task of the service broker to provide the different end-customers with the corresponding data. For that case, the service broker has its own monitoring devices to monitor incoming traffic (in form of a bundle) and also traffic that goes to every single end-customer (end-customer oriented monitoring).

10.3 Monitoring the Service Performance

Monitoring in general is applied in different areas (e.g. workflow applications, banking systems, web content) where monitoring is performed in form of data logging and reporting and simply serves for documentation purposes. However, within the agent-based monitoring framework, monitoring data is collected by SB/CSP/ISP-agents in order to evaluate the service delivery process, and as a consequence to take steps in case of service failure or shortcomings as defined in the SLA/OLAs. Therefore, we subdivided monitoring into *static monitoring* and *dynamic monitoring*[4]. Static monitoring implies a service evaluation after the service is performed, while dynamic monitoring foresees an examination of collected monitoring data 'on-the-fly' while the service delivery process is being executed. Dynamic monitoring is very useful since the corresponding service provider can be notified already as a preventive measure in case of service shortcomings. Thus, complete service failures can eventually be avoided (corresponds to the reciprocal interaction between delivery and monitoring phase as described in section 8.1 and illustrated in Figure 8.2).

The monitoring devices in Figure 10.1 scan IP traffic by filtering relevant data for the corresponding agent and service with respect to service parameters such as delay, bandwidth, packet loss, etc. With static monitoring, an evaluation of the monitoring data is performed by the SB after the service delivery and after the CSP/ISP-agents have migrated back to the SB with the monitoring data attached. Eventually, the CSP/ISP-agents undertake a pre-evaluation of the results in order to reduce the amount of data before migrating back to the SB. With dynamic monitoring, evaluation of monitoring data is performed constantly by all the monitoring agents and therefore, the CSP/ISP-agents maintain a communication path to the SB-agent and exchange 'on-the-fly' evaluated monitoring data concerning

[4]Within the extended SLM process (section 8.1), it was distinguished between *reactive* and *proactive* controlling or monitoring, respectively.

the service level. In case of service shortcomings, the local BB can additionally be notified (preventive measure to avoid service failure). That means a communication path between the CSP/ISP-agents and the SB-agent exists for exchanging monitoring information.

With static monitoring, consequences in case of non-compliance with the service contract can be taken after the evaluation of all monitoring data. With dynamic monitoring, steps in case of service failures can be taken immediately when discovered. This could result in an exchange of a malfunctioning ISP 'on-the-fly' during service delivery. An overview of differences between static and dynamic monitoring is given by the table in Figure 10.2.

	Static Monitoring	**Dynamic Monitoring**
Functionality	Evaluation of monitoring data is performed by the SB after the service delivery	Evaluation of monitoring data is performed by all monitoring agents during service delivery
Communication	There exists no communication among the SB, CSP, and ISPs	A communication path exists between the CSP/ISP-agents and the SB-agent
Type	Reactive Monitoring	Preventive Monitoring
Consequences	Delayed consequences after evaluation of monitoring data	Immediate consequences are possible

Figure 10.2: Comparison of static and dynamic monitoring within the agent-based monitoring framework (from [KSHS02]).

10.4 Implementing the Monitoring Framework

We designed the agent-based monitoring framework with one standardized agent platform. The platform is installed at the SB, CSP, and ISPs and provides a minimal set of standardized functions as described in the following. Any extension of monitoring functionality is achieved by extending the corresponding agent's functionality. This keeps the agent platform simple and small, supports a quick diffusion of the platform and guarantees a better reliability. The agent platform consists of three areas: *In/Out Area*, *Working Area* and *MD Connection Area* (see also Figure 10.3).

As described in section 10.2, agents migrate from SB to CSP and on to different ISPs. For security reasons, the agents' source code will be encrypted and digitally signed by the SB before they migrate. Therefore, it is impossible to change or to forge the agent while it is transferred from one provider to the other. After arriving in the In/Out Area the system decrypts the agent, checks its integrity, and verifies the authorization. The authorization is verified with the aid of a digital signature. After the agent's successful authorization, it can proceed to the Working Area where the agent can execute all implemented functions and communicate with the MDs with the aid of an interface provided by the MD Connection Area. The MD Connection Area is different for every provider because not all the MDs are standardized and the framework is MD-independent. Therefore, this area will not be described in more detail.

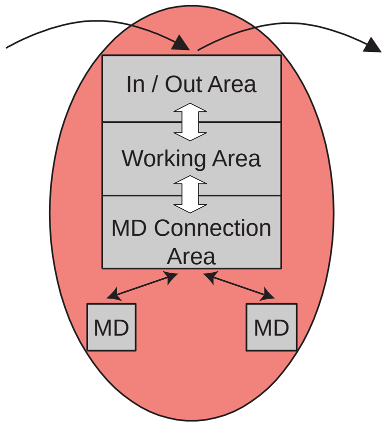

Figure 10.3: The agent platform (from [KSHS02]).

10.4.1 In/Out Area

The In/Out Area performs the following security checks on the mobile agents. The checks are implemented with standard security functions like public key encryption, message authentication code and digital signature [Sch95].

- Decryption of agents and information
- Integrity check
- Verification of Authorization
- Encryption and transfer to the next platform

10.4.2 Working Area

The Working Area serves to execute agents and to provide access to the MDs through the MD Connection Area. The agents are executed independently from each other and the area guarantees that an agent cannot change other agents and can only use the allocated resources. The Working Area must support the following functions, consisting of filters and events. All the filters and events are stored in a database. After the agent is deleted all the filters and events will be automatically deleted, too.

- **OpenMD.** This function opens a connection to the MD (of either the ingress or egress router) and allows to monitor incoming or outgoing traffic related to the corresponding SLA/OLA.

- **RegisterFilter.** The agent can define an unlimited number of filters and use them for monitoring a stream (specified by source and destination address). To create such a filter for a certain stream the agent can define the following parameters among other things:

  - *size*: This parameter defines the size of the captured data.
  - *number*: This parameter defines how often a filter is executed.
  - *interval*: The interval defines the delay between two filter accesses.

  RegisterFilter(size=500, number=10, interval=2) means this filter is executed 10 times every two minutes where 500 bytes are sent to the agent.

- **StartFilter.** With help of this function the agent can start a monitoring process with a specific filter. The function requires a reference to an already defined filter in order to open connection to a MD.

- **StopFilter.** The monitoring with a specific filter will be stopped. The function requires a reference to an already activated filter.

- **RegisterEvent.** This function allows to define an event (e.g. MD failure). An event allows to start or stop a filter when a specific event occurs. The function requires an event, a reference to an already defined filter, and an open connection to a MD. The agent can define an unlimited number of events.

- **StartEvent.** This function activates an already registered event. After the event occurs the agent will be notified and monitoring with the specific filter will be started or stopped.

- **StopEvent.** This function deactivates an already activated event.
- **DeleteFilter.** An already defined filter will be deleted.
- **DeleteEvent.** An already defined event will be deleted.

10.5 Summary

Chapter 8 described different phases of the extended SLM process within our proposed Extended Service Level Management. As an implementation of the 'negotiation phase', chapter 9 introduced our developed architecture for negotiating and trading service contracts. This chapter 10 presented our architecture for monitoring compliance with service contracts as an implementation of the 'monitoring phase'. Mobile agents were used to monitor service compliance with SLAs/OLAs. The architecture is based on assumptions about network infrastructure and resource management (e.g. resource reservation, differentiated services, QoS-based transport service) that are currently not fully applied by service providers. However, these assumptions are advantageous and will eventually be deployed for QoS-based service provisioning and provide a basis for the stipulation of service contracts and their compliance. The following chapter 11 practically illustrates how service contracts are negotiated and how their compliance is monitored with the agent-based monitoring framework.

Chapter 11

Practical Illustrations

The following practical example of use illustrates the processes of negotiating service contracts and monitoring their compliance. Furthermore, the example shows the advantages of having a service broker as the point of contact and the applicability of our agent-based monitoring framework. The example illustrates the use of our contracting protocol and monitoring framework by means of an E-Commerce scenario in which an end-customer decides to use a Video-on-Demand application, and therefore requests a video stream from a remotely located CSP. An implementation of the case requires negotiation of service levels, before the service is delivered and finally monitored (compare to the three sub-phases of the execution phase illustrated in Figure 8.2).

11.1 Negotiation Process

Figure 11.1 shows the negotiation process within a scenario with end-customer, CSP, service broker, and three ISPs. The figure illustrates all the different steps of our contracting protocol as described in section 9.2. In step 1, the end-customer specifies the Service Level Requirements (SLR) towards the service broker as the end-customer's point of contact. Besides general contractual information, the SLR contains information about service functionality and performance, monitoring and reporting, and requests about customer support.

In any case, the end-customer needs to reveal its identification. The end-customer wants to watch a movie from 8.00-10.00 pm which is equivalent with the contract duration. The end-customer wants to spend at most 8.00 Swiss Francs (CHF) after the service will have been successfully delivered. Payments are made by credit card. For specifying the service performance, the end-customer requests 256 colors, 16 frames per second with a frame size of 352 * 288 pixel (corresponding to the Common Interchange Format (CIF)). The availability of the service is important to the end-customer and is, therefore, chosen to be 99.9% of the total service duration (which is a very common value on today's services market). For specifying the reliability, the end-customer can choose for instance from three options: very good (service interrupt $< 0.1\%$ of the total service duration), medium (service interrupt $< 0.5\%$), acceptable (service interrupt $< 1\%$). The service metrics and indicators do not have to be specified separately since they are set by the chosen service level. However, the end-customer chooses to have the service monitored for 1 minute every 10 minutes to make sure that the service meets its targets. Furthermore, the end-customer wants to be notified about the quality of the delivered service hourly and in case of service failure. Finally, help desk and an information hotline is desired, in case problems occur and the end-customer needs assistance. An overview of the specified service parameters of the SLR is shown in the middle column of Figure 11.2.

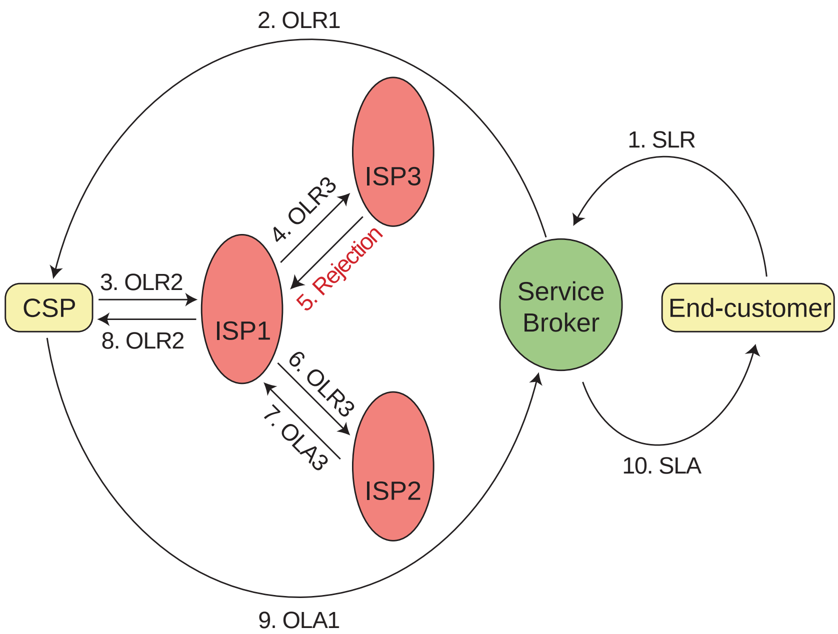

Figure 11.1: Negotiation process according to the contracting protocol as defined in section 9.2.

Step 2 implies the translation of the SLR into a technical description of service specifications as contained in the Operational Level Requirements (OLR). The translation process is executed by the service broker before OLR1 is forwarded to the CSP. Similar to the SLR, the OLR contains the service broker identification and the service duration from 8.00-10.00 pm. Charging and pricing information is changed to 6.00 CHF that the service broker has to pay the CSP after successful service delivery. It is assumed that service delivery requires four service providers whereas each charges 2.00 CHF for the service. In reality, service charges are independent from each other and in order to find out the exact price of a service, a separate price query could be envisioned where service providers along the path from CSP to end-customer add up their charges to a final price revealed to the end-customer. Payment among the service providers is performed with bank transfers using the edge billing scheme (see section 7.4.4).

A crucial point in translating the SLR into the OLR is the technical specification of performance parameters in terms of network and system resource measures. The end-customer requests for 256 colors, 16 frames per second, and a frame size of 352 * 288 pixel result in 352 * 288 (frame size) * 16 (frames/s) * 8 (bit for colors) = 1.261.568 Bit/s. This corresponds to a maximum bandwidth of 1.3 MBit/s[1] that the service broker and the other service providers have to provide in order to fulfill the OLR. Of course, this bandwidth is only realistic if the end-customer has access to a fast Internet connection. The end-to-end delay as another performance parameter can be set per default to 0.5 seconds per provider, which adds up to 2 seconds in total with four service providers. For a streaming video the end-to-end delay of 2 seconds is very realistic and does not influence the service quality. In case of video conferencing, the value has to be set much lower since this kind of interactive real-time application requires low end-to-end delays (up to 150 ms only). Jitter describes the variation of end-to-end delays of single, transmitted data packets. A jitter value of 100 ms is realistic and can be taken as a default value knowing that the value is dependent on the buffer size of the processing systems. Specification of reliability parameters by the end-customer can be translated into packet loss and error rates. A bit error rate of less than 10^{-12} is common for today's data transport with fiber cable.

The translation of the 'monitoring and reporting' section of the SLR is done by choosing static monitoring based on the fact that the end-customer only pays after successful service delivery, which implies an evaluation of the service after it has been provisioned. Other service metrics such as the end-to-end communication, average throughput, or failure rate can be taken over automatically from specifying service performance and service levels. Other internal metrics for CPU power, memory, and network response times must be calculated individually by the service providers in order to meet the targets. The table in Figure 11.2 illustrates by means of some selected service requirements how a SLR can be translated into the technical specifications of an OLR. If the service providers agree to the specifications of SLR/OLR, they will become SLA/OLA. Note that the items in the table are only a selection and do not represent the complete content of SLR/SLA and OLR/OLA as described in section 8.2.

[1]The value of 1.3 MBit/s describes the maximum bandwidth without compression. Due to video compression techniques, the required bandwidth could be decreased further.

	SLR / SLA	OLR / OLA
Identification	End-customer ec_x Service Broker sb_y	Service Broker sb_y CSP csp_z
Contract Duration	8.00 - 10.00 pm	8.00 - 10.00 pm
Charging and Pricing	8.00 CHF after successful service	6.00 CHF after successful service
Payment	Credit card payment	Bank transfer with edge billing
Performance	colors = 256 frame rate = 16 frames/s frame size = 352 x 288 pixel	bandwidth = 1.3 MBit/s end-to-end delay = 0.5 s jitter = 100 ms
Availability	99.9 % of service duration	99.9 % of service duration
Reliability	very good (service interrupt < 0.1 %)	packet loss rate < 0.1 % bit error rate < 10^{-12} (fiber)
Service Metrics / Indicators	monitoring periodicity = 1 minute every 10 minutes service metrics = service performance values	static monitoring (1 every 10 min) end-to-end comm. ≤ 2s avg. throughput ≥ 1.3 MBit/s service failure rate ≤ 0.1 % internal metrics for CPU power, memory, and network response times
Notification	hourly and in case of service failure	hourly and in case of service failure
User Support	help desk and information hotline	help desk and information hotline

Figure 11.2: Comparison of selected service requirements as specified within SLR/SLA and OLR/OLA.

Step 3 of Figure 11.1 shows the forwarding of OLR2 to ISP1. After checking its own resources, ISP1 creates OLR3 and forwards it to its neighboring ISP3. ISP3 does not have resources available to fulfill OLR3 and rejects it (step 5). Thereupon, ISP1 requests the service from ISP2 (step 6), which accepts the service conditions and responds with OLA3 (step 7). ISP1 signs OLR2 and forwards OLA2 to the CSP (step 8). The CSP returns OLA1 to the service broker that can now only sign the SLR and forward the SLA to the end-customer (step 10). After step 10 is performed, the negotiation process is terminated and the service can be delivered accordingly.

11.2 Monitoring Process

Before the service delivery process is executed, the service broker initiates the setup of the different monitoring agents for monitoring compliance with the service contracts as described in section 10.2. First, a SB-agent is created and positioned in the working area of the service broker. Then, the SB-agent creates a CSP-agent that migrates to the CSP's working area. Afterwards, the CSP-agent creates an ISP-agent that migrates to ISP1. Finally, the ISP-agent of ISP1 generates a clone of itself that migrates to ISP2. Now, the different agents open connections with the corresponding local monitoring devices (MD) as described in section 10.4. By specifying filters, the agents can request from their MDs the kind of information they are interested in. By the time service provisioning starts, the agents are ready to perform their monitoring tasks. Figure 11.3 shows the setup of the agents to monitor the SLA and OLAs.

Since streaming is very sensitive to the transmission of subsequent packets, end-to-end delay and jitter were specified within the OLA to be 0.5 seconds and 100 ms, respectively. Therefore, the ISP-agents open a connection to their ingress and egress MDs and collect monitoring information about incoming and outgoing packets as defined in the monitoring filter. Since the end-customer chose a monitoring duration of 1 minute every 10 minutes, monitoring is performed in regular time intervals. If equipped with the right functionality, the ISP-agents could calculate the delay time and jitter for the monitored data packets and preselect and store only the kind of data that is not in compliance with the OLA. This information can then be used by the service broker to take measures against the malfunctioned ISP. Since static monitoring was chosen, monitoring information of all ISPs is collected and evaluated after the service delivery. Sanctions are executed by the SB-agent after the service is finished. Sanctions are specified in the 'customer support' section of a SLA/OLA and help to build up trust between service provider and customer.

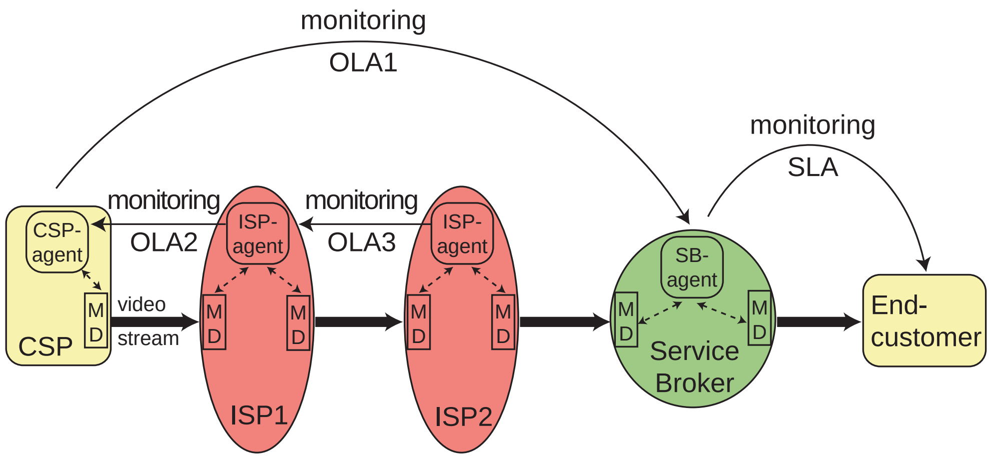

Figure 11.3: Setup of the different agents to monitor the SLA and OLAs (from [KSHS02]).

To conclude, the execution of the contracting protocol and the application of the agent-based monitoring framework provide means for the Extended Service Level Management in order to satisfy the end-customer for receiving the service he/she required, but also to satisfy service providers for provisioning services based on a systematic management of network resources.

Chapter 12

Summary

This final chapter summarizes this thesis, highlights scientific contributions and major conclusions, and provides an outlook on future work.

12.1 Scientific Contributions

The delivery of E-Commerce services over the Internet based on QoS requirements is a big problem for today's ISPs. As an approach to solve that problem, we identified different levels of abstraction to be necessary for the provisioning of streaming Internet services. On the lowest 'communication level', Internet Protocol Architectures provide the basis for reliable transport of data across the Internet. On the top 'business level', there is our Integrated Business Model with charging and accounting functionality and interacting business entities. Finally, Extended Service Level Management as controlling instance on the middle 'contracting level' is required for combining the Integrated Business Model and Internet Protocol Architectures on a contractual basis to implement service provisioning on the Internet (compare to Figure 1.1).

Within this thesis, we developed an Integrated Business Model illustrating a business process with independent and coherent phases necessary for charging and accounting of Internet services. The Integrated Business Model comprises a clear separation of different roles, a concise definition of the services they offer, and a characterization of their interactions and mechanisms within a business process. Special roles within the Integrated Business Model are played by the Access ISP, the entity that provides Internet access on the 'communication level', and the ESP, which provides E-Commerce service on the 'business level'.

Within the Integrated Business Model, we focused on two kinds of payments that have influence on the phases of the business process: pre-paid payments for anonymous end-customers and account-based post-paid payments for registered end-customers. Billing and payment are parts of an essential clearing process between service provider and customer. Clearing is part of the economic policy that regulates financial compensation of services. The policy considers registered as well as anonymous end-customers, but prefers account-based payments in case of service failures, when pre-paid payments have been made already. Account-based payments are advantageous, since clearing is performed on a pay-per-use basis where the end-customer only pays after the actual service is received.

An important step towards enabling QoS for streaming Internet services is performed by our Extended Service Level Management. Extended Service Level Management on the 'contracting level' represents the connector between the top level Integrated Business Model and the bottom level Internet Protocol Architecture. Extended Service Level Management provides means to specify, quantify, and negotiate service levels and stipulate them within contractual agreements. We proposed a structure for the content of service contracts in the form of Service Level Agreements (SLA) and Operational Level Agreements (OLA) that stipulate service conditions for cooperating business partners (e.g. neighboring ISPs).

Furthermore, we designed and implemented a contracting protocol in the form of the Simple Protocol and the Time Slice Protocol to illustrate the use of SLAs/ OLAs in a multi-provider environment. Our developed Video-on-Demand Contracting Architecture (VODCA) served to simulate a test scenario and to demonstrate practical usability of the contracting protocol. Although the network structure of the simulated E-Commerce scenario was simple, the goal of the contracting protocol to combine multiple ISPs in a corporate task was met. As an enhancement of simple SLA negotiation we designed a marketplace with service brokers to enable trade of SLAs as a new possibility to utilize network resources more efficiently and to match supply and demand of services in an optimal manner.

But what is the benefit of contracts without a monitoring system to control their fulfillment or react with punishments accordingly? Therefore, we developed a framework with mobile agents for monitoring compliance with service contracts, either statically or dynamically. Advantages of the monitoring framework are its generic usability for monitoring all kinds of different services and parameters (e.g. delay time, error rate, correct order of data packets), its dependence on a number of small and cooperative agents, and its customer-friendliness.

To summarize, the following achievements were made:

- A holistic model with different levels of abstraction for the provisioning of streaming Internet services
- An Integrated Business Model for charging and accounting of Internet services
- An Extended Service Level Management based on service contracts to stipulate QoS-based Internet services

12.2 Conclusions

The Integrated Business Model and the Extended Service Level Management were developed independently from any underlying Internet Protocol Architecture, and are therefore applicable to IntServ, DiffServ, and MPLS architectures.

Applicability in a real-world scenario requires the implementation of the contracting protocol by all the service providers involved in the scenario, e.g. by means of a handshake protocol similar to our implementation within VODCA. It does not require special communication protocols for the service negotiation but can be implemented with the TCP/IP protocol.

The design of the contracting protocol and its process flow is simple enough to be potentially acceptable for the service providers to implement in order to offer QoS-based services to their customers. In addition, the contracting protocol is not limited for use with streaming applications only but with any QoS-based services, whose Service Level Requirements need to be negotiated beforehand.

As to complexity and scalability of the contracting protocol, performance checks were not made within this thesis but scalability problems are not foreseeable since the amount of ISPs is limited and network complexity of interconnected ISPs is manageable.

The idea to reserve network bandwidth in advance and stipulate its usage within service contracts is not limited to the service provisioning of streaming Internet services but can be applied to other resources and services (e.g. human resources, materials logistics). An interesting area of application for the usage of service contracts could be envisioned within grid computing where computing power and resources are geographically distributed. Similarly, our developed agent-based monitoring framework is not limited to mobile agents only and service provisioning on the Internet but monitoring can be performed by any form of agent (e.g. human agent) for any service delivery process based on a service contract.

Potential users and beneficiaries of an implementation of the Extended Service Level Management for the Internet are those service providers that attach great importance to professional service delivery, and it is beneficial for those service customers that are willing to pay for service guarantees and their compliance.

12.3 Outlook

Further work is required in the fields of the ‘business level’, ‘contracting level’, and ‘communication level’. On the ‘business level’, it remains uncertain whether charging and accounting for the data transport will be separated from the content. Regularly changing Internet technologies and a broad variety of service providers and customers with different demands and expectations complicate the development and use of generic business models and require exact differentiation of E-Commerce scenarios. However, the business process in combination with the extended SLM process made a step towards an integration of E-Commerce and service delivery through the introduction of contractual agreements, charging and accounting mechanisms, billing, and payment. The potential of having business processes ready, efficient technology in place, and simulation experiences available for contracting and monitoring provides a basis for experimentation as well as for long-term collection of experiences.

Future work needs to be done on the ‘communication level’ in embedding technological know-how into economic models and business processes including reservation, performance, and clearing of services. Basic approaches to mapping these phases of the business process onto technological Internet Protocol Architectures were provided within this thesis but work is required to specify a detailed implementation. Reservation of network resources could be performed by using intelligent agents as bandwidth brokers. Dynamic pricing models for usage-based accounting of information also needs further investigation as payment methods are still lacking trust for a wide area of use within E-Commerce applications. The growing number of mobile and anonymous service customers and Internet users as well as new mobile Internet technology is a clear indication of changing customer expectations on today’s service market.

Continuing work is also required on the ‘contracting level’ within the framework of Extended Service Level Management. Negotiating service levels and implementing the contracting and monitoring in a multi-provider scenario with heterogeneous infrastructure is difficult and requires standards for protocols and interfaces. Both the Simple Protocol and the Time Slice Protocol were developed on an application level, which still need to be adjusted and integrated with lower level protocols (e.g. RSVP) for reservation and allocation of network resources. The contracting protocols are limited to finding an arbitrary path between end-customer and CSP. Business solutions for VoD applications would need to optimize routes since every intermediate business entity increases the total price for the service delivery. Finally, business solutions for VoD applications must include security mechanisms to ensure authenticity, non-repudiation, and integrity of exchanged contracts, service availability, and to protect the digital content and intellectual property against fraud and theft.

Appendix A

Abbreviations

AS	Autonomous System
ABR	Available Bit Rate
ADSL	Asymmetric Digital Subscriber Line
AF	Assured Forwarding
AISP	Access Internet Service Provider
API	Application Programming Interface
ARM	Application Response Monitoring
ARPA	Advanced Research Projects Agency
ASP	Application Service Provider
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
BB	Bandwidth Broker
BPM	Business Process Management
BSI	British Standards Institution
CA	Certification Authority
CBR	Constant Bit Rate
CGI	Common Gateway Interface
CIF	Common Interchange Format
CISP	Core Internet Service Provider
CPN	Customer Premises Network
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CSP	Content Service Provider
DBMS	DataBase Management System
DiffServ	architecture for Differentiated Services
DOM	Document Object Model
DSCP	DiffServ Code Point
DSL	Digital Subscriber Line
DTD	Document Type Definition

EF	Expedited Forwarding
ESP	E-Commerce Service Provider
FCFS	First-Come-First-Served
FDDI	Fiber Distributed Data Interface
FDM	Frequency-Division Multiplexing
FEC	Forwarding Equivalence Class
FIFO	First-In-First-Out
FQ	Fair Queuing
FR	Frame Relay
GQM	Goal/Question/Metric paradigm
HDSL	High-bit-rate Digital Subscriber Line
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IntServ	architecture for Integrated Services in the Internet
IP (IPv4)	Internet Protocol (version 4)
IPnG (IPv6)	Internet Protocol next Generation (version 6)
IPA	Internet Protocol Architecture
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ISP	Internet Service Provider
ITIL	IT Infrastructure Library
itSMF	IT Service Management Forum
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
IUM	Internet Usage Manager
JPEG	Joint Photographics Expert Group
JVM	Java Virtual Machine
LAN	Local Area Network
LDP	Label Distribution Protocol
LSP	Link State Packet
LSR	Label Switching Router
MAN	Metropolitan Area Network
MBone	Multicast Backbone
MD	Monitoring Device
M-JPEG	Motion-JPEG
MPEG	Moving Pictures Expert Group
MPLS	MultiProtocol Label Switching
NCSA	National Center for Supercomputing Applications
NeTraMet	Network Traffic Meter

nrt-VBR	non-real-time Variable Bit Rate
NSF	National Science Foundation
NSFNET	National Science Foundation Network
NTSC	National Television Systems Committee
OGC	Office of Government Commerce
OLA	Operational Level Agreement
OLR	Operational Level Requirements
OMG	Object Management Group
OSI	Open Systems Interconnection
PAL	Phase Alternate Line
PHB	Per-Hop-Behavior
PKI	Public Key Infrastructure
PSTN	Public Switched Telephone Network
PVC	Permanent Virtual Circuit
QoS	Quality of Service
RADSL	Rate Adaptive Digital Subscriber Line
ReRA	Resource Reservation in Advance
RFC	Request for Comments
ROI	Return On Investment
RSVP	Resource ReSerVation Protocol
rt-VBR	real-time Variable Bit Rate
RTCP	Real-Time Control Protocol
RTP	Real-time Transport Protocol
RTT	Round Trip Time
SAX	Simple Application programming interface for XML
SB	Service Broker
SCMP	Stream Control Message Protocol
SDSL	Symmetric Digital Subscriber Line
SGML	Standard Generalized Markup Language
SI	Service Indicators
SLA	Service Level Agreement
SLM	Service Level Management
SLR	Service Level Requirements
SLS	Service Level Specification
SM	Service Management
SPF	Shortest Path First
ST-II	Stream Protocol-II
STDM	Synchronous Time-Division Multiplexing
SVC	Switched Virtual Circuit
TCP	Transmission Control Protocol
TCS	Traffic Conditioning Specification

TINA	Telecommunication Information Networking Architecture
TINA-C	Telecommunication Information Networking Architecture Consortium
TMN	Telecommunications Management Network
TOS	Type of Service
TTL	Time-To-Live
TTP	Trusted Third Party
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
VCI	Virtual Circuit/Channel Identifier
VDSL	Very-high-data-rate Digital Subscriber Line
VoD	Video-on-Demand
VODCA	Video-on-Demand Contracting Architecture
VPI	Virtual Path Identifier
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WAN	Wide Area Network
WWW	World Wide Web
XML	eXtensible Markup Language
XQL	XML Query Language
XTP	Xpress Transport Protocol

Appendix B

Glossary

Definition 2.1 (Quality of Service) . . . . (chapter 2.4.1) - page 23
Quality of Service (QoS) is defined by a set of parameters that specify the performance of a service in both qualitative and measurable manner.

Definition 5.1 (Accounting) . . . . (chapter 5.3.1) - page 67
Accounting is defined as collecting user information for the utilization of a certain service or for the usage of network resources. This information includes technical parameters such as throughput for the service, service duration, delays, loss rates, etc. Accounting determines a function that transforms network resources into technical values that are stored in so-called **accounting records**. This record depends on both the underlying network infrastructure (e.g. ATM, Frame Relay) and the provided service.

Definition 5.2 (Charging) . . . . (chapter 5.3.1) - page 67
Charging is defined as the price calculation for an accounting record based on a pricing scheme for that service. Thus, charging determines a function that transforms technical values, stored in accounting records, into monetary values that are stored in so-called **charging records**.

Definition 5.3 (Pricing) . . . . (chapter 5.3.1) - page 67
Pricing is defined as the process of setting a price for a service. The price determines how much a user for example has to pay for transmitting one Megabyte of data in the afternoon over a distance of ten hops. This price depends on the business structure of the ISP or the current market situation.

Definition 5.4 (Billing) . . . . (chapter 5.3.1) - page 67
Billing is defined as the process of collecting user-specific charging records, adding them up, and generating a bill for that specific user. A bill contains the user information within a time period (e.g. one month). The bill summarizes the full cost a service user has to pay.

Definition 6.1 (Service) .... (chapter 6.1.1) - page 77
A service is an operating business activity of a service provider with beneficial value to a service customer and financial compensation.

Definition 6.2 (Service Management) .... (chapter 6.1.1) - page 78
Service management describes a framework of single processes to support, plan, control, and administrate service delivery at customer satisfaction.

Definition 6.3 (Service Delivery) .... (chapter 6.1.1) - page 78
Service delivery describes the activity of a service provider to provide a service to a customer.

Definition 6.4 (Service Provisioning) .... (chapter 6.1.1) - page 78
Service provisioning describes service delivery under compliance with QoS requirements. This might include the reservation of network resources.

Definition 6.5 (Service Level Management) .... (chapter 6.3.1) - page 89
Service Level Management (SLM) describes the process of specifying, negotiating, agreeing, and monitoring quality and quantity of service levels in form of service parameters based on a fair price and contractual agreements between service provider and customer.

Definition 6.6 (Service Level) .... (chapter 6.3.1) - page 90
A service level describes a quantifiable degree of service performance based on the specification of service parameters (QoS) including the price.

Definition 6.7 (Service Level Agreement) .... (chapter 6.3.1) - page 90
A Service Level Agreement (SLA) is a written contract between a service provider and its customer that specifies and documents the service levels in terms of content, quantity, and quality of service (QoS) including the price.

Definition 6.8 (Operational Level Agreement) ...... (chapter 6.3.1) - page 90
An Operational Level Agreement (OLA) is a written underpinning contract between two service providers that stipulates the supply of additional services to operate and fulfill the Service Level Agreement. In that case, one service provider becomes a customer of the other.

Definition 6.9 (Service Level Requirements) .... (chapter 6.3.1) - page 91
Service Level Requirements (SLR) form the starting basis in the process of service level negotiation where the service customer makes demands on the service level regarding content, quantity, and quality of service (QoS).

Definition 6.10 (Operational Level Requirements) . . (chapter 6.3.1) - page 91
Operational Level Requirements (OLR) represent a means between two service providers to describe the technical demands for supporting a service as specified within the SLR, i.e. regarding content, quantity, and quality of service (QoS).

Definition 6.11 (Metric) . . . . (chapter 6.3.1) - page 91
A metric is a quantifiable measure used to calculate the performance or efficiency of a process, system, or network component.

Definition 6.12 (Service Indicators) . . . . (chapter 6.3.1) - page 91
As opposed to direct measures for characteristics, Service Indicators (SI) are a set of pre-defined metrics used to determine both the required and the actual provided service level (performance). It thus provides an indirect measure of quality for the provided service.

Definition 6.13 (Service Quality) . . . . (chapter 6.3.1) - page 91
The service quality determines the degree of accordance between the measured service level (performance) and the stipulated service level. It thus represents the difference between actual value and target value of the service level.

Definition 7.1 (E-Commerce scenario) . . . . (chapter 7.1) - page 102
An E-Commerce scenario is an economic system of interconnected and cooperative service providers with the goal to provide services to service customers via the Internet.

Definition 7.2 (E-Commerce Service Provider) . . . . (chapter 7.2.1) - page 104
An E-Commerce Service Provider (ESP) is a business role within an E-Commerce scenario that offers products, digital contents, and services to end-customers via the Internet.

Definition 7.3 (End-customer) . . . . (chapter 7.2.1) - page 105
An End-customer is a business role within an E-Commerce scenario that buys products, digital contents, or services from the ESP via the Internet.

Definition 7.4 (Internet Service Provider) . . . . (chapter 7.2.1) - page 106
An Internet Service Provider (ISP) is a business role within an E-Commerce scenario that provides data transport services (Core ISP) and eventually Internet access (Access ISP) for end-customers.

Definition 7.5 (Customer Premises Network) . . . . . . (chapter 7.2.1) - page 107
A Customer Premises Network (CPN) within an E-Commerce scenario is a

facility of accumulated end-customers with a common policy and connection to an Access ISP.

Definition 7.6 (Service Broker) . . . . (chapter 7.2.1) - page 107
A Service Broker (SB) is a business role within an E-Commerce scenario that acts as a negotiator between service provider and service customer to match supply and demand.

Definition 7.7 (Marketplace Provider) . . . . (chapter 7.2.1) - page 107
A Marketplace Provider is a business role within an E-Commerce scenario that provides an environment (i.e. a marketplace) for service providers and service customers to meet and trade products, digital contents, and services.

Definition 7.8 (Certification Authority) . . . . (chapter 7.2.1) - page 108
A Certification Authority (CA) is a trusted business role within an E-Commerce scenario that provides security services for other business roles by issuing digital certificates.

Definition 7.9 (Payment Provider) . . . . (chapter 7.2.1) - page 108
A Payment Provider is a trusted business role within an E-Commerce scenario that provides financial services by transferring money between service providers and service customers.

Bibliography

[ADF+01] Loa Andersson, Paul Doolan, Nancy Feldman, Andre Fredette, and Bob Thomas. LDP Specification. Request for Comments RFC 3036, IETF, 2001.

[All01] Doug Allen. The Impact of Peering on ISP Performance: What's Best for You? *Network Magazine*, November 2001. http://www.networkmagazine.com.

[And00] Ragnar Andreassen. Requirements Specifications; Reference Model. Deliverable of the M3I European Project IST-1999-11429, The M3I Consortium, 2000.

[Aut00] Solcon ITM & WISS Autorenteam. Service Level Management. Lernunterlagen für die Wirtschaftsinformatikschule Schweiz (WISS), Version 0.10, September 2000.

[Bai97] Brigitte Bailer. *Geschäftsmodelle: Methoden und Qualität*. PhD thesis, Department of Information Technology, University of Zurich, 1997.

[Bal96] Larry L. Ball. *Multimedia Network Integration & Management*. McGraw-Hill, 1996.

[BBC+98] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An Architecture for Differentiated Services. Request for Comments RFC 2475, IETF, 1998.

[BC94] Robert Braden and David Clark. Integrated Services in the Internet Architecture: an Overview. Request for Comments RFC 1633, IETF, 1994.

[Bir01] Ben Birkhofer. *E-Commerce als innovativer Absatzkanal - Ein entscheidungsorientiertes Modell*. PhD thesis, Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften (HSG) der Universität St. Gallen, 2001.

[BP97] Leonard L. Berry and A. Parasuraman. Listening to the Customer - The Concept of a Service-Quality Information System. *Sloan Management Review*, pages 65–76, Spring 1997.

[BR88] Victor R. Basili and Hans Dieter Rombach. The TAME Project: Towards Improvement-Oriented Software Environments. *IEEE Transactions on Software Engineering*, 14(6):758–773, 1988.

[Bru00] Marcus Brunner. *Service Management in a Telecom Environment based on Active Network Technology*. PhD thesis, Swiss Federal Institute of Technology Zurich, 2000.

[Buc99] Thomas Buchheister. IT-Outsourcing - Analyse der Option Outsourcing im Betrieb Informatik bei der Rentenanstalt / Swiss Life. Semesterarbeit am Institut für Informatik der Universität Zürich, 1999.

[Buc00] Thomas Buchheister. Service Level Management - Theorie und Praxis. Master's thesis, Institut für Informatik der Universität Zürich, 2000.

[But99] Janet Butler. Bridging IT to the Business - SLM a language both can understand. *Software Magazine*, September 1999. http://www.softwaremag.com.

[BZP90] Leonard L. Berry, Valerie A. Zeithaml, and A. Parasuraman. Five Imperatives for Improving Service Quality. *Sloan Management Review*, pages 29–38, Spring 1990.

[CC00] Nigel Chapman and Jenny Chapman. *Digital Multimedia*. John Wiley & Sons, LTD., 2000.

[CD98] Alex Conta and Steve E. Deering. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) - Specification. Request for Comments RFC 2463, IETF, 1998.

[CHW99] Jon Crowcroft, Mark Handley, and Ian Wakeman. *Internetworking Multimedia*. Taylor & Francis and Morgan Kaufmann Publishers, 1999.

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.

[Com99] Mit Messverfahren die IT-Systeme durchleuchten. *Computerwoche No. 21*, May 28 1999. http://www.computerwoche.de.

[DC85] Steve E. Deering and David R. Cheriton. Host Groups: A Multicast Extension to the Internet Protocol. Request for Comments RFC 966, IETF, 1985.

[DC90] Steve E. Deering and David R. Cheriton. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Transactions on Computer Systems*, 8(2):85–110, 1990.

[DC99] James D. Davidson and Danny Coward. *The Java Servlet Specification, v2.2*. Sun Microsystems, 1999.

[DEF+94] Steve E. Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei. An Architecture for Wide-Area Multicast Routing. In *Proceedings of the conference on Communications architectures, protocols and applications, SIGCOMM '94*, pages 126–135, London, United Kingdom, 1994.

[DFL+98] Kevin Downes, Merilee Ford, Kim H. Lew, Steve Spanier, and Tim Stevenson. *Internetworking Technologies Handbook*. Macmillan Technical Publishing, second edition, 1998.

[DH98] Steve E. Deering and Robert Hinden. Internet Protocol, Version 6 (IPv6) - Specification. Request for Comments RFC 2460, IETF, 1998.

[DIN85] Terminology in Computing. DIN 44300, German National Standards Body, 1985.

[Ear96] Michael J. Earl. The Risks of Outsourcing IT. *Sloan Management Review*, pages 26–32, Spring 1996.

[Eng99] Fred Engel. The Role of Service Level Agreements in the Internet Service Provider Industry. *International Journal of Network Management*, 9:299–301, 1999.

[Eur99] A Common Framework for QoS/Network Performance in a multi-Provider Environment. Project p806-gi deliverable 1 - the eqos framework - version 2, The European Institute for Research and Strategic Studies in Telecommunications (EURESCOM), 1999.

[Fer99] Jacques Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Publishing Company, Inc., 1999.

[Flu95] François Fluckiger. *Understanding networked multimedia: applications and technology*. Prentice Hall Europe, 1995.

[FSVP98] George Fankhauser, Burkhard Stiller, Cristoph Vögtli, and Bernhard Plattner. Reservation-based Charging in an Integrated Services Network. In *4th INFORMS Telecommunications Conference*, Bocca Raton, Florida, USA, 1998.

[FWD+01] Françcois Le Faucheur, Liwen Wu, Bruce Davie, Shahram Davari, Pasi Vaananen, Ram Krishnan, Pierrick Cheval, and Juha Heinanen. MPLS Support of Differentiated Services. Internet Draft draft-ietf-mpls-diff-ext-09.txt, IETF, 2001.

[G0̈1] Manuel Günter. *Management of Multi-Provider Internet Services with Software Agents*. PhD thesis, Institute of Computer Science and Applied Mathematics (IAM), University of Berne, 2001.

[Gau99] Thomas Gaugler. *Interorganisatorische Informationssysteme (IOS) - Ein Analyse- und Gestaltungsrahmen für das Informationsmanagement*. PhD thesis, Department of Information Technology, University of Zurich, 1999.

[GB00] Manuel Günter and Torsten Braun. Internet Service Delivery Control with Mobile Code. In *IFIP Conference on Intelligence in Networks (SmartNet 2000)*, Vienna, Austria, 2000.

[Gib01] Jerry D. Gibson, editor. *Multimedia Communications*. Academic Press, 2001.

[Gli02] Martin Glinz. Software-Qualitätsmanagement. Folienskript zur Kernvorlesung Software Engineering an der Universität Zürich, Sommersemester 2002.

[Gro02] Dan Grossman. New Terminology and Clarifications for DiffServ. Internet Draft draft-ietf-diffserv-new-terms-08.txt, IETF, 2002.

[Gun96] Shishir Gundavaram. *CGI Programming on the World Wide Web*. O'Reilly and Associates, Inc., 1996.

[HBWW99] Juha Heinanen, Fred Baker, Walter Weiss, and John Wroclawski. Assured Forwarding PHB Group. Request for Comments RFC 2597, IETF, 1999.

[HD99] Petra Hofmann and G. Dülfer. Sicherheit in Outsourcing-Partnerschaften - Management effektiver Service Level Agreements. *Connection*, 1999. http://www.connector.de.

[Hei96] Bernd Heinrichs. *Multimedia im Netz*. Springer, 1996.

[Hei02] Lutz J. Heinrich. *Informationsmanagement*. Oldenbourg, seventh edition, 2002.

[HS98] Petra Hofmann and Beate Schmitt. IT Service Management. *Connection*, 1998. http://www.connector.de.

[ICS] Enterprise Process Management Framework - A Definition of Service Level Management. Technical report, Intelligent Communication Software (ICS).

[IIFC97] Crystaliz Inc., General Magic Inc., GMD Focus, and IBM Corp. Mobile Agent Facility Specification. Technical report, Object Management Group (OMG), 1997.

[ISO93] Information technology - digital compression and coding of continuous-tone still images. International Standard ISO/IEC JTC 1, International Standards Organization, 1993.

[ISO00] Quality Management Systems - Fundamentals and Vocabulary. International Standard ISO 9000:2000, International Standards Organization, 2000.

[ITU88] General Aspects of Services in ISDN. Recommendation I.210, International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), 1988.

[ITU92] Principles for a Telecommunications Management Network. Recommendation M.3010, International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), 1992.

[ITU94a] Data Networks and Open System Communications, Open Systems Interconnection (OSI), Model and Notion. Recommendation X.200, International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), 1994.

[ITU94b] Quality of Service, Network Management, and Traffic Engineering. Recommendation E.800, International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), 1994.

[ITU96] Framework for service quality agreement. Recommendation E.801, International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), 1996.

[ITU97] Information Technology - Quality of Service: Framework. Recommendation x.641 | international standard 13236, International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) | International Organization for Standardization - International Electrotechnical Commission (ISO/IEC), 1997.

[JGE99] Terje Jensen, Irena Grgic, and Ola Espvik. Managing Quality of Service in Multi-Provider Environment. In *World Telecom '99*, Geneva, Switzerland, 1999.

[JNP99] Van Jacobson, Kathleen Nichols, and Kedarnath Poduri. An Expedited Forwarding PHB. Request for Comments RFC 2598, IETF, 1999.

[Kai98] Urs Kaiser. Sicherheits- und Kostenaspekte in elektronischen Zahlungsystemen für RSVP. Master's thesis, Institut für Technische Informatik und Kommunikationsnetze, TIK, ETH Zürich, 1998.

[KB00] Ibrahim Khalil and Torsten Braun. Implementation of a Bandwidth Broker for Dynamic End-to-End Resource Reservation in Outsourced Virtual Private Networks. In *The Conference on Leading Edge and Practical Computer Networking (LCN 2000)*, Tampa, Florida, USA, 2000.

[KH01] Helmut Kneer and Harald Häuschen. Contracting and Monitoring. Streamcom deliverable, Department of Information Technology IFI, University of Zurich, 2001.

[KHB02] Helmut Kneer, Harald Häuschen, and Kurt Bauknecht. Tradable Service Level Agreements to Manage Network Resources for Streaming Internet Services. In *The Tenth European Conference on Information Systems (ECIS 2002)*, Gdańsk, Poland, 2002.

[KM02] Helmut Kneer and Raymond Marfurt. A Contracting Protocol for Managing Quality of Service in a Multi-Provider Environment. In *IADIS International Conference WWW/Internet 2002*, Lisbon, Portugal, 2002.

[Kne01] Helmut Kneer. Security Aspects and Solutions. Streamcom deliverable, Department of Information Technology IFI, University of Zurich, 2001.

[Kot00] Philip Kotler. *Marketing Management*. Prentice Hall, tenth edition, 2000.

[Krc00] Helmut Krcmar. *Informationsmanagement*. Springer, second edition, 2000.

[KSHS02] Helmut Kneer, Henrik Stormer, Harald Häuschen, and Burkhard Stiller. An Agent-based Framework for Monitoring Service Contracts. In *ThirdInternational Conference on Electronic Commerce and Web Technologies (EC-Web 2002)*, Aix-en-Provence, France, 2002.

[Kun98] Thomas Kunkelmann. *Sicherheit für Videodaten*. Verlag Vieweg & Sohn, 1998.

[KW96] Ravi Kalakota and Andrew B. Whinston. *Frontiers of Electronic Commerce*. Addison-Wesley Publishing Company, Inc., 1996.

[KZDS00] Helmut Kneer, Urs Zurfluh, Gabriel Dermler, and Burkhard Stiller. A Business Model for Charging and Accounting of Internet Services. In *First International Conference on Electronic Commerce and Web Technologies (EC-Web 2000)*, Greenwich, U.K., 2000.

[KZM99] Helmut Kneer, Urs Zurfluh, and Christian Matt. Security Architecture. CATI Deliverable CATI-UZH-DN-I-002-3.0, Department of Information Technology IFI, University of Zurich, 1999.

[KZM00] Helmut Kneer, Urs Zurfluh, and Christian Matt. Billing and Payment. CATI Deliverable CATI-UZH-DN-I-004-3.0, Department of Information Technology IFI, University of Zurich, 2000.

[KZS01] Helmut Kneer, Urs Zurfluh, and Burkhard Stiller. A Model for Value-added Internet Service Provisioning. In *The First IFIP Conference on E-Commerce, E-Business, and E-Government (I3E 2001)*, Zurich, Switzerland, 2001.

[Law93] L. G. Lawrence. The Role of Roles. *Computers & Security*, 12:15–21, 1993.

[LeG91] Didier J. LeGall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):305–313, 1991.

[LL99] Victor O.K. Li and Wanjiun Liao. Video-on-Demand Systems. In Borko Furht, editor, *Handbook of Internet and Multimedia systems and applications*, pages 511–529. CRC Press LLC, 1999.

[LO99] Danny B. Lange and Mitsuru Oshima. Seven Good Reasons for Mobile Agents. *Communications of the ACM*, 42(3):88–89, 1999.

[Mar01] Raymond Marfurt. Video-on-Demand Contracting Architecture (VODCA). Master's thesis, Institut für Informatik der Universität Zürich, 2001.

[Mer96] Michael Merz. *Elektronische Märkte im Internet*. International Thomson Publishing GmbH, 1996.

[Mey75] *Meyers Enzyklopädisches Lexikon, Band 15*. Bibliographisches Institut AG, Mannheim, Germany, 1975.

[MHE93] Information Technology - Coded Representation of Multimedia and Hypermedia Information, Part 1: Base notation (ASN.1). Committee draft ISO/IEC CD 13522-1, Multimedia and Hypermedia information coding Expert Group (MHEG), ISO/IEC JTC1/SC29/WG12, 1993.

[MJ99] Richard Murch and Tony Johnson. *Intelligent Software Agents*. Prentice Hall, 1999.

[MMV95] Jeffrey K. MacKie-Mason and Hal R. Varian. Pricing the Internet. In Brian Kahin and James Keller, editors, *Public Access to the Internet*, pages 269–314. MIT Press, 1995.

[MMW96] Jeffrey K. MacKie-Mason and Kimberly White. Evaluating and Selecting Digital Payment Mechanisms. In *Telecommunications Policy Research Conference*, Solomon's Island, Maryland, USA, 1996.

[Mur96] John P. Murray. Making Customer Service A Priority. *Information Systems Management*, pages 7–11, Winter 1996.

[NBBB98] Kathleen Nichols, Steven Blake, Fred Baker, and David Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. Request for Comments RFC 2474, IETF, 1998.

[Neb97] R. Nebe. Offline, Online und/oder Live: Wo liegt der Markt für geschäftliche Multimedia-Anwendungen? In Joachim Fischer, editor, *Internet, Online-Dienste, Multimedia ... Innovative Standards und Services der Geschäftskommunikation*. Verlag Online GmbH, 1997.

[NO97] Jan Niessen and Paul Oldenburg. *Service Level Management*. Central Computer and Telecommunications Agency (CCTA), 1997.

[Orl00] Harald Orlamünder. *High-Speed-Netze*. Hüthig Verlag Heidelberg, 2000.

[OS00] Huw Oliver and Dave Songhurst. Market Managed Multiservice Internet. *Telektronikk*, 2.2000.

[Par92] Craig Partridge. A Proposed Flow Specification. Request for Comments RFC 1363, IETF, 1992.

[Par95] Craig Partridge. Using the Flow Label Field in IPv6. Request for Comments RFC 1809, IETF, 1995.

[PD96] Larry L. Peterson and Bruce S. Davie. *Computer Networks - A Systems Approach*. Morgan Kaufmann Publishers, Inc., 1996.

[Per00] *IT Service Management*. Perseo Consult AG, Basel, Switzerland, third edition, 2000.

[Pie96] Duane L. Piede. Restructuring Internal Services. *Information Systems Management*, pages 12–17, Winter 1996.

[Pos81] Jon Postel. Internet Protocol. Request for Comments RFC 791, IETF, 1981.

[RLS99] Peter Reichl, Simon Leinen, and Burkhard Stiller. A Practical Review of Pricing and Cost Recovery for Internet Services. In *2nd Berlin Internet Economics Workshop (IEW '99)*, Berlin, Germany, 1999.

[Ruh01] Thomas Ruhmann. Das SLM-Dilemma. *Information Week No. 14*, June 28 2001. http://www.informationweek.de.

[RVC01] Eric Rosen, Arun Viswanathan, and Ross Callon. Multiprotocol Label Switching Architecture. Request for Comments RFC 3031, IETF, 2001.

[Say00] Khalid Sayood. *Introduction to Data Compression*. Morgan Kaufmann Publishers, second edition, 2000.

[SBGP99] Burkhard Stiller, Torsten Braun, Manuel Günter, and Bernhard Plattner. The CATI Project - Charging and Accounting Technology for the Internet. In *5th European Conference on Multimedia Applications, Services, and Techniques (ECMAST '99)*, volume 1629 of *Lecture Notes in Computer Science*, pages 281–296, Madrid, Spain, 1999. Springer.

[Sch95] Bruce Schneier. *Applied Cryptography*. Wiley, 1995.

[SDW92] Timothy W. Strayer, Bert J. Dempsey, and Asfred C. Weaver. *XTP: The Xpress Transfer Protocol*. Addison-Wesley Publishing Company, Inc., 1992.

[Ser98] Amdahl IT Services. Service-Level Management - An Introduction for Executives. Accessible at URL: http://www.amdahl.de, 1998.

[SFJ+99] Burkhard Stiller, George Fankhauser, Gabriela Joller, Peter Reichl, and Nathalie Weiler. Open Charging and QoS Interfaces for IP Telephony. In *The Internet Summit (INET '99)*, San Jose, California, USA, 1999.

[SFP98] Burkhard Stiller, George Fankhauser, and Bernhard Plattner. Charging of Multimedia Flows in an Integrated Services Network. In *Proceedings of the 8th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '98)*, pages 189–192, Cambridge, UK, 1998.

[SFPW98] Burkhard Stiller, George Fankhauser, Bernhard Plattner, and Nathalie Weiler. *Pre-study on Customer Care, Accounting, Charging, Billing, and Pricing*. Institut für Technische Informatik und Kommunikationsnetze TIK, ETHZ Zürich, Gloriastrasse 35, 8092 Zürich, Schweiz, 1998.

[SGRF01] Burkhard Stiller, Jan Gerke, Peter Reichl, and Placi Flury. Management of Differentiated Services Usage by the Cumulus Pricing Scheme and a Generic Internet Charging System. In *IEEE/IFIP Symposium on Integrated Network Management (IM 2001)*, Seattle, Washington, USA, 2001.

[SL00] Morris Sloman and Emil Lupu. Policy-based Network Management. In *IEEE/IFIP Network Operations and Management Symposium (NOMS 2000)*, Honolulu, Hawaii, USA, 2000.

[SMJ00] Rick Sturm, Wayne Morris, and Mary Jander. *Foundations of Service Level Management*. SAMS, 2000.

[SPG97] Scott Shenker, Craig Partridge, and Roch Guerin. Specification of Guaranteed Quality of Service. Request for Comments RFC 2212, IETF, 1997.

[SS00] Yun Q. Shi and Huifang Sun. *Image and Video Compression for Multimedia Engineering*. CRC Press, 2000.

[Ste99] Ralf Steinmetz. *Multimedia-Technologie: Grundlagen, Komponenten und Systeme*. Springer, second edition, 1999.

[Sti95] Burkhard Stiller. Quality of Service Issues in Networking Environments. Technical Report 380, University of Cambridge, Computer Laboratory, England, 1995.

[Tan98] Andrew S. Tanenbaum. *Computernetzwerke*. Prentice Hall, 1998.

[Tan01] Andrew S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, second edition, 2001.

[Tim98] Paul Timmers. Business Models for Electronic Markets. *Focus Theme*, 8(2):3–8, 1998.

[Tri00] Walter Tritt. *Service-Management - Prozesse, Strukturen, Logistik*. Wirtschaftsverlag Langen Müller/Herbig, 2000.

[UUN02] UUNET. Service Level Agreements. Accessible at URL: http://www1.worldcom.com/uunet/terms/sla/, April 2002.

[vDO00] Wil van der Aalst, Jörg Desel, and Andreas Oberweis, editors. *Business Process Management: Models, Techniques, and Empirical Studies*. Springer, 2000.

[VFB00] Walter Vogt, Joachim Fremmer, and Felix Buser. *Nutzen ohne Frust*. Perseo Consult AG, Basel, Switzerland, 2000.

[WJ99] Michael J. Wooldridge and Nicholas R. Jennings. Software Engineering with Agents: Pitfalls and Pratfalls. *IEEE Internet Computing*, 3(3):20–27, 1999.

[Wro97] John Wroclawski. Specification of the Controlled-Load Network Element Service. Request for Comments RFC 2211, IETF, 1997.

[WS97] Lars C. Wolf and Ralf Steinmetz. Concepts for resource reservation in advance. *Multimedia Tools and Applications*, 4(3):255–278, 1997.

[You99] Martha Young. SLAs Help Satisfy Expectations. *Microtimes Magazine No. 200*, November 1999. http://www.microtimes.com.

[YTD+97] Martin Yates, Wataru Takita, Laurence Demoudem, Rickard Jansson, and Harm Mulder. TINA Business Model and Reference Points. Tina-c deliverable, Telecommunications Information Networking Architecture (TINA) Consortium, 1997.

[ZMH98] Urs Zurfluh, Arion Meier, and Marcus Holthaus. SPP ICS Pre-study on Security. Technical report, Institut für Informatik, University of Zurich, 1998.

Lebenslauf

Name:	Helmut Kneer
Geburtsdatum:	06. Mai 1971
Geburtsort:	Münsingen, Deutschland
Staatsangehörigkeit:	Deutsch

Schul- und Universitätsausbildung

1977-1981	Grundschule in Ingstetten
1981-1990	Abitur am Albert-Schweitzer-Gymnasium in Laichingen
1990-1991	Grundwehrdienst in München und Ulm
1991-1992	Studium der Automatisierungstechnik an der Fachhochschule in Reutlingen
1992-1998	Studium der Informatik an der Universität in Ulm mit Abschluss als Diplom-Informatiker
1996-1998	Studium als Austauschstudent an der University of West Florida (UWF) in Pensacola (Florida, USA)
1998-2003	Doktorandenstudium und wissenschaftlicher Assistent am Institut für Informatik der Universität Zürich mit Abschluss als Doktor der Informatik